

# Trade Master (GMOT application)

---

This project was developed with [React js](#) and [Django](#). It is assumed that the unix/macOS machine this project runs on already has python3 and node installed.

## Installation and setup

---

### Django

To run the backend server you need to complete the following steps -

Install python dependencies (navigate to the project\_backend directory and install all python dependencies)

```
cd project_backend/  
pip3 install -r requirements.txt
```

Make the database migrations in the same directory (this created and registers database tables with the server)

```
python3 manage.py makemigrations  
python3 manage.py migrate
```

Run the server on port 8001 (this opens the server on <http://127.0.0.1:8001/>). You may get a warning screen as the project is running on a DEBUG mode (you can just ignore this and continue).

```
python3 manage.py runserver 8001
```

This starts the server and you can use the admin panel at <http://127.0.0.1:8001/admin/>. This requires you to sign in by creating a SuperUser.

```
python3 manage.py createsuperuser
```

Alternatively, you can use credentials- {username: arya002, password: Login@1234}

### React

To run the frontend React server you need to complete the following steps -

Install node dependencies (navigate to the frontend directory and install all node dependencies)

```
cd project-frontend/  
npm install
```

This starts the frontend server at <http://localhost:3000/>. The port may differ if you have another application already open at port 3000.

```
npm start
```

Remember to upload the 2 json files (GTT and Trade data) provided to the website before making queries, or the results would come back empty.

## About the project

---

This project is made for the GMOT department of the Bank of America and enables the user to query data as well as upload json to be stored in the database.

Features :

- Upload GTT data json file which gets parsed and stored in the DataBase
- Upload Trade data json file which gets parsed and stored in the DataBase
- Query records by data
- Query records by trade ID
- Query records by client ID

## Parser

---

The parser has been implemented in a way to handle duplicate entries to avoid redundancy and any possible clashes. The json schema is also checked to make sure the files follow the right format before being parsed.

## Tech

---

Trade Master uses a number of open source projects to work properly:

- [ReactJS](#) - javaScript framework for froent-end design

- [Django](#) - python based back-end framework
- [npm](#) - package manager for JavaScript
- [Twitter Bootstrap](#) - great UI boilerplate for modern web apps
- [Material UI](#) - another great UI builder
- [Axios](#) - promise based HTTP client for the browser and node.js