Backend Request and Response documentation

Project Settings

- Maven project
- Program arguments: --spring.config.location=src/main/resources/application.properties --logging.config=src/main/resources/logback-stage.xml
- JRE - 1.8
- IDE - Eclipse / IntelliJ

Host: localhost:8089/pratilipiService

Tables

- I have used h2 database which is an in-memory database as there will not be a lot of data that is stored and also there will not be any need to create a database for the evaluator. It is a best coding practice for assignments of this kind.

1. Users
   Schema:

id	Auto incremented value
username	varchar(50)
password	Encrypted password

2. Stories

id	Auto incremented value
title	varchar(50)
url	varchar(250) - url for the story page transition
live_set	Set of all distinct users who are currently reading the story
read_set	Set of all distinct users who have read the story

APIs

1. /v1/user/signUp - Used to sign up a user
2. /v1/user/validateUser - Used to validate the username, I will not allow another user with same username
3. /v1/user/login - Used to login the user
4. /v1/story/stories - Used to get all the stories
5. /v1/story/readCount - Used to get the live readers and total read count of a story
6. /v1/story/logout - Used to update live reader count

/v1/user/signUp

- Curl
```
curl -X PUT \
  http://localhost:8089/pratilipiService/v1/user/signUp \
  -H 'Content-Type: application/json' \
  -H 'X-AUTH-TOKEN: abcd' \
  -d '{
        "username": "vamsi",
        "password": "vamsi"
}'
```

- Response scenario1 -
  When there is no user with same username. Response code will be 200.
  In this case, I store the data (username and encrypted password) in Users table in DB.

- Response scenario2 - when there is already a user with same username
```
{
    "timestamp": 1600005896930,
    "status": 500,
    "exception": "java.io.IOException",
    "message": "Username already exists, please give different one",
    "path": "/pratilipiService/v1/user/signUp"
}
```

/v1/user/validateUser

- Curl
  ```
  curl -X GET \
    'http://localhost:8089/pratilipiService/v1/user/validateUser?username=vamsi' \
    -H 'Content-Type: application/json' \
    -H 'X-AUTH-TOKEN: abcd'
  ```

- Response scenario 1
  When there is a user with same username in Users table. Response code will be 200.

- Response scenario 2 - where username does not exist
  ```
  {
     "timestamp": 1600006475930,
     "status": 500,    "exception": "java.io.IOException",
     "message": "Username does not exist",
     "path": "/pratilipiService/v1/user/validateUser"
  }
  ```

/v1/user/login

- Curl
  ```
  curl -X GET \
    'http://localhost:8089/pratilipiService/v1/user/login?username=vamsi&password=vamsi'
  \
    -H 'X-AUTH-TOKEN: abcd'
  ```

- Response scenario 1
  When there is a user with this username and the password also matches with the encrypted password in DB associated with the username. Response code would be 200.

- Response scenario 2
  When there is no user with username
  ```
  {
     "timestamp": 1600006756999,
     "status": 500,
     "exception": "java.io.IOException",
     "message": "Username does not exist",
     "path": "/pratilipiService/v1/user/login"
  ```

```
}
```

- Response scenario 3
  When there is mismatch in password

```
{
   "timestamp": 1600006775174,
   "status": 500,
   "exception": "java.io.IOException",
   "message": "Password mismatch",
   "path": "/pratilipiService/v1/user/login"
}
```

/v1/story/stories

- Curl

```
curl -X GET \
  http://localhost:8089/pratilipiService/v1/story/stories \
  -H 'X-AUTH-TOKEN: abcd'
```

- Response - returns list of all Stories in the table
  I created dummy stories in DB already in the code.
  I’m using this data to display the stories and transition to particular story page.

```
[
   {
      "title": "Hare",
      "url": "Story1",
      "liveSet": "[\"vamsi\",\"praty\"]",
      "readSet": "[\"vamsi\",\"praty\"]",
      "createdAt": 1600000748440,
      "updatedAt": 1600000748440
   },
   {
      "title": "Monkey",
      "url": "Story2",
      "liveSet": null,
      "readSet": null,
      "createdAt": 1600000748440,
      "updatedAt": 1600000748440
   },
   {
      "title": "Pigeon",
      "url": "Story3",
      "liveSet": null,
      "readSet": null,
      "createdAt": 1600000748441,
      "updatedAt": 1600000748441
   }
]
```

/v1/story/readCount

- Curl

```
curl -X POST \
  http://localhost:8089/pratilipiService/v1/story/readCount \
  -H 'Content-Type: application/json' \
  -H 'X-AUTH-TOKEN: abcd' \
  -d '{
          "userName": "vamsi",
          "password": "vamsi",
          "storyTitle": "Hare"
}'
```

- Response scenario 1

```
            {
                        "live_count": 1,
                        "read_count": 2
            }
```

  - First I check the validity of the user - by comparing the username and encrypted password present in the table with data passed in the request.
  - If it is a valid user, I check if the same user is already reading the story when I got the request, if not will update the live count.
  - If user is reading the story for the first time, will update the read count

- Response scenario 2
    - If user is not valid (username does not exist or password mismatch)- I throw exception

```
{
   "timestamp": 1600007248454,
   "status": 500,
   "error": "Internal Server Error",
   "exception": "java.io.IOException",
   "message": "Password mismatch",
   "path": "/pratilipiService/v1/story/readCount"
}
```

- Response scenario 3
    - If story does not exist - which generally does not happen as it is a call from front-end and front-end would send correct story name.

```
{
   "timestamp": 1600007417381,
   "status": 500,
   "error": "Internal Server Error",
   "exception": "org.springframework.dao.EmptyResultDataAccessException",
   "message": "Incorrect result size: expected 1, actual 0",
   "path": "/pratilipiService/v1/story/readCount"
}
```

/v1/story/logout

- This call is made in the following:
    - when user logs out of the site
    - when user clicks cross / back button from one story
    - when user moves from one story to a another story

```
curl -X POST \
  http://localhost:8089/pratilipiService/v1/story/logout \
  -H 'Content-Type: application/json' \
  -H 'X-AUTH-TOKEN: abcd' \
  -d '{
        "username": "praty",
        "password": "vamsi"
}'
```

- Response scenario 1
    - Validate the user credentials, then I check for the story which user had read previously and decrement the live count for the story.
    - I am doing this assuming the number of stories are less and maintaining live count with the story is better than maintaining each user’s activity at user level.
    - Response code would be 200.