Google

Software Product Sprint

Team 32 | Datastore

Table of Contents

Datastore-API.js

Import this module to facilitate the fetch calls. All are async functions.

```
login(String username, String passwordHash){
```

Returns a json, either json.error or json.success.

Json.success contains the user matching the username and passwordHash credentials.

```
}
```

```
register(String username, String passwordHash, String email, String firstName, String middleName, String lastName){
```

Returns a json, either json.error or json.success.

Registers a new user if the username and email are not already in the database.

```
}
```

getUser(String username, String passwordHash){

If the username and passwordHash credentials are valid, returns a json matching the user defined in the database.

If not, alerts: alert("Something Went Wrong: " + response.error);

}

deleteCurrentUser(String username, String passwordHash){

Returns a json, either json.error or json.success.

Deletes the user matching the username and passwordHash credentials.

}

addDayToCalendar(String username, String passwordHash, Day:{date:long(timestamp),mood:int} day){

Returns a json, either json.error or json.success.

Adds the given day to the calendar of the user matching the username and passwordHash credentials.

}

addEntryToJournal(String username, String passwordHash, String entry){

Returns a json, either json.error or json.success.

Adds the given entry to the journal of the user matching the username and passwordHash credentials.

}

addOneToPanicButton(String username, String passwordHash){

Returns a json, either json.error or json.success.

Adds 1 to the panic button counter of the user matching the username and passwordHash credentials.

}

updateUser(String username, String passwordHash, String prevEmail,

```
User: {
      username: string,
      passwordHash: string,
      calendar: array[day],
      firstName: string,
      middleName: string,
      lastName: string,
      panicButtonPressed: int,
      journal: array[string]
      } user){
Returns a json, either json.error or json.success.
Updates the user matching the username and password, the new user data
is the data from the user parameter.
}
getUserCalendar(String username, String passwordHash){
Returns the calendar(array[day]) of the user matching the username and
password.
}
getUserJournal(String username, String passwordHash){
Returns the journal(array[string]) of the user matching the username
and password.
}
getUserPanicButton(String username, String passwordHash){
Returns the panic button counter(int) of the user matching the
username and password.
```

```
}
```

Java Servlets

/api/datastore-debug: GET method, returns a table with all the users.

/api/delete-all: GET method, deletes all the users.

/api/delete-user: POST method, deletes the user matching the username and password.

```
payload = {
      username: string,
      passwordHash: string
}
```

/api/login: POST method, returns the user matching the username and password.

```
payload = {
      username: string,
      passwordHash: string
}
```

/api/register: POST method, registers a new username if credentials are valid.

```
payload = {
      username: string,
      passwordHash: string,
      calendar: array[day],
      firstName: string,
      middleName: string,
      lastName: string,
      panicButtonPressed: int,
      journal: array[string]
}
```

/api/update-user: POST method, replaces the user matching the previous data with the data from the new user json.

```
payload = {
      user: {
            username: string,
            passwordHash: string,
            calendar: array[day],
            firstName: string,
            middleName: string,
            lastName: string,
            panicButtonPressed: int,
            journal: array[string]
            },
      prevUsername: string,
      prevPasswordHash: string,
      prevEmail: string
}
```