

Christopher Jackson
SWDV691
Week 2
Design: Service Layer(s)

Endpoint Design: RESTful principles. For this application I will be using a RESTful API design because of the main verbs needed to communicate data: GET, POST, PUT, and DELETE.

Service Endpoints:

User

- POST (/user): Add a new user.

Parameters:

- Username * required
- FirstName * required
- LastName * required
- EmailAddress * required
- Phone
- Location

i.e., Tyler goes to the website to create an account.

Request:

```
{
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

Response: 200 - ok

```
{
  "userId": 9222968140497396000,
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

Request:

```
{
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

Response: 405 – username already taken

```
{
  "userId": 9222968140497396000,
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

Request:

```
{
  "username": "",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

Response: 405 – username missing

```
{
  "userId": 9222968140497396000,
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress": "thomasanderson@fakenews.com"
}
```

• **PUT (/user): Update an existing user.**

Parameters:

- Username
- FirstName
- LastName

- EmailAddress
- Phone
- Location

i.e., Tyler goes to the website to update the email address on his account.

Request:

```
{
  "username": "tylerscott",
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 200 - ok

```
{
  "userId": 9222968140497396000,
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress ": " thomasscott@fakenews.com "
}
```

Request:

```
{
  "username": "tylerscott123456789",
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 400 – invalid username supplied

```
{
  "errorMessage": "Invalid username supplied"
}
```

Request:

```
{
  "username": "tylerscott",
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 400 – username not found

```
{
  "errorMessage": "Username not found"
}
```

Request:

```
{
  "username": "tyl",
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 400 – username input validation

```
{
  "errorMessage": "Username not enough characters"
}
```

- **GET (/user/{userId}): Find user by ID.**

Parameters:

- UserId * **required**

i.e., Tyler goes to the website but does not remember his username.

Request:

```
{
  "userId": "9222968140497396000",
}
```

Response: 200 - ok

```
{
  "userId": 9222968140497396000,
  "username": "tylerscott",
  "firstName": "Tyler",
  "lastName": "Scott",
  "emailAddress ": " thomasscott@fakenews.com "
}
```

Request:

```
{
  "userId": "92226000",
}
```

Response: 400 – Invalid ID supplied

```
{
  "errorMessage": "Invalid ID supplied",
}
```

Request:

```
{  
    "userId": "9222968140497396000",  
}
```

Response: 404 – Username not found

```
{  
    "errorMessage": "Username not found"  
}
```

- **POST (/user/{userId}): Update an existing user with form data.**

Parameters:

- UserId * **required**

i.e., Tyler goes to the website to update the email address on his account.

Request:

```
{  
    "userId": 9222968140497396000,  
    "emailAddress ": "thomasscott@fakenews.com"  
}
```

Response: 200 - ok

```
{  
    "userId": 9222968140497396000,  
    "username": "tylerscott",  
    "firstName": "Tyler",  
    "lastName": "Scott",  
    "emailAddress ": " thomasscott@fakenews.com "  
}
```

Request:

```
{  
    "userId": 90,  
    "emailAddress ": "thomasscott@fakenews.com"  
}
```

Response: 400 – invalid input

```
{  
    "errorMessage": "Invalid username supplied"  
}
```

- **DELETE (/user/{userId}): Deletes a user.**

Parameters:

- UserId * **required**

i.e., Tyler goes to the website to delete his account.

Request:

```
{
  "userId": 9222968140497396000,
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 200 - ok

```
{
  "userId": ,
  "username": "",
  "firstName": "",
  "lastName": "",
  "emailAddress ": ""
}
```

Request:

```
{
  "userId": 90,
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 400 – invalid input

```
{
  "errorMessage": "Invalid username supplied"
}
```

Request:

```
{
  "userId": 909222968140497396000
  "emailAddress ": "thomasscott@fakenews.com"
}
```

Response: 400 – account not found

```
{
  "errorMessage": "Account not found"
}
```

Post

- **POST (/post): Add a new post.**

Similar to user but replace with post.

Parameters:

- UserId * required
- Likes
- Comments
- Saves
- TravelExpenses
- LodgingExpenses
- Rating
- Review

- **PUT (/post): Update an existing post.**

Similar to user but replace with post.

Parameters:

- UserId * required
- Likes
- Comments
- Saves
- TravelExpenses
- LodgingExpenses
- Rating
- Review

- **GET (/post/{postId}): Find post by ID.**

Similar to user but replace with post.

Parameters:

- PostId * required
- UserId
- Likes
- Comments
- Saves
- TravelExpenses
- LodgingExpenses
- Rating

- Review

- **POST (/post/{postId}): Update an existing post with form data.**

Similar to user but replace with post.

Parameters:

- PostId * **required**
- UserId
- Likes
- Comments
- Saves
- TravelExpenses
- LodgingExpenses
- Rating
- Review

- **DELETE (/post/{postId}): Deletes a post.**

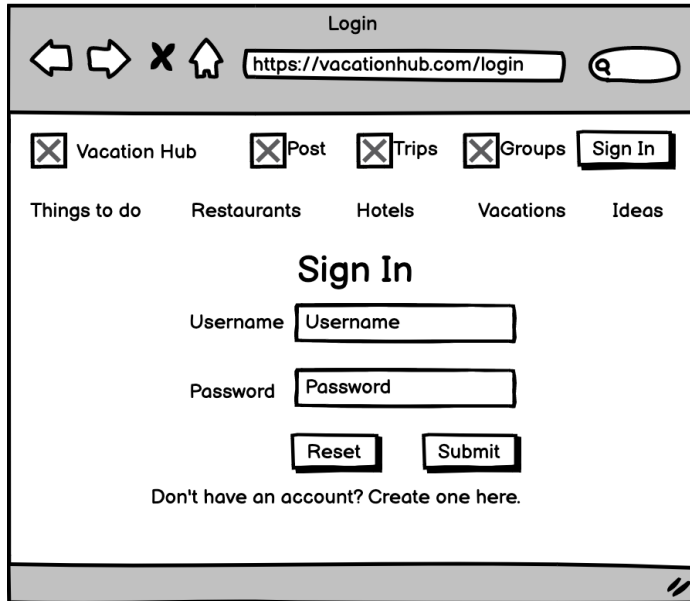
Similar to user but replace with post.

Parameters:

- PostId * **required**
- UserId
- Likes
- Comments
- Saves
- TravelExpenses
- LodgingExpenses
- Rating
- Review

Diagrams:

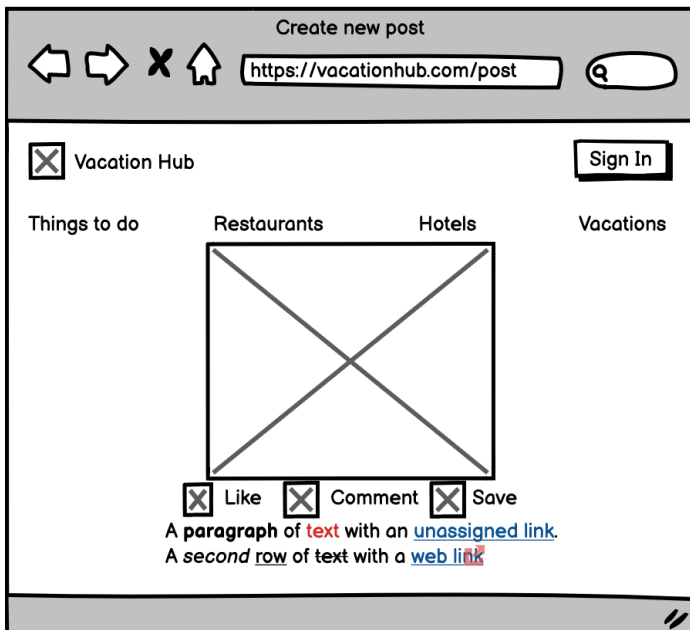
Create a User



The screenshot shows a web browser window titled "Login". The address bar contains "https://vacationhub.com/login". The page has a navigation bar with links: Vacation Hub, Post, Trips, Groups, and Sign In. Below the navigation bar are categories: Things to do, Restaurants, Hotels, Vacations, and Ideas. The main content area is titled "Sign In" and contains a form with fields for "Username" and "Password", and buttons for "Reset" and "Submit". At the bottom, there is a link: "Don't have an account? Create one here."

-----> User Service Endpoints

Create a Post



The screenshot shows a web browser window titled "Create new post". The address bar contains "https://vacationhub.com/post". The page has a navigation bar with links: Vacation Hub and Sign In. Below the navigation bar are categories: Things to do, Restaurants, Hotels, and Vacations. The main content area features a large square placeholder with a diagonal cross. Below the placeholder are buttons for "Like", "Comment", and "Save". At the bottom, there is a paragraph of text: "A paragraph of text with an unassigned link." and a second row of text: "A second row of text with a web link".

-----> Post Service Endpoints