App Aggregator

Steps to run locally:

Inside your virtual env run `pip install -r requirements.txt`

Please fill the UserExcel.xlsx file in the project folder and run `python manage.py create\_users` to generate User data.

Run `python manage.py migrate`

Available APIs(attaching Localhost apis):

Login(with generated username and password):

GET <http://localhost:8000/api/token/> - Use this jwt token as bearer for other apis.

Aggregator(Only Aggregator user type can access these apis):

Generate Apps by inserting playstore urls:

POST <http://localhost:8000/api/app/app_data/>

Payload : {‘url’: <playstore url>, ‘active’: <true/false>}

Ban Apps by its id:

PATCH http://localhost:8000/api/app/app\_data/{id}/ban\_app/

UnBan Apps by id:

PATCH http://localhost:8000/api/app/app\_data/{id}/unban\_app/

Delete Apps by id:

DELETE <http://localhost:8000/api/app/app_data/{id}/>

App List View(Can be viewed by all):

GET <http://localhost:8000/api/app/active_apps/>

User specific APIs (Every Aggregator is also a user.):

List view (Displays all purchased apps of that logged in user):

GET <http://localhost:8000/api/app/user_purchases/>

Purchase Apps by id (app id will be set by Frontend):

POST <http://localhost:8000/api/app/user_purchases/>

Payload : {‘app’: <app\_id from App list view>}

Archive Purchase (by purchase id):

PATCH http://localhost:8000/api/app/user\_purchases/{id}/archive\_purchase/

Unarchive Purchase (by purchase id):

PATCH <http://localhost:8000/api/app/user_purchases/{id}/unarchive_purchase/>

Delete Purchase (by purchase id):

DELETE http://localhost:8000/api/app/user\_purchases/{id}/delete\_purchase/

For Docker:

docker build -t django\_drf .