

Class Discription

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

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*

pullConfigFileWithConfigFileNameWithWorkflowInstanceName

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

Defined in classes/BackendServerCommunicator.ts:67

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

pullGraphForTemporaryTemplate

```
pullGraphForTemporaryTemplate(tempTemplate: Template): File
```

Defined in classes/BackendServerCommunicator.ts:32

Pulls a graph-image of the temporary template from the backendserver

Parameters

- **tempTemplate:** *Template*
The temporary template

Returns *File*

The image of the graph

pullServers

```
pullServers(): Server[]
```

Defined in classes/BackendServerCommunicator.ts:105

Pulls all the server-objects from the backendserver

Returns *Server*[]

The server-objects

pullTemplateWithName

```
pullTemplateWithName(templateName: string): Template
```

Defined in classes/BackendServerCommunicator.ts:55

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:49

Pulls all the template names from the backendserver

Returns *string*[]

List of template names

pullUsers

```
pullUsers(): User[]
```

Defined in classes/BackendServerCommunicator.ts:90

Pulls all user-objects from the backendserver

Returns [User](#)[]

The user-objects

pullVersionsWithWorkflowInstanceName

```
pullVersionsWithWorkflowInstanceName(workflowInstanceName: string): Version[]
```

Defined in classes/BackendServerCommunicator.ts:79

Pulls the versions from a workflowinstance from the backendserver

Parameters

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

Returns [Version](#)[]

the version-objects

pullWorkflowInstancesNameAndConfigFileName

```
pullWorkflowInstancesNameAndConfigFileName(): string[][]
```

Defined in classes/BackendServerCommunicator.ts:60

Pulls all workflwoInstances names the according configFiles names from the backendServer

Returns *string*[][]

WorkflowInstances names and configFiles names

pushConfigFilesWithWorkflowInstanceName

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

Defined in classes/BackendServerCommunicator.ts:73

Pushes the configFile-objects of a specific WorkflowInstance to the backendserver

Parameters

- **configFiles:** [ConfigFile](#)[]
The configFile-objects
- **workflowInstanceName:** *string*

The name fo the workflowInstace

Returns *void*

pushCreateTemplate

pushCreateTemplate(template: *Template*): *void*

Defined in classes/BackendServerCommunicator.ts:37

pushes the new template to the backendserver

Parameters

- **template:** *Template*

The new template-object

Returns *void*

pushCreateWorkflowInstanceFromTemplate

pushCreateWorkflowInstanceFromTemplate(templateName: *string*,
workflowInstanceName: *string*, configFolder: *File*): *void*

Defined in classes/BackendServerCommunicator.ts:44

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*

pushDeleteUser

pushDeleteUser(user: *User*): *void*

Defined in classes/BackendServerCommunicator.ts:100

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:85

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:110

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:95

Pushes a user-object to the backendServer

Parameters

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

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
