IOnlyGames Documentation

Front end

Make sure you're in the directory of front end and run: `npm start`

Back end

1. Using Docker:

You can dockerize every service. For ease of development, we have provided a script called `run.sh`. This option lets you dockerize every service and includes `nginx`, a load balancer. If you only want to dockerize the mysql db, then follow instructions on 2. instead.

1. Make sure you install docker and docker-compose in your computer (If docker compose does not work, install it using pip3 i.e. `pip3 install docker-compose`)
2. Download the repository, and `cd backend`
3. Run `./run.sh run` and wait until the program finishes.
4. Wait for a few minutes (this will only be needed in the first run).
5. Visit `localhost` at port `80` and your spring application should be running there.

What if there are changes to the application, does it reflect?

Yes, every time you make a change it will automatically reload in the docker container.

Database management

You can visit `localhost:8081` that will open adminer, which allows db management of the mysql data.

Stopping the application

If you want to stop the application, run `./run.sh stop`.

2. Docker for mysql but no docker for spring

You can still run the application on your computer while having docker run the database.To do this, do the following:

1. Install java 11 in your machine.
2. Run `mvn clean install`
3. Then finally run:

```
./run.sh db-only
mvn spring-boot:run
```

Database management

You can visit `localhost:8081` that will open adminer, which allows db management of the mysql data.

What if there are changes to the application, does it reflect?

Yes, every time you make a change, the spring application should automatically reload.

Stopping the application

You can kill the mvn process through hitting `ctrl+c` in the terminal. And, you can stop the db container using: `./run.sh stop`

3. Without Docker at all

1. Ensure you're using IntelliJ
2. Ensure that you have MySQL Installed
3. Ensure that the MySQL Server is running
4. Make sure that you have a database up and running with the following properties:

How do the endpoints work

1. Auth Register: Endpoint to register new users in system.

`POST` `<url>/api/v1/auth/register`

Input:

Body:

```
{
  "username": "adam",
  "password": "password"
}
```

Output:

```
{
    "timestamp": "28-10-2020 11:06:25",
    "code": null,
    "result": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJwcG9wb3BvcG9wIiwiZXhwIjoxNjAzOTAxMTg1LCJpYXQiOjE2MDM4ODMxODV9.tHN21I95lyaH696EJcSROj6
}
```

2. Auth Login: Endpoint for users to login to the System.

`POST` `<url>/api/v1/auth/login`

Input:

Same as register input (see endpoint 1)

Output:

Same as register output (see endpoint 1)

3. Matches: Endpoint for users to obtain others users they are matched to.

`GET` `<url>/api/v1/users/matches`

Input:

No input required

Output:

```
{
    "timestamp": "07-11-2020 01:42:33",
    "code": null,
    "result": [
        [
            {
                "user": {
                    "username": "jonny",
                    "firstName": "Jonathon",
                    "lastName": "Jones",
                    "email": "jonny@hotmail.com",
                    "discordId": "discord#2144",
                    "steamId": "JonnyBonesJones",
                    "bio": "I like COD",
                    "location": "Sydney",
                    "avatarURL": "image.com",
                    "numberOfReports": 0,
                    "pubGPlayerName": "JonnyBE",
                    "runescapeDisplayName": "Zezima",
                    "battlenet": "JonnyJones"
                },
                "game": {
                    "name": "CODMW",
                    "iconURL": "https://img.icons8.com/ios/452/call-of-duty-modern-warfare.png",
                    "apiURL": null
                },
                "userGameID": "JonnyPwnsNoobs",
                "statistics": "1 million kills"
            }
        ]
    ]
}
```

4. Block: Endpoint that blocks a specific user.

`POST` `<url>/api/v1/users/block`

Input:

```
{
    "username": "jonny"
}
```

Output:

```
{
    "timestamp": "07-11-2020 07:21:14",
    "code": null,
    "result": "User 'jonny' has been blocked."
}
```

5. Unblock: Endpoint to unblock users.

`POST` `<url>/api/v1/users/unblock`

Input:

```
{
    "username": "jonny"
}
```

Output:

```
{
    "timestamp": "07-11-2020 07:24:16",
    "code": null,
    "result": "User 'jonny' has been unblocked."
}
```

6. Users Blocked: Endpoint to obtain list of blocked users.

`GET` `<url>/api/v1/users/users-blocked`

Input:

No input

Output:

```
{
    "timestamp": "07-11-2020 07:27:53",
    "code": null,
    "result": [
        {
            "username": "mohamad",
            "firstName": "Mohamad",
            "lastName": "baki",
            "email": "hotmail.com",
            "discordId": "sync",
            "steamId": "big",
            "bio": "i like cats",
            "location": "home",
            "avatarURL": "somepic.com",
            "numberOfReports": 2,
            "pubGPlayerName": pubglord,
            "runescapeDisplayName": Noob,
            "battlenet": null
        },
        {
            "username": "jonny",
            "firstName": "Jonathon",
            "lastName": "Jones",
            "email": "jonny@hotmail.com",
            "discordId": "discord#2144",
            "steamId": "JonnyBonesJones",
            "bio": "I like COD",
            "location": "Sydney",
            "avatarURL": "image.com",
            "numberOfReports": 0,
            "pubGPlayerName": "JonnyBE",
            "runescapeDisplayName": "Zezima",
            "battlenet": "JonnyJones"
        }
    ]
}
```

7. Search: Endpoint to obtain list of users like a given string

`GET` `<url>/api/v1/users/search`

Input:

Params:

```
<url>/api/v1/users/search?username=jon
```

Output:

```
{
    "timestamp": "08-11-2020 10:56:45",
    "code": null,
    "result": [
        "jon",
        "jonny"
    ]
}
```

8. Like: Endpoint to allow current user to like another user

`POST` `<url>/api/v1/users/like`

Input:

Body

```
{
    "username": "jonny"
}
```

Output:

```
{
    "timestamp": "08-11-2020 11:09:32",
    "code": null,
    "result": "User 'jonny' has been liked."
}
```

9. Dislike: Endpoint to allow current user to dislike a previously liked user.

`POST` `<url>/api/v1/users/like`

Input:

Body

```
{
    "username": "jonny"
}
```

Output:

```
{
    "timestamp": "08-11-2020 11:11:46",
    "code": null,
    "result": "jonny has been disliked."
}
```

10. Users Liked: Endpoint to allow current user to obtain list of liked users

`GET` `<url>/api/v1/users/users-liked`

Input:

No input

Output:

```
{
    "timestamp": "08-11-2020 11:13:35",
    "code": null,
    "result": [
        {
            "username": "jonny",
            "firstName": "Jonathon",
            "lastName": "Jones",
            "email": "jonny@hotmail.com",
            "discordId": "discord#2144",
            "steamId": "JonnyBonesJones",
            "bio": "I like COD",
            "location": "Sydney",
            "avatarURL": "image.com",
            "numberOfReports": 0,
            "pubGPlayerName": "JonnyBE",
            "runescapeDisplayName": "Zezima",
            "battlenet": "JonnyJones"
        }
    ]
}
```

11. Get Profile: Endpoint to allow a user to get another users profile

`GET` `<url>/api/v1/users/profile`

Input:

Params

```
<url>/api/v1/users/profile?username=jonny
```

Output:

```
{
    "timestamp": "08-11-2020 11:18:29",
    "code": null,
    "result": {
        "username": "jonny",
        "firstName": "Jonathon",
        "lastName": "Jones",
        "email": null,
        "discordId": null,
        "steamId": null,
        "bio": "I like COD",
        "location": "Sydney",
        "avatarURL": "image.com",
        "numberOfReports": 0,
        "pubGPlayerName": null,
        "runescapeDisplayName": null,
        "battlenet": null
    }
}
```

12. Update: Endpoint that allows a user to update their own profile details.

`PUT` `<url>/api/v1/users/update`

Input:

Body

```
{
    "firstName": "Jon",
    "lastName": "Jones",
    "email": "updatedEmail@hotmail.com",
    "discordId": "update-bones#3212"
}
```

Output:

```
{
    "timestamp": "08-11-2020 11:29:02",
    "code": null,
    "result": {
        "username": "jon",
        "firstName": "Jon",
        "lastName": "Jones",
        "email": "updatedEmail@hotmail.com",
        "discordId": "update-bones#3212",
        "steamId": null,
        "bio": "i love coca cola",
        "location": "Sydney",
        "avatarURL": "https://i.pinimg.com/474x/ab/b1/16/abb11623b8ff29604b1eceaa1e3894e0.jpg",
        "numberOfReports": 1,
        "pubGPlayerName": null,
        "runescapeDisplayName": null,
        "battlenet": null
    }
}
```

13. Report: Endpoint that allows a user to report another user.

`POST` `<url>/api/v1/users/report`

Input:

Body

```
{
    "username": "jonny"
}
```

Output:

```
{
    "timestamp": "08-11-2020 11:31:34",
    "code": null,
    "result": "User 'jonny' has been reported."
}
```

14. Add Availability: Endpoint that adds an availability to current user

`POST` `<url>/api/v1/users/availability/add`

Input:

Body - day (coded from 0 to 6), timeStart, timeEnd (coded as minutes from 0 to 1440)

```
{
    "day": "0",
    "timeStart": "600",
    "timeEnd": "630"
}
```

Output:

```
{
    "timestamp": "08-11-2020 12:13:01",
    "code": null,
    "result": "Availability has been successfully added."
}
```

15. Availability: Endpoint that obtains a list of a given users availabilities

`GET` `<url>/api/v1/users/availability`

Input:

Params `<url>/api/v1/users/availability?user=mohamad`

Output:

```
{
    "timestamp": "08-11-2020 12:13:51",
    "code": null,
    "result": [
        {
            "id": 85,
            "day": 0,
            "timeStart": 600,
            "timeEnd": 630
        },
        {
            "id": 86,
            "day": 1,
            "timeStart": 800,
            "timeEnd": 860
        }
    ]
}
```

16: Remove Availability: endpoint that allows a user to remove a previously added availability

`POST` `<url>/api/v1/users/availability/remove`

Input:

Body

```
{
    "id": 85
}
```

Output:

```
{
    "timestamp": "08-11-2020 12:17:52",
    "code": null,
    "result": "Availability successfully removed"
}
```

17. Get Games: Endpoint that returns current users details including games played and statistics

`GET` `<url>/api/v1/users/games`

Input:

No input

Output

```
{
    "timestamp": "08-11-2020 12:50:05",
    "code": null,
    "result": [
        {
            "name": "CODMW",
            "iconURL": "https://img.icons8.com/ios/452/call-of-duty-modern-warfare.png",
            "apiURL": null
        },
        {
            "name": "PUBG",
            "iconURL": "https://i.dlpng.com/static/png/6707559_preview.png",
            "apiURL": null
        }
    ]
}
```