

Index

Modules

 **app**

Modules

app

 app: *app*

Defined in PersistentEntity.ts:1
Defined in /home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:2
Defined in Repository.ts:4

Index

Modules

 **domain**

Modules

domain

Defined in [PersistentEntity.ts:1](#)
Defined in [/home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:2](#)
Defined in [Repository.ts:4](#)

Index

Modules

 [factory](#)
 [repository](#)

Modules

factory

 factory: [factory](#)

Defined in [/home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:2](#)

Index

Classes

 [Empty](#)
 [ObjectFactory](#)

Classes

 Empty: *Empty*

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:3>

ObjectFactory

 ObjectFactory: *ObjectFactory*

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:5>

static createFromJson

 createFromJson(type: *any*, data: *any*): *any*

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:15>

create an prototype-object from object data recursive

Parameters

- **type:** *any*
e.g. String, Number, Object, Array, YourPrototype, ... -> Own Prototypes must implement static property factoryConfiguration (core.FactoryConfiguration)

- **data:** *any*
e.g. { "id": 5, "name": "DummyObject" }

Returns *any*

- New object

static createObject

 createObject(constructor: *any*, args: *any*): *any*

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:90>


properties will fail on access, so use an empty dummy class

Parameters

- **constructor:** *any*
- **args:** *any*

Returns *any*

static **createProperty**

 `createProperty(dataItem: any, property: any): any`

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:64>

create an object property from a data item

Parameters

- **dataItem:** *any*
e.g. "DummyObject1"
- **property:** *any*
e.g. { name: "id", type: Number, subType: null }

Returns *any*

static **updateFromJson**

 `updateFromJson(item: any, type: any, data: any): any`

Defined in </home/tobias/Business/studium/modules/BA/HSR.BA.Project/app/assets/scripts/classes/domain/factory/ObjectFactory.ts:45>

updates an object with some new data

Parameters

- **item:** *any*
the item to be updated
- **type:** *any*
e.g. String, Number, Object, Array, YourPrototype, ... -> Own

- **data:** *any*
e.g. { "id": 5, "name": "DummyObject" }

Returns *any*

- Updated object

repository

 repository: *repository*

Defined in [PersistentEntity.ts:1](#)
Defined in [Repository.ts:4](#)

Index

Modules

 core

Modules

core

 core: *core*

Defined in [PersistentEntity.ts:1](#)
Defined in [Repository.ts:4](#)

Index

 **PersistentEntity**

Classes

 **Repository**

Interfaces

PersistentEntity

 PersistentEntity: *PersistentEntity*

Defined in [PersistentEntity.ts:2](#)

id

 id: *number*

Defined in [PersistentEntity.ts:3](#)

Classes

Repository

 Repository: *Repository*

Defined in [Repository.ts:5](#)

constructor

 constructor(httpService: *any*): *Repository*

Defined in [Repository.ts:67](#)

- `httpClient: any`

- Angular \$http service, used to call a remote api

Returns [Repository](#)

dataList

• `dataList: string`

Defined in [Repository.ts:35](#)

Name of the item list inside the returned js object from remote api

example

```
// api returns:

{ "items": [

  { "name": "item 1"

}, { "name": "item 2" }

] }
```

filter

• `filter: Function`

Defined in [Repository.ts:46](#)

Function to filter the items returned by the remote api

example

```
function(element) {

  return element.type

=== "domain.model.Asset";
```

```
}
```

filter function signature

```
() (element: any): boolean
```

Defined in [Repository.ts:46](#)

Parameters

- **element:** *any*
 - The current item inside the items list

Returns *boolean*

host

• host: *string*

Defined in [Repository.ts:16](#)

The host url of the remote api, will be concatenated in front of the relative resource path. Should be used in combination of proxy or host has to allow cross origin calls.

example "http://dks.eepi.ch"

httpService

• httpService: *any*

Defined in [Repository.ts:48](#)

itemCache

appDomain/repository/repositoryConsistency-

Defined in [Repository.ts:8](#)

proxy

● proxy: *Object*

Defined in [Repository.ts:24](#)

Path with {target} to be replaced by the remote url. If proxy is set, /path/to/proxy/{<http://example.host.tld/path/to/resource/>} will be called instead of <http://example.host.tld/path/to/resource/>

example "/dks/getFromDKS?url={target}"

proxy.url

● proxy.url: *string*

Defined in [Repository.ts:24](#)

resources

● resources: *Object*

Defined in [Repository.ts:62](#)

Dictionary with resource paths

example {

```

        list: {
method: 'GET', url: '/dksMapping' },

        detail: { method:
'GET', url: '/dksMapping/{id}' },

```

```
'POST', url: '/dksMapping' },

                                update: { method:
'POST', url: '/dksMapping/{id}' },

                                remove: { method:
'POST', url: '/dksMapping/{id}/delete' }

                                }
```

resources[]

```
[] (index: string): { method: string; url: string; }
```

Defined in [Repository.ts:62](#)

Parameters

- **index:** *string*

Returns *{ method: string; url: string; }*

type

● type: *any*

Defined in [Repository.ts:7](#)

add

```
⚡ add(item: T in app.domain.repository.core.Repository<T
  extends PersistentEntity>, callback: Function)
```

Defined in [Repository.ts:136](#)

Parameters

- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*
 - The item to persist using the remote api
- **callback:** *Function*
 - The callback is called with (true, item) on success and with (false, null) on error

- **callback function signature**

```
() (success: boolean, item: T in app.domain.repository.core.Repository<T extends PersistentEntity>)
```

Defined in [Repository.ts:136](#)

Parameters

- **success:** *boolean*
- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*

findAll

```
() findAll(callback: Function, doCache?: boolean = false)
```

Defined in [Repository.ts:92](#)

Find all items in remote repository or local cache.

Parameters

- **callback:** *Function*
 - Will be called with (true, items) on success successful remote/cache call and with (false, []) on error

- **callback function signature**

```
() (success: boolean, items: Array<T extends app.domain.repository.core.PersistentEntity>)
```

Defined in [Repository.ts:92](#)

- **success:** *boolean*
- **items:** *Array<T extends app.domain.repository.core.PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns items from remote api and updates the items cache if false, returns items from local cache if true.

findAllWithNodesAndSubNodes

```
() findAllWithNodesAndSubNodes(propertyName: string, repository: Repository, callback: Function, doCache?: boolean = false)
```

Defined in [Repository.ts:315](#)

Find all items and its related properties

Parameters

- **propertyName:** *string*
 - The name of the item property
- **repository:** *Repository*
 - The repository to fetch the the property
- **callback:** *Function*
 - Will be called with (true, items) on success successful remote/cache call and with (false, []) on error
- **callback function signature**

```
() (success: boolean, items: Array<T extends app.domain.repository.core.PersistentEntity>)
```

Defined in [Repository.ts:316](#)

Parameters

- **success:** *boolean*
- **items:** *Array<T extends app.domain.repository.core.PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns items from remote api and updates the item cache if

findByPropertyId

```
() findByPropertyId(propertyName: string, property: any, callback: Function, doCache?: boolean = false)
```

Defined in [Repository.ts:244](#)

Find items by a property (object) id

Parameters

- **propertyName:** *string*
 - The name of the property of the item
- **property:** *any*
 - The value of the property to search
- **callback:** *Function*
 - The callback is called with (true, item) on success and with (false, null) on error
- **callback function signature**

```
() (success: boolean, items: Array<T extends app.domain.repository.core.PersistentEntity>)
```

Defined in [Repository.ts:244](#)

Parameters

- **success:** *boolean*
- **items:** *Array<T extends app.domain.repository.core.PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns item from remote api and updates the item in the cache if false, returns item from local cache if true.

findOneBy

```
() findOneBy(propertyName: string, property: any, callback:
```

Defined in [Repository.ts:204](#)

Find an item by a property value

Parameters

- **propertyName:** *string*
 - The name of the property of the item
- **property:** *any*
 - The value of the property to search
- **callback:** *Function*
 - The callback is called with (true, item) on success and with (false, null) on error
- **callback function signature**

```
() (success: boolean, item: T in  
  app.domain.repository.core.Repository<T extends  
  PersistentEntity>)
```

Defined in [Repository.ts:204](#)

Parameters

- **success:** *boolean*
- **item:** *T* in *app.domain.repository.core.Repository<T extends PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns item from remote api and updates the item in the cache if false, returns item from local cache if true.

findOneById

```
() findOneById(id: number, callback: Function, doCache?:  
  boolean = false)
```

Defined in [Repository.ts:164](#)

Search item by id on remote resource or in cache. Updates item in item cache on success.

Parameters

- **callback:** *Function*

- The callback is called with (true, item) on success and with (false, null) on error

- **callback function signature**

```
() (success: boolean, item: T in  
  app.domain.repository.core.Repository<T extends  
  PersistentEntity>)
```

Defined in [Repository.ts:164](#)

Parameters

- **success:** *boolean*
- **item:** *T* in *app.domain.repository.core.Repository<T extends PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns item from remote api and updates the item in the cache if false, returns item from local cache if true.

findOneByPropertyId

```
() findOneById(propertyName: string, property: any,  
  callback: Function, doCache?: boolean = false)
```

Defined in [Repository.ts:224](#)

Find an item by a property (object) id

Parameters

- **propertyName:** *string*
 - The name of the property of the item
- **property:** *any*
 - The value of the property to search
- **callback:** *Function*
 - The callback is called with (true, item) on success and with (false, null) on error
- **callback function signature**

```
app.domain.repository.core.repository.T extends  
PersistentEntity>)
```

Defined in [Repository.ts:224](#)

Parameters

- **success:** *boolean*
- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns item from remote api and updates the item in the cache if false, returns item from local cache if true.

findSubNodes

```
() findSubNodes(nodes: Array<T extends  
app.domain.repository.core.PersistentEntity>,  
propertyName: string, repository: Repository, callback:  
Function, doCache?: boolean = false)
```

Defined in [Repository.ts:337](#)

Find related properties of nodes

Parameters

- **nodes:** *Array<T extends app.domain.repository.core.PersistentEntity>*
 - A list with items to find its properties
- **propertyName:** *string*
 - The name of the item property
- **repository:** [Repository](#)
 - The repository to fetch the the property
- **callback:** *Function*
 - Will be called with (true, items) on success successful remote/cache call and with (false, []) on error
- **callback function signature**

```
() (success: boolean, items: Array<T extends
```


Defined in [Repository.ts:341](#)

Parameters

- **success:** *boolean*
- **items:** *Array<T extends app.domain.repository.core.PersistentEntity>*
- **doCache?:** *boolean* optional
 - Returns items from remote api and updates the item cache if false, returns items from local cache if true.

getResourcePath

```
() getResourcePath(resource: string): string
```

Defined in [Repository.ts:78](#)

Get url for resource

Parameters

- **resource:** *string*

Returns *string*

path - Gets the path for the requested resource. Concatenates host & path and uses proxy of set

remove

```
() remove(item: T in app.domain.repository.core.Repository<T extends PersistentEntity>, callback: Function)
```

Defined in [Repository.ts:262](#)

Remove item from remote collection and item cache

Parameters

- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*
 - The item to remove

error

- **callback function signature**

```
() (success: boolean)
```

Defined in [Repository.ts:262](#)

Parameters

- **success:** *boolean*

update

```
() update(item: T in app.domain.repository.core.Repository<T extends PersistentEntity>, callback: Function)
```

Defined in [Repository.ts:289](#)

Update item in remote collection

Parameters


- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*
 - The item to remove
- **callback:** *Function*
 - The callback is called with (true, item) on success and with (false, null) on error
- **callback function signature**


```
() (success: boolean, item: T in app.domain.repository.core.Repository<T extends PersistentEntity>)
```

Defined in [Repository.ts:289](#)

Parameters

- **success:** *boolean*
- **item:** *T in app.domain.repository.core.Repository<T extends PersistentEntity>*

 Container, dynamic module

 Enumeration

 Enumeration member

 Object literal

 Constructor

 Variable


 Function, call signature, accessor

 Index signature

 Interface

 Constructor

 Property


 Member, accessor

 Index signature


 Class


 Constructor

 Property

 Member, accessor

 Index signature

 Static property

 Static member

Index

Modules

 **app**

Modules

app

 app: *app*

Defined in TemplateProcessor.ts:1

Index

Modules

 **service**

Modules

service

 service: *service*

Index

Classes

-  [ProcessorPattern](#)
-  [TemplateProcessor](#)
-  [VariablePattern](#)

Classes

ProcessorPattern

 ProcessorPattern: [ProcessorPattern](#)

Defined in [TemplateProcessor.ts:2](#)

name

 name: *Object*

Defined in [TemplateProcessor.ts:4](#)

name.pattern

 name.pattern: *string*

Defined in [TemplateProcessor.ts:5](#)

name.postSignLength

 name.postSignLength: *number*

Defined in [TemplateProcessor.ts:9](#)

● `name.preSignLength`: *number*

Defined in [TemplateProcessor.ts:7](#)

parameter

● `parameter`: *Object*

Defined in [TemplateProcessor.ts:11](#)

parameter.pattern

● `parameter.pattern`: *string*

Defined in [TemplateProcessor.ts:12](#)

parameter.postSignLength

● `parameter.postSignLength`: *number*

Defined in [TemplateProcessor.ts:16](#)

parameter.preSignLength

● `parameter.preSignLength`: *number*

Defined in [TemplateProcessor.ts:14](#)

pattern

● `pattern`: *string*

Defined in [TemplateProcessor.ts:3](#)

 **TemplateProcessor:** [TemplateProcessor](#)

Defined in [TemplateProcessor.ts:29](#)

constructor

• constructor(*data*: *Object*, *template*: *string*, *processors*: *Object*): [TemplateProcessor](#)

Defined in [TemplateProcessor.ts:101](#)

Parameters

- **data:** *Object*
 - A dictionary used by variables and processors for template rendering
- **template:** *string*
 - A text template containing markers and processors to replace
- **processors:** *Object*
 - A dictionary with processor functions
- **processors[]**

```
[ ] (index: string): any
```

Defined in [TemplateProcessor.ts:101](#)

Parameters

- **index:** *string*

Returns *any*

Returns [TemplateProcessor](#)

primaryProcessorPattern

• primaryProcessorPattern: [ProcessorPattern](#)

Defined in [TemplateProcessor.ts:67](#)

processor pattern: \$processorName:(param1, param2)\$

example : \$processor:(abc1, abc, "ef")\$ \$processor:(path.to.variable, "ab")\$ \$processor:(abc1)\$ \$processor:(abc1)\$ \$processor:()\$

primaryVariablePattern

● primaryVariablePattern: *VariablePattern*

Defined in [TemplateProcessor.ts:39](#)

example : \${path.to.something} \${variable}

secondaryProcessorPattern

● secondaryProcessorPattern: *ProcessorPattern*


Defined in [TemplateProcessor.ts:80](#)

secondaryVariablePattern

● secondaryVariablePattern: *VariablePattern*

Defined in [TemplateProcessor.ts:45](#)

parseProcessors

 parseProcessors(text: *string*, processorPattern: *ProcessorPattern*, executer: *Function*): *string*

Defined in [TemplateProcessor.ts:154](#)

Parse processors and fetch text to replace the processor tags

Parameters

- **text:** *string*
 - Template to search for processors

▪ **executer: function**

- Function to be execute for every found processor - should return the text which will replace the processor tag

▪ **executer function signature**

```
(processorName: string, processorParameters: Array<string>, startIndex: number, length: number): string
```

Defined in [TemplateProcessor.ts:155](#)

Parameters

- **processorName:** *string*
- **processorParameters:** *Array<string>*
- **startIndex:** *number*
- **length:** *number*

Returns *string*

Returns *string*

- Template with replaced processor tags

parseVariables

```
() parseVariables(variablePattern: VariablePattern, text: string): string
```

Defined in [TemplateProcessor.ts:276](#)


find patterns like \${var} or \${var.auto.name} or \${!var}

Parameters

- **variablePattern:** *VariablePattern*
- **text:** *string*
 - The template containing variable markers to replace

Returns *string*

textToReplace - The template with replaced variable markers

 `process(): string`


Defined in [TemplateProcessor.ts:112](#)

process member template using member data and member processors

Returns *string*

template - The rendered text template

processSecondary

 `processSecondary(): string`


Defined in [TemplateProcessor.ts:131](#)

process member template using member data and member secondary processors

Returns *string*

template - The rendered text template

runProcessor

 `runProcessor(processorName: string, processorParameters: Array<string>): string`

Defined in [TemplateProcessor.ts:190](#)

Execute a processor

`example` : `runProcessor('concat', ["path.to.title", "\":\"", "path.to.value"]);`

Parameters

- **processorName:** *string*
- **processorParameters:** *Array<string>*
 - A list with string and path variables

string variables: start **and end with "**, e.g.

path variables: e.q.
`"path.to.variable"`

Returns *string*

- The rendered processor or ""

VariablePattern

 VariablePattern: *VariablePattern*

Defined in [TemplateProcessor.ts:19](#)

pattern

● pattern: *string*

Defined in [TemplateProcessor.ts:20](#)

postSignLength


● postSignLength: *number*

Defined in [TemplateProcessor.ts:24](#)

preSignLength

● preSignLength: *number*


Defined in [TemplateProcessor.ts:22](#)

 Container, dynamic module

 Enumeration

 Enumeration member

 Object literal

 Constructor

 Variable


 Function, call signature, accessor

 Index signature

 Interface

 Constructor

 Property


 Member, accessor

 Index signature

 Class


 Constructor

 Property

 Member, accessor

 Index signature

 Static property

 Static member