

search the docs

ANEXD

(api/ANEXD)

module

service

ANEXDService

(api/ANEXD.ANEXDService)

APIService

(api/ANEXD.APIService)

SessionService

(api/ANEXD.SessionService)

SocketService

(api/ANEXD.SocketService)

# ANEXDService

service in module **ANEXD**

## Description

Provides a simple interface for application developers to send and receive websocket messages from the server

## Dependencies

- \$rootScope (api/ng.\$rootScope)
- \$timeout (api/ng.\$timeout)
- SocketService (api/SocketService)

## Methods

### expect(event)

Call once for each message expected from the server that may be received at any time Different from sendToServer in that it isn't in response to any other event Results are gathered from the getFromServer function

#### Parameters

Param	Type	Details
event	string ()	The event to listen on

### getFromServer()

returns the most recent websocket message received being listened on by expect Use \$scope.\$watch to access on the \$digest cycle

#### Returns

string ()    Object of the event and data received from the most recent event

### sendToServer(event, val)

Send a message to the server and wait for a reply using SocketService promise

#### Parameters

Param	Type	Details
event	string ()	The event to send the websocket message on
val	string ()	Contents of message

search the docs

- ANEXD

(api/ANEXD)
- module
- service
- ANEXDService
- (api/ANEXD.ANEXDService)
- APIService
- (api/ANEXD.APIService)
- SessionService
- (api/ANEXD.SessionService)
- SocketService
- (api/ANEXD.SocketService)

API Documentation (api) / ANEXD (api/ANEXD) / APIService

# APIService

service in module ANEXD

## Description

APIService provides a simple interface for sending HTTP POST requests

## Dependencies

- \$rootScope (api/ng.\$rootScope)
- \$http (api/ng.\$http)
- CONST (api/CONST)

## Methods

### post(event, data, error)

Post an HTTP POST request

#### Parameters

Param	Type	Details
event	string ()	The URL on the API to POST on
data	string ()	Any data to POST in JSON form
error	boolean=true ()	Should successful POSTs that return negative responses display an error message?

#### Returns

string () Either HTTP response on successful post or false otherwise

API Documentation (api) / ANEXD (api/ANEXD) / SocketService

**ANEXD**  
(api/ANEXD)

module

service

ANEXDService  
(api/ANEXD.ANEXDService)APIService  
(api/ANEXD.APIService)SessionService  
(api/ANEXD.SessionService)SocketService  
(api/ANEXD.SocketService)

# SocketService

service in module **ANEXD**

## Description

SocketService provides an interface for sending and receiving websocket events. Also provides promises for sending then receiving a message on the same event.

## Dependencies

- `$rootScope` (api/ng.\$rootScope)
- `$q` (api/ng.\$q)
- `$timeout` (api/ng.\$timeout)
- `CONST` (api/CONST)
- `socketFactory` (api/socketFactory)

## Methods

### **emit(event, val)**

Sends a message on the provided event (also possible using `socket.default.emit`)

#### Parameters

Param	Type	Details
event	<code>string ()</code>	The event to send the websocket message on
val	<code>string ()</code>	Contents of message

### **promise(event, val, error)**

Promises a response on the same event sent

#### Parameters

Param	Type	Details
event	<code>string ()</code>	The event to send the websocket message on
val	<code>string ()</code>	Contents of message
error	<code>boolean ()</code>	Whether to show errors on timeout (3 seconds)

#### Returns

`string ()` Promise of the return message

search the docs

ANEXD

(api/ANEXD)

module

service

ANEXDService

(api/ANEXD.ANEXDService)

APIService

(api/ANEXD.APIService)

SessionService

(api/ANEXD.SessionService)

SocketService

(api/ANEXD.SocketService)

# SessionService

service in module ANEXD

## Description

SessionService provides application-wide access to logging in, logging out, creating and removing sessions, and accessing their states

## Dependencies

- \$rootScope (api/ng.\$rootScope)
- \$cookies (api/ng.\$cookies)
- \$http (api/ng.\$http)
- md5 (api/md5)
- CONST (api/CONST)
- APIService (api/APIService)

## Methods

### close()

End a running session

### create(lobbyId, appId)

Create a new session holding the lobby and application ids

#### Parameters

Param	Type	Details
lobbyId	string ()	◦ The lobby's id
appId	string ()	◦ The application's id

### createUser()

Request to create a new user. If successful, call the login function on the new user.

#### Returns

boolean () True if the user is successfully created and logged in, false otherwise

### details()

Return the lobby and app ids of a running service

**string ()**    Object of two strings; {'lobby': lobby, 'app', app}

### **getUser()**

Get the email of the user currently logged in, if any

#### **Returns**

**string ()**    User's email if logged in, false otherwise

### **getUserId()**

Get the id of the currently logged in user, if any

#### **Returns**

**string ()**    User's id if logged in, false otherwise

### **isLoggedIn()**

Returns whether a user is currently logged in or not

#### **Returns**

**boolean ()**    True if the user is logged in, false otherwise

### **login(email, password)**

attempt to log a user in given a validated existing email and password

#### **Parameters**

Param	Type	Details
email	<b>string ()</b>	◦ The user's email (confirmed by checkEmail to be a valid user email)
password	<b>string ()</b>	◦ The user's supposed password

#### **Returns**

**boolean ()**    True if the user successfully logs in, false otherwise

### **logout()**

Log the user out, clear cookies, and broadcast the 'logout' event

#### **Returns**

**undefined ()**    No return as all actions are taken locally

## **running()**

Check if a session is currently running

### **Returns**

**boolean ()** True if a session is running, false otherwise