

Router/API Document - HabitConnect

//{something}: something would be a variable in URL

Pages we are going to have:

- HomePage
- UserInfoPage
- Sign-in page
- Sign-up page
- LeaderboardPage
- Goal details Page

Router:

- www.habitConnect.com/signin
- www.habitConnect.com/signup
- www.habitConnect.com/{user_id}/home
- www.habitConnect.com/{user_id}/leaderboard
- www.habitConnect.com/{user_id}/goal/{goal_id}

Session Storage:

we can access following information at any page by:

- user_id: sessionStorage.getItem("userID")
- token: sessionStorage.getItem("userToken") **(done)**
- userName: sessionStorage.getItem("userName") **(done)**
- Streak: sessionStorage.getItem("userStreak") **(done)**
- FirstName: sessionStorage.getItem("userFname")
- LastName: sessionStorage.getItem("userLname")

User Information: GET+POST

Sign in: POST

Sign up: POST

API document:

API we need in the front-end:

Requirements:

1. GET User Info (for profile page) **(done)**
Current: http://localhost:8081/api/user/userProfileInfo?user_id={userId}
provide: user_id
receive: picture, email, friends (user_id, username, streak)
2. We need a POST method for user sign-up. **(done)**
Current: <http://localhost:8081/api/user/signup>
3. We need a POST method for user login. **(done)**

Current: http://localhost:8081/api/user/login

4. We need one or many POST methods to change user information (not password). **(done)**

parameters: user_id, field_name, data

return: bool

Current: http://localhost:8081/api/user/updateUserInfo

5. We need a GET method to get a user's goals (habits). **(done)**

parameters: user_id

expect return: a list of all habits

Current: localhost:8081/api/habit/getHabits?user_id={userId}

example return:

```
{
  "habits": [
    {
      "id": 1,
      "name": "Drink Water",
      "streak": 10
    },
    {
      "id": 2,
      "name": "Morning Exercise",
      "streak": 7
    },
    {
      "id": 3,
      "name": "Read a Book",
      "streak": 5
    },
    {
      "id": 4,
      "name": "Meditation",
      "streak": 12
    }
  ]
}
```

6. We want a POST method to update the habit. **(Done)**

Current: localhost:8081/api/habit/updateHabit

parameters: habit_id, user_id, data

Sample Request:

```
{
  "HabitID": "65d7e7819d8773504385ce8a",
  "UserID": "65cfd752107086c89a13eace",
  "data": {
    "Goal": "8",
    "Title": "Workout"
  }
}
```

return: bool

7. We want a POST method to add a new habit **(Done)**
Current: localhost:8081/api/habit/createHabit
Sample Request:

```
{
  "Owner": "65cfd752107086c89a13eace",
  "Title": "Morning Exercise",
  "PrivacyType": "0",
  "MeasurementType": "0",
  "Goal": "1"
}
```

parameters: user_id, data(all fields of a habit)
return: bool
8. We want a DELETE method to delete habits. **(Done)**
Current: localhost:8081/api/habit/deleteHabit
parameters: habit_id, user_id
return: bool
9. We want a POST method for password updating **(Done)**
Current: http://localhost:8081/api/user/updatePassword
parameters: user_id, old_password, new_password
if the old password matches, then update the password with the new password, else
returns an error
10. We want a GET method to get a list of user's friends**(Done)**
parameters: user_id
return: a list of friend objects
11. We want a POST method to send a friend request by user_name**(Done)**
Current: http://localhost:8081/api/Friends/sendFriendRequest
parameters: current_user_id, friend_username
return: bool
Sample request:

```
{
  "User": "65f6902c8d4313103042bbeb",
  "FriendsWith": "testfriends2"
}
```
12. We want a DELETE method to delete a friend**(Done)**
parameters: current_user_id, friend_user_id
return: bool
notice: we need to remove the friend from both users' friend list.
Current: http://localhost:8081/api/Friends/deleteFriend
Sample Request:

```
{
```

```

    "User": "65f690238d4313103042bbe7",
    "FriendsWith": "65f6902c8d4313103042bbeb"
  }

```

13. We want a POST method to accept a friend request(**Done**)

parameters: current_user_id, friend_user_id

return: bool

notice: we need to add the friend to both users' friend list

Current: <http://localhost:8081/api/Friends/acceptFriendRequest>

Sample Request:

```

{
  "User": "65f690238d4313103042bbe7",
  "FriendsWith": "65f6902c8d4313103042bbeb"
}

```

14. We want a POST method to deny a friend request(**Done**)

Current: <http://localhost:8081/api/Friends/declineFriendRequest>

Sample request:

```

{
  "User": "65f6902c8d4313103042bbeb",
  "FriendsWith": "65f690238d4313103042bbe7"
}

```

parameters: current_user_id, friend_user_id

return: bool

15. We want a GET method to return a list of habits and check-in status.

parameters: user_id, Date

return: a list of habit_id and check-in status

example:

```

{
  {habit_id:123, check-in: true},
  .....
}

```

- 16.

Notifications api brainstorming

Types of notifications

- Friend Request
- You have been unfriended by...
- when login generate habit and todo notifications

What I will have to do for each type of notification

- Friend request
 - Adds notification to notification table when a friend request is made
 - if emails are on for friend requests send email to recipient
- You have been unfriended by...
 - Adds notification to notification table to recipient when they are unfriended
 - if emails are on for friend requests send email to recipient
- To do reminder, Habit reminder notification
 - when login generate habit and todo notifications

Additional Api required

- Check Notification as seen
 - remove it from the notification table
- Returns notifications
- Anything else?