RockFIIT API Documentation
Louis Doherty

API Server IP Address : https://rockFIITapi.lousterrd.repl.co (Testing site)

exercises API

Purpose
The mobile app client is delivered with a local copy of the exercise library, and this API enables the local database to be updateds at regular intervals. This enables centralized distribution of data to the mobile app installation.

Request
URL-path: "/exercises"
Method Type: GET, POST
Authentication required: No
Body of request: Empty

Response
The exercises route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : <data>
}
```

The 'info' field is a text field and will always be present. When a successful response is sent (status code 200) , it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 5xx), it may contain the error information regarding the issue.

The 'data' field contains the exercise library as a list of lists, with each of the inner lists representing a row of the table. The elements of the list will be ordered corresponding to the table's field order.

addUser API

Purpose
This API creates a new user in the userTable within the database.

Request
URL-path: "/addUser"
Method Type: POST
Authentication required: No
Body of request: Contains user information

Format:

```
{
'userName'  : <text>,
'password'  : <text>,
'firstName' : <text>,
'weight'    : <real>
}
```

The `'userName'` field is a text field that contains the email address of a new user.

The `'password'` field is a text field that contains the plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. The server generates a unique salt for each new user and hashes the password, then stores that hashed password in the centralized database.

The `'firstName'` field is a text field that contains the first name of the user.

The `'weight'` field is a real field, which in SQL translates to any floating point number. This is the user's current weight in pounds.

Response
The addUser route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The `'info'` field is a text field and will always be present. When a successful response triggered by the creation of a new user is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.

changePassword API

Purpose
This API updates a user's password in the userTable within the database.

Request
URL-path: "/changePassword"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'  : <text>,
'oldPwd'  : <text>,
'newPwd' : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'oldPwd'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server.

The `'newPwd'` field is a text field that contains the user's new plaintext password. Because the request uses HTTPS, the data is secure while being sent to the server. This will become the user's new password to login.

Response
The changePassword route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The `'info'` field is a text field and will always be present. When a successful response triggered by the updating of the user's password is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.

activities API

Purpose
This API retrieves a user's logged activities.

Request
URL-path: "/activities"
Method Type: GET, POST1
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'  : <text>,
'password'  : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The activities route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : <data>
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the updating of the user's password is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field contains the activities as a list of lists, with each of the inner lists representing a row of the table. The elements of the list will be ordered corresponding to the table's field order.

logActivity API

Purpose
This API creates a new entry in the exerciseLog table within the database that contains the information of a user's completed workout.

Request
URL-path: "/logActivity"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'       : <text>,
'exerciseID'     : <integer>,
'setsCompleted'  : <integer>,
'repsCompleted'  : <integer>,
'weight'         : <real>,
'notes'          : <text>,
'date'           : <text>,
'password'       : <text>
}
```

The 'userName' field is a text field that contains the email address of a user.

The 'exerciseID' is an integer field that contains the exerciseID of the completed exercise.

The 'setsCompleted' field is an integer field that contains the user's number of completed sets.

The 'repsCompleted' field is an integer field that contains the user's number of completed reps.

The 'weight' field is a real field, which in SQL translates to any floating point number. This is the user's weight used for the exercise.

The 'notes' field is a text field that contains the user's notes and comments about the logged exercise.

The 'date' field is a text field that contains the date that the exercise was logged .

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response

The logActivity route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the insertion of a newly logged exercise is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.

programs API

Purpose
This API retrieves a user's programs.

Request
URL-path: "/programs"
Method Type: GET, POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'  : <text>,
'password'  : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The programs route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : <data>
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the retrieval of the user's programs is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field contains the programs as a list of lists, with each of the inner lists representing a row of the table. The elements of the list will be ordered corresponding to the table's field order.

addProgram API

Purpose
This API creates a new entry in the programTable table within the database that contains the information of a user's new program, containing up to 5 exercises.

Request
URL-path: "/addProgram"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'      : <text>,
'programName'   : <text>,
'exercise1'     : <integer>,
'exercise2'     : <integer>,
'exercise3'     : <integer>,
'exercise4'     : <integer>,
'exercise5'     : <integer>,
'password'      : <text>
}
```

The 'userName' field is a text field that contains the email address of a user.

The 'programName' field is a text field that contains the name of the user's program.

The 'exercise1' is an integer field that contains the exerciseID of the added exercise.

The 'exercise2' is an integer field that contains the exerciseID of the added exercise.

The 'exercise3' is an integer field that contains the exerciseID of the added exercise.

The 'exercise4' is an integer field that contains the exerciseID of the added exercise.

The 'exercise5' is an integer field that contains the exerciseID of the added exercise.

The 'password' field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The addProgram route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the insertion of a new program is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The 'data' field is a False boolean as it is not utilized in this response.

delProgram API

Purpose
This API deletes an entry in the programTable within the database.

Request
URL-path: "/delProgram"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'      : <text>,
'programID'     : <integer>,
'password'      : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'programID'` field is an integer field that contains the ID of the user's program.

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The delProgram route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The `'info'` field is a text field and will always be present. When a successful response triggered by the deletion of a program is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.

goals API

Purpose
This API retrieves a user's goals.

Request
URL-path: "/goals"
Method Type: GET, POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'  : <text>,
'password'  : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The goals route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : <data>
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the retrieval of the user's goals is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field contains the goals as a list of lists, with each of the inner lists representing a row of the table. The elements of the list will be ordered corresponding to the table's field order.

addGoal API

Purpose
This API creates a new entry in the goalTable within the database that contains the information of a user's new goal.

Request
URL-path: "/addGoal"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'      : <text>,
'goalName'      : <text>,
'completed'     : <integer>,
'date'          : <text>,
'password'      : <text>
}
```

The 'userName' field is a text field that contains the email address of a user.

The 'goalName' field is a text field that contains the name of the user's goal.

The 'completed' field is an integer field that contains either a 0 (false) or 1 (true) that shows whether or not a user has completed that goal.

The 'date' is a text field that contains the date the goal was created.

The 'password' field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The addGoal route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the insertion of a new goal is sent (status code 201), it will contain a

'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.

updateGoal API

Purpose
This API updates the completion status of an entry in the goalTable within the database.

Request
URL-path: "/updateGoal"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'      : <text>,
'goalName'      : <integer>,
'completed'     : <integer>,
'password'      : <text>
}
```

The 'userName' field is a text field that contains the email address of a user.

The 'goalName' field is a text field that contains the name of the user's goal.

The 'completed' field is an integer field that contains either a 0 (false) or 1 (true) that shows whether or not a user has completed that goal.

The 'password' field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The updateGoal route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the updating of a goal is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The 'data' field is a False boolean as it is not utilized in this response.

delGoal API

Purpose
This API deletes an entry in the goalTable within the database.

Request
URL-path: "/delGoal"
Method Type: POST
Authentication required: Yes
Body of request: Contains user information

Format:

```
{
'userName'      : <text>,
'goalID'        : <integer>,
'password'      : <text>
}
```

The `'userName'` field is a text field that contains the email address of a user.

The `'goalID'` field is an integer field that contains the ID of the user's goal.

The `'password'` field is a text field that contains the current plaintext password of the user. Because the request uses HTTPS, the data is secure while being sent to the server. This is used for authentication purposes.

Response
The delGoal route API sends a response in JSON format:

```
{
'info' : <text>,
'data' : False
}
```

The 'info' field is a text field and will always be present. When a successful response triggered by the deletion of a goal is sent (status code 201), it will contain a 'successful' message. However, when an unsuccessful response is sent (status code 4xx or 5xx), it may contain the error information regarding the issue.

The `'data'` field is a False boolean as it is not utilized in this response.