
MICO Developer Documentation

Release 0.0.1

MICO Authors

Jun 05, 2020

Contents:

1	MICO-Admin UI	1
1.1	local testing	1
1.2	configuration	1
1.3	Documentation of Core Components	1
1.3.1	TypeScript Doc	1
1.4	Mockups	16
1.5	Design Decisions	16
2	MICO-Core	17
2.1	Integration Tests	17
2.1.1	Implementation	17
2.1.2	Special Dockerfiles	17
2.1.3	Requirements	17
2.1.4	Run locally	18
2.2	Javadoc	18
2.2.1	io.github.ust.mico.core	18
2.2.2	io.github.ust.mico.core.broker	19
2.2.3	io.github.ust.mico.core.configuration	39
2.2.4	io.github.ust.mico.core.configuration.extension	42
2.2.5	io.github.ust.mico.core.dto.request	42
2.2.6	io.github.ust.mico.core.dto.response	46
2.2.7	io.github.ust.mico.core.dto.response.internal	53
2.2.8	io.github.ust.mico.core.dto.response.status	53
2.2.9	io.github.ust.mico.core.exception	55
2.2.10	io.github.ust.mico.core.model	69
2.2.11	io.github.ust.mico.core.persistence	90
2.2.12	io.github.ust.mico.core.resource	99
2.2.13	io.github.ust.mico.core.service	118
2.2.14	io.github.ust.mico.core.service.imagebuilder	128
2.2.15	io.github.ust.mico.core.util	130
	Index	137

CHAPTER 1

MICO-Admin UI

The MICO-Admin UI uses the [Angular](#) framework (v7) with the [Angular Material](#) component library.

1.1 local testing

To run the frontend locally first install all dependencies and then start the development server.

```
cd mico-admin
npm install
npm run start
```

Then open <http://localhost:4200> in your browser.

Hint: The frontend (in development mode) expects the backend at <http://localhost:8080>. You can use **kubect1** to forward traffic to a backend running in a cluster.

1.2 configuration

The backend api url can be set in the `/mico-admin/src/environments/environment.ts`.

The docker container uses the environment variable `MICO_REST_API` to setup a forwarding proxy in nginx.

1.3 Documentation of Core Components

1.3.1 TypeScript Doc

app/api/api.service

API Service

class ApiService()

exported from src/app/api/api.service

Service to interact with the mico api.

ApiService.canonizeStreamUrl (streamURL)

Canonize a resource url.

(Remove schema, host, port, api path prefix and leading/trailing slashes.)

Arguments

- **streamURL (string)** – resource url

Returns string –

ApiService.deleteAllApplicationVersions (shortName)

delete all versions of an application uses: DELETE application/{shortName}

Arguments

- **shortName (string)** – shortName of the application versions to be deleted

Returns Observable –

ApiService.deleteAllServiceVersions (shortName)

delete all versions of a service uses: DELETE services/{shortName}

Arguments

- **shortName (string)** – shortName of the service versions to be deleted

Returns Observable –

ApiService.deleteApplication (shortName, version)

Deletes a specific application version uses: DELETE applications/{shortName}/{version}

Arguments

- **shortName (string)** – shortName of the application
- **version (string)** – version of the application

Returns Observable –

ApiService.deleteApplicationKafkaFaasConnector (applicationShortName, applicationVersion, instanceId)

Deletes an includes relationship from an application to a service uses: DELETE applications/{applicationShortName}/{applicationVersion}/kafka-faas-connector/{instanceId}

Arguments

- **applicationShortName (string)** – the applications shortName
- **applicationVersion (string)** – the applications version
- **instanceId (string)** – the kafka faas connector instance id

Returns Observable –

`ApiService.deleteApplicationServices` (*applicationShortName*, *applicationVersion*, *serviceShortName*)

Deletes an includes relationship from an application to a service uses: DELETE applications/{shortName}/{version}/services

Arguments

- **applicationShortName** (*string*) – the applications shortName
- **applicationVersion** (*string*) – the applications version
- **serviceShortName** (*any*) – the services shortName

Returns Observable –

`ApiService.deleteService` (*shortName*, *version*)

Arguments

- **shortName** (*any*) –
- **version** (*any*) –

Returns Observable –

`ApiService.deleteServiceDependee` (*serviceShortName*, *serviceVersion*, *dependeeShortName*, *dependeeVersion*)

deletes a depends on relation of a service uses: DELETE services/{shortName}/{version}/dependees

Arguments

- **serviceShortName** (*string*) – shortName of the depending service
- **serviceVersion** (*string*) – version of the depending service
- **dependeeShortName** (*string*) – shortName of the depended service
- **dependeeVersion** (*string*) – version of the depended service

Returns Observable –

`ApiService.deleteServiceInterface` (*shortName*, *version*, *serviceInterfaceName*)

Deletes a service interface uses: DELETE services/{shortName}/{version}/interfaces/{serviceInterfaceName}

Arguments

- **shortName** (*string*) – the services shortName
- **version** (*string*) – the services version
- **serviceInterfaceName** (*string*) – the interfaces name

Returns Observable –

`ApiService.getApplication` (*shortName*, *version*)

get an application based on its shortName and version uses: GET applications/{shortName}/{version}

Arguments

- **shortName** (*string*) – of the application
- **version** (*string*) – of the application

Returns Observable –

`ApiService.getApplicationDeploymentStatus` (*applicationShortName*, *applicationVersion*)

Returns the deployment status of a specified application uses: GET applications/{applicationShortName}/{applicationVersion}/deploymentStatus

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application

Returns Observable –

ApiService.getApplicationKafkaFaasConnector (*applicationShortName, applicationVersion, instanceId*)

Get an existing kafka faas connector deployment info uses: GET applications/{applicationShortName}/{applicationVersion}/kafka-faas-connector/{instanceId}

Arguments

- **applicationShortName** (*string*) – the applications shortName
- **applicationVersion** (*string*) – the applications version
- **instanceId** (*string*) – the kafka faas connector instance id this gets a specific instance

Returns Observable –

ApiService.getApplicationStatus (*shortName, version*)

returns runtime information (status) about a specific application and its services uses GET applications/{shortName}/{version}/status

Arguments

- **shortName** (*string*) – the applications shortName
- **version** (*string*) – the applications version

Returns Observable –

ApiService.getApplicationVersions (*shortName*)

Get all versions of an application based on its shortName uses: GET application/{shortName}

Arguments

- **shortName** (*string*) – the shortName of the applicaton

Returns Observable –

ApiService.getApplications ()

Get application list uses: GET application

Returns Observable –

ApiService.getJobStatus (*applicationShortName, applicationVersion*)

retrieves the status of a deployment uses: GET jobs/{applicationShortName}/{applicationVersion}/status

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application

Returns Observable –

ApiService.getModelDefinitions ()

Returns Observable –

ApiService.getOpenFaaSFunctions ()

Retrieves the function list of the OpenFaaS Portal from the backend.

Returns Observable –

`ApiService.getOpenFaaSIP()`

Retrieves the IP address of the OpenFaaS Portal from the backend.

Returns Observable –

`ApiService.getService(shortName, version)`

Get a specific version of a service uses: GET services/{shortName}/{version}

Arguments

- **shortName** (*any*) – unique short name of the service
- **version** (*any*) – service version to be returned

Returns Observable –

`ApiService.getServiceDependees(shortName, version)`

Get all services a specific service depends on. uses: GET services/{shortName}/{version}/dependees

Arguments

- **shortName** (*any*) – unique short name of the service
- **version** (*any*) – service version to be returned

Returns Observable –

`ApiService.getServiceDependencyGraph(shortName, version)`

Get full dependency graph of service uses: GET services/{shortName}/{version}/dependencyGraph

Arguments

- **shortName** (*any*) – unique short name of the service
- **version** (*any*) – service version to be returned

Returns Observable –

`ApiService.getServiceDependers(shortName, version)`

Get all services depending on a specific service uses: GET services/{shortName}/{version}/dependers

Arguments

- **shortName** (*any*) – unique short name of the service
- **version** (*any*) – service version to be returned

Returns Observable –

`ApiService.getServiceDeploymentInformation(applicationShortName, applicationVersion, serviceShortName)`

Returns the deployment information of an applications included service uses: GET applications/{applicationShortName}/{applicationVersion}/deploymentInformation/{serviceShortName}

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application
- **serviceShortName** (*any*) – shortName of the service

Returns Observable –

`ApiService.getServiceInterfacePublicIp(serviceShortName, serviceVersion, instanceId, interfaceShortName)`

Gets the public IP of a deployed service uses: services/{shortName}/{version}/interaces/{interfaceName}/publicIP

Arguments

- **serviceShortName** (*string*) – shortName of the service
- **serviceVersion** (*string*) – version of the service
- **instanceId** (*string*) – the instance id of the service deployment information
- **interfaceShortName** (*string*) – name of the serviceInterface

Returns Observable –

`ApiService.getServiceInterfaces (shortName, version)`

get all service interfaces of a specified service uses: GET services/{shortName}/{version}/interfaces

Arguments

- **shortName** (*any*) – shortName of the service
- **version** (*any*) – version of the service

Returns Observable –

`ApiService.getServiceStatus (shortName, version)`

returns runtime information (status) about a specific service uses GET services/{shortName}/{version}/status

Arguments

- **shortName** (*string*) – the services shortName
- **version** (*string*) – the services version

Returns Observable –

`ApiService.getServiceVersions (shortName)`

Get all versions of a service based on its shortName uses: GET services/{shortName}

Arguments

- **shortName** (*any*) – the serices shortName

Returns Observable –

`ApiService.getServiceVersionsViaGithub (url)`

takes an url to a github repository and returns the available versions of the repository. uses: GET services/import/github

Arguments

- **url** (*string*) – url to the github repository

Returns Observable –

`ApiService.getServiceYamlConfig (shortName, version)`

Get a yaml string containing a services kubeconfig uses: GET services/{shortName}/{version}/yaml

Arguments

- **shortName** (*string*) – unique short name of the service
- **version** (*string*) – service version

Returns Observable –

`ApiService.getServices ()`

Get a list of all services uses: GET services

Returns Observable –

`ApiService.getStreamSource (streamURL, defaultSubject)`

Fetch the stream source for the given resource url.

Arguments

- **streamURL** (*string*) – resource url
- **defaultSubject** (*<TODO>*) – function to create a new streamSource if needed (default: BehaviourSubject)

Returns Subject –

`ApiService.getTopics ()`

Get all known used kafka topics from the backend.

Returns Observable –

`ApiService.new ApiService (rest, snackBar)`

Arguments

- **rest** (*src/app/api/api-base-function.service.ApiBaseFunctionService*) –
- **snackBar** (*MatSnackBar*) –

Returns src/app/api/api.service.ApiService –

`ApiService.pollDeploymentJobStatus (applicationShortName, applicationVersion)`

Polls the deployment job of a given application and provides feedback if the deployment failed/ was successful. Also stops polling after 10 minutes

Arguments

- **applicationShortName** (*string*) – applicationShortName of the application to be polled
- **applicationVersion** (*string*) – applicationVersion of the application to be polled

Returns Subscription –

`ApiService.pollServiceInterfacePublicIp (serviceShortName, serviceVersion, instanceId, interfaceName)`

Polls the public ip of a services interface until a public ip is available. Also ends the polling after 3 minutes.

Arguments

- **serviceShortName** (*string*) – serviceShortName of the interface to be polled
- **serviceVersion** (*string*) – serviceVersion of the interface to be polled
- **instanceId** (*string*) – the instance id of the service deployment information
- **interfaceName** (*string*) – interfaceName of the interface to be polled

Returns Subscription –

`ApiService.polling`

type: Map

`ApiService.postApplication (data)`

Creates a new application uses: POST applications

Arguments

- **data** (*any*) – object holding the applications information

Returns Observable –

`ApiService.postApplicationDeployCommand(shortName, version)`

commands the mico-core application to deploy an application uses: POST applications/{shortName}/{version}/deploy

Arguments

- **shortName** (*string*) – the applications shortName
- **version** (*string*) – the applications version

Returns Observable –

`ApiService.postApplicationKafkaFaasConnector(applicationShortName, applicationVersion, connectorVersion, instanceId)`

Adds a new kafka faas connector to the application or update an existing one uses: POST applications/{applicationShortName}/{applicationVersion}/kafka-faas-connector/{instanceId}?version={kafkaFaasConnectorVersion}

Arguments

- **applicationShortName** (*string*) – the applications shortName
- **applicationVersion** (*string*) – the applications version
- **connectorVersion** (*string*) – the kafka faas connector version if omitted latest version is used
- **instanceId** (*string*) – the kafka faas connector instance id if present this updates a specific instance to this version instead of adding a new instance

Returns Observable –

`ApiService.postApplicationServices(applicationShortName, applicationVersion, serviceShortName, serviceVersion)`

Adds an includes relationship from an application to a service uses: POST applications/{applicationShortName}/{applicationVersion}/services/{serviceShortName}/{serviceVersion}

Arguments

- **applicationShortName** (*string*) – the applications shortName
- **applicationVersion** (*string*) – the applications version
- **serviceShortName** (*string*) – the services shortName
- **serviceVersion** (*string*) – the services version

Returns Observable –

`ApiService.postApplicationUndeployCommand(shortName, version)`

commands the mico-core application to undeploy an application uses: POST applications/{shortName}/{version}/undeploy

Arguments

- **shortName** (*string*) – the applications shortName
- **version** (*string*) – the applications version

Returns Observable –

`ApiService.postService(data)`

Creates a new service uses: POST services

Arguments

- **data** (*any*) – object holding the services information

Returns Observable –

`ApiService.postServiceDependee(serviceShortName, serviceVersion, dependee)`

Adds a depends on relation from a service to an other service (dependee) uses: POST services/{shortName}/{version}/dependees

Arguments

- **serviceShortName** (*any*) – shortName of the depending service
- **serviceVersion** (*any*) – version of the depending service
- **dependee** (*any*) – object holding the dependee

Returns Observable –

`ApiService.postServiceInterface(shortName, version, data)`

Adds a new service interface to a service uses: POST services/{shortName}/{version}/interfaces

Arguments

- **shortName** (*any*) – shortName of the service
- **version** (*any*) – version of the service
- **data** (*any*) – object holding the interfaces information

Returns Observable –

`ApiService.postServiceViaGithub(url, version)`

crawls a specific version from a github repository uses: POST services/import/github

Arguments

- **url** (*string*) – url to the github repository
- **version** (*string*) – version to be crawled

Returns Observable –

`ApiService.promoteApplication(shortName, version, newVersion)`

Creates a new application version from an existing application uses: POST applications/{shortName}/{version}/promote

Arguments

- **shortName** (*any*) – shortName of the application
- **version** (*any*) – version of the application
- **newVersion** (*string*) – the new version of the application

Returns Observable –

`ApiService.promoteService(shortName, version, newVersion)`

Creates a new service version from an existing service uses: POST services/{shortName}/{version}/promote

Arguments

- **shortName** (*any*) – shortName of the service
- **version** (*any*) – version of the service
- **newVersion** (*string*) – the new version of the service

Returns Observable –

`ApiService.putApplication (shortName, version, data)`

Updates an existing application uses: PUT applications/{shortName}/{version}

Arguments

- **shortName** (*any*) – shortName of the application
- **version** (*any*) – version of the application
- **data** (*any*) – object holding the updated application information

Returns Observable –

`ApiService.putApplicationKafkaFaasConnector (applicationShortName, applicationVersion, instanceId, data)`

Updates the deployment information of a kafka faas connector instance. uses: PUT applications/{applicationShortName}/{applicationVersion}/kafka-faas-connector/{instanceId}

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application
- **instanceId** (*string*) – instance id of the kafka faas connector
- **data** (*any*) – object holding the updated deployment information

Returns Observable –

`ApiService.putService (shortName, version, data)`

Updates a services information uses: PUT service/{shortName}/{version}

Arguments

- **shortName** (*any*) – the services shortName
- **version** (*any*) – the services version
- **data** (*any*) – object holding the services updated version

Returns Observable –

`ApiService.putServiceDeploymentInformation (applicationShortName, applicationVersion, serviceShortName, data)`

Updates the deployment information of an applications service uses: PUT applications/{applicationShortName}/{applicationVersion}/deploymentInformation/{serviceShortName}

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application
- **serviceShortName** (*any*) – shortName of the service
- **data** (*any*) – object holding the updated deployment information

Returns Observable –

`ApiService.putServiceInterface (shortName, version, serviceInterfaceName, serviceData)`

Updates a service interface uses: PUT services/{shortName}/{version}/interfaces/{serviceInterfaceName}

Arguments

- **shortName** (*string*) – the services shortName
- **version** (*string*) – the services version

- **serviceInterfaceName** (*string*) – the interfaces name
- **serviceData** (*any*) –

Returns Observable –

ApiService.**rest**

type: src/app/api/api-base-function.service.ApiBaseFunctionService

ApiService.**snackBar**

type: MatSnackBar

ApiService.**startApplicationStatusPolling** (*applicationShortName*, *applicationVersion*)

Starts the polling for 'pollApplicationStatus'. Only one subscriptions is returned at a time, to avoid multiple polling of the same information. Otherwise 'undefined' is returned.

Arguments

- **applicationShortName** (*string*) – shortName of the application
- **applicationVersion** (*string*) – version of the application

Returns Subscription –

ApiService.**streams**

type: Map

Model Service

class ModelsService ()

exported from src/app/api/models.service

ModelsService.**apiService**

type: src/app/api/api.service.ApiService

ModelsService.**canonicalizeModelUri** (*modelUri*)

Canonize a resource url.

(Remove schema, host, port, api path prefix and leading/trailing slashes.)

Arguments

- **modelUri** (*string*) –

Returns string –

ModelsService.**filterModel** (*properties*, *isBlacklist*)

Return a stream filter for api models. (use with map in observable pipe)

Arguments

- **properties** (*Iterable*) – the property keys to filter for (array/set or other iterable)
Use Empty iterable or null to deactivate filter
- **isBlacklist** (*boolean*) – if true the filter ill be applied as blacklist. (default=whitelest/false)

Returns <TODO> –

ModelsService.**getCacheSource** (*cacheURL*)

Fetch the cache source for the given model url.

Arguments

- **cacheURL** (*string*) –

Returns AsyncSubject –

`ModelsService.getModel(modelUrl)`

Get a model for the modelUrl.

Observable only sends a value if the model was found. Times out after 2s

Arguments

- **modelUrl** (*any*) – modelUrl

Returns Observable –

`ModelsService.handleArrayProperties(propRef)`

Handle array type properties.

Replaces items with ApiModelRef to nestedModelCache if needed

Arguments

- **propRef** (*src/app/api/models.service.PropertyRef*) – input PropertyRef

Returns Observable –

`ModelsService.handleComplexProperties(model)`

Check all properties of model for complex properties like arrays or objects.

Replaces all nested models with ApiModelRefs to nestedModelCache

Arguments

- **model** (*src/app/api/apimodel.ApiModel*) – input model

Returns Observable –

`ModelsService.handleNumericPropertyKeys(property)`

Convert all property keys that should have numeric values (like ‘minimum’).

Arguments

- **property** (*src/app/api/models.service.PropertyRef*) – input PropertyRef

Returns Observable –

`ModelsService.handleObjectProperties(propRef)`

Handle object type properties.

Replaces prop with ApiModelRef to nestedModelCache if needed

Arguments

- **propRef** (*src/app/api/models.service.PropertyRef*) – input PropertyRef

Returns Observable –

`ModelsService.localModels`

type: <TODO>

`ModelsService.mergeModels(targetModel, sourceModel)`

Merge two ApiModels into one model.

Arguments

- **targetModel** (*src/app/api/apimodel.ApiModel*) – the model to be merged into

- **sourceModel** (*src/app/api/apimodel.ApiModel*) – the model to be merged

Returns *src/app/api/apimodel.ApiModel* –

ModelsService.mergeProperties (*targetModel, sourceModel*)

Merge properties of two ApiModels into one merged properties object.

Arguments

- **targetModel** (*src/app/api/apimodel.ApiModel*) – the model to be merged into
- **sourceModel** (*src/app/api/apimodel.ApiModel*) – the model to be merged

Returns **<TODO>** –

ModelsService.modelCache

type: Map

ModelsService.nestedModelCache

type: Map

ModelsService.new ModelsService (*apiService*)

Arguments

- **apiService** (*src/app/api/api.service.ApiService*) –

Returns *src/app/api/models.service.ModelsService* –

ModelsService.onlyRequired (*model*)

Stream filter that filters out all not required properties of the model.

Arguments

- **model** (*src/app/api/apimodel.ApiModel*) – input api model

Returns **<TODO>** –

ModelsService.resolveModel (*modelUrl*)

Resolve the modelUrl and return the corresponding model.

Arguments

- **modelUrl** (*string*) – resource url

Returns **Observable** –

ModelsService.resolveModelLinks (*model*)

Resolve all model links and return an observable of pure ApiModels

starting with the first model of an allOf and recursively applying itself

Arguments

- **| src/app/api/apimodel.ApiModelRef | src/app/api/apimodel.ApiModel model** (*src/app/api/apimodel.ApiModelAllOf*) – input model

Returns **Observable** –

ModelsService.resolveProperties (*model*)

Resolves all properties to actual ApiModels.

Arguments

- **model** (*src/app/api/apimodel.ApiModel*) – input model

Returns **Observable** –

app/forms

FormGroup Service

```
class FormGroupService()  
    exported from src/app/forms/form-group.service
```

```
FormGroupService.modelToFormGroup(model)  
    Create a new FormGroup for use with reactive forms.
```

Arguments

- **model** (*src/app/api/apimodel.ApiModel*) – the source ApiModel to create a FormGroup for

Returns FormGroup –

```
FormGroupService.new FormGroupService()
```

Returns `src/app/forms/form-group.service.FormGroupService` –

app/forms/mico-form

Mico Form Component

Component Usage

```
@Input() modelUrl: string  
    A string identifying the jsonschema model to use. Example: local/servicePOST  
  
@Input() filter: string[] = []  
    A list of properties to filter by.  
  
@Input() isBlacklist: boolean = false  
    If property filter is whitelist or blacklist.  
  
@Input() debug: boolean = false  
    If form renders debug data.  
  
@Output() valid: EventEmitter<boolean>  
    If form is valid.  
  
@Output() data: EventEmitter<any>  
    Form data as js object.
```

Example Usage

```
<mico-form [modelUrl]='local/servicePOST' [filter]='[name]' (data)="data = $event  
    <"/></mico-form>
```

Class

```
class MicoFormComponent()  
    exported from src/app/forms/mico-form/mico-form.component
```

Implements:

- `OnInit()`
- `OnChanges()`

Dynamic form component that renders a form from the given `modelUrl`.

`MicoFormComponent._startData`

type: `<TODO>`

`MicoFormComponent.data`

type: `EventEmitter`

`MicoFormComponent.debug`

type: `boolean`

`MicoFormComponent.filter`

type: `string[]`

`MicoFormComponent.form`

type: `FormGroup`

`MicoFormComponent.formGroup`

type: `src/app/forms/form-group.service.FormGroupService`

`MicoFormComponent.formSubscription`

type: `Subscription`

`MicoFormComponent.isBlacklist`

type: `boolean`

`MicoFormComponent.model`

type: `src/app/api/apimodel.ApiModel`

`MicoFormComponent.modelUrl`

type: `string`

`MicoFormComponent.models`

type: `src/app/api/models.service.ModelsService`

`MicoFormComponent.new MicoFormComponent (models, formGroup)`

Arguments

- **models** (`src/app/api/models.service.ModelsService`) –
- **formGroup** (`src/app/forms/form-group.service.FormGroupService`) –

Returns `src/app/forms/mico-form/mico-form.component.MicoFormComponent` –

`MicoFormComponent.ngOnChanges (changes)`

Arguments

- **changes** (`SimpleChanges`) –

`MicoFormComponent.ngOnInit ()`

`MicoFormComponent.onlyRequired`

type: `boolean`

`MicoFormComponent.properties`

type: `src/app/api/apimodel.ApiModel | src/app/api/apimodel.ApiModelRef[]`

`MicoFormComponent.startData`

type: `<TODO>`

`MicoFormComponent.trackByFn(index, item)`

Arguments

- **index** (*any*) –
- **item** (*any*) –

Returns *any* –

`MicoFormComponent.valid`
type: EventEmitter

1.4 Mockups

- Mico Dashboard
- App-Detail Overview
- App-Detail Deploystatus
- App-Detail Deploysettings
- Service-Detail Overview
- Service-Detail Deploystatus

1.5 Design Decisions

- Web-UI Framework
- Browser Compatibility
- Features first
- JSON+hal as serialization Format for the API
- Re-use capabilities of Winery Topology Modeler

2.1 Integration Tests

Integration Tests are implemented with JUnit 4.

Hint: CI/CD pipeline

The setup of the CI/CD pipeline is described in [CI/CD Pipeline](#).

2.1.1 Implementation

Integration tests are marked with the annotation `@Category(IntegrationTests.class)`. To add a new integration test, write a JUnit test and annotate the class with the annotation above.

2.1.2 Special Dockerfiles

To run integration tests on Jenkins there exists a special Dockerfile, named *Dockerfile.mico-core.integrationtests*, which runs the integration tests on 3.6-jdk-alpine Docker container. Build the Dockerfile and run the image to test same as Jenkins does. The Dockerfile runs all tests with the annotation above using the following command: .. code::

```
bash
mvn failsafe:integration-test
```

2.1.3 Requirements

ImageBuilderIntegrationTests require credentials to Docker Hub. Provide the credentials base64 encoded as environment variables:

- `DOCKERHUB_USERNAME_BASE64`

- `DOCKERHUB_PASSWORD_BASE64`

To encode the username and the password with base64 you could use

```
echo -n "username" | base64 -w 0
echo -n "password" | base64 -w 0
```

There are also integration tests which need some Kubernetes credentials. At the moment we cannot support this within our integration test environment. So until fixed, integration tests using Kubernetes won't work. You can ignore them with adding the `@Ignore` annotation on class level.

2.1.4 Run locally

To run the integration tests locally execute

```
mvn failsafe:integration-test
```

2.2 Javadoc

2.2.1 io.github.ust.mico.core

MicoCoreApplication

public class **MicoCoreApplication** implements `ApplicationListener<ApplicationReadyEvent>`
Entry point for the MICO core application.

Fields

gitHubCrawler

GitHubCrawler **gitHubCrawler**

kafkaFaasConnectorConfig

KafkaFaasConnectorConfig **kafkaFaasConnectorConfig**

micoServiceBroker

MicoServiceBroker **micoServiceBroker**

Methods

main

public static void **main** (`String[] args`)

onApplicationEvent

public void **onApplicationEvent** (ApplicationReadyEvent *event*)

Runs when application is ready. Add KafkaFaasConnector in latest version to the database, if the current profile is not 'unit-testing' (e.g. Travis CI).

Parameters

- **event** – the ApplicationReadyEvent

restTemplate

public RestTemplate **restTemplate** (RestTemplateBuilder *builder*)

Parameters

- **builder** –

See also: [RealDeanZhao/autowire-resttemplate.md](#)

2.2.2 io.github.ust.mico.core.broker

BackgroundJobBroker

public class **BackgroundJobBroker**

Broker to operate with jobs.

Constructors

BackgroundJobBroker

public **BackgroundJobBroker** (*MicoBackgroundJobRepository jobRepository*, *MicoApplicationRepository applicationRepository*)

Methods

deleteAllJobs

public void **deleteAllJobs** ()

Delete all jobs in the database. If a future of a job is still running, it will be cancelled.

deleteJob

public void **deleteJob** (*String id*)

Deletes a job in the database. If the future is still running, it will be cancelled.

Parameters

- **id** – the id of the job.

getAllJobs

```
public List<MicoServiceBackgroundJob> getAllJobs ()
```

Retrieves all jobs saved in database.

Returns a List of *MicoServiceBackgroundJob*.

getJobById

```
public Optional<MicoServiceBackgroundJob> getJobById (String id)
```

Retrieves a job by id.

Parameters

- **id** – the id of the job.

Returns a *MicoServiceBackgroundJob*.

getJobByMicoServiceInstanceId

```
public Optional<MicoServiceBackgroundJob> getJobByMicoServiceInstanceId (String instanceId,  
                                                                           MicoServiceBack-  
                                                                           groundJob.Type  
                                                                           type)
```

Return a *MicoServiceBackgroundJob* for a given *instanceId* and *MicoServiceBackgroundJob.Type*.

Parameters

- **instanceId** – instance ID of a *MicoServiceDeploymentInfo*
- **type** – the *MicoServiceBackgroundJob.Type*

Returns the optional job. Is empty if no job exists for the given *instanceId*

getJobStatusByApplicationShortNameAndVersion

```
public MicoApplicationJobStatus getJobStatusByApplicationShortNameAndVersion (String  
                                                                           short-  
                                                                           Name,  
                                                                           String  
                                                                           version)
```

Retrieves the job status of a *MicoApplication*.

Parameters

- **shortName** – the short name of the *MicoApplication*.
- **version** – the version of the *MicoApplication*.

Returns the *MicoApplicationJobStatus* with the status and jobs.

getJobsByMicoService

```
public List<MicoServiceBackgroundJob> getJobsByMicoService (String micoServiceShortName, String  
                                                             micoServiceVersion, MicoService-  
                                                             BackgroundJob.Type type)
```

Return MicoServiceBackgroundJobs for a given MicoService and
MicoServiceBackgroundJob.Type.

Parameters

- **micoServiceShortName** – the short name of a *MicoService*
- **micoServiceVersion** – the version of a *MicoService*
- **type** – the *MicoServiceBackgroundJob.Type*

Returns the job list. Is empty if no job exists for the given *MicoService*

saveFutureOfJob

```
void saveFutureOfJob (MicoServiceDeploymentInfo micoServiceDeploymentInfo, MicoServiceBack-  
                      groundJob.Type type, CompletableFuture<?> future)
```

Saves a future of a job to the database.

Parameters

- **micoServiceDeploymentInfo** – the *MicoServiceDeploymentInfo*
- **future** – the future as a *CompletableFuture*
- **type** – the *MicoServiceBackgroundJob.Type*

saveJob

```
public MicoServiceBackgroundJob saveJob (MicoServiceBackgroundJob job)
```

Save a job to the database.

Parameters

- **job** – the *MicoServiceBackgroundJob*

Returns the saved *MicoServiceBackgroundJob*

saveNewStatus

```
void saveNewStatus (MicoServiceDeploymentInfo micoServiceDeploymentInfo, MicoServiceBackground-  
                    Job.Type type, MicoServiceBackgroundJob.Status newStatus)
```

Saves a new status of a job to the database.

Parameters

- **micoServiceDeploymentInfo** – the *MicoServiceDeploymentInfo*
- **type** – the *MicoServiceBackgroundJob.Type*
- **newStatus** – the new *MicoServiceBackgroundJob.Status*

saveNewStatus

void **saveNewStatus** (*MicoServiceDeploymentInfo* micoServiceDeploymentInfo, *MicoServiceBackgroundJob.Type* type, *MicoServiceBackgroundJob.Status* newStatus, String errorMessage)
Saves a new status of a job to the database.

Parameters

- **micoServiceDeploymentInfo** – the *MicoServiceDeploymentInfo*
- **type** – the *MicoServiceBackgroundJob.Type*
- **newStatus** – the new *MicoServiceBackgroundJob.Status*
- **errorMessage** – the optional error message if the job has failed

DeploymentBroker

public class **DeploymentBroker**

Methods

checkIfKafkaFaasConnectorIsDeployable

public void **checkIfKafkaFaasConnectorIsDeployable** (*MicoServiceDeploymentInfo* kfConnectorDeploymentInfo)
Checks if the properties of the *MicoServiceDeploymentInfo* are valid so the corresponding KafkaFaasConnector is considered deployable.

Parameters

- **kfConnectorDeploymentInfo** – the *MicoServiceDeploymentInfo*

Throws

- *DeploymentRequirementsOfKafkaFaasConnectorNotMetException* – if the requirements are not met

deployApplication

public *MicoApplicationJobStatus* **deployApplication** (String shortName, String version, boolean rebuildImages)
Deploys an application with all its included services and KafkaFaasConnector instances.

Parameters

- **shortName** – the short name of the *MicoApplication*
- **version** – the version of the *MicoApplication*

Throws

- *MicoApplicationNotFoundException* – if the *MicoApplication* does not exist
- *MicoServiceInterfaceNotFoundException* – if the *MicoServiceInterface* does not exist

Returns the *MicoApplicationJobStatus*

undeployApplication

public void **undeployApplication** (*String shortName*, *String version*)

Undeploys an application with all its included services and KafkaFaasConnector instances.

Parameters

- **shortName** – the short name of the *MicoApplication*
- **version** – the version of the *MicoApplication*

Throws

- ***MicoApplicationNotFoundException*** – if the *MicoApplication* does not exist
- ***MicoApplicationIsDeployingException*** – if the *MicoApplication* is currently deploying

KafkaFaasConnectorDeploymentInfoBroker

public class **KafkaFaasConnectorDeploymentInfoBroker**

Methods

getKafkaFaasConnectorDeploymentInformation

public *List<MicoServiceDeploymentInfo>* **getKafkaFaasConnectorDeploymentInformation** (*String mi-coAp-
pli-
ca-
tion-
Short-
Name*, *String mi-
coAp-
pli-
ca-
tion-
Ver-
sion*)

Fetches a list of *MicoServiceDeploymentInfos* of all KafkaFaasConnector instances associated with the specified *MicoApplication*.

Parameters

- **micoApplicationShortName** – the shortName of the micoApplication
- **micoApplicationVersion** – the version of the micoApplication

Throws

- ***MicoApplicationNotFoundException*** – if there is no such micoApplication

Returns the list of *MicoServiceDeploymentInfos*

getKafkaFaasConnectorDeploymentInformation

```
public Optional<MicoServiceDeploymentInfo> getKafkaFaasConnectorDeploymentInformation (String
mi-
coAp-
pli-
ca-
tion-
Short-
Name,
String
mi-
coAp-
pli-
ca-
tion-
Ver-
sion,
String
in-
stan-
ceId)
```

Filters the list of *MicoServiceDeploymentInfo* from *KafkaFaasConnectorDeploymentInfoBroker*.
getKafkaFaasConnectorDeploymentInformation (String, String) for a specific
instanceId.

Parameters

- **micoApplicationShortName** – the short name of the *MicoApplication*
- **micoApplicationVersion** – the version of the *MicoApplication*
- **instanceId** – the instance ID of the *MicoServiceDeploymentInfo*

Throws

- ***MicoApplicationNotFoundException*** – if the *MicoApplication* does not exist.

Returns a single *MicoServiceDeploymentInfo* with an instance ID equal to the give one.

updateKafkaFaasConnectorDeploymentInformation

```
public MicoServiceDeploymentInfo updateKafkaFaasConnectorDeploymentInformation (String
                                                                    in-
                                                                    stan-
                                                                    ceId,
                                                                    KF-
                                                                    Con-
                                                                    nec-
                                                                    torDe-
                                                                    ploy-
                                                                    mentIn-
                                                                    foRe-
                                                                    quest-
                                                                    DTO
                                                                    kf-
                                                                    Con-
                                                                    nec-
                                                                    torDe-
                                                                    ploy-
                                                                    mentIn-
                                                                    foRe-
                                                                    quest-
                                                                    DTO)
```

Updates an existing *MicoServiceDeploymentInfo* in the database based on the values of a *KFConnectorDeploymentInfoRequestDTO* object.

Parameters

- **instanceId** – the instance ID of the *MicoServiceDeploymentInfo*
- **kfConnectorDeploymentInfoRequestDTO** – the *KFConnectorDeploymentInfoRequestDTO*

Throws

- ***KafkaFaasConnectorInstanceNotFoundException*** – if there is no *MicoServiceDeploymentInfo* for the requested *instanceId* stored in the database

Returns the new *MicoServiceDeploymentInfo* stored in the database

MicoApplicationBroker

```
public class MicoApplicationBroker
```

Methods

addKafkaFaasConnectorInstanceToMicoApplicationByVersion

```
public MicoServiceDeploymentInfo addKafkaFaasConnectorInstanceToMicoApplicationByVersion (String  
ap-  
pli-  
ca-  
tion-  
Short-  
Name,  
String  
ap-  
pli-  
ca-  
tion-  
Ver-  
sion,  
String  
kf-  
Con-  
nec-  
torVer-  
sion)
```

Adds a new `KafkaFaasConnector` instance to the `kafkaFaasConnectorDeploymentInfos` of the `MicoApplication`. An unique instance ID will be created that is returned as part of a `MicoServiceDeploymentInfo`.

Parameters

- **applicationShortName** – the short name of the `MicoApplication`
- **applicationVersion** – the version of the `MicoApplication`
- **kfConnectorVersion** – the version of the `KafkaFaasConnector` (`MicoService`

Throws

- **`MicoApplicationNotFoundException`** – if the `MicoApplication` does not exist
- **`MicoApplicationIsNotUndeployedException`** – if the `MicoApplication` is not undeployed
- **`KafkaFaasConnectorVersionNotFoundException`** – if the version of the `KafkaFaasConnector` does not exist in MICO

Returns the `MicoServiceDeploymentInfo` including the newly created instance ID

addMicoServiceToMicoApplicationByShortNameAndVersion

```
public MicoServiceDeploymentInfo addMicoServiceToMicoApplicationByShortNameAndVersion (String  
                                                                                       ap-  
                                                                                       pli-  
                                                                                       ca-  
                                                                                       tion-  
                                                                                       Short-  
                                                                                       Name,  
                                                                                       String  
                                                                                       ap-  
                                                                                       pli-  
                                                                                       ca-  
                                                                                       tion-  
                                                                                       Ver-  
                                                                                       sion,  
                                                                                       String  
                                                                                       ser-  
                                                                                       viceShort-  
                                                                                       Name,  
                                                                                       String  
                                                                                       ser-  
                                                                                       viceV-  
                                                                                       er-  
                                                                                       sion,  
                                                                                       Op-  
                                                                                       tional<String>  
                                                                                       in-  
                                                                                       stan-  
                                                                                       ceI-  
                                                                                       dOp-  
                                                                                       tional)
```

Adds a *MicoService* to a *MicoApplication*. If an instance ID is provided, the existing *MicoServiceDeploymentInfo* will be reused.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **serviceShortName** – the short name of the *MicoService*
- **serviceVersion** – the version of the *MicoService*
- **instanceIdOptional** – the optional instance if of the *MicoServiceDeploymentInfo*

Throws

- ***MicoApplicationNotFoundException*** – if the *MicoApplication* does not exist
- ***MicoServiceNotFoundException*** – if the *MicoService* does not exist
- ***MicoServiceAddedMoreThanOnceToMicoApplicationException*** – if the *MicoService* is added more than once to the *MicoApplication*

- ***MicoApplicationIsNotUndeployedException*** – if the *MicoApplication* is not undeployed
- ***MicoTopicRoleUsedMultipleTimesException*** – if a role of a *MicoTopicRole* is used multiple times
- ***MicoServiceDeploymentInformationNotFoundException*** – if the *MicoServiceDeploymentInfo* does not exist
- ***KubernetesResourceException*** – if there is an error with Kubernetes
- ***MicoApplicationDoesNotIncludeMicoServiceException*** – if the *MicoApplication* does not include the *MicoService*, if it is expected
- ***KafkaFaasConnectorNotAllowedHereException*** – if a *KafkaFaasConnector* is provided instead of a normal *MicoService*

Returns the *MicoServiceDeploymentInfo* that was created or reused

copyAndUpgradeMicoApplicationByShortNameAndVersion

```
public MicoApplication copyAndUpgradeMicoApplicationByShortNameAndVersion (String
                                                                    short-
                                                                    Name,
                                                                    String
                                                                    version,
                                                                    String
                                                                    newVer-
                                                                    sion)
```

createMicoApplication

```
public MicoApplication createMicoApplication (MicoApplication micoApplication)
```

deleteMicoApplicationByShortNameAndVersion

```
public void deleteMicoApplicationByShortNameAndVersion (String shortName, String version)
```

deleteMicoApplicationsByShortName

```
public void deleteMicoApplicationsByShortName (String shortName)
```

getApplicationDeploymentStatus

```
public MicoApplicationDeploymentStatus getApplicationDeploymentStatus (String shortName,
                                                                    String version)
```

getApplicationStatus

```
public MicoApplicationStatusResponseDTO getApplicationStatus (String shortName, String version)
```


getLinksOfMicoApplication

```
public Iterable<Link> getLinksOfMicoApplication (MicoApplication application)
```

getMicoApplicationByShortNameAndVersion

```
public MicoApplication getMicoApplicationByShortNameAndVersion (String shortName, String version)
```

getMicoApplicationForMicoService

```
MicoApplication getMicoApplicationForMicoService (String applicationShortName, String applicationVersion, String serviceShortName)
```

Returns the *MicoApplication* for the provided short name and version if it exists and if it includes the *MicoService* with the provided short name.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **serviceShortName** – the short name of the *MicoService*

Throws

- *MicoApplicationNotFoundException* – if the *MicoApplication* does not exist
- *MicoApplicationDoesNotIncludeMicoServiceException* – if the *MicoApplication* does not include the *MicoService* with the provided short name

Returns the *MicoApplication*

getMicoApplications

```
public List<MicoApplication> getMicoApplications ()
```

getMicoApplicationsByShortName

```
public List<MicoApplication> getMicoApplicationsByShortName (String shortName)
```

getMicoApplicationsUsingMicoService

```
public List<MicoApplication> getMicoApplicationsUsingMicoService (String serviceShortName, String serviceVersion)
```

getMicoServicesOfMicoApplicationByShortNameAndVersion

```
public List<MicoService> getMicoServicesOfMicoApplicationByShortNameAndVersion (String short-  
Name, String ver-  
sion)
```

removeAllKafkaFaasConnectorInstancesFromMicoApplication

```
public void removeAllKafkaFaasConnectorInstancesFromMicoApplication (String applica-  
tionShort-  
Name, String applica-  
tionVersion)
```

Removes all KafkaFaasConnector instances from the *MicoApplication*.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*

Throws

- *MicoApplicationNotFoundException* – if the *MicoApplication* does not exist
- *MicoApplicationIsNotUndeployedException* – if the *MicoApplication* is not undeployed

removeKafkaFaasConnectorInstanceFromMicoApplicationByInstanceId

```
public void removeKafkaFaasConnectorInstanceFromMicoApplicationById (String applica-  
tionShort-  
Name, String applica-  
tionVer-  
sion, String in-  
stanceId)
```

Removes a KafkaFaasConnector instance that has the requested instance ID from the *MicoApplication*.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **instanceId** – the instance ID of the *MicoServiceDeploymentInfo*

Throws

- ***MicoApplicationNotFoundException*** – if the *MicoApplication* does not exist
- ***MicoApplicationIsNotUndeployedException*** – if the *MicoApplication* is not undeployed
- ***KafkaFaasConnectorInstanceNotFoundException*** – if the instance of the *KafkaFaasConnector* does not exist in MICO
- ***MicoApplicationDoesNotIncludeKFConnectorInstanceException*** – if the *MicoApplication* does not include the *KafkaFaasConnector* deployment with the provided instance ID

removeMicoServiceFromMicoApplicationByShortNameAndVersion

```
public MicoApplication removeMicoServiceFromMicoApplicationByShortNameAndVersion (String  
ap-  
pli-  
ca-  
tion-  
Short-  
Name,  
String  
ap-  
pli-  
ca-  
tion-  
Ver-  
sion,  
String  
ser-  
viceShort-  
Name)
```

updateKafkaFaasConnectorInstanceOfMicoApplicationByVersionAndInstanceId

```
public MicoServiceDeploymentInfo updateKafkaFaasConnectorInstanceOfMicoApplicationByVersionAndInsta
```

Updates an existing KafkaFaasConnector instance of the *MicoApplication* to a new version.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **kfConnectorVersion** – the version of the KafkaFaasConnector (*MicoService*
- **instanceId** – the instance ID of an existing *MicoServiceDeploymentInfo*. It will be reused to update its version

Throws

- *MicoApplicationNotFoundException* – if the *MicoApplication* does not exist
- *MicoApplicationIsNotUndeployedException* – if the *MicoApplication* is not undeployed
- *KafkaFaasConnectorVersionNotFoundException* – if the version of the KafkaFaasConnector does not exist in MICO
- *KafkaFaasConnectorInstanceNotFoundException* – if there is no instance for the provided instance ID

Returns the existing *MicoServiceDeploymentInfo* with the new version

updateMicoApplication

```
public MicoApplication updateMicoApplication (String shortName, String version, MicoApplication  
                                             micoApplication)
```

MicoServiceBroker

```
public class MicoServiceBroker
```

Methods

checkIfDependencyAlreadyExists

```
public boolean checkIfDependencyAlreadyExists (MicoService service, MicoService serviceDe-  
pendee)
```

deleteAllDependees

```
public MicoService deleteAllDependees (MicoService service)
```

deleteAllVersionsOfService

```
public void deleteAllVersionsOfService (String shortName)
```

deleteDependencyBetweenServices

```
public MicoService deleteDependencyBetweenServices (MicoService service, MicoService service-  
ToDelete)
```

deleteService

```
public void deleteService (MicoService service)
```

findDependers

```
public List<MicoService> findDependers (MicoService service)
```

getAllServicesAsList

```
public List<MicoService> getAllServicesAsList ()
```

getAllVersionsOfServiceFromDatabase

```
public List<MicoService> getAllVersionsOfServiceFromDatabase (String shortName)
```

getDependeesByMicoService

```
public List<MicoService> getDependeesByMicoService (MicoService service)
```

getDependencyGraph

```
public MicoServiceDependencyGraphResponseDTO getDependencyGraph (MicoService micoService-  
Root)
```

getDependers

```
public List<MicoService> getDependers (MicoService serviceToLookFor)
```

getLatestKFConnectorVersion

```
public String getLatestKFConnectorVersion ()  
Returns the latest version of the KafkaFaaSConnector (according to the database)
```

Throws

- *KafkaFaasConnectorLatestVersionNotFound* – if no KafkaFaasConnector can be found

Returns the latest version of the KafkaFaaSConnector

getServiceById

```
public MicoService getServiceById (Long id)
```

getServiceFromDatabase

```
public MicoService getServiceFromDatabase (String shortName, String version)
```

getServiceInstanceFromDatabase

```
public MicoServiceDeploymentInfo getServiceInstanceFromDatabase (String shortName, String  
version, String instanceId)
```

getServiceYamlByShortNameAndVersion

```
public String getServiceYamlByShortNameAndVersion (String shortName, String version)  
Return yaml for a MicoService for the give shortName and version.
```

Parameters

- **shortName** – the short name of the *MicoService*.
- **version** – version the version of the *MicoService*.

Returns the kubernetes YAML for the *MicoService*.

persistNewDependencyBetweenServices

```
public MicoService persistNewDependencyBetweenServices (MicoService service, MicoService  
serviceDependee)
```

persistService

```
public MicoService persistService (MicoService newService)
```

promoteService

```
public MicoService promoteService (MicoService service, String newVersion)
```

updateExistingService

```
public MicoService updateExistingService (MicoService service)
```

MicoServiceDeploymentInfoBroker

```
public class MicoServiceDeploymentInfoBroker
```

Methods

cleanUpTanglingNodes

```
public void cleanUpTanglingNodes ()
```

Cleans up tangling nodes related to a *MicoServiceDeploymentInfo* in the database. In case addition properties (stored as separate node entity) such as labels, environment variables have been removed from a service deployment information, the standard `save ()` function of the service deployment information repository will not delete those “tangling” (without relationships) labels (nodes), hence the manual clean up.

createOrReuseOpenFaaSFunctionsInDatabase

```
MicoServiceDeploymentInfo createOrReuseOpenFaaSFunctionsInDatabase (MicoServiceDeploymentInfo  
                                                                    serviceDeploy-  
                                                                    mentInfo)
```

Checks if the given OpenFaaS function name is already present in the database. If so it will be reused. Otherwise a new node will be created.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns the updated *MicoServiceDeploymentInfo*

createOrReuseTopic

```
Optional<MicoTopic> createOrReuseTopic (String topicName)
```

createOrReuseTopics

```
List<MicoTopic> createOrReuseTopics (List<String> topicNames)
```

createOrReuseTopicsInDatabase

MicoServiceDeploymentInfo **createOrReuseTopicsInDatabase** (*MicoServiceDeploymentInfo* *serviceDeploymentInfo*)

Checks if topics with the same name already exists in the database. If so reuse them by setting the id of the existing Neo4j node and save them. If not create them in the database.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo* containing topics

getExistingServiceDeploymentInfo

public *MicoServiceDeploymentInfo* **getExistingServiceDeploymentInfo** (*MicoApplication* *micoApplication*, *MicoService* *micoService*)

Retrieves the *MicoServiceDeploymentInfo* that is used for the deployment of the requested *MicoService* as part of a *MicoApplication*. There must not be zero or more than one service deployment information stored. If that's the case, an *IllegalStateException* will be thrown.

Parameters

- **micoApplication** – the *MicoApplication*
- **micoService** – the *MicoService*

Throws

- **IllegalStateException** – if there is no or more than one service deployment information stored

Returns the one and only existing *MicoServiceDeploymentInfo*

getMicoServiceDeploymentInformation

public *MicoServiceDeploymentInfo* **getMicoServiceDeploymentInformation** (String *applicationShortName*, String *applicationVersion*, String *serviceShortName*)

Returns the *MicoServiceDeploymentInfo* stored in the database.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **serviceShortName** – the short name of the *MicoService*

Throws

- **MicoServiceDeploymentInformationNotFoundException** – if there is no *MicoServiceDeploymentInfo* stored in the database
- **MicoApplicationNotFoundException** – if there is no *MicoApplication* with the specified short name and version

- ***MicoApplicationDoesNotIncludeMicoServiceException*** – if there is no service included in the specified *MicoApplication* with the particular short name

Returns the *MicoServiceDeploymentInfo* stored in the database

setDefaultDeploymentInformationForKafkaEnabledService

```
void setDefaultDeploymentInformationForKafkaEnabledService (MicoServiceDeploymentInfo
                                                         micoServiceDeploymentInfo)
```

Sets the default environment variables for Kafka-enabled MicoServices. See *MicoEnvironmentVariable.DefaultNames* for a complete list.

Parameters

- ***micoServiceDeploymentInfo*** – The *MicoServiceDeploymentInfo* with an corresponding MicoService

updateMicoServiceDeploymentInformation

```
public MicoServiceDeploymentInfo updateMicoServiceDeploymentInformation (String applicationShortName,
                                                                           String applicationVersion,
                                                                           String serviceShortName,
                                                                           MicoServiceDeploymentInfoRequestDTO
                                                                           serviceDeploymentInfoDTO)
```

Updates an existing *MicoServiceDeploymentInfo* in the database based on the values of a *MicoServiceDeploymentInfoRequestDTO* object.

Parameters

- ***applicationShortName*** – the short name of the *MicoApplication*
- ***applicationVersion*** – the version of the *MicoApplication*
- ***serviceShortName*** – the short name of the *MicoService*
- ***serviceDeploymentInfoDTO*** – the *MicoServiceDeploymentInfoRequestDTO*

Throws

- ***MicoApplicationNotFoundException*** – if there is no *MicoApplication* with the specified short name and version
- ***MicoApplicationDoesNotIncludeMicoServiceException*** – if there is no service included in the specified *MicoApplication* with the particular short name
- ***MicoServiceDeploymentInformationNotFoundException*** – if there is no *MicoServiceDeploymentInfo* stored in the database
- ***KubernetesResourceException*** – if there are problems with retrieving Kubernetes resource information

- *MicoTopicRoleUsedMultipleTimesException* – if a *MicoTopicRole* is used multiple times

Returns the new *MicoServiceDeploymentInfo* stored in the database

MicoServiceInterfaceBroker

```
public class MicoServiceInterfaceBroker
```

Methods

deleteMicoServiceInterface

```
public void deleteMicoServiceInterface (MicoService micoService, String serviceInterfaceName)
```

getInterfaceOfServiceByName

```
public MicoServiceInterface getInterfaceOfServiceByName (String shortName, String version, String interfaceName)
```

getInterfacesOfService

```
public List<MicoServiceInterface> getInterfacesOfService (String shortName, String version)
```

persistMicoServiceInterface

```
public MicoServiceInterface persistMicoServiceInterface (MicoService micoService, MicoServiceInterface micoServiceInterface)
```

updateMicoServiceInterface

```
public MicoServiceInterface updateMicoServiceInterface (MicoService micoService, String serviceInterfaceName, MicoServiceInterface micoServiceInterface)
```

OpenFaasBroker

```
public class OpenFaasBroker
```

Fields

OPEN_FAAS_UI_PROTOCOL

```
public static final String OPEN_FAAS_UI_PROTOCOL  
    The supported protocol of the OpenFaaS UI
```

micoKubernetesClient

MicoKubernetesClient **micoKubernetesClient**

micoKubernetesConfig

MicoKubernetesConfig **micoKubernetesConfig**

openFaaSConfig

OpenFaaSConfig **openFaaSConfig**

Methods

getExternalAddress

public [Optional](#)<[URL](#)> **getExternalAddress** ()

Requests the external address of the OpenFaaS UI and returns it or `null` if OpenFaaS does not exist.

Throws

- **MalformedURLException** – if the address is not in the URL format.
- [KubernetesResourceException](#) – if the IP address or the ports of the external gateway svc can't be retrieved

Returns the external address of the OpenFaaS UI or `null`.

TopicBroker

public class **TopicBroker**

Methods

getAllTopics

public [List](#)<[String](#)> **getAllTopics** ()

Reads all topics from the topic repository

2.2.3 io.github.ust.mico.core.configuration

CorsConfig

public class **CorsConfig**

KafkaConfig

public class **KafkaConfig**

Configuration of the Kafka connection.

Methods

getDefaultEnvironmentVariablesForKafka

```
public List<MicoEnvironmentVariable> getDefaultEnvironmentVariablesForKafka ()
```

getDefaultTopics

```
public List<MicoTopicRole> getDefaultTopics (MicoServiceDeploymentInfo sdi)
```

KafkaFaasConnectorConfig

```
public class KafkaFaasConnectorConfig  
    Configuration of the KafkaFaasConnector deployment.
```

MicoKubernetesBuildBotConfig

```
public class MicoKubernetesBuildBotConfig  
    Configuration of the build bot (TektonPipelinesController)
```

MicoKubernetesConfig

```
public class MicoKubernetesConfig  
    Configuration that includes information about the MICO Kubernetes cluster
```

MicoRedisConfiguration

```
public class MicoRedisConfiguration  
    Configuration that includes information about Redis Database
```

Methods

redisConnectionFactory

```
public LettuceConnectionFactory redisConnectionFactory ()
```

redisTemplate

```
public RedisTemplate<?, ?> redisTemplate ()
```

OpenFaaSConfig

```
public class OpenFaaSConfig  
    Configuration for the OpenFaaS connection.
```

Methods

getDefaultEnvironmentVariablesForOpenFaaS

```
public List<MicoEnvironmentVariable> getDefaultEnvironmentVariablesForOpenFaaS ()
```

PrometheusConfig

```
public class PrometheusConfig
```

RequestLoggingFilterConfig

```
public class RequestLoggingFilterConfig
```

Methods

logFilter

```
public Filter logFilter ()
```

RequestLoggingFilterConfig.MicoRequestLoggingFilter

```
public static class MicoRequestLoggingFilter extends CommonsRequestLoggingFilter
```

Methods

shouldNotFilter

```
protected boolean shouldNotFilter (HttpServletRequest request)
```

SwaggerConfig

```
public class SwaggerConfig
```

Methods

api

```
public Docket api ()
```

WebConfig

```
public class WebConfig extends WebMvcConfigurerAdapter
```

Fields

`corsUserConfig`

CorsConfig `corsUserConfig`

Methods

`corsFilter`

public FilterRegistrationBean<CorsFilter> **corsFilter** ()

Based on <https://github.com/springfox/springfox/issues/2215#issuecomment-446178059>

2.2.4 io.github.ust.mico.core.configuration.extension

`CustomOpenApiExtentionsPlugin`

public class **CustomOpenApiExtentionsPlugin** implements ModelPropertyBuilderPlugin

Fields

`X_MICO_CUSTOM_EXTENSION`

public static final String **X_MICO_CUSTOM_EXTENSION**

Methods

`apply`

public void **apply** (ModelPropertyContext *context*)

`supports`

public boolean **supports** (DocumentationType *delimiter*)

2.2.5 io.github.ust.mico.core.dto.request

`CrawlingInfoRequestDTO`

public class **CrawlingInfoRequestDTO**

DTO for the information needed by a Crawler (e.g., *GitHubCrawler*) for crawling a service from a remote repository intended to use with requests only.

KFConnectorDeploymentInfoRequestDTO

public class **KFConnectorDeploymentInfoRequestDTO**

DTO for *MicoServiceDeploymentInfo* specialised for a KafkaFaasConnector intended to use with requests only.

Constructors

KFConnectorDeploymentInfoRequestDTO

public **KFConnectorDeploymentInfoRequestDTO** (*MicoServiceDeploymentInfo* kfConnectorDeploymentInfo)

Creates an instance of KFConnectorDeploymentInfoRequestDTO based on a MicoServiceDeploymentInfo.

Parameters

- **kfConnectorDeploymentInfo** – the *MicoServiceDeploymentInfo*.

MicoApplicationRequestDTO

public class **MicoApplicationRequestDTO**

DTO for a *MicoApplication* intended to use with requests only. Note that the services are not included.

Constructors

MicoApplicationRequestDTO

public **MicoApplicationRequestDTO** (*MicoApplication* application)

Creates an instance of MicoApplicationRequestDTO based on a MicoApplication.

Parameters

- **application** – the *MicoApplication*.

MicoEnvironmentVariableRequestDTO

public class **MicoEnvironmentVariableRequestDTO**

DTO for a *MicoEnvironmentVariable* intended to use with requests only.

Constructors

MicoEnvironmentVariableRequestDTO

public **MicoEnvironmentVariableRequestDTO** (*MicoEnvironmentVariable* environmentVariable)

Creates an instance of MicoEnvironmentVariableRequestDTO based on a MicoEnvironmentVariable.

Parameters

- **environmentVariable** – the *MicoEnvironmentVariable*.

MicoInterfaceConnectionRequestDTO

public class **MicoInterfaceConnectionRequestDTO**

DTO for the information needed to connect a *MicoService* to an *MicoServiceInterface* of another *MicoService* intended to use with requests only.

Constructors

MicoInterfaceConnectionRequestDTO

public **MicoInterfaceConnectionRequestDTO** (*MicoInterfaceConnection* interfaceConnection)

Creates an instance of MicoInterfaceConnectionRequestDTO based on a MicoInterfaceConnection.

Parameters

- **interfaceConnection** – the *MicoInterfaceConnection*.

MicoLabelRequestDTO

public class **MicoLabelRequestDTO**

DTO for a *MicoLabel* intended to use with requests only.

Constructors

MicoLabelRequestDTO

public **MicoLabelRequestDTO** (*MicoLabel* label)

Creates an instance of MicoLabelRequestDTO based on a MicoLabel.

Parameters

- **label** – the *MicoLabel*.

MicoServiceDeploymentInfoRequestDTO

public class **MicoServiceDeploymentInfoRequestDTO**

DTO for *MicoServiceDeploymentInfo* intended to use with requests only.

Constructors

MicoServiceDeploymentInfoRequestDTO

public **MicoServiceDeploymentInfoRequestDTO** (*MicoServiceDeploymentInfo* serviceDeploymentInfo)

Creates an instance of MicoServiceDeploymentInfoRequestDTO based on a MicoServiceDeploymentInfo.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*.

MicoServiceInterfaceRequestDTO

public class **MicoServiceInterfaceRequestDTO**
DTO for a *MicoServiceInterface* intended to use with requests only.

Constructors

MicoServiceInterfaceRequestDTO

public **MicoServiceInterfaceRequestDTO** (*MicoServiceInterface* serviceInterface)
Creates an instance of MicoServiceInterfaceRequestDTO based on a MicoServiceInterface.

Parameters

- **serviceInterface** – the *MicoServiceInterface*.

MicoServicePortRequestDTO

public class **MicoServicePortRequestDTO**
DTO for a *MicoServicePort* intended to use with requests only.

Constructors

MicoServicePortRequestDTO

public **MicoServicePortRequestDTO** (*MicoServicePort* servicePort)
Creates an instance of MicoServicePortRequestDTO based on a MicoServicePort.

Parameters

- **servicePort** – the *MicoServicePort*.

MicoServiceRequestDTO

public class **MicoServiceRequestDTO**
DTO for a *MicoService* intended to use with requests only. Note that the *MicoServiceDependencies* and *MicoServiceInterfaces* are not included.

Constructors

MicoServiceRequestDTO

public **MicoServiceRequestDTO** (*MicoService* service)
Creates an instance of MicoServiceRequestDTO based on a MicoService.

Parameters

- **service** – the *MicoService*.

MicoTopicRequestDTO

public class **MicoTopicRequestDTO**

Constructors

MicoTopicRequestDTO

public **MicoTopicRequestDTO** (*MicoTopicRole* *micoTopicRole*)

Creates an instance of MicoTopicRequestDTO based on a MicoTopicRole that includes the MicoTopic and a role.

Parameters

- **micoTopicRole** – the *MicoTopicRole*.

MicoVersionRequestDTO

public class **MicoVersionRequestDTO**

DTO for a version intended to use with requests only, e.g., with a request to promote a new version of a *MicoApplication*.

Constructors

MicoVersionRequestDTO

public **MicoVersionRequestDTO** (*String* *version*)

Creates an instance of MicoVersionRequestDTO based on the String value *version*.

Parameters

- **version** – the version.

2.2.6 io.github.ust.mico.core.dto.response

ExternalUrlDTO

public class **ExternalUrlDTO**

Fields

externalUrl

URL **externalUrl**

An url for a service which is externally reachable.

isExternalUrlAvailable

boolean **isExternalUrlAvailable**

Is true if the url is available.

KFConnectorDeploymentInfoResponseDTO

public class **KFConnectorDeploymentInfoResponseDTO** extends *KFConnectorDeploymentInfoRequestDTO*
DTO for *MicoServiceDeploymentInfo* intended to use with responses only.

Constructors

KFConnectorDeploymentInfoResponseDTO

public **KFConnectorDeploymentInfoResponseDTO** (*MicoServiceDeploymentInfo* *kfConnectorDeploymentInfo*)
Creates an instance of **KFConnectorDeploymentInfoResponseDTO** based on a *MicoServiceDeploymentInfo*.

Parameters

- **kfConnectorDeploymentInfo** – the *MicoServiceDeploymentInfo*.

KubernetesDeploymentInfoResponseDTO

public class **KubernetesDeploymentInfoResponseDTO**
DTO for *KubernetesDeploymentInfo* intended to use with responses only.

Constructors

KubernetesDeploymentInfoResponseDTO

public **KubernetesDeploymentInfoResponseDTO** (*KubernetesDeploymentInfo* *kubernetesDeploymentInfo*)
Creates an instance of **KubernetesDeploymentInfoResponseDTO** based on a *KubernetesDeploymentInfo*.

Parameters

- **kubernetesDeploymentInfo** – the *KubernetesDeploymentInfo*.

MicoApplicationJobStatusResponseDTO

public class **MicoApplicationJobStatusResponseDTO**
DTO for a *MicoApplicationJobStatus* intended to use with responses only.

Constructors

MicoApplicationJobStatusResponseDTO

public **MicoApplicationJobStatusResponseDTO** (*MicoApplicationJobStatus* *applicationJobStatus*)
Creates a **MicoApplicationJobStatusDTO** based on a *MicoApplicationJobStatus*.

Parameters

- **applicationJobStatus** – the *MicoApplicationJobStatus*.

MicoApplicationResponseDTO

public class **MicoApplicationResponseDTO** extends *MicoApplicationRequestDTO*
DTO for a *MicoApplication* intended to use with responses only. Note that neither the services nor their deployment information is included. Contains the current deployment status of this application (may be unknown).

Constructors

MicoApplicationResponseDTO

public **MicoApplicationResponseDTO** (*MicoApplication* application)
Creates an instance of MicoApplicationResponseDTO based on a MicoApplication. Note that the deployment status is not set since it cannot be inferred from the MicoApplication itself

Parameters

- **application** – the *MicoApplication*.

MicoApplicationResponseDTO

public **MicoApplicationResponseDTO** (*MicoApplication* application, *MicoApplicationDeploymentStatus* deploymentStatus)
Creates an instance of MicoApplicationResponseDTO based on a MicoApplication and a MicoApplicationDeploymentStatus.

Parameters

- **application** – the *MicoApplication*.
- **deploymentStatus** – the *MicoApplicationDeploymentStatus*.

MicoApplicationWithServicesResponseDTO

public class **MicoApplicationWithServicesResponseDTO** extends *MicoApplicationResponseDTO*
DTO for a *MicoApplication* intended to use with responses only. Additionally includes all of services of the application.

Constructors

MicoApplicationWithServicesResponseDTO

public **MicoApplicationWithServicesResponseDTO** (*MicoApplication* application)
Creates an instance of MicoApplicationWithServicesResponseDTO based on a MicoApplication. Note that the deployment status is not set since it cannot be inferred from the MicoApplication itself

Parameters

- **application** – the *MicoApplication*.

MicoApplicationWithServicesResponseDTO

public **MicoApplicationWithServicesResponseDTO** (*MicoApplication* application, *MicoApplicationDeploymentStatus* deploymentStatus)
Creates an instance of MicoApplicationWithServicesResponseDTO based on a MicoApplication and a MicoApplicationDeploymentStatus.

Parameters

- **application** – the *MicoApplication*.
- **deploymentStatus** – the *MicoApplicationDeploymentStatus*.

MicoEnvironmentVariableResponseDTO

public class **MicoEnvironmentVariableResponseDTO** extends *MicoEnvironmentVariableRequestDTO*
DTO for a *MicoEnvironmentVariable* intended to use with responses only.

Constructors

MicoEnvironmentVariableResponseDTO

public **MicoEnvironmentVariableResponseDTO** (*MicoEnvironmentVariable* environmentVariable)
Creates an instance of MicoEnvironmentVariableResponseDTO based on a MicoEnvironmentVariable.

Parameters

- **environmentVariable** – the *MicoEnvironmentVariable*.

MicoInterfaceConnectionResponseDTO

public class **MicoInterfaceConnectionResponseDTO** extends *MicoInterfaceConnectionRequestDTO*
DTO for a *MicoInterfaceConnection* intended to use with responses only.

Constructors

MicoInterfaceConnectionResponseDTO

public **MicoInterfaceConnectionResponseDTO** (*MicoInterfaceConnection* interfaceConnection)
Creates an instance of MicoInterfaceConnectionResponseDTO based on a MicoInterfaceConnection.

Parameters

- **interfaceConnection** – the *MicoInterfaceConnection*.

MicoLabelResponseDTO

public class **MicoLabelResponseDTO** extends *MicoLabelRequestDTO*
DTO for a *MicoLabel* intended to use with responses only.

Constructors

MicoLabelResponseDTO

public **MicoLabelResponseDTO** (*MicoLabel* label)

Creates an instance of *MicoLabelResponseDTO* based on a *MicoLabel*.

Parameters

- **label** – the *MicoLabel*.

MicoServiceBackgroundJobResponseDTO

public class **MicoServiceBackgroundJobResponseDTO**

DTO for a *MicoServiceBackgroundJob* intended to use with responses only.

Constructors

MicoServiceBackgroundJobResponseDTO

public **MicoServiceBackgroundJobResponseDTO** (*MicoServiceBackgroundJob* serviceBackgroundJob)

Creates a *MicoBackgroundJobResponseDTO* based on a *MicoServiceBackgroundJob*.

Parameters

- **serviceBackgroundJob** – the *MicoServiceBackgroundJob*.

Return a *MicoServiceBackgroundJobResponseDTO* with all the values of the given *MicoServiceBackgroundJob*.

MicoServiceDependencyGraphEdgeResponseDTO

public class **MicoServiceDependencyGraphEdgeResponseDTO**

DTO for the edge of the dependency graph of a *MicoService*.

Constructors

MicoServiceDependencyGraphEdgeResponseDTO

public **MicoServiceDependencyGraphEdgeResponseDTO** (*MicoService* source, *MicoService* target)

Creates an instance of *MicoServiceDependencyGraphEdgeResponseDTO* based on a source *MicoService* and a target *MicoService*.

Parameters

- **source** – the source *MicoService*.
- **target** – the target *MicoService*.

MicoServiceDependencyGraphResponseDTO

public class **MicoServiceDependencyGraphResponseDTO**
DTO for the dependency graph of a *MicoService*.

MicoServiceDeploymentInfoResponseDTO

public class **MicoServiceDeploymentInfoResponseDTO** extends *MicoServiceDeploymentInfoRequestDTO*
DTO for *MicoServiceDeploymentInfo* intended to use with responses only.

Constructors

MicoServiceDeploymentInfoResponseDTO

public **MicoServiceDeploymentInfoResponseDTO** (*MicoServiceDeploymentInfo* serviceDeploymentInfo)
Creates an instance of *MicoServiceDeploymentInfoResponseDTO* based on a *MicoServiceDeploymentInfo*.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*.

MicoServiceInterfaceResponseDTO

public class **MicoServiceInterfaceResponseDTO** extends *MicoServiceInterfaceRequestDTO*
DTO for a *MicoServiceInterface* intended to use with responses only.

Constructors

MicoServiceInterfaceResponseDTO

public **MicoServiceInterfaceResponseDTO** (*MicoServiceInterface* serviceInterface)
Creates an instance of *MicoServiceInterfaceResponseDTO* based on a *MicoServiceInterface*.

Parameters

- **serviceInterface** – the *MicoServiceInterface*.

MicoServicePortResponseDTO

public class **MicoServicePortResponseDTO** extends *MicoServicePortRequestDTO*
DTO for a *MicoServicePort* intended to use with responses only.

Constructors

MicoServicePortResponseDTO

public **MicoServicePortResponseDTO** (*MicoServicePort* servicePort)
Creates an instance of *MicoServicePortResponseDTO* based on a *MicoServicePort*.

Parameters

- **servicePort** – the *MicoServicePort*.

MicoServiceResponseDTO

public class **MicoServiceResponseDTO** extends *MicoServiceRequestDTO*
DTO for a *MicoService* intended for use with responses only. Note that the
MicoServiceDependencies and *MicoServiceInterfaces* are not included.

Constructors

MicoServiceResponseDTO

public **MicoServiceResponseDTO** (*MicoService* service)
Creates an instance of MicoServiceResponseDTO based on a MicoService.

Parameters

- **service** – the *MicoService*.

MicoTopicResponseDTO

public class **MicoTopicResponseDTO** extends *MicoTopicRequestDTO*
DTO for a *MicoTopic* for response only use

Constructors

MicoTopicResponseDTO

public **MicoTopicResponseDTO** (*MicoTopicRole* micoTopicRole)
Creates an instance of MicoTopicResponseDTO based on a MicoTopicRole.

Parameters

- **micoTopicRole** – *MicoTopicRole*.

MicoYamlResponseDTO

public class **MicoYamlResponseDTO**

Fields

yaml

String **yaml**
Yaml from Kubernetes resource

TopicDTO

```
public class TopicDTO
```

Fields

name

`String` **name**

The name of the topic

ValidationErrorResponseDTO

```
public class ValidationErrorResponseDTO
```

Fields

HTTP_STATUS

```
public static final HttpStatus HTTP_STATUS
```

HTTP status code that is used for validation errors.

“422 Unprocessable Entity” is used instead of “400 Bad Request” because the request was syntactically correct, but semantically incorrect.

Constructors

ValidationErrorResponseDTO

```
public ValidationErrorResponseDTO (List<FieldError> fieldErrorList)
```

2.2.7 io.github.ust.mico.core.dto.response.internal

PrometheusResponseDTO

```
public class PrometheusResponseDTO
```

Internal DTO for a response from Prometheus. It contains a status field and the value field for the CPU load / memory usage.

2.2.8 io.github.ust.mico.core.dto.response.status

KubernetesNodeMetricsResponseDTO

```
public class KubernetesNodeMetricsResponseDTO
```

DTO for the average CPU load and the average memory usage of all Pods running on a Kubernetes Node intended to use with responses only.

KubernetesPodInformationResponseDTO

public class **KubernetesPodInformationResponseDTO**

Represents basic information for a Pod in Kubernetes intended to use with responses only..

KubernetesPodMetricsResponseDTO

public class **KubernetesPodMetricsResponseDTO**

Contains information about CPU/memory load of a Pod intended to use with responses only..

MicoApplicationDeploymentStatusResponseDTO

public class **MicoApplicationDeploymentStatusResponseDTO**

DTO for a *MicoApplicationDeploymentStatus* intended to use with responses only.

Constructors

MicoApplicationDeploymentStatusResponseDTO

public **MicoApplicationDeploymentStatusResponseDTO** (*MicoApplicationDeploymentStatus* *applicationDeploymentStatus*)

Creates an instance of *MicoApplicationDeploymentStatusResponseDTO* based on a *MicoApplicationDeploymentStatus*.

Parameters

- **applicationDeploymentStatus** – the *applicationDeploymentStatus*.

MicoApplicationStatusResponseDTO

public class **MicoApplicationStatusResponseDTO**

DTO for the status information of a *MicoApplication* intended to use with responses only.

MicoMessageResponseDTO

public class **MicoMessageResponseDTO**

DTO for a *MicoMessage* intended to use with responses only.

Constructors

MicoMessageResponseDTO

public **MicoMessageResponseDTO** (*MicoMessage* *message*)

Creates an instance of *MicoMessageResponseDTO* based on a *MicoMessage*.

Parameters

- **message** – the *message*.

MicoServiceInterfaceStatusResponseDTO

public class **MicoServiceInterfaceStatusResponseDTO**

DTO for the status information of a *MicoServiceInterface* that is mapped to a Kubernetes Service intended to use with responses only.

MicoServiceStatusResponseDTO

public class **MicoServiceStatusResponseDTO**

DTO for the status information of a *MicoService* intended to use with responses only..

2.2.9 io.github.ust.mico.core.exception

DeploymentException

public class **DeploymentException** extends *Exception*

Constructors

DeploymentException

public **DeploymentException** ()

DeploymentException

public **DeploymentException** (*String message*)

DeploymentRequirementsOfKafkaFaasConnectorNotMetException

public class **DeploymentRequirementsOfKafkaFaasConnectorNotMetException** extends *Exception*

Constructors

DeploymentRequirementsOfKafkaFaasConnectorNotMetException

public **DeploymentRequirementsOfKafkaFaasConnectorNotMetException** (*MicoServiceDeploymentInfo serviceDeploymentInfo, String reason*)

DeploymentRequirementsOfKafkaFaasConnectorNotMetException

public **DeploymentRequirementsOfKafkaFaasConnectorNotMetException** (*MicoServiceDeploymentInfo serviceDeploymentInfo*)

DeploymentRequirementsOfKafkaFaasConnectorNotMetException

```
public DeploymentRequirementsOfKafkaFaasConnectorNotMetException ()
```

GlobalExceptionHandler

```
public class GlobalExceptionHandler
```

Global exception handler

Inspired by Petri Kainulainen <https://www.petrikainulainen.net/programming/spring-framework/spring-from-the-trenches-adding-validation-to-a-rest-api/>

Methods

methodArgumentNotValidException

```
public ValidationErrorResponseDTO methodArgumentNotValidException (MethodArgumentNotValidException  
ex)
```

ImageBuildException

```
public class ImageBuildException extends Exception
```

Constructors

ImageBuildException

```
public ImageBuildException ()
```

ImageBuildException

```
public ImageBuildException (String message)
```

KafkaFaasConnectorInstanceNotFoundException

```
public class KafkaFaasConnectorInstanceNotFoundException extends Exception
```

Constructors

KafkaFaasConnectorInstanceNotFoundException

```
public KafkaFaasConnectorInstanceNotFoundException (String instanceId, String version)
```

KafkaFaasConnectorInstanceNotFoundException

```
public KafkaFaasConnectorInstanceNotFoundException (String instanceId)
```

KafkaFaasConnectorInstanceNotFoundException

```
public KafkaFaasConnectorInstanceNotFoundException ()
```

KafkaFaasConnectorLatestVersionNotFound

```
public class KafkaFaasConnectorLatestVersionNotFound extends Exception
```

Constructors

KafkaFaasConnectorLatestVersionNotFound

```
public KafkaFaasConnectorLatestVersionNotFound ()
```

KafkaFaasConnectorNotAllowedHereException

```
public class KafkaFaasConnectorNotAllowedHereException extends Exception
```

Constructors

KafkaFaasConnectorNotAllowedHereException

```
public KafkaFaasConnectorNotAllowedHereException ()
```

KafkaFaasConnectorVersionNotFoundException

```
public class KafkaFaasConnectorVersionNotFoundException extends Exception
```

Constructors

KafkaFaasConnectorVersionNotFoundException

```
public KafkaFaasConnectorVersionNotFoundException (String version)
```

KafkaFaasConnectorVersionNotFoundException

```
public KafkaFaasConnectorVersionNotFoundException ()
```

KubernetesResourceException

```
public class KubernetesResourceException extends Exception
```

Used to indicate that there is a problem concerning a Kubernetes resource, e.g., a Deployment cannot be found or there are multiple results for a query for a resource that is expected to be unique.

Constructors

KubernetesResourceException

```
public KubernetesResourceException ()
```

KubernetesResourceException

```
public KubernetesResourceException (String message)
```

KubernetesResourceException

```
public KubernetesResourceException (Throwable cause)
```

KubernetesResourceException

```
public KubernetesResourceException (String message, Throwable cause)
```

MicoApplicationAlreadyExistsException

```
public class MicoApplicationAlreadyExistsException extends Exception
```

Constructors

MicoApplicationAlreadyExistsException

```
public MicoApplicationAlreadyExistsException (String shortName, String version)
```

MicoApplicationAlreadyExistsException

```
public MicoApplicationAlreadyExistsException (Long id)
```

MicoApplicationAlreadyExistsException

```
public MicoApplicationAlreadyExistsException ()
```

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public class MicoApplicationDoesNotIncludeKFConnectorInstanceException extends Exception
```

Constructors

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public MicoApplicationDoesNotIncludeKFConnectorInstanceException (String applicationShortName,  
                                                                    String applicationVersion,  
                                                                    String instanceId, String kfConnectorVersion)
```

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public MicoApplicationDoesNotIncludeKFConnectorInstanceException (String applicationShortName,  
                                                                    String applicationVersion,  
                                                                    String instanceId)
```

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public MicoApplicationDoesNotIncludeKFConnectorInstanceException (String applicationShortName,  
                                                                    String applicationVersion)
```

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public MicoApplicationDoesNotIncludeKFConnectorInstanceException (Long applicationId, Long serviceId)
```

MicoApplicationDoesNotIncludeKFConnectorInstanceException

```
public MicoApplicationDoesNotIncludeKFConnectorInstanceException ()
```

MicoApplicationDoesNotIncludeMicoServiceException

```
public class MicoApplicationDoesNotIncludeMicoServiceException extends Exception
```

Constructors

MicoApplicationDoesNotIncludeMicoServiceException

```
public MicoApplicationDoesNotIncludeMicoServiceException (String applicationShort-  
Name, String applica-  
tionVersion, String ser-  
viceShortName, String  
serviceVersion)
```

MicoApplicationDoesNotIncludeMicoServiceException

```
public MicoApplicationDoesNotIncludeMicoServiceException (String applicationShort-  
Name, String applica-  
tionVersion, String serviceShort-  
Name)
```

MicoApplicationDoesNotIncludeMicoServiceException

```
public MicoApplicationDoesNotIncludeMicoServiceException (String applicationShort-  
Name, String applicationVer-  
sion)
```

MicoApplicationDoesNotIncludeMicoServiceException

```
public MicoApplicationDoesNotIncludeMicoServiceException (Long applicationId, Long  
serviceId)
```

MicoApplicationDoesNotIncludeMicoServiceException

```
public MicoApplicationDoesNotIncludeMicoServiceException ()
```

MicoApplicationIsDeployingException

```
public class MicoApplicationIsDeployingException extends Exception
```

Constructors

MicoApplicationIsDeployingException

```
public MicoApplicationIsDeployingException (String shortName, String version)
```

MicoApplicationIsDeployingException

```
public MicoApplicationIsDeployingException ()
```

MicoApplicationIsNotUndeployedException

```
public class MicoApplicationIsNotUndeployedException extends Exception
```


Constructors

MicoApplicationIsNotUndeployedException

```
public MicoApplicationIsNotUndeployedException (String shortName, String version)
```

MicoApplicationIsNotUndeployedException

```
public MicoApplicationIsNotUndeployedException ()
```

MicoApplicationNotFoundException

```
public class MicoApplicationNotFoundException extends Exception
```

Constructors

MicoApplicationNotFoundException

```
public MicoApplicationNotFoundException (String shortName, String version)
```

MicoApplicationNotFoundException

```
public MicoApplicationNotFoundException (String shortName)
```

MicoApplicationNotFoundException

```
public MicoApplicationNotFoundException (Long id)
```

MicoApplicationNotFoundException

```
public MicoApplicationNotFoundException ()
```

MicoServiceAddedMoreThanOnceToMicoApplicationException

```
public class MicoServiceAddedMoreThanOnceToMicoApplicationException extends Exception
```

Constructors

MicoServiceAddedMoreThanOnceToMicoApplicationException

```
public MicoServiceAddedMoreThanOnceToMicoApplicationException (String application-  
ShortName, String  
applicationVersion,  
String serviceShort-  
Name, String ser-  
viceVersion)
```

MicoServiceAddedMoreThanOnceToMicoApplicationException

```
public MicoServiceAddedMoreThanOnceToMicoApplicationException (String applicationShortName, String applicationVersion)
```

MicoServiceAddedMoreThanOnceToMicoApplicationException

```
public MicoServiceAddedMoreThanOnceToMicoApplicationException ()
```

MicoServiceAlreadyAddedToMicoApplicationException

```
public class MicoServiceAlreadyAddedToMicoApplicationException extends Exception
```

Constructors

MicoServiceAlreadyAddedToMicoApplicationException

```
public MicoServiceAlreadyAddedToMicoApplicationException (String applicationShortName, String applicationVersion, String serviceShortName, String serviceVersion)
```

MicoServiceAlreadyAddedToMicoApplicationException

```
public MicoServiceAlreadyAddedToMicoApplicationException (Long applicationId, Long serviceId)
```

MicoServiceAlreadyAddedToMicoApplicationException

```
public MicoServiceAlreadyAddedToMicoApplicationException ()
```

MicoServiceAlreadyExistsException

```
public class MicoServiceAlreadyExistsException extends Exception
```

Constructors

MicoServiceAlreadyExistsException

```
public MicoServiceAlreadyExistsException (String shortName, String version)
```

MicoServiceAlreadyExistsException

```
public MicoServiceAlreadyExistsException (Long id)
```

MicoServiceAlreadyExistsException

```
public MicoServiceAlreadyExistsException ()
```

MicoServiceDeploymentInformationNotFoundException

```
public class MicoServiceDeploymentInformationNotFoundException extends Exception
```

Constructors

MicoServiceDeploymentInformationNotFoundException

```
public MicoServiceDeploymentInformationNotFoundException (String applicationShort-  
Name, String application-  
Version, String serviceShort-  
Name)
```

MicoServiceDeploymentInformationNotFoundException

```
public MicoServiceDeploymentInformationNotFoundException (String applicationShort-  
Name, String applica-  
tionVersion, String ser-  
viceShortName, String  
serviceVersion)
```

MicoServiceDeploymentInformationNotFoundException

```
public MicoServiceDeploymentInformationNotFoundException (Long applicationId, Long  
serviceId)
```

MicoServiceHasDependersException

```
public class MicoServiceHasDependersException extends Exception
```

Constructors

MicoServiceHasDependersException

```
public MicoServiceHasDependersException (String shortName, String version)
```

MicoServiceHasDependersException

```
public MicoServiceHasDependersException (Long id)
```

MicoServiceHasDependersException

```
public MicoServiceHasDependersException ()
```

MicoServiceInstanceDoesNotMatchShortNameAndVersionException

```
public class MicoServiceInstanceDoesNotMatchShortNameAndVersionException extends Exception
```

Constructors

MicoServiceInstanceDoesNotMatchShortNameAndVersionException

```
public MicoServiceInstanceDoesNotMatchShortNameAndVersionException (String instanceId, String providedShortName, String providedVersion, String actualShortName, String actualVersion)
```

MicoServiceInstanceNotFoundException

```
public class MicoServiceInstanceNotFoundException extends Exception
```

Constructors

MicoServiceInstanceNotFoundException

```
public MicoServiceInstanceNotFoundException (String shortName, String version, String instanceId)
```

MicoServiceInstanceNotFoundException

```
public MicoServiceInstanceNotFoundException ()
```

MicoServiceInterfaceAlreadyExistsException

```
public class MicoServiceInterfaceAlreadyExistsException extends Exception
```

Constructors

MicoServiceInterfaceAlreadyExistsException

```
public MicoServiceInterfaceAlreadyExistsException (String serviceName, String serviceVersion, String serviceInterfaceName)
```

MicoServiceInterfaceAlreadyExistsException

```
public MicoServiceInterfaceAlreadyExistsException (Long serviceId, String serviceInterfaceName)
```

MicoServiceInterfaceAlreadyExistsException

```
public MicoServiceInterfaceAlreadyExistsException ()
```

MicoServiceInterfaceNotFoundException

```
public class MicoServiceInterfaceNotFoundException extends Exception
```

Constructors

MicoServiceInterfaceNotFoundException

```
public MicoServiceInterfaceNotFoundException (String shortName, String version, String interfaceName)
```

MicoServiceInterfaceNotFoundException

```
public MicoServiceInterfaceNotFoundException (String shortName, String version)
```

MicoServicesDeployedException

```
public class MicoServiceIsDeployedException extends Exception
```

Constructors

MicoServicesDeployedException

```
public MicoServiceIsDeployedException (String shortName, String version)
```

MicoServicesDeployedException

```
public MicoServiceIsDeployedException (String shortName)
```

MicoServiceIsDeployedException

```
public MicoServiceIsDeployedException (Long id)
```

MicoServiceIsDeployedException

```
public MicoServiceIsDeployedException ()
```

MicoServiceIsUsedByMicoApplicationsException

```
public class MicoServiceIsUsedByMicoApplicationsException extends Exception
```

Constructors

MicoServiceIsUsedByMicoApplicationsException

```
public MicoServiceIsUsedByMicoApplicationsException (String shortName, String version,  
List<MicoApplication> applications)
```

MicoServiceIsUsedByMicoApplicationsException

```
public MicoServiceIsUsedByMicoApplicationsException (String shortName, String version)
```

MicoServiceIsUsedByMicoApplicationsException

```
public MicoServiceIsUsedByMicoApplicationsException ()
```

MicoServiceNotFoundException

```
public class MicoServiceNotFoundException extends Exception
```

Constructors

MicoServiceNotFoundException

```
public MicoServiceNotFoundException (String shortName, String version)
```

MicoServiceNotFoundException

```
public MicoServiceNotFoundException (String shortName)
```

MicoServiceNotFoundException

```
public MicoServiceNotFoundException (Long id)
```

MicoServiceNotFoundException

```
public MicoServiceNotFoundException ()
```

MicoTopicRoleUsedMultipleTimesException

```
public class MicoTopicRoleUsedMultipleTimesException extends Exception
```

Constructors

MicoTopicRoleUsedMultipleTimesException

```
public MicoTopicRoleUsedMultipleTimesException (MicoTopicRole.Role role)
```

MicoTopicRoleUsedMultipleTimesException

```
public MicoTopicRoleUsedMultipleTimesException ()
```

NotInitializedException

```
public class NotInitializedException extends Exception
```

Constructors

NotInitializedException

```
public NotInitializedException ()
```

NotInitializedException

```
public NotInitializedException (String message)
```

PrometheusRequestFailedException

```
public class PrometheusRequestFailedException extends Exception
```

Fields

serialVersionUID

```
public static final long serialVersionUID
```

Constructors

PrometheusRequestFailedException

```
public PrometheusRequestFailedException (String message)
```

PrometheusRequestFailedException

```
public PrometheusRequestFailedException ()
```

ShortNameOfMicoApplicationDoesNotMatchException

```
public class ShortNameOfMicoApplicationDoesNotMatchException extends Exception
```

Constructors

ShortNameOfMicoApplicationDoesNotMatchException

```
public ShortNameOfMicoApplicationDoesNotMatchException ()
```

VersionNotSupportedException

```
public class VersionNotSupportedException extends Exception  
    Indicates that a certain version is not supported.
```

Constructors

VersionNotSupportedException

```
public VersionNotSupportedException ()
```

VersionNotSupportedException

```
public VersionNotSupportedException (String message)
```

VersionOfMicoApplicationDoesNotMatchException

```
public class VersionOfMicoApplicationDoesNotMatchException extends Exception
```

Constructors

VersionOfMicoApplicationDoesNotMatchException

```
public VersionOfMicoApplicationDoesNotMatchException ()
```


2.2.10 io.github.ust.mico.core.model

KubernetesDeploymentInfo

public class **KubernetesDeploymentInfo**

Information about the Kubernetes resources that are created through an actual deployment of a *MicoService*.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

applyValuesFrom

public *KubernetesDeploymentInfo* **applyValuesFrom** (*KubernetesDeploymentInfoResponseDTO* *kubernetesDeploymentInfoDto*)

Applies the values of all properties of a *KubernetesDeploymentInfoResponseDTO* to this *KubernetesDeploymentInfo*. Note that the id will not be affected.

Parameters

- **kubernetesDeploymentInfoDto** – the *KubernetesDeploymentInfoResponseDTO*.

Returns this *KubernetesDeploymentInfo* with the values of the properties of the given *KubernetesDeploymentInfoResponseDTO*.

valueOf

public static *KubernetesDeploymentInfo* **valueOf** (*KubernetesDeploymentInfoResponseDTO* *kubernetesDeploymentInfoDto*)

Creates a new *KubernetesDeploymentInfo* based on a *KubernetesDeploymentInfoResponseDTO*. Note that the id will be set to null.

Parameters

- **kubernetesDeploymentInfoDto** – the *KubernetesDeploymentInfoResponseDTO*.

Returns a *KubernetesDeploymentInfo*.

MicoApplication

public class **MicoApplication**

Represents an application as a set of *MicoServices* in the context of MICO.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

getMicoVersion

public *MicoVersion* **getMicoVersion** ()

valueOf

public static *MicoApplication* **valueOf** (*MicoApplicationRequestDTO* applicationDto)

Creates a new *MicoApplication* based on a *MicoApplicationRequestDTO*. Note that the id will be set to null.

Parameters

- **applicationDto** – the *MicoApplicationRequestDTO*.

Returns a *MicoApplication*.

MicoApplicationDeploymentStatus

public class **MicoApplicationDeploymentStatus**

Wraps the deployment status of a *MicoApplication* and some messages (optional) with more detailed information.

Note that this class is only used for business logic purposes and instances are not persisted.

Methods

deployed

public static final *MicoApplicationDeploymentStatus* **deployed** (String... messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.DEPLOYED* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – one or messages.

Returns a *MicoApplicationDeploymentStatus*.

deployed

public static final *MicoApplicationDeploymentStatus* **deployed** (List<String> messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.DEPLOYED* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – the *List* of messages as *String*.

Returns a *MicoApplicationDeploymentStatus*.

deployed

public static final *MicoApplicationDeploymentStatus* **deployed** (String message, Type messageType)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.DEPLOYED* as well as a *MicoMessage* with the given message content and type.

Parameters

- **message** – the content of the message.

- **messageType** – the *Type* of the message.

Returns a *MicoApplicationDeploymentStatus*.

incomplete

public static final *MicoApplicationDeploymentStatus* **incomplete** (String... messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.INCOMPLETE* as well as a *MicoMessage* with the given message content and type *Error*.

Parameters

- **messages** – one or messages.

Returns a *MicoApplicationDeploymentStatus*.

incomplete

public static final *MicoApplicationDeploymentStatus* **incomplete** (List<String> messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.INCOMPLETE* as well as a *MicoMessage* with the given message content and type *Error*.

Parameters

- **messages** – the List of messages as String.

Returns a *MicoApplicationDeploymentStatus*.

incomplete

public static final *MicoApplicationDeploymentStatus* **incomplete** (String message, Type messageType)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.INCOMPLETE* as well as a *MicoMessage* with the given message content and type.

Parameters

- **message** – the content of the message.
- **messageType** – the *Type* of the message.

Returns a *MicoApplicationDeploymentStatus*.

pending

public static final *MicoApplicationDeploymentStatus* **pending** (String... messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.PENDING* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – one or messages.

Returns a *MicoApplicationDeploymentStatus*.

pending

public static final *MicoApplicationDeploymentStatus* **pending** (List<String> messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.PENDING* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – the List of messages as String.

Returns a *MicoApplicationDeploymentStatus*.

pending

public static final *MicoApplicationDeploymentStatus* **pending** (String message, Type messageType)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.PENDING* as well as a *MicoMessage* with the given message content and type.

Parameters

- **message** – the content of the message.
- **messageType** – the *Type* of the message.

Returns a *MicoApplicationDeploymentStatus*.

undeployed

public static final *MicoApplicationDeploymentStatus* **undeployed** (String... messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.UNDEPLOYED* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – one or messages.

Returns a *MicoApplicationDeploymentStatus*.

undeployed

public static final *MicoApplicationDeploymentStatus* **undeployed** (List<String> messages)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.UNDEPLOYED* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – the List of messages as String.

Returns a *MicoApplicationDeploymentStatus*.

undeployed

public static final *MicoApplicationDeploymentStatus* **undeployed** (String message, Type messageType)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value.UNDEPLOYED* as well as a *MicoMessage* with the given message content and type.

Parameters

- **message** – the content of the message.
- **messageType** – the *Type* of the message.

Returns a *MicoApplicationDeploymentStatus*.

unknown

public static final *MicoApplicationDeploymentStatus* **unknown** (*String*... *messages*)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value*.*UNKNOWN* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – one or messages.

Returns a *MicoApplicationDeploymentStatus*.

unknown

public static final *MicoApplicationDeploymentStatus* **unknown** (*List*<*String*> *messages*)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value*.*UNKNOWN* as well as a *MicoMessage* with the given message content and type *Info*.

Parameters

- **messages** – the *List* of messages as *String*.

Returns a *MicoApplicationDeploymentStatus*.

unknown

public static final *MicoApplicationDeploymentStatus* **unknown** (*String* *message*, *Type* *messageType*)

Creates a new *MicoApplicationDeploymentStatus* instance with the value *Value*.*UNKNOWN* as well as a *MicoMessage* with the given message content and type.

Parameters

- **message** – the content of the message.
- **messageType** – the *Type* of the message.

Returns a *MicoApplicationDeploymentStatus*.

MicoApplicationDeploymentStatus.Value

public enum **Value**

Enumeration for the different values of a *MicoApplicationDeploymentStatus*.

Enum Constants

DEPLOYED

public static final *MicoApplicationDeploymentStatus.Value* **DEPLOYED**

Indicates that a *MicoApplication* with all its *MicoService* has been deployed successfully.

INCOMPLETE

public static final *MicoApplicationDeploymentStatus.Value* **INCOMPLETE**

Indicates that the deployment / undeployment of a *MicoApplication* did not complete due to at least one *MicoService* of the *MicoApplication* that couldn't be deployed / undeployed successfully.

PENDING

public static final *MicoApplicationDeploymentStatus.Value* **PENDING**

Indicates that a *MicoApplication* is currently being deployed / undeployed.

UNDEPLOYED

public static final *MicoApplicationDeploymentStatus.Value* **UNDEPLOYED**

Indicates that a *MicoApplication* with all its *MicoService* has been undeployed successfully.

UNKNOWN

public static final *MicoApplicationDeploymentStatus.Value* **UNKNOWN**

Indicates that the current deployment status of a *MicoApplication* is not known.

Methods

toString

public *String* **toString** ()

MicoApplicationJobStatus

public class **MicoApplicationJobStatus**

Represents the job status for a *MicoApplication*. Contains a list of jobs.

Note that this class is only used for business logic purposes and instances are not persisted.

MicoEnvironmentVariable

public class **MicoEnvironmentVariable**

An environment variable represented as a simple key-value pair. Necessary since Neo4j does not allow to persist properties of composite types.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

valueOf

```
public static MicoEnvironmentVariable valueOf(MicoEnvironmentVariableRequestDTO environmentVariableDto)
```

Creates a new *MicoEnvironmentVariable* based on a *MicoEnvironmentVariableRequestDTO*.
Note that the id will be set to null.

Parameters

- **environmentVariableDto** – the *MicoEnvironmentVariableRequestDTO*.

Returns a *MicoEnvironmentVariable*.

MicoEnvironmentVariable.DefaultNames

```
public enum DefaultNames
```

The default environment variables for a Kafka-enabled MicoServices.

Enum Constants

KAFKA_BOOTSTRAP_SERVERS

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_BOOTSTRAP_SERVERS
```

KAFKA_GROUP_ID

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_GROUP_ID
```

KAFKA_TOPIC_DEAD_LETTER

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_TOPIC_DEAD_LETTER
```

KAFKA_TOPIC_INPUT

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_TOPIC_INPUT
```

KAFKA_TOPIC_INVALID_MESSAGE

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_TOPIC_INVALID_MESSAGE
```

KAFKA_TOPIC_OUTPUT

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_TOPIC_OUTPUT
```

KAFKA_TOPIC_TEST_MESSAGE_OUTPUT

```
public static final MicoEnvironmentVariable.DefaultNames KAFKA_TOPIC_TEST_MESSAGE_OUTPUT
```

OPENFAAS_FUNCTION_NAME

public static final *MicoEnvironmentVariable.DefaultNames* **OPENFAAS_FUNCTION_NAME**

OPENFAAS_GATEWAY

public static final *MicoEnvironmentVariable.DefaultNames* **OPENFAAS_GATEWAY**

MicoInterfaceConnection

public class **MicoInterfaceConnection**

An interface connection contains the the information needed to connect a *MicoService* to an *MicoServiceInterface* of another *MicoService*.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

valueOf

public static *MicoInterfaceConnection* **valueOf** (*MicoInterfaceConnectionRequestDTO* *interfaceConnectionDTO*)

Creates a new *MicoInterfaceConnection* based on a *MicoInterfaceConnectionRequestDTO*.

Parameters

- **interfaceConnectionDTO** – the *MicoInterfaceConnectionRequestDTO*.

Returns a *MicoInterfaceConnection*.

MicoLabel

public class **MicoLabel**

A label represented as a simple key-value pair. Necessary since Neo4j does not allow to persist properties of composite types.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

valueOf

public static *MicoLabel* **valueOf** (*MicoLabelRequestDTO* *labelDto*)

Creates a new *MicoLabel* based on a *MicoLabelRequestDTO*. Note that the id will be set to null.

Parameters

- **labelDto** – the *MicoLabelRequestDTO*.

Returns a *MicoLabel*.

MicoMessage

public class **MicoMessage**

A simple message associated with a *Type*.

Note that this class is only used for business logic purposes and instances are not persisted.

Methods

error

public static final *MicoMessage* **error** (*String content*)

Creates a new MicoMessage instance with the type *Type.ERROR* and the given message content.

Parameters

- **content** – the message content as *String*.

Returns a *MicoMessage*.

info

public static final *MicoMessage* **info** (*String content*)

Creates a new MicoMessage instance with the type *Type.INFO* and the given message content.

Parameters

- **content** – the message content as *String*.

Returns a *MicoMessage*.

warning

public static final *MicoMessage* **warning** (*String content*)

Creates a new MicoMessage instance with the type *Type.WARNING* and the given message content.

Parameters

- **content** – the message content as *String*.

Returns a *MicoMessage*.

MicoMessage.Type

public enum **Type**

Enumeration for all types of a MicoInfoMessage.

Enum Constants

ERROR

public static final *MicoMessage.Type* **ERROR**

INFO

public static final *MicoMessage.Type* **INFO**

WARNING

public static final *MicoMessage.Type* **WARNING**

Methods

toString

public *String* **toString** ()

MicoPortType

public enum **MicoPortType**
Enumeration for all port types, e.g., TCP, supported by MICO.

Enum Constants

TCP

public static final *MicoPortType* **TCP**
Transmission Control Protocol.

UDP

public static final *MicoPortType* **UDP**
User Datagram Protocol.

MicoService

public class **MicoService**
Represents a service in the context of MICO.
Instances of this class are persisted as nodes in the Neo4j database.

Methods

getMicoVersion

public *MicoVersion* **getMicoVersion** ()

valueOf

public static *MicoService* **valueOf** (*MicoServiceRequestDTO* serviceDto)

Creates a new *MicoService* based on a *MicoServiceRequestDTO*. Note that the id will be set to null.

Parameters

- **serviceDto** – the *MicoServiceRequestDTO*.

Returns a *MicoService*.

MicoServiceBackgroundJob

public class **MicoServiceBackgroundJob** implements *Serializable*

Background job for a *MicoService*.

Instances of this class are persisted in the Redis database.

MicoServiceBackgroundJob.Status

public enum **Status**

Enum Constants

DONE

public static final *MicoServiceBackgroundJob.Status* **DONE**

ERROR

public static final *MicoServiceBackgroundJob.Status* **ERROR**

PENDING

public static final *MicoServiceBackgroundJob.Status* **PENDING**

RUNNING

public static final *MicoServiceBackgroundJob.Status* **RUNNING**

UNDEFINED

public static final *MicoServiceBackgroundJob.Status* **UNDEFINED**

MicoServiceBackgroundJob.Type

public enum **Type**

Enum Constants

BUILD

public static final *MicoServiceBackgroundJob.Type* **BUILD**

MicoServiceCrawlingOrigin

public enum **MicoServiceCrawlingOrigin**
Enumeration for the various places a service may originate from.

Enum Constants

DOCKER

public static final *MicoServiceCrawlingOrigin* **DOCKER**
Indicates that a service originates from Docker.

GITHUB

public static final *MicoServiceCrawlingOrigin* **GITHUB**
Indicates that a service originates from some GitHub repository.

NOT_DEFINED

public static final *MicoServiceCrawlingOrigin* **NOT_DEFINED**
Undefined.

MicoServiceDependency

public class **MicoServiceDependency**
Represents a dependency of a *MicoService*.

Instances of this class are persisted as relationships between nodes of the type *MicoService* in the Neo4j database.

MicoServiceDeploymentInfo

public class **MicoServiceDeploymentInfo**
Represents the information necessary for deploying a *MicoApplication*. DTO is *MicoServiceDeploymentInfoResponseDTO*.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

applyValuesFrom

```
public MicoServiceDeploymentInfo applyValuesFrom (MicoServiceDeploymentInfoRequestDTO serviceDeploymentInfoDto)
```

Applies the values of all properties of a *MicoServiceDeploymentInfoRequestDTO* to this *MicoServiceDeploymentInfo*. Note that the id will not be affected.

Parameters

- **serviceDeploymentInfoDto** – the *MicoServiceDeploymentInfoRequestDTO*.

Returns this *MicoServiceDeploymentInfo* with the values of the properties of the given *MicoServiceDeploymentInfoRequestDTO*.

applyValuesFrom

```
public MicoServiceDeploymentInfo applyValuesFrom (MicoServiceDeploymentInfoResponseDTO serviceDeploymentInfoDto)
```

Applies the values of all properties of a *MicoServiceDeploymentInfoResponseDTO* to this *MicoServiceDeploymentInfo*. Note that the id will not be affected.

Parameters

- **serviceDeploymentInfoDto** – the *MicoServiceDeploymentInfoResponseDTO*.

Returns this *MicoServiceDeploymentInfo* with the values of the properties of the given *MicoServiceDeploymentInfoResponseDTO*.

MicoServiceDeploymentInfo.ImagePullPolicy

```
public enum ImagePullPolicy
```

Enumeration for the different policies specifying when to pull an image.

Enum Constants

ALWAYS

```
public static final MicoServiceDeploymentInfo.ImagePullPolicy ALWAYS
```

IF_NOT_PRESENT

```
public static final MicoServiceDeploymentInfo.ImagePullPolicy IF_NOT_PRESENT
```

NEVER

```
public static final MicoServiceDeploymentInfo.ImagePullPolicy NEVER
```

Methods

toString

```
public String toString ()
```

MicoServiceInterface

```
public class MicoServiceInterface
```

Represents a interface, e.g., REST API, of a *MicoService*.

Instances of this class are persisted as nodes in the Neo4j database.

Methods

valueOf

```
public static MicoServiceInterface valueOf (MicoServiceInterfaceRequestDTO serviceInterfaceDto)
```

Creates a new *MicoServiceInterface* based on a *MicoServiceInterfaceRequestDTO*. Note that the id will be set to null.

Parameters

- **serviceInterfaceDto** – the *MicoServiceInterfaceRequestDTO*.

Returns a *MicoServiceInterface*.

MicoServicePort

```
public class MicoServicePort
```

Represents a basic port with a port number and port type (protocol).

Instances of this class are persisted as nodes in the Neo4j database.

Methods

valueOf

```
public static MicoServicePort valueOf (MicoServicePortRequestDTO servicePortDto)
```

Creates a new *MicoServicePort* based on a *MicoServicePortRequestDTO*. Note that the id will be set to null.

Parameters

- **servicePortDto** – the *MicoServicePortRequestDTO*.

Returns a *MicoServicePort*.

MicoTopic

public class **MicoTopic**

A Topic represented a kafka-topic Instances of this class are persisted as nodes in the neo4j database Possible roles for the topics are INPUT, OUTPUT, DEAD_LETTER, INVALID_MESSAGE, TEST_MESSAGE_OUTPUT

MicoTopicRole

public class **MicoTopicRole**

Represents a role of a *MicoTopic*.

An instance of this class is persisted as a relationship between a *MicoServiceDeploymentInfo* and a *MicoTopic* node in the Neo4j database.

Methods

valueOf

public static *MicoTopicRole* **valueOf** (*MicoTopicRequestDTO* topicDto, *MicoServiceDeploymentInfo* serviceDeploymentInfo)

Creates a new MicoTopicRole based on a MicoTopicRequestDTO.

MicoTopicRole.Role

public enum **Role**

Enum Constants

DEAD_LETTER

public static final *MicoTopicRole.Role* **DEAD_LETTER**

INPUT

public static final *MicoTopicRole.Role* **INPUT**

INVALID_MESSAGE

public static final *MicoTopicRole.Role* **INVALID_MESSAGE**

OUTPUT

public static final *MicoTopicRole.Role* **OUTPUT**

TEST_MESSAGE_OUTPUT

public static final *MicoTopicRole.Role* **TEST_MESSAGE_OUTPUT**

MicoVersion

public class **MicoVersion** implements *Comparable<MicoVersion>*

Wrapper for a *Version* that adds the functionality for a version prefix, so that versions like, e.g., ‘v1.2.3’ are possible.

Note that this class is only used for business logic purposes and instances are not persisted.

Methods

compareTo

public int **compareTo** (*MicoVersion other*)

Compares this version to the other version.

This method does not take into account the versions’ build metadata. If you want to compare the versions’ build metadata use the *Version.compareTo* method or the *Version.BUILD_AWARE_ORDER* comparator.

Parameters

- **other** – the other version to compare to.

Returns a negative integer, zero or a positive integer if this version is less than, equal to or greater than the specified version.

See also: *.compareTo(MicoVersion other)*

compareToWithBuildsTo

public int **compareToWithBuildsTo** (*MicoVersion other*)

Compare this version to the other version taking into account the build metadata.

The method makes use of the *Version.BUILD_AWARE_ORDER* comparator.

Parameters

- **other** – the other version to compare to

Returns integer result of comparison compatible with that of the *Comparable.compareTo* method

See also: *.BUILD_AWARE_ORDER*

equals

public boolean **equals** (*Object other*)

Checks if this version equals the other version.

Parameters

- **other** – the other version to compare to.

Returns `true` if this version equals the other version or `false` otherwise.

See also: `.compareTo(MicoVersion other)`

forIntegers

public static *MicoVersion* **forIntegers** (int *major*, int *minor*, int *patch*)

Creates a new instance of *MicoVersion* for the specified version numbers.

Parameters

- **major** – the major version number.
- **minor** – the minor version number.
- **patch** – the patch version number.

Throws

- ***IllegalArgumentException*** – if a negative integer is passed.

Returns a new instance of the *MicoVersion* class.

forIntegersWithPrefix

public static *MicoVersion* **forIntegersWithPrefix** (*String* *prefix*, int *major*, int *minor*, int *patch*)

Creates a new instance of *MicoVersion* for the specified version numbers with the specified prefix string.

Parameters

- **prefix** – the prefix string.
- **major** – the major version number.
- **minor** – the minor version number.
- **patch** – the patch version number.

Throws

- ***IllegalArgumentException*** – if a negative integer is passed.

Returns a new instance of the *MicoVersion* class.

getBuildMetadata

public *String* **getBuildMetadata** ()

Returns the string representation of the build metadata.

Returns the string representation of the build metadata

getMajorVersion

public int **getMajorVersion** ()

Returns the major version number.

Returns the major version number

getMinorVersion

```
public int getMinorVersion ()
```

Returns the minor version number.

Returns the minor version number

getPatchVersion

```
public int getPatchVersion ()
```

Returns the patch version number.

Returns the patch version number

getPreReleaseVersion

```
public String getPreReleaseVersion ()
```

Returns the string representation of the pre-release version.

Returns the string representation of the pre-release version

greaterThan

```
public boolean greaterThan (MicoVersion other)
```

Checks if this version is greater than the other version.

Parameters

- **other** – the other version to compare to.

Returns `true` if this version is greater than the other version or `false` otherwise.

greaterThanOrEqualTo

```
public boolean greaterThanOrEqualTo (MicoVersion other)
```

Checks if this version is greater than or equal to the other version.

Parameters

- **other** – the other version to compare to.

Returns `true` if this version is greater than or equal to the other version or `false` otherwise.

hashCode

```
public int hashCode ()
```

```
{ @inheritDoc }
```

incrementBuildMetadata

public *MicoVersion* **incrementBuildMetadata** ()

Increments the build metadata.

Returns the updated instance of the *MicoVersion* class.

incrementMajorVersion

public *MicoVersion* **incrementMajorVersion** ()

Increments the major version.

Returns the updated instance of the *MicoVersion* class.

incrementMajorVersion

public *MicoVersion* **incrementMajorVersion** (*String preRelease*)

Increments the major version and appends the pre-release version.

Parameters

- **preRelease** – the pre-release version to append.

Throws

- **IllegalArgumentException** – if the input string is NULL or empty.
- **ParseException** – when invalid version string is provided.
- **UnexpectedCharacterException** – is a special case of **ParseException**.

Returns the updated instance of the *MicoVersion* class.

incrementMinorVersion

public *MicoVersion* **incrementMinorVersion** ()

Increments the minor version.

Returns the updated instance of the *MicoVersion* class.

incrementMinorVersion

public *MicoVersion* **incrementMinorVersion** (*String preRelease*)

Increments the minor version and appends the pre-release version.

Parameters

- **preRelease** – the pre-release version to append.

Throws

- **IllegalArgumentException** – if the input string is NULL or empty.
- **ParseException** – when invalid version string is provided.
- **UnexpectedCharacterException** – is a special case of **ParseException**.

Returns the updated instance of the *MicoVersion* class.

incrementPatchVersion

public *MicoVersion* **incrementPatchVersion** ()

Increments the patch version.

Returns the updated instance of the *MicoVersion* class.

incrementPatchVersion

public *MicoVersion* **incrementPatchVersion** (*String preRelease*)

Increments the patch version and appends the pre-release version.

Parameters

- **preRelease** – the pre-release version to append.

Throws

- **IllegalArgumentException** – if the input string is NULL or empty.
- **ParseException** – when invalid version string is provided.
- **UnexpectedCharacterException** – is a special case of *ParseException*.

Returns the updated instance of the *MicoVersion* class.

incrementPreReleaseVersion

public *MicoVersion* **incrementPreReleaseVersion** ()

Increments the pre-release version.

Returns the updated instance of the *MicoVersion* class.

lessThan

public boolean **lessThan** (*MicoVersion other*)

Checks if this version is less than the other version.

Parameters

- **other** – the other version to compare to.

Returns `true` if this version is less than the other version or `false` otherwise.

lessThanOrEqualTo

public boolean **lessThanOrEqualTo** (*MicoVersion other*)

Checks if this version is less than or equal to the other version.

Parameters

- **other** – the other version to compare to.

Returns `true` if this version is less than or equal to the other version or `false` otherwise.

setBuildMetadata

public *MicoVersion* **setBuildMetadata** (*String* build)

Sets the build metadata.

Parameters

- **build** – the build metadata to set.

Throws

- **IllegalArgumentException** – if the input string is NULL or empty.
- **ParseException** – when invalid version string is provided.
- **UnexpectedCharacterException** – is a special case of **ParseException**.

Returns the updated instance of the *MicoVersion* class.

setPreReleaseVersion

public *MicoVersion* **setPreReleaseVersion** (*String* preRelease)

Sets the pre-release version.

Parameters

- **preRelease** – the pre-release version to set.

Throws

- **IllegalArgumentException** – if the input string is NULL or empty.
- **ParseException** – when invalid version string is provided.
- **UnexpectedCharacterException** – is a special case of **ParseException**.

Returns the updated instance of the *MicoVersion* class.

toString

```
public String toString ()  
    { @inheritDoc }
```

valueOf

public static *MicoVersion* **valueOf** (*String* version)

Creates a new instance of *MicoVersion* as a result of parsing the specified version string. Prefixes are possible as everything before the first digit in the given version string is treated as a prefix to the actual semantic version. Note that the prefix can only consist of letters.

Parameters

- **version** – the version string to parse (may include a prefix).

Throws

- **VersionNotSupportedException** – if the version is not a semantic version with a string prefix.

Returns a new instance of the *MicoVersion* class.

OpenFaaSFunction

public class **OpenFaaSFunction**

Represents an OpenFaaS function as used by the KafkaFaaSConnector

Instances of this class are persisted as nodes in the Neo4j database.

2.2.11 io.github.ust.mico.core.persistence

KubernetesDeploymentInfoRepository

public interface **KubernetesDeploymentInfoRepository** extends `Neo4jRepository<KubernetesDeploymentInfo, Long>`

Methods

cleanUp

void **cleanUp** ()

Deletes all *KubernetesDeploymentInfo* nodes that do **not** have any relationship to another node.

MicoApplicationRepository

public interface **MicoApplicationRepository** extends `Neo4jRepository<MicoApplication, Long>`

Methods

findAll

`List<MicoApplication>` **findAll** ()

findAll

`List<MicoApplication>` **findAll** (int *depth*)

findAllByUsedService

`List<MicoApplication>` **findAllByUsedService** (String *shortName*, String *version*)

Find all applications that are using the given service.

Parameters

- **shortName** – the shortName of the *MicoService*
- **version** – the version of the *MicoService*

Returns a list of *MicoApplication*

findAllByUsedServiceInstance

List<*MicoApplication*> **findAllByUsedServiceInstance** (*String* instanceId)

Find all applications that are using the given service.

Parameters

- **instanceId** – the instance ID of the *MicoServiceDeploymentInfo*

Returns a list of *MicoApplication*

findByShortName

List<*MicoApplication*> **findByShortName** (*String* shortName)

findByShortNameAndVersion

Optional<*MicoApplication*> **findByShortNameAndVersion** (*String* shortName, *String* version)

MicoBackgroundJobRepository

public interface **MicoBackgroundJobRepository** extends CrudRepository<*MicoServiceBackgroundJob*, *String*>

Methods

findAll

List<*MicoServiceBackgroundJob*> **findAll** ()

findById

List<*MicoServiceBackgroundJob*> **findById** (*String* instanceId)

findByIdAndType

Optional<*MicoServiceBackgroundJob*> **findByIdAndType** (*String* instanceId, *MicoServiceBackgroundJob.Type* type)

findByServiceShortNameAndServiceVersion

List<*MicoServiceBackgroundJob*> **findByServiceShortNameAndServiceVersion** (*String* micoServiceShortName, *String* micoServiceVersion)

findByServiceShortNameAndServiceVersionAndType

```
List<MicoServiceBackgroundJob> findByServiceShortNameAndServiceVersionAndType (String
                                                                    mi-
                                                                    coSer-
                                                                    viceShort-
                                                                    Name,
                                                                    String
                                                                    mi-
                                                                    coSer-
                                                                    viceV-
                                                                    er-
                                                                    sion,
                                                                    Mi-
                                                                    coSer-
                                                                    vice-
                                                                    Back-
                                                                    ground-
                                                                    Job.Type
                                                                    type)
```

MicoEnvironmentVariableRepository

public interface **MicoEnvironmentVariableRepository** extends Neo4jRepository<*MicoEnvironmentVariable*, Long>

Methods

cleanUp

```
void cleanUp ()
    Deletes all environment variables that do not have any relationship to another node.
```

MicoInterfaceConnectionRepository

public interface **MicoInterfaceConnectionRepository** extends Neo4jRepository<*MicoInterfaceConnection*, Long>

Methods

cleanUp

```
void cleanUp ()
    Deletes all interface connections that do not have any relationship to another node.
```

MicoLabelRepository

public interface **MicoLabelRepository** extends Neo4jRepository<*MicoLabel*, Long>

Methods

cleanUp

void **cleanUp** ()

Deletes all labels that do **not** have any relationship to another node.

MicoServiceDependencyRepository

public interface **MicoServiceDependencyRepository** extends Neo4jRepository<*MicoServiceDependency*, Long>

MicoServiceDeploymentInfoRepository

public interface **MicoServiceDeploymentInfoRepository** extends Neo4jRepository<*MicoServiceDeploymentInfo*, Long>

Methods

deleteAllByApplication

void **deleteAllByApplication** (*String applicationShortName*)

Deletes all deployment information for all versions of an application including the ones for the KafkaFaasConnectors. All additional properties of a *MicoServiceDeploymentInfo* that are stored as a separate node entity and connected to it via a [:HAS] relationship will be deleted too, if they are used exclusively by this deployment information. Exclusively means that there must only be one single edge connected to the particular property (relatedNode, see size operator in WHERE clause). If that's the case, it's possible to delete this related node safely.

Also works with a KafkaFaasConnector instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.

deleteAllByApplication

void **deleteAllByApplication** (*String applicationShortName*, *String applicationVersion*)

Deletes all deployment information for a particular application including the ones for the KafkaFaasConnectors. All additional properties of a *MicoServiceDeploymentInfo* that are stored as a separate node entity and connected to it via a [:HAS] relationship will be deleted too, if they are used exclusively by this deployment information. Exclusively means that there must only be one single edge connected to the particular property (relatedNode, see size operator in WHERE clause). If that's the case, it's possible to delete this related node safely.

Also works with a KafkaFaasConnector instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.

deleteByApplicationAndService

```
void deleteByApplicationAndService (String applicationShortName, String applicationVersion,  
                                   String serviceShortName)
```

Deletes the deployment information for a particular application and service. All additional properties of a *MicoServiceDeploymentInfo* that are stored as a separate node entity and connected to it via a [:HAS] relationship will be deleted too, if they are used exclusively by this deployment information. Exclusively means that there must only be one single edge connected to the particular property (*relatedNode*, see *size* operator in *WHERE* clause). If that's the case, it's possible to delete this related node safely.

Also works with a *KafkaFaasConnector* instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.
- **serviceShortName** – the short name of the *MicoService*.

deleteByApplicationAndService

```
void deleteByApplicationAndService (String applicationShortName, String applicationVersion,  
                                   String serviceShortName, String serviceVersion)
```

Deletes the deployment information for a particular application and service. All additional properties of a *MicoServiceDeploymentInfo* that are stored as a separate node entity and connected to it via a [:HAS] relationship will be deleted too, if they are used exclusively by this deployment information. Exclusively means that there must only be one single edge connected to the particular property (*relatedNode*, see *size* operator in *WHERE* clause). If that's the case, it's possible to delete this related node safely.

Also works with a *KafkaFaasConnector* instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.
- **serviceShortName** – the short name of the *MicoService*.
- **serviceVersion** – the version of the *MicoService*.

findAllByApplication

```
List<MicoServiceDeploymentInfo> findAllByApplication (String applicationShortName, String appli-  
                                                    cationVersion)
```

Retrieves all service deployment information of a particular application. Also includes these which are used for the deployments of *KafkaFaasConnector* instances.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.

Returns a *List* of *MicoServiceDeploymentInfo* instances.

findAllByService

List<*MicoServiceDeploymentInfo*> **findAllByService** (String *serviceShortName*, String *serviceVersion*)

Retrieves all service deployment information of a service. Note that one service can be used by (included in) multiple applications. Also works with a KafkaFaasConnector instance.

Parameters

- **serviceShortName** – the short name of the *MicoService*.
- **serviceVersion** – the version of the *MicoService*.

Returns a List of *MicoServiceDeploymentInfo* instances.

findByApplicationAndService

List<*MicoServiceDeploymentInfo*> **findByApplicationAndService** (String *applicationShortName*,
String *applicationVersion*,
String *serviceShortName*)

Retrieves the deployment information for a particular application and service. Also works with a KafkaFaasConnector instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.
- **serviceShortName** – the short name of the *MicoService*.

Returns an Optional of *MicoServiceDeploymentInfo*.

findByApplicationAndService

List<*MicoServiceDeploymentInfo*> **findByApplicationAndService** (String *applicationShortName*,
String *applicationVersion*,
String *serviceShortName*, String *serviceVersion*)

Retrieves the deployment information for a particular application and service. Also works with a KafkaFaasConnector instance.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.
- **serviceShortName** – the short name of the *MicoService*.
- **serviceVersion** – the version of the *MicoService*.

Returns an Optional of *MicoServiceDeploymentInfo*.

findById

Optional<*MicoServiceDeploymentInfo*> **findById** (String *instanceId*)

findKFConnectorSDIsByApplication

List<*MicoServiceDeploymentInfo*> **findKFConnectorSDIsByApplication** (*String* applicationShortName, *String* applicationVersion)

Retrieves all service deployment information that are used for KafkaFaasConnectors of a particular application.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.

Returns a List of *MicoServiceDeploymentInfo* instances.

findMicoServiceSDIsByApplication

List<*MicoServiceDeploymentInfo*> **findMicoServiceSDIsByApplication** (*String* applicationShortName, *String* applicationVersion)

Retrieves all service deployment information that are used for normal MicoServices of a particular application.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.

Returns a List of *MicoServiceDeploymentInfo* instances.

MicoServiceInterfaceRepository

public interface **MicoServiceInterfaceRepository** extends Neo4jRepository<*MicoServiceInterface*, Long>

Methods

deleteByServiceAndName

void **deleteByServiceAndName** (*String* shortName, *String* version, *String* serviceInterfaceName)

findByService

List<*MicoServiceInterface*> **findByService** (*String* shortName, *String* version)

findByServiceAndName

Optional<*MicoServiceInterface*> **findByServiceAndName** (*String* shortName, *String* version, *String* serviceInterfaceName)

MicoServicePortRepository

public interface **MicoServicePortRepository** extends Neo4jRepository<*MicoServicePort*, Long>

MicoServiceRepository

public interface **MicoServiceRepository** extends Neo4jRepository<*MicoService*, Long>

Methods

deleteServiceByShortNameAndVersion

void **deleteServiceByShortNameAndVersion** (*String shortName*, *String version*)

findAll

List<*MicoService*> **findAll** ()

findAll

List<*MicoService*> **findAll** (int *depth*)

findAllByApplication

List<*MicoService*> **findAllByApplication** (*String applicationShortName*, *String applicationVersion*)

Finds all services that are included by a given application.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.

Returns a list of *MicoServices*.

findAllByApplicationAndServiceShortName

Optional<*MicoService*> **findAllByApplicationAndServiceShortName** (*String applicationShortName*, *String applicationVersion*, *String serviceShortName*)

Finds the service that is included in a given application for a given service short name.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*.
- **applicationVersion** – the version of the *MicoApplication*.
- **serviceShortName** – the short name of the *MicoService*.

Returns a list of *MicoServices*.

findByShortName

List<*MicoService*> **findByShortName** (*String shortName*)

findByShortNameAndVersion

Optional<*MicoService*> **findByShortNameAndVersion** (*String* shortName, *String* version)

findDependees

List<*MicoService*> **findDependees** (*String* shortName, *String* version)

Finds all services (dependees) the given service (depender) depends on.

Parameters

- **shortName** – the short name of the *MicoService* (depender).
- **version** – the version of the *MicoService* (depender).

Returns a list of *MicoServices*.

findDependeesIncludeDepender

List<*MicoService*> **findDependeesIncludeDepender** (*String* shortName, *String* version)

Finds all services (dependees) the given service (depender) depends on as well as the service (depender) itself.

Parameters

- **shortName** – the short name of the *MicoService* (depender).
- **version** – the version of the *MicoService* (depender).

Returns a list of *MicoServices* including all dependees as well as the depender.

findDependers

List<*MicoService*> **findDependers** (*String* shortName, *String* version)

Finds all services (dependers) that depend on the given service (dependee).

Parameters

- **shortName** – the short name of the *MicoService* (dependee).
- **version** – the version of the *MicoService* (dependee).

Returns a list of *MicoServices*.

MicoTopicRepository

public interface **MicoTopicRepository** extends Neo4jRepository<*MicoTopic*, Long>

Methods

cleanUp

void **cleanUp** ()

Deletes all topics that do **not** have any relationship to another node.

findAllByName

List<*MicoTopic*> **findAllByName** (*String name*)

findByName

Optional<*MicoTopic*> **findByName** (*String name*)

findFirstByName

MicoTopic **findFirstByName** (*String name*)

OpenFaaSFunctionRepository

public interface **OpenFaaSFunctionRepository** extends Neo4jRepository<*OpenFaaSFunction*, Long>

Methods

cleanUp

void **cleanUp** ()

Deletes all OpenFaaS functions that do **not** have any relationship to another node.

findByName

Optional<*OpenFaaSFunction*> **findByName** (*String name*)

2.2.12 io.github.ust.mico.core.resource

ApplicationResource

public class **ApplicationResource**

Fields

PATH_APPLICATIONS

public static final *String* **PATH_APPLICATIONS**

PATH_KAFKA_FAAS_CONNECTOR

public static final *String* **PATH_KAFKA_FAAS_CONNECTOR**

PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_INSTANCE_ID

static final *String* **PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_INSTANCE_ID**

PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_VERSION

public static final *String* **PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_VERSION**

PATH_VARIABLE_SHORT_NAME

static final *String* **PATH_VARIABLE_SHORT_NAME**

PATH_VARIABLE_VERSION

static final *String* **PATH_VARIABLE_VERSION**

Methods

addKafkaFaasConnectorInstanceToApplication

public ResponseEntity<Resource<*KFConnectorDeploymentInfoResponseDTO*>> **addKafkaFaasConnectorInstanceToApp1**

addServiceToApplication

```
public ResponseEntity<Resource<MicoServiceDeploymentInfoResponseDTO>> addServiceToApplication (String  
                                                                                   ap-  
                                                                                   pli-  
                                                                                   ca-  
                                                                                   tion-  
                                                                                   Short-  
                                                                                   Name,  
                                                                                   String  
                                                                                   ap-  
                                                                                   pli-  
                                                                                   ca-  
                                                                                   tion-  
                                                                                   Ver-  
                                                                                   sion,  
                                                                                   String  
                                                                                   ser-  
                                                                                   viceShort-  
                                                                                   Name,  
                                                                                   String  
                                                                                   ser-  
                                                                                   viceV-  
                                                                                   er-  
                                                                                   sion)
```

addServiceToApplication

```
public ResponseEntity<Resource<MicoServiceDeploymentInfoResponseDTO>> addServiceToApplication (String  
                                                                                               ap-  
                                                                                               pli-  
                                                                                               ca-  
                                                                                               tion-  
                                                                                               Short-  
                                                                                               Name,  
                                                                                               String  
                                                                                               ap-  
                                                                                               pli-  
                                                                                               ca-  
                                                                                               tion-  
                                                                                               Ver-  
                                                                                               sion,  
                                                                                               String  
                                                                                               ser-  
                                                                                               viceShort-  
                                                                                               Name,  
                                                                                               String  
                                                                                               ser-  
                                                                                               viceV-  
                                                                                               er-  
                                                                                               sion,  
                                                                                               String  
                                                                                               in-  
                                                                                               stan-  
                                                                                               ceId)
```

Currently we don't support multiple instance deployment for normal MICO services. Covered by MICO#743.
Therefore this API endpoint is not required at the moment.

createApplication

```
public ResponseEntity<Resource<MicoApplicationWithServicesResponseDTO>> createApplication (MicoApplicationRequestD  
                                                                                               ap-  
                                                                                               pli-  
                                                                                               ca-  
                                                                                               tionDto)
```

deleteAllVersionsOfApplication

```
public ResponseEntity<Void> deleteAllVersionsOfApplication (String shortName)
```

deleteApplication

```
public ResponseEntity<Void> deleteApplication (String shortName, String version)
```

deleteKafkaFaasConnectorInstanceFromApplication

```
public ResponseEntity<Void> deleteKafkaFaasConnectorInstanceFromApplication (String
                                                                    short-
                                                                    Name,
                                                                    String
                                                                    version,
                                                                    String
                                                                    instan-
                                                                    ceId)
```

deleteKafkaFaasConnectorInstancesFromApplication

```
public ResponseEntity<Void> deleteKafkaFaasConnectorInstancesFromApplication (String
                                                                    short-
                                                                    Name,
                                                                    String
                                                                    ver-
                                                                    sion)
```

deleteServiceFromApplication

```
public ResponseEntity<Void> deleteServiceFromApplication (String shortName, String version,
                                                                    String serviceShortName)
```

getAllApplications

```
public ResponseEntity<Resources<Resource<MicoApplicationWithServicesResponseDTO>>> getAllApplications ()
```

getApplicationByShortNameAndVersion

```
public ResponseEntity<Resource<MicoApplicationWithServicesResponseDTO>> getApplicationByShortNameAndVersion
```

getApplicationDeploymentStatus

```
public ResponseEntity<Resource<MicoApplicationDeploymentStatusResponseDTO>> getApplicationDeploymentStatus (S
                                                                    sh
                                                                    M
                                                                    S
                                                                    ve
                                                                    si
```

getApplicationsByShortName

```
public ResponseEntity<Resources<Resource<MicoApplicationWithServicesResponseDTO>>> getApplicationsByShortName (
```

getServicesOfApplication

```
public ResponseEntity<Resources<Resource<MicoServiceResponseDTO>>> getServicesOfApplication (String  
short-  
Name,  
String  
ver-  
sion)
```

getStatusOfApplication

```
public ResponseEntity<Resource<MicoApplicationStatusResponseDTO>> getStatusOfApplication (String  
short-  
Name,  
String  
ver-  
sion)
```

promoteApplication

```
public ResponseEntity<Resource<MicoApplicationWithServicesResponseDTO>> promoteApplication (String  
short-  
Name,  
String  
ver-  
sion,  
Mi-  
coV-  
er-  
sion-  
Re-  
quest-  
DTO  
newVer-  
sionDto)
```

updateApplication

```
public ResponseEntity<Resource<MicoApplicationWithServicesResponseDTO>> updateApplication (String  
                                                                 short-  
                                                                 Name,  
                                                                 String  
                                                                 ver-  
                                                                 sion,  
                                                                 Mi-  
                                                                 coAp-  
                                                                 pli-  
                                                                 ca-  
                                                                 tion-  
                                                                 Re-  
                                                                 quest-  
                                                                 DTO  
                                                                 ap-  
                                                                 pli-  
                                                                 ca-  
                                                                 tion-  
                                                                 Re-  
                                                                 quest-  
                                                                 Dto)
```

updateKafkaFaasConnectorInstanceOfApplication

```
public ResponseEntity<Resource<KFConnectorDeploymentInfoResponseDTO>> updateKafkaFaasConnectorInstanceOfAp
```

BackgroundJobResource

```
public class BackgroundJobResource
```

Methods

deleteAllJobs

```
public ResponseEntity<Void> deleteAllJobs ()
```

deleteJob

```
public ResponseEntity<Void> deleteJob (String id)
```

getAllJobs

```
public ResponseEntity<Resources<Resource<MicoServiceBackgroundJobResponseDTO>>> getAllJobs ()
```

getJobById

```
public ResponseEntity<Resource<MicoServiceBackgroundJobResponseDTO>> getJobById (String id)
```

getJobStatusByApplicationShortNameAndVersion

```
public ResponseEntity<Resource<MicoApplicationJobStatusResponseDTO>> getJobStatusByApplicationShortNameAndVersion (String shortName, String version)
```

DeploymentResource

```
public class DeploymentResource
```

Methods

deploy

```
public ResponseEntity<Resource<MicoApplicationJobStatusResponseDTO>> deploy (String shortName,
                                                                              String version,
                                                                              boolean rebuildImages)
```

undeploy

```
public ResponseEntity<Void> undeploy (String shortName, String version)
```

KafkaFaasConnectorDeploymentInfoResource

```
public class KafkaFaasConnectorDeploymentInfoResource
```

Methods

getKafkaFaasConnectorDeploymentInformation

```
public ResponseEntity<Resources<Resource<KFConnectorDeploymentInfoResponseDTO>>> getKafkaFaasConnectorDeploy
```

getKafkaFaasConnectorDeploymentInformationInstance

```
public ResponseEntity<Resource<KFConnectorDeploymentInfoResponseDTO>> getKafkaFaasConnectorDeploymentInfo
```

getKfConnectorDeploymentInfoResponseDTOResource

protected static Resource<*KFConnectorDeploymentInfoResponseDTO*> **getKfConnectorDeploymentInfoResponseDTORes**

Wraps a *KFConnectorDeploymentInfoResponseDTO* into a HATEOAS resource with a link to the application and a self-link.

Parameters

- **applicationShortName** – the short name of the *MicoApplication*
- **applicationVersion** – the version of the *MicoApplication*
- **micoServiceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns The resource containing the *KFConnectorDeploymentInfoResponseDTO*.

updateKafkaFaasConnectorDeploymentInfo

```
public ResponseEntity<Resource<KFConnectorDeploymentInfoResponseDTO>> updateKafkaFaasConnectorDeploymentInfo
```

OpenFaasResource

```
public class OpenFaasResource
```

Fields

FUNCTIONS_PATH

```
public static final String FUNCTIONS_PATH
```

OPEN_FAAS_BASE_PATH

```
public static final String OPEN_FAAS_BASE_PATH
```

OPEN_FAAS_FUNCTION_LIST_PATH

```
public static final String OPEN_FAAS_FUNCTION_LIST_PATH
```

openFaaSConfig

OpenFaaSConfig **openFaaSConfig**

openFaasBroker

OpenFaasBroker **openFaasBroker**

restTemplate

RestTemplate **restTemplate**

Methods

getOpenFaasFunctions

```
public ResponseEntity<String> getOpenFaasFunctions ()
```

getOpenFaasURL

```
public ResponseEntity<ExternalUrlDTO> getOpenFaasURL ()
```

ServiceDeploymentInfoResource

```
public class ServiceDeploymentInfoResource
```

Methods

getServiceDeploymentInformation

```
public ResponseEntity<Resource<MicoServiceDeploymentInfoResponseDTO>> getServiceDeploymentInformation (String  
short-  
Name,  
String  
ver-  
sion,  
String  
ser-  
viceSh  
Name)
```

updateServiceDeploymentInformation

```
public ResponseEntity<Resource<MicoServiceDeploymentInfoResponseDTO>> updateServiceDeploymentInformation (St
```

sh
No
St
ve
si
St
se
vi
No
M
co
vi
pl
m
fo
qu
D
se
vi
pl
m
fo
qu
D

ServiceInterfaceResource

```
public class ServiceInterfaceResource
```

Methods

createServiceInterface

```
public ResponseEntity<Resource<MicoServiceInterfaceResponseDTO>> createServiceInterface (String
                                                                    short-
                                                                    Name,
                                                                    String
                                                                    ver-
                                                                    sion,
                                                                    Mi-
                                                                    coSer-
                                                                    vi-
                                                                    ceIn-
                                                                    ter-
                                                                    fac-
                                                                    eRequest-
                                                                    DTO
                                                                    ser-
                                                                    vi-
                                                                    ceIn-
                                                                    ter-
                                                                    fac-
                                                                    eRequest-
                                                                    Dto)
```

This is not transactional. At the moment we have only one user. If this changes transactional support is a must.
FIXME Add transactional support

Parameters

- **shortName** – the name of the MICO service
- **version** – the version of the MICO service
- **serviceInterfaceRequestDto** – the *MicoServiceInterfaceRequestDTO*

Returns the created MICO service interface

deleteServiceInterface

```
public ResponseEntity<Void> deleteServiceInterface (String shortName, String version, String ser-
                                                                    viceInterfaceName)
```

getInterfaceByName

```
public ResponseEntity<Resource<MicoServiceInterfaceResponseDTO>> getInterfaceByName (String
                                                                    short-
                                                                    Name,
                                                                    String
                                                                    version,
                                                                    String
                                                                    servi-
                                                                    ceIn-
                                                                    terface-
                                                                    Name)
```

getInterfacePublicIpByName

```
public ResponseEntity<MicoServiceInterfaceStatusResponseDTO> getInterfacePublicIpByName (String
                                                                 short-
                                                                 Name,
                                                                 String
                                                                 ver-
                                                                 sion,
                                                                 String
                                                                 ser-
                                                                 vi-
                                                                 ceIn-
                                                                 ter-
                                                                 face-
                                                                 Name,
                                                                 String
                                                                 in-
                                                                 stan-
                                                                 ceId)
```

getInterfacesOfService

```
public ResponseEntity<Resources<Resource<MicoServiceInterfaceResponseDTO>>> getInterfacesOfService (String
                                                                 short-
                                                                 Name,
                                                                 String
                                                                 ver-
                                                                 sion)
```

updateServiceInterface

```
public ResponseEntity<Resource<MicoServiceInterfaceResponseDTO>> updateServiceInterface (String
short-
Name,
String
ver-
sion,
String
ser-
vi-
ceIn-
ter-
face-
Name,
Mi-
coSer-
vi-
ceIn-
ter-
fac-
eRequest-
DTO
up-
dat-
ed-
Ser-
vi-
ceIn-
ter-
fac-
eRequest-
Dto)
```

Updates an existing MICO service interface.

Parameters

- **shortName** – the name of a *MicoService*
- **version** – the version a *MicoService*
- **serviceInterfaceName** – the name of a *MicoServiceInterface*
- **updatedServiceInterfaceRequestDto** – the *MicoServiceInterfaceRequestDTO*

Returns the updated *MicoServiceInterfaceResponseDTO*

ServiceResource

```
public class ServiceResource
```

Fields

PATH_VARIABLE_INSTANCE_ID

static final `String` **PATH_VARIABLE_INSTANCE_ID**

PATH_VARIABLE_SHORT_NAME

static final `String` **PATH_VARIABLE_SHORT_NAME**

PATH_VARIABLE_VERSION

static final `String` **PATH_VARIABLE_VERSION**

Methods

createNewDependee

public ResponseEntity<Void> **createNewDependee** (`String` *shortName*, `String` *version*, `String` *dependeeShortName*, `String` *dependeeVersion*)
Creates a new dependency edge between the Service and the depended service.

createService

public ResponseEntity<Resource<*MicoServiceResponseDTO*>> **createService** (*MicoServiceRequestDTO* *serviceDto*)

deleteAllDependees

public ResponseEntity<Void> **deleteAllDependees** (`String` *shortName*, `String` *version*)

deleteAllVersionsOfService

public ResponseEntity<Void> **deleteAllVersionsOfService** (`String` *shortName*)

deleteDependee

public ResponseEntity<Void> **deleteDependee** (`String` *shortName*, `String` *version*, `String` *dependeeShortName*, `String` *dependeeVersion*)

deleteService

public ResponseEntity<Void> **deleteService** (`String` *shortName*, `String` *version*)

getDependees

```
public ResponseEntity<Resources<Resource<MicoServiceResponseDTO>>> getDependees (String short-  
Name, String  
version)
```

getDependencyGraph

```
public ResponseEntity<Resource<MicoServiceDependencyGraphResponseDTO>> getDependencyGraph (String  
short-  
Name,  
String  
ver-  
sion)
```

getDependers

```
public ResponseEntity<Resources<Resource<MicoServiceResponseDTO>>> getDependers (String short-  
Name, String  
version)
```

getServiceByShortNameAndVersion

```
public ResponseEntity<Resource<MicoServiceResponseDTO>> getServiceByShortNameAndVersion (String  
short-  
Name,  
String  
ver-  
sion)
```

getServiceLinks

```
static Iterable<Link> getServiceLinks (MicoService service)
```

getServiceList

```
public ResponseEntity<Resources<Resource<MicoServiceResponseDTO>>> getServiceList ()
```

getServiceResponseDTOResource

```
static Resource<MicoServiceResponseDTO> getServiceResponseDTOResource (MicoService ser-  
vice)
```

getServiceResponseDTOResourcesList

```
static List<Resource<MicoServiceResponseDTO>> getServiceResponseDTOResourcesList (List<MicoService>  
ser-  
vices)
```


getServiceYamlByShortNameAndVersion

```
public ResponseEntity<Resource<MicoYamlResponseDTO>> getServiceYamlByShortNameAndVersion (String
                                                                 short-
                                                                 Name,
                                                                 String
                                                                 ver-
                                                                 sion)
```

Return yaml for a *MicoService* for the give shortName and version.

Parameters

- **shortName** – the short name of the *MicoService*.
- **version** – version the version of the *MicoService*.

Returns the kubernetes YAML for the *MicoService*.

getStatusListOfService

```
public ResponseEntity<Resources<Resource<MicoServiceStatusResponseDTO>>> getStatusListOfService (String
                                                                 short-
                                                                 Name,
                                                                 String
                                                                 ver-
                                                                 sion)
```

getStatusOfServiceInstance

```
public ResponseEntity<Resource<MicoServiceStatusResponseDTO>> getStatusOfServiceInstance (String
                                                                 short-
                                                                 Name,
                                                                 String
                                                                 ver-
                                                                 sion,
                                                                 String
                                                                 in-
                                                                 stan-
                                                                 ceId)
```

getVersionsFromGitHub

```
public ResponseEntity<Resources<Resource<MicoVersionRequestDTO>>> getVersionsFromGitHub (String
                                                                 url)
```

getVersionsOfService

```
public ResponseEntity<Resources<Resource<MicoServiceResponseDTO>>> getVersionsOfService (String
                                                                 short-
                                                                 Name)
```

importMicoServiceFromGitHub

```
public ResponseEntity<Resource<MicoServiceResponseDTO>> importMicoServiceFromGitHub (CrawlingInfoRequestDTO  
                                                                                      crawling-  
                                                                                      Info)
```

promoteService

```
public ResponseEntity<Resource<MicoServiceResponseDTO>> promoteService (String      shortName,  
                                                                 String  version,  Mi-  
                                                                 coVersionRequestDTO  
                                                                 newVersionDto)
```

updateService

```
public ResponseEntity<Resource<MicoServiceResponseDTO>> updateService (String      shortName,  
                                                                 String  version,  Mi-  
                                                                 coServiceRequestDTO  
                                                                 serviceDto)
```

TopicResource

```
public class TopicResource
```

Fields

TOPIC_BASE_PATH

```
public static final String TOPIC_BASE_PATH
```

Methods

getAllTopics

```
public ResponseEntity<Resources<Resource<TopicDTO>>> getAllTopics ()
```

2.2.13 io.github.ust.mico.core.service

GitHubCrawler

```
public class GitHubCrawler
```

Constructors

GitHubCrawler

```
public GitHubCrawler (RestTemplate restTemplate, KubernetesNameNormalizer kubernetesNameNormalizer)
```

Methods

adaptUriForGitHubApi

```
public String adaptUriForGitHubApi (String url)
```

crawlGitHubRepoLatestRelease

```
public MicoService crawlGitHubRepoLatestRelease (String gitHubRepoUrl, String dockerfilePath)
```

crawlGitHubRepoLatestRelease

```
public MicoService crawlGitHubRepoLatestRelease (String gitHubRepoUrl)
```

crawlGitHubRepoSpecificRelease

```
public MicoService crawlGitHubRepoSpecificRelease (String gitHubRepoUrl, String version, String dockerfilePath)
```

crawlGitHubRepoSpecificRelease

```
public MicoService crawlGitHubRepoSpecificRelease (String gitHubRepoUrl, String version)
```

getVersionsFromGitHubRepo

```
public List<String> getVersionsFromGitHubRepo (String gitHubRepoUrl)
```

ImageBuilderHealthIndicator

```
public class ImageBuilderHealthIndicator implements HealthIndicator
```

Constructors

ImageBuilderHealthIndicator

```
public ImageBuilderHealthIndicator (TektonPipelinesController imageBuilder)
```

Methods

health

public Health **health** ()

MicoKubernetesClient

public class **MicoKubernetesClient**

Provides accessor methods for creating deployments and services in Kubernetes as well as getter methods to retrieve existing Kubernetes deployments and services.

Fields

OPEN_FAAS_SECRET_DATA_PASSWORD_NAME

public static final [String](#) **OPEN_FAAS_SECRET_DATA_PASSWORD_NAME**

The name of the data element which holds the OpenFaaS password inside the secret.

OPEN_FAAS_SECRET_DATA_USERNAME_NAME

public static final [String](#) **OPEN_FAAS_SECRET_DATA_USERNAME_NAME**

The name of the data element which holds the OpenFaaS username inside the secret

OPEN_FAAS_SECRET_NAME

public static final [String](#) **OPEN_FAAS_SECRET_NAME**

The name of the secret which holds the OpenFaaS username and password.

Constructors

MicoKubernetesClient

public **MicoKubernetesClient** (*MicoKubernetesConfig* micoKubernetesConfig, *MicoKubernetesBuildBotConfig* buildBotConfig, *KubernetesClient* kubernetesClient, *TektonPipelinesController* imageBuilder, *BackgroundJobBroker* backgroundJobBroker, *MicoApplicationRepository* applicationRepository, *MicoServiceDeploymentInfoRepository* serviceDeploymentInfoRepository, *KubernetesDeploymentInfoRepository* kubernetesDeploymentInfoRepository)

Methods

createMicoServiceInstance

public Deployment **createMicoServiceInstance** (*MicoServiceDeploymentInfo* serviceDeploymentInfo)

Create a Kubernetes deployment based on a *MicoServiceDeploymentInfo*.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns the Kubernetes Deployment resource object

createMicoServiceInterface

public Service **createMicoServiceInterface** (*MicoServiceInterface* micoServiceInterface, *MicoServiceDeploymentInfo* micoServiceDeploymentInfo)

Create a Kubernetes service based on a MICO service interface.

Parameters

- **micoServiceInterface** – the *MicoServiceInterface*
- **micoServiceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns the Kubernetes Service resource

createOrUpdateInterfaceConnections

public void **createOrUpdateInterfaceConnections** (*MicoApplication* micoApplication)

Creates or updates all interface connections of the given MicoApplication.

Parameters

- **micoApplication** – the *MicoApplication*

createServiceName

public String **createServiceName** (*MicoServiceDeploymentInfo* serviceDeploymentInfo, *MicoServiceInterface* serviceInterface)

Creates the name of the Kubernetes service based on the serviceDeploymentInfo and the serviceInterfaceName.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*
- **serviceInterface** – the *MicoServiceInterface*

Returns the created string that should be used as the name of the Kubernetes service

getApplicationDeploymentStatus

public *MicoApplicationDeploymentStatus* **getApplicationDeploymentStatus** (*MicoApplication* micoApplication)

Indicates whether a MicoApplication is currently deployed.

In order to determine the application deployment status of the given MicoApplication the following points are checked:

- the current *MicoApplicationJobStatus* (deployment may be scheduled, running or finished with an error)
- the stored *MicoServiceDeploymentInfo* and *KubernetesDeploymentInfo*
- the actual information retrieved from Kubernetes regarding deployments for *MicoServices* and Kubernetes Services for *MicoServiceInterfaces*

Note that the returned *MicoApplicationDeploymentStatus* contains info messages with further information in case the *MicoApplication* currently is not deployed.

Parameters

- **micoApplication** – the *MicoApplication*.

Returns the *MicoApplicationDeploymentStatus*.

getDeploymentOfMicoServiceInstance

public *Optional*<Deployment> **getDeploymentOfMicoServiceInstance** (*MicoServiceDeploymentInfo* *serviceDeploymentInfo*)

Returns a Kubernetes Deployment instance that corresponds to the provided *MicoServiceDeploymentInfo*, if it is already deployed to the Kubernetes cluster. Labels are used for the lookup.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns an *Optional* with the Deployment of the Kubernetes service, or an empty *Optional* if there is no Kubernetes deployment of the *MicoService*.

getDeploymentsOfMicoService

public *List*<Deployment> **getDeploymentsOfMicoService** (*MicoService* *micoService*)

Returns a list of Kubernetes Deployment instances that corresponds to the *MicoService*. Labels are used for the lookup.

Parameters

- **micoService** – the *MicoService*

Returns a list of Kubernetes Deployments. It is empty if there is no Kubernetes deployment of the *MicoService*.

getInterfaceByNameOfMicoServiceInstance

public *Optional*<Service> **getInterfaceByNameOfMicoServiceInstance** (*MicoServiceDeploymentInfo* *serviceDeploymentInfo*,
String *micoServiceInterfaceName*)

Check if the *MicoServiceInterface* is already created for the *MicoService* in the Kubernetes cluster. Labels are used for the lookup.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*
- **micoServiceInterfaceName** – the name of a *MicoServiceInterface*

Returns an `Optional` with the Kubernetes Service, or an empty `Optional` if there is no Kubernetes Service for this `MicoServiceInterface`.

`getInterfacesOfMicoService`

```
public List<Service> getInterfacesOfMicoService (MicoService micoService)
```

Looks up if there are any interfaces created for the `MicoService` in the Kubernetes cluster. If so, it returns them as a list of Kubernetes Service objects. Labels are used for the lookup.

Parameters

- **micoService** – the `MicoService`

Returns the list of Kubernetes Service objects

`getInterfacesOfMicoServiceInstance`

```
public List<Service> getInterfacesOfMicoServiceInstance (MicoServiceDeploymentInfo serviceDeploymentInfo)
```

Looks up if there are any interfaces created for the `MicoServiceDeploymentInfo` in the Kubernetes cluster. If so, it returns them as a list of Kubernetes Service objects. Labels are used for the lookup.

Parameters

- **serviceDeploymentInfo** – the `MicoServiceDeploymentInfo`

Returns the list of Kubernetes Service objects

`getOpenFaasCredentials`

```
public PasswordAuthentication getOpenFaasCredentials ()
```

Requests the OpenFaaS credentials from a Kubernetes secret.

Returns the username and the password

`getPodsCreatedByDeploymentOfMicoServiceInstance`

```
public List<Pod> getPodsCreatedByDeploymentOfMicoServiceInstance (MicoServiceDeploymentInfo serviceDeploymentInfo)
```

Looks up if the `MicoServiceDeploymentInfo` is already deployed to the Kubernetes cluster. If so, it returns the list of Kubernetes Pod objects that belongs to the Deployment. Labels are used for the lookup.

Parameters

- **serviceDeploymentInfo** – the `MicoServiceDeploymentInfo`

Returns the list of Kubernetes Pod objects

`getPublicIpOfKubernetesService`

```
public Optional<String> getPublicIpOfKubernetesService (String name, String namespace)
```

Requests the public IP of a Kubernetes service and returns it or an empty `Optional` if the service has no public IP.

Parameters

- **name** – the name of the service.
- **namespace** – the namespace which contains the service.

Throws

- *KubernetesResourceException* – if there is no such service.

Returns the public ip of a service or an empty optional.

getPublicPortsOfKubernetesService

public *List<Integer>* **getPublicPortsOfKubernetesService** (*String name*, *String namespace*)
Requests the list of public ports of a service. It returns the list of ports or an empty list if there are none.

Parameters

- **name** – the name of the service.
- **namespace** – the namespace which contains the service.

Throws

- *KubernetesResourceException* – if there is no such service.

Returns a list of ports or an empty list.

getService

public *Optional<Service>* **getService** (*String name*, *String namespace*)
Requests the service with the given name in the given namespace or *null* if there is no such service

Parameters

- **name** – the name of the service.
- **namespace** – the namespace which contains the service.

Returns the service in the namespace and with the given name or *null*.

getYaml

public *String* **getYaml** (*MicoService micoService*)
Retrieves the yaml(s) for all Kubernetes deployments of a MicoService and the yaml(s) for all Kubernetes services of the including interfaces (if there are any).

Parameters

- **micoService** – the *MicoService*

Throws

- *JsonProcessingException* – if there is a error processing the content.

Returns the kubernetes YAML for the *MicoService*.

getYaml

public **String** **getYaml** (*MicoServiceDeploymentInfo* serviceDeploymentInfo)

Retrieves the yaml for a MicoServiceDeploymentInfo, contains the interfaces if they exist.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Throws

- **JsonProcessingException** – if there is a error processing the content.

Returns the kubernetes YAML for the *MicoService*.

isApplicationDeployed

public boolean **isApplicationDeployed** (*MicoApplication* micoApplication)

Checks whether a given MicoApplication is currently deployed.

Parameters

- **micoApplication** – the *MicoApplication*.

Returns true if and only if *getApplicationDeploymentStatus (MicoApplication)* returns a *MicoApplicationDeploymentStatus* with *Deployed*; false otherwise.

isApplicationUndeployed

public boolean **isApplicationUndeployed** (*MicoApplication* micoApplication)

Checks whether a given MicoApplication is currently undeployed.

Parameters

- **micoApplication** – the *MicoApplication*.

Returns true if and only if *getApplicationDeploymentStatus (MicoApplication)* returns a *MicoApplicationDeploymentStatus* with *Undeployed*; false otherwise.

isMicoServiceDeployed

public boolean **isMicoServiceDeployed** (*MicoService* micoService)

Checks if a *MicoService* is already deployed at least with one instance.

Parameters

- **micoService** – the *MicoService*

Returns true if the *MicoService* is deployed.

isMicoServiceInstanceDeployed

public boolean **isMicoServiceInstanceDeployed** (*MicoServiceDeploymentInfo* serviceDeploymentInfo)

Checks if a *MicoService* instance is already deployed.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*

Returns `true` if the *MicoService* is deployed.

scaleIn

```
public Optional<Deployment> scaleIn (MicoServiceDeploymentInfo serviceDeploymentInfo, int numberOfReplicas)
```

Performs a scale in of a Kubernetes deployment based on some service deployment information by a given number of replicas to remove.

Note that the Kubernetes deployment will be undeployed if and only if the given number of replicas is less than or equal to 0.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*.
- **numberOfReplicas** – the number of replicas to remove.

Throws

- *KubernetesResourceException* – if the Kubernetes deployment can't be found

scaleOut

```
public Optional<Deployment> scaleOut (MicoServiceDeploymentInfo serviceDeploymentInfo, int numberOfReplicas)
```

Performs a scale out of a Kubernetes deployment based on some service deployment information by a given number of replicas to add.

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*.
- **numberOfReplicas** – the number of replicas to add.

Returns the Kubernetes `Deployment`.

undeployApplication

```
public void undeployApplication (MicoApplication application)
```

Undeploys an application. Note that *MicoServices* included in this application will not be undeployed, if and only if they are included in at least one other application. In this case the corresponding Kubernetes deployment will be scaled in.

Parameters

- **application** – the *MicoApplication*.

MicoStatusService

```
public class MicoStatusService
```

Provides functionality to retrieve status information for a *MicoApplication* or a particular *MicoService*.

Constructors

MicoStatusService

```
public MicoStatusService (PrometheusConfig prometheusConfig, MicoKubernetesClient micoKuber-
                           netesClient, RestTemplate restTemplate, MicoServiceDeploymentInfoRepos-
                           itory serviceDeploymentInfoRepository, MicoApplicationRepository micoAp-
                           plicationRepository)
```

Methods

getApplicationStatus

```
public MicoApplicationStatusResponseDTO getApplicationStatus (MicoApplication micoApplica-
                                                                tion)
```

Get status information for a *MicoApplication*.

Parameters

- **micoApplication** – the application the status is requested for

Returns *MicoApplicationStatusResponseDTO* containing a list of *MicoServiceStatusResponseDTO* for status information of a single *MicoService*.

getPublicIpOfKubernetesService

```
public MicoServiceInterfaceStatusResponseDTO getPublicIpOfKubernetesService (MicoServiceDeploymentInfo
                                                                                   serviceDeploy-
                                                                                   mentInfo,
                                                                                   MicoServi-
                                                                                   ceInterface
                                                                                   serviceInter-
                                                                                   face)
```

Get the public IP of a *MicoServiceInterface* by providing the corresponding Kubernetes Service.

Parameters

- **serviceDeploymentInfo** – is the *MicoServiceDeploymentInfo*, that has a *MicoServiceInterface*, which is deployed on Kubernetes
- **serviceInterface** – the *MicoServiceInterface*, that is deployed as a Kubernetes service

Throws

- *KubernetesResourceException* – if it's not possible to get the Kubernetes service

Returns the public IP of the provided Kubernetes Service

getServiceInstanceStatus

```
public MicoServiceStatusResponseDTO getServiceInstanceStatus (MicoServiceDeploymentInfo ser-
                                                                                   viceDeploymentInfo)
```

Get status information for a single *MicoServiceDeploymentInfo* if it is deployed: # available replicas, # requested replicas, pod metrics (CPU load, memory usage).

Parameters

- **serviceDeploymentInfo** – the *MicoServiceDeploymentInfo*.

Returns the *MicoServiceStatusResponseDTO* which contains status information for a specific instance of a *MicoService*.

getServiceInterfaceStatus

```
public List<MicoServiceInterfaceStatusResponseDTO> getServiceInterfaceStatus (MicoServiceDeploymentInfo
                                                                              serviceDe-
                                                                              ploymentInfo,
                                                                              List<MicoMessageResponseDTO>
                                                                              errorMes-
                                                                              sages)
```

Get the status information for all *MicoServiceInterfaces* of the *MicoService*.

Parameters

- **serviceDeploymentInfo** – is the *MicoServiceDeploymentInfo* that includes the service for which the status information of the *MicoServiceInterfaces* is requested.
- **errorMessages** – is the list of error messages, which is empty if no error occurs.

Returns a list of *MicoServiceInterfaceStatusResponseDTO*, one DTO per *MicoServiceInterface*.

getServiceStatus

```
public List<MicoServiceStatusResponseDTO> getServiceStatus (MicoService micoService)
```

Get status information for all instances of a *MicoService* and return them as a list: # available replicas, # requested replicas, pod metrics (CPU load, memory usage).

Parameters

- **micoService** – the *MicoService*.

Returns the list of *MicoServiceStatusResponseDTOs* which contains status information for all instances of a *MicoService*.

2.2.14 io.github.ust.mico.core.service.imagebuilder

ImageBuilder

```
public interface ImageBuilder
```

Methods**build**

```
CompletableFuture<String> build (MicoService micoService)
```

init

void **init** (ContextRefreshedEvent *cre*)

init

void **init** ()

isInitialized

boolean **isInitialized** ()

TektonPipelinesController

public class **TektonPipelinesController** implements *ImageBuilder*

Builds container images by using Tekton Pipelines and Kaniko.

Constructors

TektonPipelinesController

public **TektonPipelinesController** (KubernetesClient *kubernetesClient*, *MicoKubernetesBuildBotConfig* *buildBotConfig*, *KubernetesNameNormalizer* *kubernetesNameNormalizer*)

Create a *ImageBuilder* to be able to build Docker images in the cluster.

Parameters

- **kubernetesClient** – the *KubernetesClient*
- **buildBotConfig** – the build bot configuration for the image builder
- **kubernetesNameNormalizer** – the *KubernetesNameNormalizer*

Methods

build

public *CompletableFuture<String>* **build** (*MicoService* *micoService*)

Builds an OCI image based on a Git repository provided by a *MicoService*. The result of the returned *CompletableFuture* is the Docker image URI.

Parameters

- **micoService** – the MICO service for which the image should be build

Throws

- *NotInitializedException* – if the image builder was not initialized

Returns the *CompletableFuture* that executes the build. The result is the Docker image URI.

createImageUrl

```
public String createImageUrl (String serviceName)
```

Creates an image name based on the DockerHub registry name and service's short name.

Parameters

- **serviceName** – the short name of the *MicoService*.

Returns the image name.

init

```
public void init (ContextRefreshedEvent cre)
```

Initialize the image builder every time the application context is refreshed.

Parameters

- **cre** – the ContextRefreshedEvent

init

```
public void init ()
```

Initialize the image builder. This is required to be able to use the image builder. It's not required to trigger the initialization manually, because the method is triggered by application context refresh events.

Throws

- *NotInitializedException* – if there are errors during initialization

initilizeBuildPipeline

```
public void initilizeBuildPipeline (String namespace)
```

Initialize the Tekton build-and-push pipeline. This is required to be able to use the image builder.

Parameters

- **namespace** – the namespace for Tekton pipeline resources.

Throws

- *IOException* – if there are errors when reading Tekton definition files

2.2.15 io.github.ust.mico.core.util

CollectionUtils

```
public class CollectionUtils
```

Provides some utility functions for easy creation of collections.

Methods

distinctByKey

public static <T> Predicate<T> **distinctByKey** (Function<? super T, ?> *keyExtractor*)
Inspired by Stuart Marks <https://stackoverflow.com/a/27872852/9556565>

listOf

public final <T> List<T> **listOf** (T... *items*)

mapOf

public final <K, V> Map<K, V> **mapOf** (K *key*, V *value*)

mapOf

public final <K, V> Map<K, V> **mapOf** (K *k1*, V *v1*, K *k2*, V *v2*)

mapOf

public final <K, V> Map<K, V> **mapOf** (K *k1*, V *v1*, K *k2*, V *v2*, K *k3*, V *v3*)

mapOf

public final <K, V> Map<K, V> **mapOf** (K *k1*, V *v1*, K *k2*, V *v2*, K *k3*, V *v3*, K *k4*, V *v4*)

FutureUtils

public class **FutureUtils**
Provides some utility functions for `Future` / `CompletableFuture`.

Methods

all

public static <T> CompletableFuture<List<T>> **all** (List<CompletableFuture<T>> *futures*)
Waits for *all* futures to complete and returns a list of results. If *any* future completes exceptionally then the resulting future will also complete exceptionally.

Parameters

- **futures** – the `CompletableFuture`s
- **<T>** – the generic type of the `CompletableFuture`s

Returns the list of `CompletableFuture`s

KubernetesNameNormalizer

public class **KubernetesNameNormalizer**
Normalizes names to be valid Kubernetes resource names.

Fields

MICO_NAME_MAX_SIZE

public static final int **MICO_NAME_MAX_SIZE**
A max limit of the MICO names (*MicoApplication*, *MicoService* and *MicoServiceInterface*) is required because they are used as values of Kubernetes labels that have a limit of 63. Furthermore the name is used to create a UID that adds 9 characters to it. Therefore the limit have to be set to 54.

Methods

createBuildName

public *String* **createBuildName** (*String* serviceShortName, *String* serviceVersion)
Creates a build name based on the short name and version of a service.

Parameters

- **serviceShortName** – the short name of the *MicoService*.
- **serviceVersion** – the version of the *MicoService*.

Returns the name of the *//@link* Build.

createBuildName

public *String* **createBuildName** (*MicoService* service)
Creates a build name based on a service.

Parameters

- **service** – the *MicoService*.

Returns the name of the *//@link* Build.

normalizeName

public *String* **normalizeName** (*String* name)
Normalizes a name so it is a valid Kubernetes resource name.

Returns the normalized name

Patterns

public class **Patterns**
Contains regular expressions that are used for pattern matching.

Fields

KAFKA_TOPIC_NAME_MESSAGE

public static final `String` **KAFKA_TOPIC_NAME_MESSAGE**

Message is used if a match with the `Patterns.KAFKA_TOPIC_NAME_REGEX` fails.

KAFKA_TOPIC_NAME_REGEX

public static final `String` **KAFKA_TOPIC_NAME_REGEX**

Kafka topic names must only contain letters, numbers, dots, underscores and minus symbols.

KUBERNETES_ENV_VAR_NAME_MESSAGE

public static final `String` **KUBERNETES_ENV_VAR_NAME_MESSAGE**

Message is used if a match with the `Patterns.KUBERNETES_ENV_VAR_NAME_REGEX` fails.

KUBERNETES_ENV_VAR_NAME_REGEX

public static final `String` **KUBERNETES_ENV_VAR_NAME_REGEX**

Kubernetes environment variable names must only contain letters, numbers and underscores, and must not start with a digit.

KUBERNETES_LABEL_KEY_MESSAGE

public static final `String` **KUBERNETES_LABEL_KEY_MESSAGE**

Message is used if a match with the `Patterns.KUBERNETES_LABEL_KEY_REGEX` fails.

KUBERNETES_LABEL_KEY_REGEX

public static final `String` **KUBERNETES_LABEL_KEY_REGEX**

Valid label keys have two segments: an optional prefix and name, separated by a slash (/). The name segment is required and must be 63 characters or less, beginning and ending with an alphanumeric character ([a-z0-9A-Z]) with dashes (-), underscores (_), dots (.), and alphanumerics between. The prefix is optional. If specified, the prefix must be a DNS subdomain: a series of DNS labels separated by dots (.), not longer than 253 characters in total, followed by a slash (/).

KUBERNETES_LABEL_VALUE_MESSAGE

public static final `String` **KUBERNETES_LABEL_VALUE_MESSAGE**

Message is used if a match with the `Patterns.KUBERNETES_LABEL_VALUE_REGEX` fails.

KUBERNETES_LABEL_VALUE_REGEX

public static final `String` **KUBERNETES_LABEL_VALUE_REGEX**

Kubernetes label values must be 63 characters or less and must be empty or begin and end with an alphanumeric character ([a-z0-9A-Z]) with dashes (-), underscores (_), dots (.), and alphanumerics between.

KUBERNETES_NAMING_MESSAGE

public static final **String** **KUBERNETES_NAMING_MESSAGE**
Message is used if a match with the *Patterns.KUBERNETES_NAMING_REGEX* fails.

KUBERNETES_NAMING_REGEX

public static final **String** **KUBERNETES_NAMING_REGEX**
Kubernetes resource names must be a valid DNS-1123 subdomain.

NOT_EMPTY_REGEX

public static final **String** **NOT_EMPTY_REGEX**
Regex for strings that MUST NOT be empty.

ONLY_LETTERS_OR_EMPTY_REGEX

public static final **String** **ONLY_LETTERS_OR_EMPTY_REGEX**
Regex to ensure to only use letters (may be empty).

OPEN_FAAS_FUNCTION_NAME_MESSAGE

public static final **String** **OPEN_FAAS_FUNCTION_NAME_MESSAGE**
Message is used if a match with the *Patterns.OPEN_FAAS_FUNCTION_NAME_REGEX* fails.

OPEN_FAAS_FUNCTION_NAME_REGEX

public static final **String** **OPEN_FAAS_FUNCTION_NAME_REGEX**
OpenFaaS function names must be a valid DNS-1123 subdomain.

RELATIVE_PATH_REGEX

public static final **String** **RELATIVE_PATH_REGEX**
Regex for strings that MUST be a relative path.

SEMANTIC_VERSIONING_MESSAGE

public static final **String** **SEMANTIC_VERSIONING_MESSAGE**
Message is used if a match with the *Patterns.SEMANTIC_VERSION_WITH_PREFIX_REGEX* fails.

SEMANTIC_VERSION_REGEX

public static final **String** **SEMANTIC_VERSION_REGEX**
Regex for a semantic version.

SEMANTIC_VERSION_WITH_PREFIX_REGEX

public static final [String](#) **SEMANTIC_VERSION_WITH_PREFIX_REGEX**
Regex for a semantic version with a prefix (optional) consisting of letters.

PrometheusValueDeserializer

public class **PrometheusValueDeserializer** extends StdDeserializer<[Integer](#)>
Custom deserializer for a response, which is received from Prometheus for CPU load / memory usage requests.

Constructors

PrometheusValueDeserializer

public **PrometheusValueDeserializer** ()

Methods

deserialize

public [Integer](#) **deserialize** ([JsonParser](#) *parser*, [DeserializationContext](#) *context*)

RandomStringFactory

public class **RandomStringFactory**
Provides functionality to generate random [String](#).

Methods

randomAlphanumeric

public final [String](#) **randomAlphanumeric** ()

RestTemplates

public class **RestTemplates**

Fields

QUALIFIER_AUTHENTICATED_OPEN_FAAS_REST_TEMPLATE

public static final [String](#) **QUALIFIER_AUTHENTICATED_OPEN_FAAS_REST_TEMPLATE**

micoKubernetesClient

[MicoKubernetesClient](#) **micoKubernetesClient**

Methods

getAuthenticatedOpenFaaSRestTemplate

public RestTemplate **getAuthenticatedOpenFaaSRestTemplate** (RestTemplateBuilder *builder*)

Constructs the rest template to be able to connect the OpenFaaS Portal. It uses the OpenFaaS credentials that are stored inside a Kubernetes secret. The Spring Bean Request Scope in proxy mode `ScopedProxyMode.TARGET_CLASS` is used, so that it will be instantiated when it is needed (prevents errors during application context is loading).

Parameters

- **builder** – the RestTemplateBuilder

Returns the RestTemplate

getRestTemplate

public RestTemplate **getRestTemplate** (RestTemplateBuilder *builder*)

Prefer the not authenticated rest template

Parameters

- **builder** –

UIDUtils

public class **UIDUtils**

Methods

uidFor

public final String **uidFor** (*MicoApplication application*)

uidFor

public final String **uidFor** (*MicoService service*)

uidFor

public final String **uidFor** (*MicoServiceInterface serviceInterface*)

Code Documentation for the MICO-Core.

User documentation for [MICO](#).

List of [Architectural Design Decisions](#).

A

- `adaptUriForGitHubApi(String)` (Java method), 119
- `addKafkaFaasConnectorInstanceToApplication(String, String, String)` (Java method), 100
- `addKafkaFaasConnectorInstanceToMicoApplicationByVersion(String, String, String)` (Java method), 26
- `addMicoServiceToMicoApplicationByShortNameAndVersion(String, String, String, String, Optional)` (Java method), 27
- `addServiceToApplication(String, String, String, String)` (Java method), 101
- `addServiceToApplication(String, String, String, String, String)` (Java method), 102
- `all(List)` (Java method), 131
- `ALWAYS` (Java field), 81
- `api()` (Java method), 41
- `ApiService()` (class), 2
- `ApiService.canonizeStreamUrl()` (ApiService method), 2
- `ApiService.deleteAllApplicationVersions()` (ApiService method), 2
- `ApiService.deleteAllServiceVersions()` (ApiService method), 2
- `ApiService.deleteApplication()` (ApiService method), 2
- `ApiService.deleteApplicationKafkaFaasConnector()` (ApiService method), 2
- `ApiService.deleteApplicationServices()` (ApiService method), 2
- `ApiService.deleteService()` (ApiService method), 3
- `ApiService.deleteServiceDependee()` (ApiService method), 3
- `ApiService.deleteServiceInterface()` (ApiService method), 3
- `ApiService.getApplication()` (ApiService method), 3
- `ApiService.getApplicationDeploymentStatus()` (ApiService method), 3
- `ApiService.getApplicationKafkaFaasConnector()` (ApiService method), 4
- `ApiService.getApplications()` (ApiService method), 4
- `ApiService.getApplicationStatus()` (ApiService method), 4
- `ApiService.getApplicationVersions()` (ApiService method), 4
- `ApiService.getJobStatus()` (ApiService method), 4
- `ApiService.getModelDefinitions()` (ApiService method), 4
- `ApiService.getOpenFaaSFunctions()` (ApiService method), 4
- `ApiService.getOpenFaaSIP()` (ApiService method), 4
- `ApiService.getService()` (ApiService method), 5
- `ApiService.getServiceDependees()` (ApiService method), 5
- `ApiService.getServiceDependencyGraph()` (ApiService method), 5
- `ApiService.getServiceDependers()` (ApiService method), 5
- `ApiService.getServiceDeploymentInformation()` (ApiService method), 5
- `ApiService.getServiceInterfacePublicIp()` (ApiService method), 5
- `ApiService.getServiceInterfaces()` (ApiService method), 6
- `ApiService.getServices()` (ApiService method), 6
- `ApiService.getServiceStatus()` (ApiService method), 6
- `ApiService.getServiceVersions()` (ApiService method), 6
- `ApiService.getServiceVersionsViaGithub()` (ApiService method), 6
- `ApiService.getServiceYamlConfig()` (ApiService method), 6
- `ApiService.getStreamSource()` (ApiService method), 6
- `ApiService.getTopics()` (ApiService method), 7
- `ApiService.new ApiService()` (ApiService method), 7
- `ApiService.pollDeploymentJobStatus()` (ApiService method), 7
- `ApiService.polling` (ApiService attribute), 7
- `ApiService.pollServiceInterfacePublicIp()` (ApiService method), 7
- `ApiService.postApplication()` (ApiService method), 7
- `ApiService.postApplicationDeployCommand()` (ApiService method), 7

ApiService.postApplicationKafkaFaasConnector()
(ApiService method), 8

ApiService.postApplicationServices() (ApiService
method), 8

ApiService.postApplicationUndeployCommand() (ApiS-
ervice method), 8

ApiService.postService() (ApiService method), 8

ApiService.postServiceDependee() (ApiService method),
9

ApiService.postServiceInterface() (ApiService method),
9

ApiService.postServiceViaGithub() (ApiService
method), 9

ApiService.promoteApplication() (ApiService method), 9

ApiService.promoteService() (ApiService method), 9

ApiService.putApplication() (ApiService method), 9

ApiService.putApplicationKafkaFaasConnector() (ApiS-
ervice method), 10

ApiService.putService() (ApiService method), 10

ApiService.putServiceDeploymentInformation() (ApiS-
ervice method), 10

ApiService.putServiceInterface() (ApiService method),
10

ApiService.rest (ApiService attribute), 11

ApiService.snackBar (ApiService attribute), 11

ApiService.startApplicationStatusPolling() (ApiService
method), 11

ApiService.streams (ApiService attribute), 11

app/api/api.service (module), 2

app/forms (module), 14

app/forms/mico-forms (module), 14

ApplicationResource (Java class), 99

apply(ModelPropertyContext) (Java method), 42

applyValuesFrom(KubernetesDeploymentInfoResponseDTO)
(Java method), 69

applyValuesFrom(MicoServiceDeploymentInfoRequestDTO)
(Java method), 81

applyValuesFrom(MicoServiceDeploymentInfoResponseDTO)
(Java method), 81

B

BackgroundJobBroker (Java class), 19

BackgroundJobBroker(MicoBackgroundJobRepository,
MicoApplicationRepository) (Java construc-
tor), 19

BackgroundJobResource (Java class), 106

BUILD (Java field), 80

build(MicoService) (Java method), 128, 129

C

checkIfDependencyAlreadyExists(MicoService, Mi-
coService) (Java method), 33

checkIfKafkaFaasConnectorIsDeployable(MicoServiceDeploymentInfo)
(Java method), 22

cleanUp() (Java method), 90, 92, 93, 98, 99

cleanUpTanglingNodes() (Java method), 35

CollectionUtils (Java class), 130

compareTo(MicoVersion) (Java method), 84

compareWithBuildsTo(MicoVersion) (Java method), 84

copyAndUpgradeMicoApplicationByShortNameAndVersion(String,
String, String) (Java method), 28

CorsConfig (Java class), 39

corsFilter() (Java method), 42

corsUserConfig (Java field), 42

crawlGitHubRepoLatestRelease(String) (Java method),
119

crawlGitHubRepoLatestRelease(String, String) (Java
method), 119

crawlGitHubRepoSpecificRelease(String, String) (Java
method), 119

crawlGitHubRepoSpecificRelease(String, String, String)
(Java method), 119

CrawlingInfoRequestDTO (Java class), 42

createApplication(MicoApplicationRequestDTO) (Java
method), 102

createBuildName(MicoService) (Java method), 132

createBuildName(String, String) (Java method), 132

createImageUrl(String) (Java method), 130

createMicoApplication(MicoApplication) (Java method),
28

createMicoServiceInstance(MicoServiceDeploymentInfo)
(Java method), 121

createMicoServiceInterface(MicoServiceInterface, Mi-
coServiceDeploymentInfo) (Java method),
121

createNewDependee(String, String, String, String) (Java
method), 115

createOrReuseOpenFaaSFunctionsInDatabase(MicoServiceDeploymentInfo)
(Java method), 35

createOrReuseTopic(String) (Java method), 35

createOrReuseTopics(List) (Java method), 35

createOrReuseTopicsInDatabase(MicoServiceDeploymentInfo)
(Java method), 36

createOrUpdateInterfaceConnections(MicoApplication)
(Java method), 121

createService(MicoServiceRequestDTO) (Java method),
115

createServiceInterface(String, String, MicoServiceInter-
faceRequestDTO) (Java method), 112

createServiceName(MicoServiceDeploymentInfo, Mi-
coServiceInterface) (Java method), 121

CustomOpenApiExtentionsPlugin (Java class), 42

D

DEAD_LETTER (Java field), 83

DefaultNames (Java enum), 75

deleteAllByApplication(String) (Java method), 93

deleteAllByApplication(String, String) (Java method), 93

- deleteAllDependees(MicoService) (Java method), 33
- deleteAllDependees(String, String) (Java method), 115
- deleteAllJobs() (Java method), 19, 106
- deleteAllVersionsOfApplication(String) (Java method), 102
- deleteAllVersionsOfService(String) (Java method), 33, 115
- deleteApplication(String, String) (Java method), 102
- deleteByApplicationAndService(String, String, String) (Java method), 94
- deleteByApplicationAndService(String, String, String, String) (Java method), 94
- deleteByServiceAndName(String, String, String) (Java method), 96
- deleteDependee(String, String, String, String) (Java method), 115
- deleteDependencyBetweenServices(MicoService, MicoService) (Java method), 33
- deleteJob(String) (Java method), 19, 106
- deleteKafkaFaasConnectorInstanceFromApplication(String, String, String) (Java method), 103
- deleteKafkaFaasConnectorInstancesFromApplication(String, String) (Java method), 103
- deleteMicoApplicationByShortNameAndVersion(String, String) (Java method), 28
- deleteMicoApplicationsByShortName(String) (Java method), 28
- deleteMicoServiceInterface(MicoService, String) (Java method), 38
- deleteService(MicoService) (Java method), 33
- deleteService(String, String) (Java method), 115
- deleteServiceByShortNameAndVersion(String, String) (Java method), 97
- deleteServiceFromApplication(String, String, String) (Java method), 103
- deleteServiceInterface(String, String, String) (Java method), 112
- deploy(String, String, boolean) (Java method), 106
- deployApplication(String, String, boolean) (Java method), 22
- DEPLOYED (Java field), 73
- deployed(List) (Java method), 70
- deployed(String) (Java method), 70
- deployed(String, Type) (Java method), 70
- DeploymentBroker (Java class), 22
- DeploymentException (Java class), 55
- DeploymentException() (Java constructor), 55
- DeploymentException(String) (Java constructor), 55
- DeploymentRequirementsOfKafkaFaasConnectorNotMetException (Java class), 55
- DeploymentRequirementsOfKafkaFaasConnectorNotMetException() (Java constructor), 56
- DeploymentRequirementsOfKafkaFaasConnectorNotMetException(String) (Java constructor), 55
- DeploymentRequirementsOfKafkaFaasConnectorNotMetException(String) (Java constructor), 55
- DeploymentResource (Java class), 106
- deserialize(JsonParser, DeserializationContext) (Java method), 135
- distinctByKey(Function) (Java method), 131
- DOCKER (Java field), 80
- DONE (Java field), 79
- ## E
- environment variable
- MICO_REST_API, 1
- equals(Object) (Java method), 84
- ERROR (Java field), 77, 79
- error(String) (Java method), 77
- externalUrl (Java field), 46
- ExternalUrlDTO (Java class), 46
- ## F
- findAll() (Java method), 90, 91, 97
- findAll(int) (Java method), 90, 97
- findAllByApplication(String, String) (Java method), 94, 97
- findAllByApplicationAndServiceShortName(String, String, String) (Java method), 97
- findAllByName(String) (Java method), 99
- findAllByService(String, String) (Java method), 95
- findAllByUsedService(String, String) (Java method), 90
- findAllByUsedServiceInstance(String) (Java method), 91
- findByApplicationAndService(String, String, String) (Java method), 95
- findByApplicationAndService(String, String, String, String) (Java method), 95
- findById(String) (Java method), 91, 95
- findByIdAndType(String, MicoServiceBackgroundJob.Type) (Java method), 91
- findByName(String) (Java method), 99
- findByService(String, String) (Java method), 96
- findByServiceAndName(String, String, String) (Java method), 96
- findByServiceShortNameAndServiceVersion(String, String) (Java method), 91
- findByServiceShortNameAndServiceVersionAndType(String, String, MicoServiceBackgroundJob.Type) (Java method), 92
- findByShortName(String) (Java method), 91, 97
- findByShortNameAndVersion(String, String) (Java method), 91, 98
- findDependees(String, String) (Java method), 98
- findDependeesIncludeDepender(String, String) (Java method), 98
- findDependers(MicoService) (Java method), 33
- findDependers(MicoServiceDeploymentInfo) (Java method), 98
- findFirstByName(String) (Java method), 99

`findKfConnectorSDIsByApplication(String, String)` (Java method), 96

`findMicoServiceSDIsByApplication(String, String)` (Java method), 96

`forIntegers(int, int, int)` (Java method), 85

`forIntegersWithPrefix(String, int, int, int)` (Java method), 85

`FormGroupService()` (class), 14

`FormGroupService.modelToFormGroup()` (FormGroupService method), 14

`FormGroupService.new FormGroupService()` (FormGroupService method), 14

`FUNCTIONS_PATH` (Java field), 109

`FutureUtils` (Java class), 131

G

`getAllApplications()` (Java method), 103

`getAllJobs()` (Java method), 20, 106

`getAllServicesAsList()` (Java method), 33

`getAllTopics()` (Java method), 39, 118

`getAllVersionsOfServiceFromDatabase(String)` (Java method), 33

`getApplicationByShortNameAndVersion(String, String)` (Java method), 103

`getApplicationDeploymentStatus(MicoApplication)` (Java method), 121

`getApplicationDeploymentStatus(String, String)` (Java method), 28, 103

`getApplicationsByShortName(String)` (Java method), 104

`getApplicationStatus(MicoApplication)` (Java method), 127

`getApplicationStatus(String, String)` (Java method), 28

`getAuthenticatedOpenFaaSRestTemplate(RestTemplateBuilder)` (Java method), 136

`getBuildMetadata()` (Java method), 85

`getDefaultEnvironmentVariablesForKafka()` (Java method), 40

`getDefaultEnvironmentVariablesForOpenFaaS()` (Java method), 41

`getDefaultTopics(MicoServiceDeploymentInfo)` (Java method), 40

`getDependees(String, String)` (Java method), 116

`getDependeesByMicoService(MicoService)` (Java method), 33

`getDependencyGraph(MicoService)` (Java method), 34

`getDependencyGraph(String, String)` (Java method), 116

`getDependers(MicoService)` (Java method), 34

`getDependers(String, String)` (Java method), 116

`getDeploymentOfMicoServiceInstance(MicoServiceDeploymentInfo)` (Java method), 122

`getDeploymentsOfMicoService(MicoService)` (Java method), 122

`getExistingServiceDeploymentInfo(MicoApplication, MicoService)` (Java method), 36

`getExternalAddress()` (Java method), 39

`getInterfaceByName(String, String, String)` (Java method), 112

`getInterfaceByNameOfMicoServiceInstance(MicoServiceDeploymentInfo, String)` (Java method), 122

`getInterfaceOfServiceByName(String, String, String)` (Java method), 38

`getInterfacePublicIpByName(String, String, String, String)` (Java method), 113

`getInterfacesOfMicoService(MicoService)` (Java method), 123

`getInterfacesOfMicoServiceInstance(MicoServiceDeploymentInfo)` (Java method), 123

`getInterfacesOfService(String, String)` (Java method), 38, 113

`getJobById(String)` (Java method), 20, 106

`getJobByMicoServiceInstanceId(String, MicoServiceBackgroundJob.Type)` (Java method), 20

`getJobsByMicoService(String, String, MicoServiceBackgroundJob.Type)` (Java method), 21

`getJobStatusByApplicationShortNameAndVersion(String, String)` (Java method), 20, 106

`getKafkaFaasConnectorDeploymentInformation(String, String)` (Java method), 23, 107

`getKafkaFaasConnectorDeploymentInformation(String, String, String)` (Java method), 24

`getKafkaFaasConnectorDeploymentInformationInstance(String, String, String)` (Java method), 107

`getKfConnectorDeploymentInfoResponseDTOResource(String, String, MicoServiceDeploymentInfo)` (Java method), 108

`getLatestKfConnectorVersion()` (Java method), 34

`getLinksOfMicoApplication(MicoApplication)` (Java method), 29

`getMajorVersion()` (Java method), 85

`getMicoApplicationByShortNameAndVersion(String, String)` (Java method), 29

`getMicoApplicationForMicoService(String, String, String)` (Java method), 29

`getMicoApplications()` (Java method), 29

`getMicoApplicationsByShortName(String)` (Java method), 29

`getMicoApplicationsUsingMicoService(String, String)` (Java method), 29

`getMicoServiceDeploymentInformation(String, String, String)` (Java method), 36

`getMicoServicesOfMicoApplicationByShortNameAndVersion(String, String)` (Java method), 30

`getMinorVersion()` (Java method), 69, 78

`getMinorVersion()` (Java method), 86

`getOpenFaasCredentials()` (Java method), 123

`getOpenFaasFunctions()` (Java method), 110

`getOpenFaasURL()` (Java method), 110

`getPatchVersion()` (Java method), 86

- getPodsCreatedByDeploymentOfMicoServiceInstance(MicoServiceDeploymentInfo) (Java method), 123
- getPreReleaseVersion() (Java method), 86
- getPublicIpOfKubernetesService(MicoServiceDeploymentInfo, MicoServiceInterface) (Java method), 127
- getPublicIpOfKubernetesService(String, String) (Java method), 123
- getPublicPortsOfKubernetesService(String, String) (Java method), 124
- getRestTemplate(RestTemplateBuilder) (Java method), 136
- getService(String, String) (Java method), 124
- getServiceById(Long) (Java method), 34
- getServiceByShortNameAndVersion(String, String) (Java method), 116
- getServiceDeploymentInformation(String, String, String) (Java method), 110
- getServiceFromDatabase(String, String) (Java method), 34
- getServiceInstanceFromDatabase(String, String, String) (Java method), 34
- getServiceInstanceStatus(MicoServiceDeploymentInfo) (Java method), 127
- getServiceInterfaceStatus(MicoServiceDeploymentInfo, List) (Java method), 128
- getServiceLinks(MicoService) (Java method), 116
- getServiceList() (Java method), 116
- getServiceResponseDTOResource(MicoService) (Java method), 116
- getServiceResponseDTOResourcesList(List) (Java method), 116
- getServicesOfApplication(String, String) (Java method), 104
- getServiceStatus(MicoService) (Java method), 128
- getServiceYamlByShortNameAndVersion(String, String) (Java method), 34, 117
- getStatusListOfService(String, String) (Java method), 117
- getStatusOfApplication(String, String) (Java method), 104
- getStatusOfServiceInstance(String, String, String) (Java method), 117
- getVersionsFromGitHub(String) (Java method), 117
- getVersionsFromGitHubRepo(String) (Java method), 119
- getVersionsOfService(String) (Java method), 117
- getYaml(MicoService) (Java method), 124
- getYaml(MicoServiceDeploymentInfo) (Java method), 125
- GITHUB (Java field), 80
- GitHubCrawler (Java class), 118
- gitHubCrawler (Java field), 18
- GitHubCrawler(RestTemplate, KubernetesNameNormalizer) (Java constructor), 119
- GlobalExceptionHandler (Java class), 56
- GreaterThanOrEqualTo(MicoVersion) (Java method), 86
- ## H
- hashCode() (Java method), 86
- health() (Java method), 120
- HTTP_STATUS (Java field), 53
- ## I
- IF_NOT_PRESENT (Java field), 81
- ImageBuilder (Java interface), 128
- ImageBuilderHealthIndicator (Java class), 119
- ImageBuilderHealthIndicator(TektonPipelinesController) (Java constructor), 119
- ImageBuildException (Java class), 56
- ImageBuildException() (Java constructor), 56
- ImageBuildException(String) (Java constructor), 56
- ImagePullPolicy (Java enum), 81
- importMicoServiceFromGitHub(CrawlingInfoRequestDTO) (Java method), 118
- INCOMPLETE (Java field), 74
- incomplete(List) (Java method), 71
- incomplete(String) (Java method), 71
- incomplete(String, Type) (Java method), 71
- incrementBuildMetadata() (Java method), 87
- incrementMajorVersion() (Java method), 87
- incrementMajorVersion(String) (Java method), 87
- incrementMinorVersion() (Java method), 87
- incrementMinorVersion(String) (Java method), 87
- incrementPatchVersion() (Java method), 88
- incrementPatchVersion(String) (Java method), 88
- incrementPreReleaseVersion() (Java method), 88
- INFO (Java field), 78
- info(String) (Java method), 77
- init() (Java method), 129, 130
- init(ContextRefreshedEvent) (Java method), 129, 130
- initilizeBuildPipeline(String) (Java method), 130
- INPUT (Java field), 83
- INVALID_MESSAGE (Java field), 83
- io.github.ust.mico.core (package), 18
- io.github.ust.mico.core.broker (package), 19
- io.github.ust.mico.core.configuration (package), 39
- io.github.ust.mico.core.configuration.extension (package), 42
- io.github.ust.mico.core.dto.request (package), 42
- io.github.ust.mico.core.dto.response (package), 46
- io.github.ust.mico.core.dto.response.internal (package), 53
- io.github.ust.mico.core.dto.response.status (package), 53
- io.github.ust.mico.core.exception (package), 55
- io.github.ust.mico.core.model (package), 69
- io.github.ust.mico.core.persistence (package), 90
- io.github.ust.mico.core.resource (package), 99
- io.github.ust.mico.core.service (package), 118

`io.github.ust.mico.core.service.imagebuilder` (package), 128
`io.github.ust.mico.core.util` (package), 130
`isApplicationDeployed(MicoApplication)` (Java method), 125
`isApplicationUndeployed(MicoApplication)` (Java method), 125
`isExternalUrlAvailable` (Java field), 46
`isInitialized()` (Java method), 129
`isMicoServiceDeployed(MicoService)` (Java method), 125
`isMicoServiceInstanceDeployed(MicoServiceDeploymentInfo)` (Java method), 125

K

`KAFKA_BOOTSTRAP_SERVERS` (Java field), 75
`KAFKA_GROUP_ID` (Java field), 75
`KAFKA_TOPIC_DEAD_LETTER` (Java field), 75
`KAFKA_TOPIC_INPUT` (Java field), 75
`KAFKA_TOPIC_INVALID_MESSAGE` (Java field), 75
`KAFKA_TOPIC_NAME_MESSAGE` (Java field), 133
`KAFKA_TOPIC_NAME_REGEX` (Java field), 133
`KAFKA_TOPIC_OUTPUT` (Java field), 75
`KAFKA_TOPIC_TEST_MESSAGE_OUTPUT` (Java field), 75
`KafkaConfig` (Java class), 39
`KafkaFaasConnectorConfig` (Java class), 40
`kafkaFaasConnectorConfig` (Java field), 18
`KafkaFaasConnectorDeploymentInfoBroker` (Java class), 23
`KafkaFaasConnectorDeploymentInfoResource` (Java class), 107
`KafkaFaasConnectorInstanceNotFoundException` (Java class), 56
`KafkaFaasConnectorInstanceNotFoundException()` (Java constructor), 57
`KafkaFaasConnectorInstanceNotFoundException(String)` (Java constructor), 56
`KafkaFaasConnectorInstanceNotFoundException(String, String)` (Java constructor), 56
`KafkaFaasConnectorLatestVersionNotFound` (Java class), 57
`KafkaFaasConnectorLatestVersionNotFound()` (Java constructor), 57
`KafkaFaasConnectorNotAllowedHereException` (Java class), 57
`KafkaFaasConnectorNotAllowedHereException()` (Java constructor), 57
`KafkaFaasConnectorVersionNotFoundException` (Java class), 57
`KafkaFaasConnectorVersionNotFoundException()` (Java constructor), 57
`KafkaFaasConnectorVersionNotFoundException(String)` (Java constructor), 57

`KFConnectorDeploymentInfoRequestDTO` (Java class), 43
`KFConnectorDeploymentInfoRequestDTO(MicoServiceDeploymentInfo)` (Java constructor), 43
`KFConnectorDeploymentInfoResponseDTO` (Java class), 47
`KFConnectorDeploymentInfoResponseDTO(MicoServiceDeploymentInfo)` (Java constructor), 47
`KUBERNETES_ENV_VAR_NAME_MESSAGE` (Java field), 133
`KUBERNETES_ENV_VAR_NAME_REGEX` (Java field), 133
`KUBERNETES_LABEL_KEY_MESSAGE` (Java field), 133
`KUBERNETES_LABEL_KEY_REGEX` (Java field), 133
`KUBERNETES_LABEL_VALUE_MESSAGE` (Java field), 133
`KUBERNETES_LABEL_VALUE_REGEX` (Java field), 133
`KUBERNETES_NAMING_MESSAGE` (Java field), 134
`KUBERNETES_NAMING_REGEX` (Java field), 134
`KubernetesDeploymentInfo` (Java class), 69
`KubernetesDeploymentInfoRepository` (Java interface), 90
`KubernetesDeploymentInfoResponseDTO` (Java class), 47
`KubernetesDeploymentInfoResponseDTO(KubernetesDeploymentInfo)` (Java constructor), 47
`KubernetesNameNormalizer` (Java class), 132
`KubernetesNodeMetricsResponseDTO` (Java class), 53
`KubernetesPodInformationResponseDTO` (Java class), 54
`KubernetesPodMetricsResponseDTO` (Java class), 54
`KubernetesResourceException` (Java class), 57
`KubernetesResourceException()` (Java constructor), 58
`KubernetesResourceException(String)` (Java constructor), 58
`KubernetesResourceException(String, Throwable)` (Java constructor), 58
`KubernetesResourceException(Throwable)` (Java constructor), 58

L

`lessThan(MicoVersion)` (Java method), 88
`lessThanOrEqualTo(MicoVersion)` (Java method), 88
`listOf(T)` (Java method), 131
`logFilter()` (Java method), 41

M

`main(String[])` (Java method), 18
`mapOf(K, V)` (Java method), 131
`mapOf(K, V, K, V)` (Java method), 131
`mapOf(K, V, K, V, K, V)` (Java method), 131
`mapOf(K, V, K, V, K, V, K, V)` (Java method), 131

[methodArgumentNotValidException\(MethodArgumentNotValidException\) \(Java method\), 56](#)
[MICO_NAME_MAX_SIZE \(Java field\), 132](#)
[MICO_REST_API, 1](#)
[MicoApplication \(Java class\), 69](#)
[MicoApplicationAlreadyExistsException \(Java class\), 58](#)
[MicoApplicationAlreadyExistsException\(\) \(Java constructor\), 58](#)
[MicoApplicationAlreadyExistsException\(Long\) \(Java constructor\), 58](#)
[MicoApplicationAlreadyExistsException\(String, String\) \(Java constructor\), 58](#)
[MicoApplicationBroker \(Java class\), 25](#)
[MicoApplicationDeploymentStatus \(Java class\), 70](#)
[MicoApplicationDeploymentStatusResponseDTO \(Java class\), 54](#)
[MicoApplicationDeploymentStatusResponseDTO\(MicoApplicationDeploymentStatus\) \(Java constructor\), 54](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException \(Java class\), 58](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException\(\) \(Java constructor\), 59](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException\(Long\) \(Java constructor\), 59](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException\(String\) \(Java constructor\), 59](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException\(String, String\) \(Java constructor\), 59](#)
[MicoApplicationDoesNotIncludeKFConnectorInstanceException\(String, String, String\) \(Java constructor\), 59](#)
[MicoApplicationDoesNotIncludeMicoServiceException \(Java class\), 59](#)
[MicoApplicationDoesNotIncludeMicoServiceException\(\) \(Java constructor\), 60](#)
[MicoApplicationDoesNotIncludeMicoServiceException\(Long\) \(Java constructor\), 60](#)
[MicoApplicationDoesNotIncludeMicoServiceException\(String, String\) \(Java constructor\), 60](#)
[MicoApplicationDoesNotIncludeMicoServiceException\(String, String, String\) \(Java constructor\), 60](#)
[MicoApplicationDoesNotIncludeMicoServiceException\(String, String, String, String\) \(Java constructor\), 60](#)
[MicoApplicationIsDeployingException \(Java class\), 60](#)
[MicoApplicationIsDeployingException\(\) \(Java constructor\), 60](#)
[MicoApplicationIsDeployingException\(String, String\) \(Java constructor\), 60](#)
[MicoApplicationIsNotUndeployedException \(Java class\), 60](#)
[MicoApplicationIsNotUndeployedException\(\) \(Java constructor\), 61](#)
[MicoApplicationIsNotUndeployedException\(String, String\) \(Java constructor\), 61](#)
[MicoApplicationJobStatus \(Java class\), 74](#)
[MicoApplicationJobStatusResponseDTO \(Java class\), 47](#)
[MicoApplicationJobStatusResponseDTO\(MicoApplicationJobStatus\) \(Java constructor\), 47](#)
[MicoApplicationNotFoundException \(Java class\), 61](#)
[MicoApplicationNotFoundException\(\) \(Java constructor\), 61](#)
[MicoApplicationNotFoundException\(Long\) \(Java constructor\), 61](#)
[MicoApplicationNotFoundException\(String\) \(Java constructor\), 61](#)
[MicoApplicationNotFoundException\(String, String\) \(Java constructor\), 61](#)
[MicoApplicationRepository \(Java interface\), 90](#)
[MicoApplicationRequestDTO \(Java class\), 43](#)
[MicoApplicationRequestDTO\(MicoApplication\) \(Java constructor\), 43](#)
[MicoApplicationResponseDTO \(Java class\), 48](#)
[MicoApplicationResponseDTO\(MicoApplication\) \(Java constructor\), 48](#)
[MicoApplicationResponseDTO\(MicoApplication, MicoApplicationDeploymentStatus\) \(Java constructor\), 48](#)
[MicoApplicationStatusResponseDTO \(Java class\), 54](#)
[MicoApplicationWithServicesResponseDTO \(Java class\), 48](#)
[MicoApplicationWithServicesResponseDTO\(MicoApplication\) \(Java constructor\), 48](#)
[MicoApplicationWithServicesResponseDTO\(MicoApplication, MicoApplicationDeploymentStatus\) \(Java constructor\), 49](#)
[MicoBackgroundJobRepository \(Java interface\), 91](#)
[MicoCoreApplication \(Java class\), 18](#)
[MicoEnvironmentVariable \(Java class\), 74](#)
[MicoEnvironmentVariableRepository \(Java interface\), 92](#)
[MicoEnvironmentVariableRequestDTO \(Java class\), 43](#)
[MicoEnvironmentVariableRequestDTO\(MicoEnvironmentVariable\) \(Java constructor\), 43](#)
[MicoEnvironmentVariableResponseDTO \(Java class\), 49](#)
[MicoEnvironmentVariableResponseDTO\(MicoEnvironmentVariable\) \(Java constructor\), 49](#)
[MicoFormComponent\(\) \(class\), 14](#)
[MicoFormComponent._startData \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.data \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.debug \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.filter \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.form \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.formGroup \(MicoFormComponent attribute\), 15](#)
[MicoFormComponent.formSubscription \(MicoForm-](#)

- Component attribute), 15
- MicoFormComponent.isBlacklist (MicoFormComponent attribute), 15
- MicoFormComponent.model (MicoFormComponent attribute), 15
- MicoFormComponent.models (MicoFormComponent attribute), 15
- MicoFormComponent.modelUrl (MicoFormComponent attribute), 15
- MicoFormComponent.new MicoFormComponent() (MicoFormComponent method), 15
- MicoFormComponent.ngOnChanges() (MicoFormComponent method), 15
- MicoFormComponent.ngOnInit() (MicoFormComponent method), 15
- MicoFormComponent.onlyRequired (MicoFormComponent attribute), 15
- MicoFormComponent.properties (MicoFormComponent attribute), 15
- MicoFormComponent.startData (MicoFormComponent attribute), 15
- MicoFormComponent.trackByFn() (MicoFormComponent method), 15
- MicoFormComponent.valid (MicoFormComponent attribute), 16
- MicoInterfaceConnection (Java class), 76
- MicoInterfaceConnectionRepository (Java interface), 92
- MicoInterfaceConnectionRequestDTO (Java class), 44
- MicoInterfaceConnectionRequestDTO(MicoInterfaceConnectionRequestDTO) (Java constructor), 44
- MicoInterfaceConnectionResponseDTO (Java class), 49
- MicoInterfaceConnectionResponseDTO(MicoInterfaceConnectionResponseDTO) (Java constructor), 49
- MicoKubernetesBuildBotConfig (Java class), 40
- MicoKubernetesClient (Java class), 120
- micoKubernetesClient (Java field), 39, 135
- MicoKubernetesClient(MicoKubernetesConfig, MicoKubernetesBuildBotConfig, KubernetesClient, TektonPipelinesController, BackgroundJobBroker, MicoApplicationRepository, MicoServiceDeploymentInfoRepository, KubernetesDeploymentInfoRepository) (Java constructor), 120
- MicoKubernetesConfig (Java class), 40
- micoKubernetesConfig (Java field), 39
- MicoLabel (Java class), 76
- MicoLabelRepository (Java interface), 92
- MicoLabelRequestDTO (Java class), 44
- MicoLabelRequestDTO(MicoLabel) (Java constructor), 44
- MicoLabelResponseDTO (Java class), 49
- MicoLabelResponseDTO(MicoLabel) (Java constructor), 50
- MicoMessage (Java class), 77
- MicoMessageResponseDTO (Java class), 54
- MicoMessageResponseDTO(MicoMessage) (Java constructor), 54
- MicoPortType (Java enum), 78
- MicoRedisConfiguration (Java class), 40
- MicoRequestLoggingFilter (Java class), 41
- MicoService (Java class), 78
- MicoServiceAddedMoreThanOnceToMicoApplicationException (Java class), 61
- MicoServiceAddedMoreThanOnceToMicoApplicationException() (Java constructor), 62
- MicoServiceAddedMoreThanOnceToMicoApplicationException(String, String) (Java constructor), 62
- MicoServiceAddedMoreThanOnceToMicoApplicationException(String, String, String, String) (Java constructor), 61
- MicoServiceAlreadyAddedToMicoApplicationException (Java class), 62
- MicoServiceAlreadyAddedToMicoApplicationException() (Java constructor), 62
- MicoServiceAlreadyAddedToMicoApplicationException(Long, Long) (Java constructor), 62
- MicoServiceAlreadyAddedToMicoApplicationException(String, String, String, String) (Java constructor), 62
- MicoServiceAlreadyExistsException (Java class), 62
- MicoServiceAlreadyExistsException() (Java constructor), 63
- MicoServiceAlreadyExistsException(Long) (Java constructor), 62
- MicoServiceAlreadyExistsException(String, String) (Java constructor), 62
- MicoServiceAlreadyExistsException(String, String, String) (Java constructor), 62
- MicoServiceBackgroundJob (Java class), 79
- MicoServiceBackgroundJobResponseDTO (Java class), 50
- MicoServiceBackgroundJobResponseDTO(MicoServiceBackgroundJob) (Java constructor), 50
- MicoServiceBroker (Java class), 33
- micoServiceBroker (Java field), 18
- MicoServiceCrawlingOrigin (Java enum), 80
- MicoServiceDependency (Java class), 80
- MicoServiceDependencyGraphEdgeResponseDTO (Java class), 50
- MicoServiceDependencyGraphEdgeResponseDTO(MicoService, MicoService) (Java constructor), 50
- MicoServiceDependencyGraphResponseDTO (Java class), 51
- MicoServiceDependencyRepository (Java interface), 93
- MicoServiceDeploymentInfo (Java class), 80
- MicoServiceDeploymentInfoBroker (Java class), 35
- MicoServiceDeploymentInfoRepository (Java interface), 93
- MicoServiceDeploymentInfoRequestDTO (Java class), 44
- MicoServiceDeploymentInfoRequestDTO(MicoServiceDeploymentInfo) (Java constructor), 44

MicoServiceDeploymentInfoResponseDTO (Java class), 51	MicoServiceIsDeployedException() (Java constructor), 66
MicoServiceDeploymentInfoResponseDTO(MicoServiceDeploymentInfo) (Java constructor), 51	MicoServiceIsDeployedException(Long) (Java constructor), 66
MicoServiceDeploymentInformationNotFoundException (Java class), 63	MicoServiceIsDeployedException(String) (Java constructor), 65
MicoServiceDeploymentInformationNotFoundException(Long) (Java constructor), 63	MicoServiceIsDeployedException(String, String) (Java constructor), 65
MicoServiceDeploymentInformationNotFoundException(String, String) (Java constructor), 63	MicoServiceIsUsedByMicoApplicationsException (Java class), 66
MicoServiceDeploymentInformationNotFoundException(String, String, String) (Java constructor), 63	MicoServiceIsUsedByMicoApplicationsException() (Java constructor), 66
MicoServiceHasDependersException (Java class), 63	MicoServiceIsUsedByMicoApplicationsException(String, String) (Java constructor), 66
MicoServiceHasDependersException() (Java constructor), 64	MicoServiceIsUsedByMicoApplicationsException(String, String, List) (Java constructor), 66
MicoServiceHasDependersException(Long) (Java constructor), 63	MicoServiceNotFoundException (Java class), 66
MicoServiceHasDependersException(String, String) (Java constructor), 63	MicoServiceNotFoundException() (Java constructor), 67
MicoServiceInstanceDoesNotMatchShortNameAndVersionException (Java class), 64	MicoServiceNotFoundException(Long) (Java constructor), 66
MicoServiceInstanceDoesNotMatchShortNameAndVersionException(String, String, String, String) (Java constructor), 64	MicoServiceNotFoundException(String) (Java constructor), 66
MicoServiceInstanceNotFoundException (Java class), 64	MicoServiceNotFoundException(String, String) (Java constructor), 66
MicoServiceInstanceNotFoundException() (Java constructor), 64	MicoServicePort (Java class), 82
MicoServiceInstanceNotFoundException(String, String, String) (Java constructor), 64	MicoServicePortRepository (Java interface), 96
MicoServiceInterface (Java class), 82	MicoServicePortRequestDTO (Java class), 45
MicoServiceInterfaceAlreadyExistsException (Java class), 64	MicoServicePortRequestDTO(MicoServicePort) (Java constructor), 45
MicoServiceInterfaceAlreadyExistsException() (Java constructor), 65	MicoServicePortResponseDTO (Java class), 51
MicoServiceInterfaceAlreadyExistsException(Long, String) (Java constructor), 65	MicoServicePortResponseDTO(MicoServicePort) (Java constructor), 51
MicoServiceInterfaceAlreadyExistsException(String, String, String) (Java constructor), 65	MicoServiceRepository (Java interface), 97
MicoServiceInterfaceBroker (Java class), 38	MicoServiceRequestDTO (Java class), 45
MicoServiceInterfaceNotFoundException (Java class), 65	MicoServiceRequestDTO(MicoService) (Java constructor), 45
MicoServiceInterfaceNotFoundException(String, String) (Java constructor), 65	MicoServiceResponseDTO (Java class), 52
MicoServiceInterfaceNotFoundException(String, String, String) (Java constructor), 65	MicoServiceResponseDTO(MicoService) (Java constructor), 52
MicoServiceInterfaceRepository (Java interface), 96	MicoServiceStatusResponseDTO (Java class), 55
MicoServiceInterfaceRequestDTO (Java class), 45	MicoStatusService (Java class), 126
MicoServiceInterfaceRequestDTO(MicoServiceInterface) (Java constructor), 45	MicoStatusService(PrometheusConfig, MicoKubernetesClient, RestTemplate, MicoServiceDeploymentInfoRepository, MicoApplicationRepository) (Java constructor), 127
MicoServiceInterfaceResponseDTO (Java class), 51	MicoTopic (Java class), 83
MicoServiceInterfaceResponseDTO(MicoServiceInterface) (Java constructor), 51	MicoTopicRepository (Java interface), 98
MicoServiceInterfaceStatusResponseDTO (Java class), 55	MicoTopicRequestDTO (Java class), 46
MicoServiceIsDeployedException (Java class), 65	MicoTopicRequestDTO(MicoTopicRole) (Java constructor), 46
	MicoTopicResponseDTO (Java class), 52
	MicoTopicResponseDTO(MicoTopicRole) (Java constructor), 52
	MicoTopicRole (Java class), 83

MicoTopicRoleUsedMultipleTimesException (Java class), 67
MicoTopicRoleUsedMultipleTimesException() (Java constructor), 67
MicoTopicRoleUsedMultipleTimesException(MicoTopicRole Role) (Java constructor), 67
MicoVersion (Java class), 84
MicoVersionRequestDTO (Java class), 46
MicoVersionRequestDTO(String) (Java constructor), 46
MicoYamlResponseDTO (Java class), 52
ModelsService() (class), 11
ModelsService.apiService (ModelsService attribute), 11
ModelsService.canonizeModelUri() (ModelsService method), 11
ModelsService.filterModel() (ModelsService method), 11
ModelsService.getCacheSource() (ModelsService method), 11
ModelsService.getModel() (ModelsService method), 12
ModelsService.handleArrayProperties() (ModelsService method), 12
ModelsService.handleComplexProperties() (ModelsService method), 12
ModelsService.handleNumericPropertyKeys() (ModelsService method), 12
ModelsService.handleObjectProperties() (ModelsService method), 12
ModelsService.localModels (ModelsService attribute), 12
ModelsService.mergeModels() (ModelsService method), 12
ModelsService.mergeProperties() (ModelsService method), 13
ModelsService.modelCache (ModelsService attribute), 13
ModelsService.nestedModelCache (ModelsService attribute), 13
ModelsService.new ModelsService() (ModelsService method), 13
ModelsService.onlyRequired() (ModelsService method), 13
ModelsService.resolveModel() (ModelsService method), 13
ModelsService.resolveModelLinks() (ModelsService method), 13
ModelsService.resolveProperties() (ModelsService method), 13

N

name (Java field), 53
NEVER (Java field), 81
normalizeName(String) (Java method), 132
NOT_DEFINED (Java field), 80
NOT_EMPTY_REGEX (Java field), 134
NotInitializedException (Java class), 67

NotInitializedException() (Java constructor), 67
NotInitializedException(String) (Java constructor), 67

O

onApplicationEvent(ApplicationReadyEvent) (Java method), 19
ONLY_LETTERS_OR_EMPTY_REGEX (Java field), 134
OPEN_FAAS_BASE_PATH (Java field), 109
OPEN_FAAS_FUNCTION_LIST_PATH (Java field), 109
OPEN_FAAS_FUNCTION_NAME_MESSAGE (Java field), 134
OPEN_FAAS_FUNCTION_NAME_REGEX (Java field), 134
OPEN_FAAS_SECRET_DATA_PASSWORD_NAME (Java field), 120
OPEN_FAAS_SECRET_DATA_USERNAME_NAME (Java field), 120
OPEN_FAAS_SECRET_NAME (Java field), 120
OPEN_FAAS_UI_PROTOCOL (Java field), 38
OPENFAAS_FUNCTION_NAME (Java field), 76
OPENFAAS_GATEWAY (Java field), 76
OpenFaasBroker (Java class), 38
openFaasBroker (Java field), 110
OpenFaaSConfig (Java class), 40
openFaaSConfig (Java field), 39, 110
OpenFaaSFunction (Java class), 90
OpenFaaSFunctionRepository (Java interface), 99
OpenFaasResource (Java class), 109
OUTPUT (Java field), 83

P

PATH_APPLICATIONS (Java field), 99
PATH_KAFKA_FAAS_CONNECTOR (Java field), 99
PATH_VARIABLE_INSTANCE_ID (Java field), 115
PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_INSTANCE_ID (Java field), 100
PATH_VARIABLE_KAFKA_FAAS_CONNECTOR_VERSION (Java field), 100
PATH_VARIABLE_SHORT_NAME (Java field), 100, 115
PATH_VARIABLE_VERSION (Java field), 100, 115
Patterns (Java class), 132
PENDING (Java field), 74, 79
pending(List) (Java method), 72
pending(String) (Java method), 71
pending(String, Type) (Java method), 72
persistMicoServiceInterface(MicoService, MicoServiceInterface) (Java method), 38
persistNewDependencyBetweenServices(MicoService, MicoService) (Java method), 34
persistService(MicoService) (Java method), 35
PrometheusConfig (Java class), 41

PrometheusRequestFailedException (Java class), 67
 PrometheusRequestFailedException() (Java constructor), 68
 PrometheusRequestFailedException(String) (Java constructor), 68
 PrometheusResponseDTO (Java class), 53
 PrometheusValueDeserializer (Java class), 135
 PrometheusValueDeserializer() (Java constructor), 135
 promoteApplication(String, String, MicoVersionRequestDTO) (Java method), 104
 promoteService(MicoService, String) (Java method), 35
 promoteService(String, String, MicoVersionRequestDTO) (Java method), 118

Q

QUALIFIER_AUTHENTICATED_OPEN_FAAS_REST_TEMPLATE (Java field), 135

R

randomAlphanumeric() (Java method), 135
 RandomStringFactory (Java class), 135
 redisConnectionFactory() (Java method), 40
 redisTemplate() (Java method), 40
 RELATIVE_PATH_REGEX (Java field), 134
 removeAllKafkaFaasConnectorInstancesFromMicoApplication(String) (Java method), 30
 removeKafkaFaasConnectorInstanceFromMicoApplicationByInstanceName(String, String) (Java method), 30
 removeMicoServiceFromMicoApplicationByShortNameAndVersion(String, String) (Java method), 31
 RequestLoggingFilterConfig (Java class), 41
 restTemplate (Java field), 110
 restTemplate(RestTemplateBuilder) (Java method), 19
 RestTemplates (Java class), 135
 Role (Java enum), 83
 RUNNING (Java field), 79

S

saveFutureOfJob(MicoServiceDeploymentInfo, MicoServiceBackgroundJob.Type, CompletableFuture) (Java method), 21
 saveJob(MicoServiceBackgroundJob) (Java method), 21
 saveNewStatus(MicoServiceDeploymentInfo, MicoServiceBackgroundJob.Type, MicoServiceBackgroundJob.Status) (Java method), 21
 saveNewStatus(MicoServiceDeploymentInfo, MicoServiceBackgroundJob.Type, MicoServiceBackgroundJob.Status, String) (Java method), 22
 scaleIn(MicoServiceDeploymentInfo, int) (Java method), 126
 scaleOut(MicoServiceDeploymentInfo, int) (Java method), 126
 SEMANTIC_VERSION_REGEX (Java field), 134

SEMANTIC_VERSION_WITH_PREFIX_REGEX (Java field), 135
 SEMANTIC_VERSIONING_MESSAGE (Java field), 134
 serialVersionUID (Java field), 67
 ServiceDeploymentInfoResource (Java class), 110
 ServiceInterfaceResource (Java class), 111
 ServiceResource (Java class), 114
 setBuildMetadata(String) (Java method), 89
 setDefaultDeploymentInformationForKafkaEnabledService(MicoServiceDeploymentInfo) (Java method), 37
 setPreReleaseVersion(String) (Java method), 89
 ShortNameOfMicoApplicationDoesNotMatchException (Java class), 68
 ShortNameOfMicoApplicationDoesNotMatchException() (Java constructor), 68
 shouldNotFilter(HttpServletRequest) (Java method), 41
 Status (Java enum), 79
 supports(DocumentationType) (Java method), 42
 SwaggerConfig (Java class), 41

T

TCP (Java field), 78
 TektonPipelinesController (Java class), 129
 TektonPipelinesController(KubernetesClient, MicoKubernetesBuildBotConfig, KubernetesNameNormalizer) (Java constructor), 129
 TEST_MESSAGE_OUTPUT (Java field), 84
 TOPIC_BASE_PATH (Java field), 118
 TopicBroker (Java class), 39
 TopicDTO (Java class), 53
 TopicResource (Java class), 118
 toString() (Java method), 74, 78, 82, 89
 Type (Java enum), 77, 79

U

UDP (Java field), 78
 uidFor(MicoApplication) (Java method), 136
 uidFor(MicoService) (Java method), 136
 uidFor(MicoServiceInterface) (Java method), 136
 UIDUtils (Java class), 136
 UNDEFINED (Java field), 79
 undeploy(String, String) (Java method), 107
 undeployApplication(MicoApplication) (Java method), 126
 undeployApplication(String, String) (Java method), 23
 UNDEPLOYED (Java field), 74
 undeployed(List) (Java method), 72
 undeployed(String) (Java method), 72
 undeployed(String, Type) (Java method), 72
 UNKNOWN (Java field), 74
 unknown(List) (Java method), 73
 unknown(String) (Java method), 73
 unknown(String, Type) (Java method), 73

`updateApplication(String, String, MicoApplicationRequestDTO)` (Java method), [105](#)
`updateExistingService(MicoService)` (Java method), [35](#)
`updateKafkaFaasConnectorDeploymentInfo(String, String, String, KFConnectorDeploymentInfoRequestDTO)` (Java method), [109](#)
`updateKafkaFaasConnectorDeploymentInformation(String, KFConnectorDeploymentInfoRequestDTO)` (Java method), [25](#)
`updateKafkaFaasConnectorInstanceOfApplication(String, String, String, String)` (Java method), [105](#)
`updateKafkaFaasConnectorInstanceOfMicoApplicationByVersionAndInstanceId(String, String, String, String)` (Java method), [32](#)
`updateMicoApplication(String, String, MicoApplication)` (Java method), [32](#)
`updateMicoServiceDeploymentInformation(String, String, String, MicoServiceDeploymentInfoRequestDTO)` (Java method), [37](#)
`updateMicoServiceInterface(MicoService, String, MicoServiceInterface)` (Java method), [38](#)
`updateService(String, String, MicoServiceRequestDTO)` (Java method), [118](#)
`updateServiceDeploymentInformation(String, String, String, MicoServiceDeploymentInfoRequestDTO)` (Java method), [111](#)
`updateServiceInterface(String, String, String, MicoServiceInterfaceRequestDTO)` (Java method), [114](#)

V

`ValidationErrorResponseDTO` (Java class), [53](#)
`ValidationErrorResponseDTO(List)` (Java constructor), [53](#)
`Value` (Java enum), [73](#)
`valueOf(KubernetesDeploymentInfoResponseDTO)` (Java method), [69](#)
`valueOf(MicoApplicationRequestDTO)` (Java method), [70](#)
`valueOf(MicoEnvironmentVariableRequestDTO)` (Java method), [75](#)
`valueOf(MicoInterfaceConnectionRequestDTO)` (Java method), [76](#)
`valueOf(MicoLabelRequestDTO)` (Java method), [76](#)
`valueOf(MicoServiceInterfaceRequestDTO)` (Java method), [82](#)
`valueOf(MicoServicePortRequestDTO)` (Java method), [82](#)
`valueOf(MicoServiceRequestDTO)` (Java method), [79](#)
`valueOf(MicoTopicRequestDTO, MicoServiceDeploymentInfo)` (Java method), [83](#)
`valueOf(String)` (Java method), [89](#)
`VersionNotSupportedException` (Java class), [68](#)
`VersionNotSupportedException()` (Java constructor), [68](#)
`VersionNotSupportedException(String)` (Java constructor), [68](#)

`VersionOfMicoApplicationDoesNotMatchException` (Java class), [68](#)
`VersionOfMicoApplicationDoesNotMatchException()` (Java constructor), [68](#)

W

`WARNING` (Java field), [78](#)
`warning(String)` (Java method), [77](#)
`WebConfig` (Java class), [41](#)

X

`X_MICO_CUSTOM_EXTENSION` (Java field), [42](#)

Y

`yaml` (Java field), [52](#)