
Manages all the communication between the backendserver and the frontend-application

Index

Constructors

constructor

Properties

singleBackendServerCommunicator

Methods

[login](#)
[pullConfigFileFromConfigFileNameFromWorkflowInstanceName](#)
[pullServers](#)
[pullTemplateFromName](#)
[pullTemplatesName](#)
[pullUsers](#)
[pullVersionsFromWorkflowInstance](#)
[pullWorkflowInstancesAndConfigFilesName](#)
[pushConfigFileForConfigFileNameFromWorkflowInstanceName](#)
[pushCreateTemplate](#)
[pushDeleteUser](#)
[pushReplaceActiveVersionOfWorkflowInstance](#)
[pushServer](#)
[pushUser](#)
[pushWorkflowInstanceFromTemplate](#)
[register](#)
[returnBackendServerCommunicator](#)

Constructors

Private **constructor**

```
new BackendServerCommunicator(): BackendServerCommunicator
```

Defined in classes/BackendServerCommunicator.ts:8

Returns *BackendServerCommunicator*

Properties

Private **singleBackendServerCommunicator**

```
singleBackendServerCommunicator: BackendServerCommunicator
```

Defined in classes/BackendServerCommunicator.ts:6

Methods

login

```
login(userName: string, userPassword: string): void
```

Defined in classes/BackendServerCommunicator.ts:20

A person signs in to his account

Parameters

- **userName:** *string*
the persons userName
- **userPassword:** *string*
the persons userPassword

Returns *void*

pullConfigFileFromConfigFileNameFromWorkflowInstanceName

```
pullConfigFileFromConfigFileNameFromWorkflowInstanceName(configFileName: string, workflowInstanceName: string): ConfigFile
```

Defined in classes/BackendServerCommunicator.ts:61

Pulls a configFile-object from backendserver

Parameters

- **configFileName:** *string*
The name of the configFile
- **workflowInstanceName:** *string*
The name of the workflowInstance

Returns *ConfigFile*

The configFile-object

pullServers

pullServers(): *Server* []

Defined in classes/BackendServerCommunicator.ts:99

Pulls all the server-objects from the backendserver

Returns *Server* []

The server-objects

pullTemplateFromName

pullTemplateFromName(templateName: *string*): *Template*

Defined in classes/BackendServerCommunicator.ts:49

Pulls a template object from the backendserver

Parameters

- **templateName:** *string*
The template name

Returns *Template*

The template-object

pullTemplatesName

pullTemplatesName(): *string* []

Defined in classes/BackendServerCommunicator.ts:43

Pulls all the template names from the backendserver

Returns *string*[]

List of template names

pullUsers

pullUsers(): *User*[]

Defined in classes/BackendServerCommunicator.ts:84

Pulls all user-objects from the backendserver

Returns *User*[]

The user-objects

pullVersionsFromWorkflowInstance

pullVersionsFromWorkflowInstance(workflowInstanceName: *string*):
WorkflowInstance

Defined in classes/BackendServerCommunicator.ts:73

Pulls a workflowInstance from the server

Parameters

- **workflowInstanceName:** *string*
The name of the workflowInstance

Returns *WorkflowInstance*

the workflowInstance-object

pullWorkflowInstancesAndConfigFileName

pullWorkflowInstancesAndConfigFileName(): *string*[][]

Defined in classes/BackendServerCommunicator.ts:54

Pulls all workflowInstances names the according configFiles names from the backendServer

Returns *string*[][]

pushConfigFileForConfigFilesNameFromWorkflowInstanceName

```
pushConfigFileForConfigFilesNameFromWorkflowInstanceName(configFiles:
  ConfigFile[], workflowInstanceName: string): void
```

Defined in classes/BackendServerCommunicator.ts:67

Pushes the configFile-objects of a specificWorkflowInstance to the backendserver

Parameters

- **configFiles:** *ConfigFile*[]
The configFile-objects
- **workflowInstanceName:** *string*
The name of the workflowInstance

Returns *void*

pushCreateTemplate

```
pushCreateTemplate(template: Template): void
```

Defined in classes/BackendServerCommunicator.ts:31

pushes the new template to the backendserver

Parameters

- **template:** *Template*
The new template-object

Returns *void*

pushDeleteUser

```
pushDeleteUser(user: User): void
```

Defined in classes/BackendServerCommunicator.ts:94

Pushes a delete-request to the backendserver

Parameters

- **user:** *User*
The user-object

Returns *void*

pushReplaceActiveVersionOfWorkflowInstance

```
pushReplaceActiveVersionOfWorkflowInstance(workflowInstanceName: string,  
versionNumber: string): void
```

Defined in classes/BackendServerCommunicator.ts:79

Parameters

- **workflowInstanceName:** *string*
The name of the workflowInstance
- **versionNumber:** *string*
The name of the versionNumber

Returns *void*

pushServer

```
pushServer(server: Server): void
```

Defined in classes/BackendServerCommunicator.ts:104

Pushes a server-object to the backend

Parameters

- **server:** *Server*
The server-object

Returns *void*

pushUser

```
pushUser(user: User): void
```

Defined in classes/BackendServerCommunicator.ts:89

Pushes a user-object to the backendServer

Parameters

- **user:** *User*

The user-object

Returns *void*

pushWorkflowInstanceFromTemplate

```
pushWorkflowInstanceFromTemplate(templateName: string, workflowInstanceName:
string, configFolder: File): void
```

Defined in classes/BackendServerCommunicator.ts:38

pushes a new workflowInstance to the backendserver

Parameters

- **templateName:** *string*
The name of the choosen template
- **workflowInstanceName:** *string*
The name for the new workflowInstance
- **configFolder:** *File*
The configFolder encoded as zip-file

Returns *void*

register

```
register(userName: string, userPassword: string): void
```

Defined in classes/BackendServerCommunicator.ts:26

A person registers a new account

Parameters

- **userName:** *string*
The choosen userName of the person
- **userPassword:** *string*
The choosen userPassword of the person

Returns *void*

returnBackendServerCommunicator

`returnBackendServerCommunicator(): BackendServerCommunicator`

Defined in classes/BackendServerCommunicator.ts:14

Returns the only instance of BackendServerCommunicator (Singelton-Pattern)

Returns *BackendServerCommunicator*

singleBackendServerCommunicator
