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) / SessionService

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

Returns

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	**string ()**	◦ The user's email (confirmed by checkEmail to be a valid user email)
password	**string ()**	◦ 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