**Designing RESTful API’s**

**1- Required UML and API’s for Follows**

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User **Follows** Another User | **POST** /users/:uid1/follows/:uid2 | User1 Pk and User2 FK | New follow JSON |
| User **Unfollows** Another User | **DELETE** /users/:uid1/unfollows/:uid2 | Delete status |
| User retrieves **Followed** user | **GET** /users/:uid/follows | User's PK | User Following JSON array and Followed UserId |
| User retrieves **Following** user | **GET** /users/:uid/following | User’s PK | User Followed JSON array and Following UserId |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| ***1- User Follows Another User*** |  |
| ***Request*** | POST  http://localhost:4000/api/users/6216773b8afb8157c7c855ac/follows/6216773b8afb8157c7c855ae |
|  | |
|  | |
| ***Response*** |  |
| {  "userFollowed": "6216773b8afb8157c7c855ac",  "userFollowing": "6216773b8afb8157c7c855ae",  "\_id": "6219914bb8895e9aca665452",  "\_\_v": 0  } | |

|  |  |
| --- | --- |
| ***2- User Unfollows Another User*** |  |
| ***Request*** | DELETE http://localhost:4000/api/users/6216773b8afb8157c7c855ac/unfollows/6216773b8afb8157c7c855ae |
|  | |
|  | |
| ***Response*** |  |
| {  "deletedCount": 1  } | |

|  |  |
| --- | --- |
| ***3- User Retrieves Followed User*** |  |
| ***Request*** | GET http://localhost:4000/api/users/6216773b8afb8157c7c855ac/follows |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "6217e9f0de4ca441db3a9d4b",  "userFollowed": "6216773b8afb8157c7c855ac",  "userFollowing": {  "\_id": "6216773b8afb8157c7c855ae",  "username": "charles",  "password": "9091",  "firstName": "charles",  "lastName": "waters",  "email": "charles@hotmail.com",  "salary": 50000,  "\_\_v": 0  },  "\_\_v": 0  }  ] | |

|  |  |
| --- | --- |
| ***4- User Retrieves Following User*** |  |
| ***Request*** | GET http://localhost:4000/api/users/6216773b8afb8157c7c855ae/following |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "6217e9f0de4ca441db3a9d4b",  "userFollowed": {  "\_id": "6216773b8afb8157c7c855ac",  "username": "bob",  "password": "9098",  "firstName": "bob",  "lastName": "ochani",  "email": "bob@hotmail.com",  "salary": 50000,  "\_\_v": 0  },  "userFollowing": "6216773b8afb8157c7c855ae",  "\_\_v": 0  }  ] | |

**2- Alternative UML and API’s for Follows**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User retrieves All **Followed** users | **GET** /followed |  | All Followed UserId’s |
| User retrieves All **Following** user | **GET** /following |  | User Followed JSON array and Following UserId |

**Pros**

GET : All Followed UserId’s

This will retrieve every user’s information that are followed without mentioning any userid in the request

GET : All Followed UserId’s

This will retrieve every user’s information that are following other user’s without mentioning any userid in the request

**Cons**

It will be hard to track the information of all the users, keeping in mind that all the users have many to many relationships in their following list and not mentioning user ID as parameter makes it difficult to fetch.

The highlighted one are the alternative method:

Diagram

Description automatically generated

**1- Required UML and API’s for Bookmarks**

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User **Bookmarks** a tuit | **POST** /users/:uid/bookmarks/:tid | User's and Tuit's PK | New bookmark JSON |
| User **Unbookmarks** Tuit | **DELETE** /users/:uid/unbookmarks/:tid | Delete status |
| User retrieves his **Bookmarked** tuits | **GET** /users/:uid/bookmarks | User's PK | Bookmarked JSON array |
| Retrieve user’s details who **Bookmarked** tuit | **GET** /tuits/:tid/bookmarks | User’s PK | User JSON array |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| ***1- User Bookmarks Tuit*** |  |
| ***Request*** | POST  http://localhost:4000/api/users/6216773b8afb8157c7c855ae/bookmarks/62168a0e42f625a2b490876b |
|  | |
|  | |
| ***Response*** |  |
| {  "bookmarkedTuit": "62168a0e42f625a2b490876b",  "bookmarkedBy": "6216773b8afb8157c7c855ae",  "\_id": "621998afb8895e9aca665454",  "\_\_v": 0  } | |

|  |  |
| --- | --- |
| ***2- User Unbookmarks Tuit*** |  |
| ***Request*** | DELETE  http://localhost:4000/api/users/6216773b8afb8157c7c855ae/unbookmarks/62168a0e42f625a2b490876b |
|  | |
|  | |
| ***Response*** |  |
| {  "deletedCount": 1  } | |

|  |  |
| --- | --- |
| ***3- User retrieves bookmarks Tuit*** |  |
| ***Request*** | GET  http://localhost:4000/api/users/6216773b8afb8157c7c855ae/bookmarks |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "6217e02ab1728cdbd84097f9",  "bookmarkedTuit": {  "\_id": "62168a0e42f625a2b490876b",  "tuit": "charles's 1st tuit regarding his wife",  "postedBy": "6216773b8afb8157c7c855ae",  "postedOn": "2022-02-23T19:25:02.223Z",  "\_\_v": 0  },  "bookmarkedBy": "6216773b8afb8157c7c855ae",  "\_\_v": 0  }  ] | |

|  |  |
| --- | --- |
| ***4- Retrieves User’s that Bookmarked a tuit*** |  |
| ***Request*** | GET  http://localhost:4000/api/tuits/62168a0e42f625a2b490876b/bookmarks |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "6217e02ab1728cdbd84097f9",  "bookmarkedTuit": "62168a0e42f625a2b490876b",  "bookmarkedBy": {  "\_id": "6216773b8afb8157c7c855ae",  "username": "charles",  "password": "9091",  "firstName": "charles",  "lastName": "waters",  "email": "charles@hotmail.com",  "salary": 50000,  "\_\_v": 0  },  "\_\_v": 0  }  ] | |

**2- Alternative UML and API’s for Bookmarks**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User retrieves All **Bookmarked** tuits | **GET** /bookmarks |  | All Bookmarked Tuits |
| User retrieves latest **Bookmark** tuits | **GET** uid/postedOn/bookmarks | Date | User latest Bookmarked JSON array |

The highlighted one are the alternative method

Diagram

Description automatically generated

**Pros**

GET : All Bookmarked tuits

This will retrieve every user’s information that are bookmarked without mentioning any userid in the request

GET : Latest Bookmarked tuits

This will retrieve latest bookmarked tuits with the parameter of Date

**Cons**

It will be hard to track the information of all the latest bookmarked tuit without the use of user id as parameter.

**1- Required UML and API’s for Message**

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User sent **Message** to User | **POST** /users/:uid1/messages/:uid2 | User1 PK and User2 FK | New Message JSON |
| User Delete sent **Message** | **DELETE** /messages/:mid | Message’s mid Pk | Delete Status |
| User retrieves his sent **Messages** | **GET** /users/:uid/messagesSent | User's PK | Message Sent |
| Retrieves Received **Message** | **GET**  /users/:uid/messagesReceived | User’s PK | Message Received |

Diagram

Description automatically generated

|  |  |
| --- | --- |
| ***1- User Message Another User*** |  |
| ***Request*** | POST  http://localhost:4000/api/users/6216773b8afb8157c7c855ac/messages/6216773b8afb8157c7c855ae |
|  | |
|  | |
| ***Response*** |  |
| {  "message": "bob's sends message to his friend",  "sentTo": "6216773b8afb8157c7c855ae",  "receivedFrom": "6216773b8afb8157c7c855ac",  "\_id": "621926eda91e890b134a323e",  "sentOn": "2022-02-25T18:58:53.437Z",  "\_\_v": 0  } | |

|  |  |
| --- | --- |
| ***2- User Delete Sent Message*** |  |
| ***Request*** | DELETE  http://localhost:4000/api/messages/621926eda91e890b134a323e |
|  | |
|  | |
| ***Response*** |  |
| {  "deletedCount": 1  } | |

|  |  |
| --- | --- |
| ***3- User Retrieve Sent Message*** |  |
| ***Request*** | GET  http://localhost:4000/api/users/6216773b8afb8157c7c855ae/messagesSent |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "621926eda91e890b134a323e",  "message": "bob's sends message to his friend",  "sentTo": "6216773b8afb8157c7c855ae",  "receivedFrom": "6216773b8afb8157c7c855ac",  "sentOn": "2022-02-25T18:58:53.437Z",  "\_\_v": 0  }  ] | |

|  |  |
| --- | --- |
| ***4- User Retrieves Received Message*** |  |
| ***Request*** | GET  http://localhost:4000/api/users/6216773b8afb8157c7c855ac/messagesReceived |
|  | |
|  | |
| ***Response*** |  |
| [  {  "\_id": "621926eda91e890b134a323e",  "message": "bob's sends message to his friend",  "sentTo": "6216773b8afb8157c7c855ae",  "receivedFrom": "6216773b8afb8157c7c855ac",  "sentOn": "2022-02-25T18:58:53.437Z",  "\_\_v": 0  }  ] | |

**2- Alternative UML and API’s for Message**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Method /resource/path*** | ***Request*** | ***Response Body*** |
| User retrieves All sent **Messages** | **GET** /messages |  | All sent Messages |
| User retrieves latest sent **Messages** | **GET** uid/sentOn/messages | Date | User latest sent Messages JSON array |

The highlighted one are the alternative methods

Diagram

Description automatically generated

**Pros**

GET : All sent Message

This will retrieve every user’s sent messages without mentioning any userid in the request

GET : Latest sent Messages

This will retrieve latest sent messages with the parameter of Date

**Cons**

It will be hard to track the information of all the latest sent messages without the use of user id as parameter.