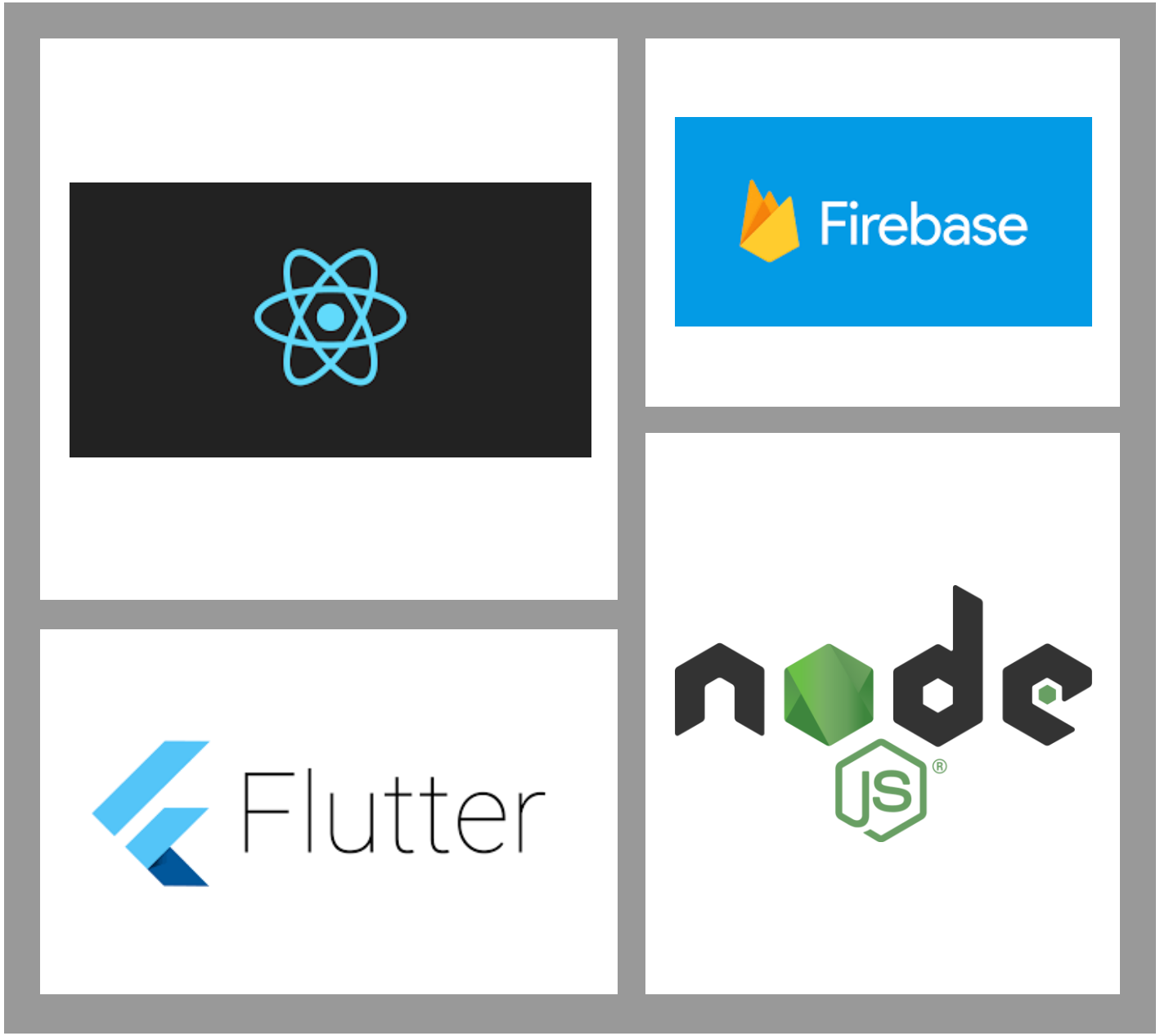
Area Potter

Technical Documentation



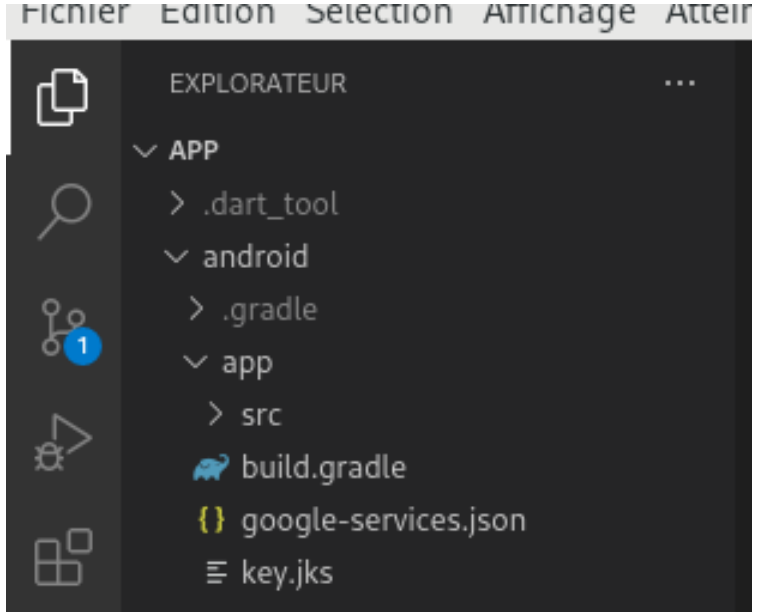
Summary

- Mobile requirements
- OAuth2 Flow / Mobile and Web
- User management
- Area management
- About.json
- How we use about.json in Front Web
- About.json use to getAllParams Front Web
- About.json use to getParamsFromAboutJson Front Mobile
- Server technical documentation

Mobile requirements

- Android Emulator compile Sdk Version must be 31 or greater
- Create a sha1 signature and key.jks file in client_mobile/android/app/key.jks
- Register your sha1 signature in Firebase
- Download and copy the googles-services.json in client_mobile/android/app/googles-services.json

- Make "adb reverse tcp:8080 tcp:8080"
  - It is to link server port to emulator android port
  - It is only util for google OAuth



Mobile config

- You can configurate the address and port of the Server for the mobile App

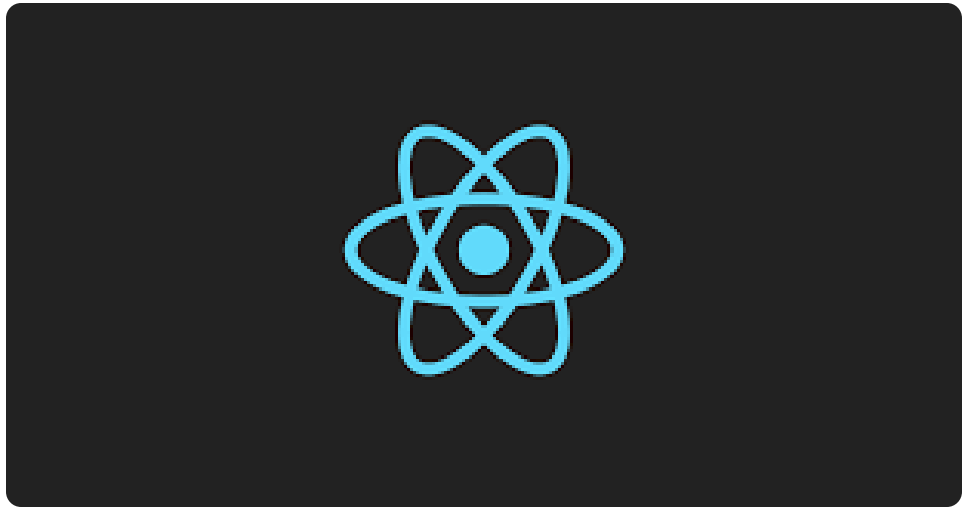
- You can easily set a new House Background (new theme)
- You can change the default House Background

app > lib > utils > config.dart > ...

```dart
String servFromMobileIp = "10.0.2.2";
String port = "8080";

// All House Background
Map<String, String> allHousesBackground = {
  "gryffindor": "assets/potter/house/gryffindor_house.jpg",
  "hufflepuff": "assets/potter/house/hufflepuff_house.jpg",
  "ravenclaw": "assets/potter/house/ravenclaw_house.jpg",
  "slytherin": "assets/potter/house/slytherin_house.JPG"
};

// Default House Background
String defaultHouseBackground = "assets/potter/house/slytherin_house.JPG";
```



OAuth2 Flow / Mobile and Web

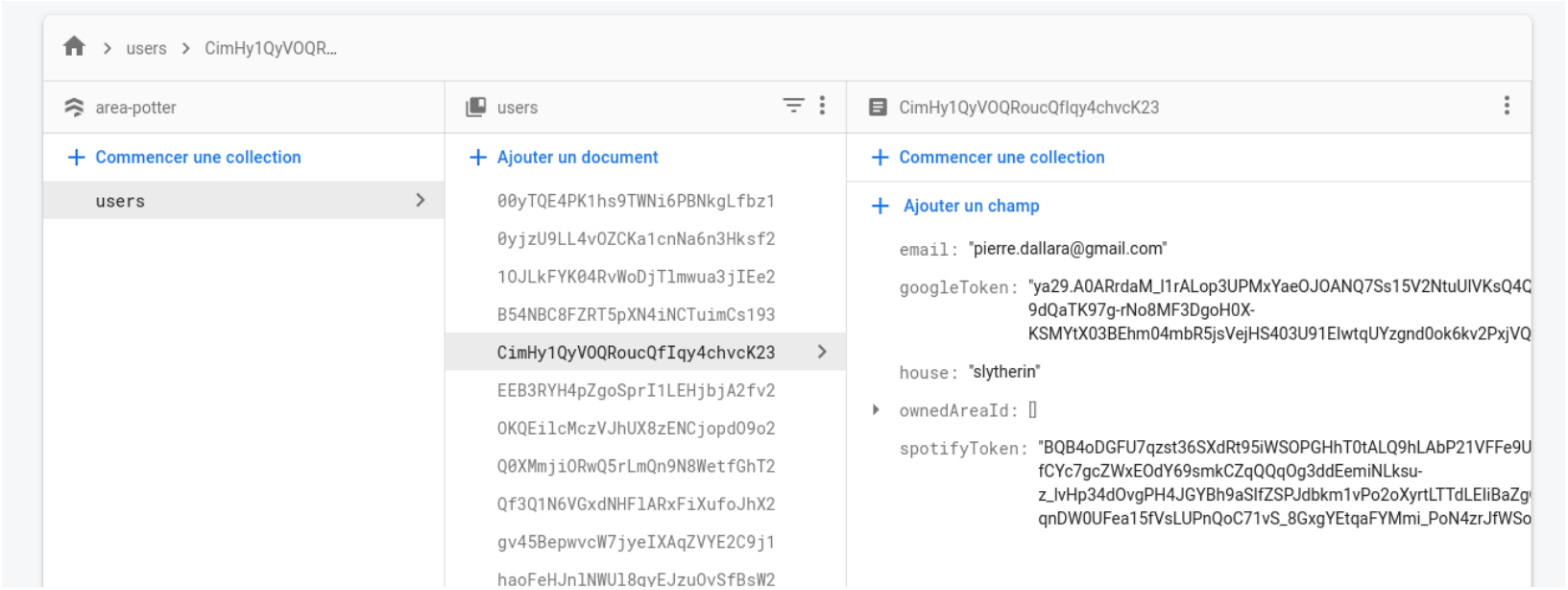
- GET to server/service/oauth2/authorize
- Web handle callback
- Callback is handled by Server for mobile

- After the user is connected to service, mobile and web ask Server to save Token:
  - POST server/savedb

Flutter

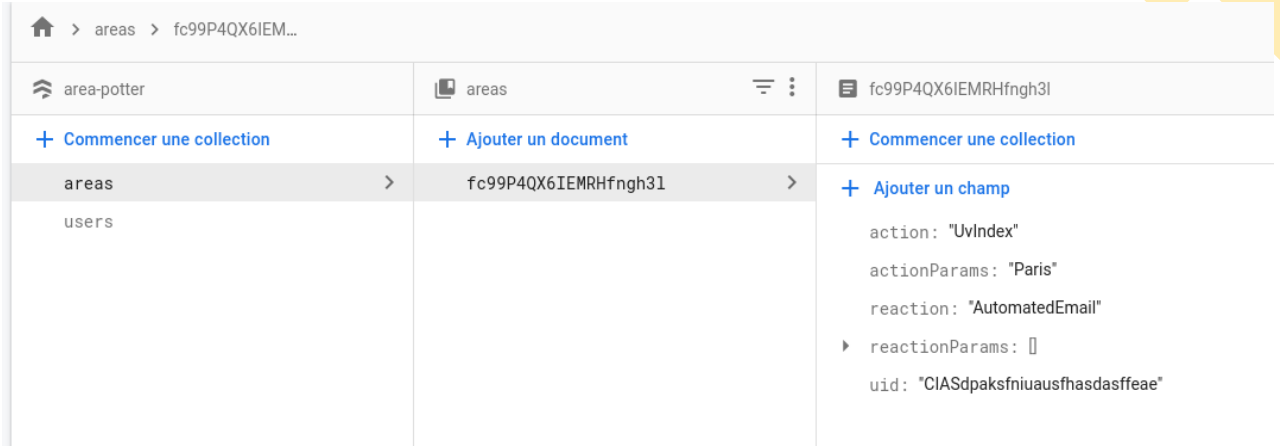
User management

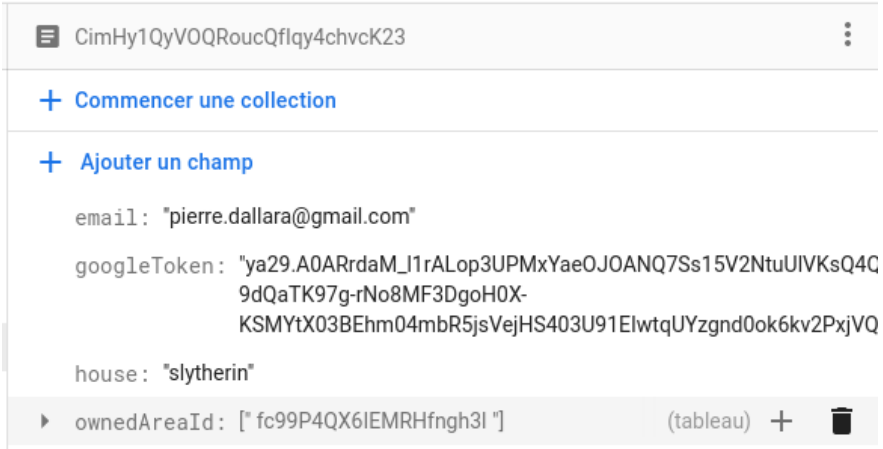
- Web and Mobile use Firebase Auth
- After a user signin with Google or withEmailAndPassword, we must:
  - Create a new user in FireStore Bdd
  - Collection "users"
  - New document with id = uid of new connected user
  - Fields:
    - "email": mail of user
    - "house": default house



Area management

- Add new doc in collection "areas"
- Add new id of doc in "ownedAreaId" of user
- Fields:
  - "uid": id of user who create Area
  - "action": name of action in about.json
  - "actionParams": Arguments of action if needed, optional in about.json
  - "reaction": name of reaction in about.json
  - "reactionParams": Arguments in array of reaction if needed, optional in about.json





about.json

- Mobile and Web get the about.json from Server
- Display Action, Action Arguments depends on the about.json
- Display Reaction, Reaction Arguments depends on the about.json
- Action Arguments and Reaction Arguments are optional value in about.json
- Display depends if Service isNotOauth, optional value in about.json


```
"services": [{
    "name": "weather",
    "actions": [{
        "name": "UvIndex",
        "description": "Uv index change in city changed",
        "args": [{
            "argName": "City"
        }]
    }, {
        "name": "Temperature",
        "description": "Temperature in city changed",
        "args": [{
            "argName": "City"
        }]
    }, {
        "name": "Weather ",
        "description": "Weather in a city changed",
        "args": [{
            "argName": "City"
        }]
    }],
    "isNotOauth": true
}, {
    "name": "github",
    "actions": [{
        "name": "Repos",
        "description": "User has a new repository"
    }, {
        "name": "Followers",
        "description": "User gets a new Follower"
    }],
    "reactions": [{
        "name": "Block",
        "description": "Block  a user",
        "args": [{
            "argName": "Username"
        }]
    }, {
        "name": "Follow",
        "description": "Follow a user",
        "args": [{
            "argName": "Username"
        }]
    }, {
```


How we use about.json in Front Mobile

- For each service form create we use the same function in "app/lib/utils/functions.dart": Future<List<List<Params>>>getParamsFrom AboutJson(String fromPage, String uid) async

- Each form for Create Area use class"app/lib/services/formCreate.dart"

Flutter

How we use about.json in Front Mobile

app > lib > models > params.dart > ...

```dart
// Class Params
//  Name                 define the text display of params name
//  tokenName            define the fields name of token in FireStore Bdd
class Params {
  String serviceName;
  String name;
  String description;
  List<String> args;

  Params(this.serviceName, this.name, this.description) : args = [];
}

```

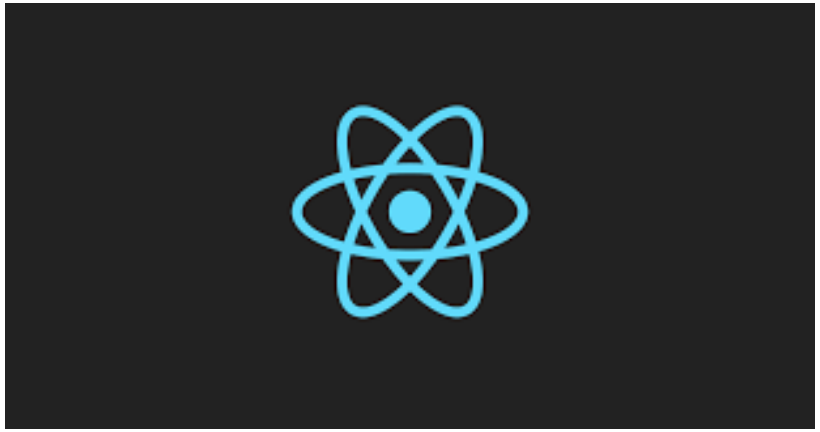
Function async Future<List<List<Params>>> getParamsFromAboutJson(String fromPage, String uid)

- FromPage is the name of Page
- Uid is id of current user connected
- Get about.json from Server
- Get doc of user in Firestore Bdd
- Each action and reaction are represented by a class Params

- Get all action of a service:
  - If page == service name
  - If field "service name" + "Token" is here in UserData ("spotifyToken", discordToken", ...)
  - Or if service isNotOauth is set to true

- Get all reaction of a service:
  - If field "service name" + "Token" is here in UserData ("spotifyToken", discordToken", ...)
  - Or if service isNotOauth is set to true

How we use about.json in Front Web



- For each form create Area, we use the same function to get all action and reaction infos

- In client/src/functions.jsx

async getAllParams(userData, page)

This functions returns an array with all actions infos at index 0 and all reactions infos at index 1

How we use about.json in Front Web

client > src > components > models > params.jsx > Action

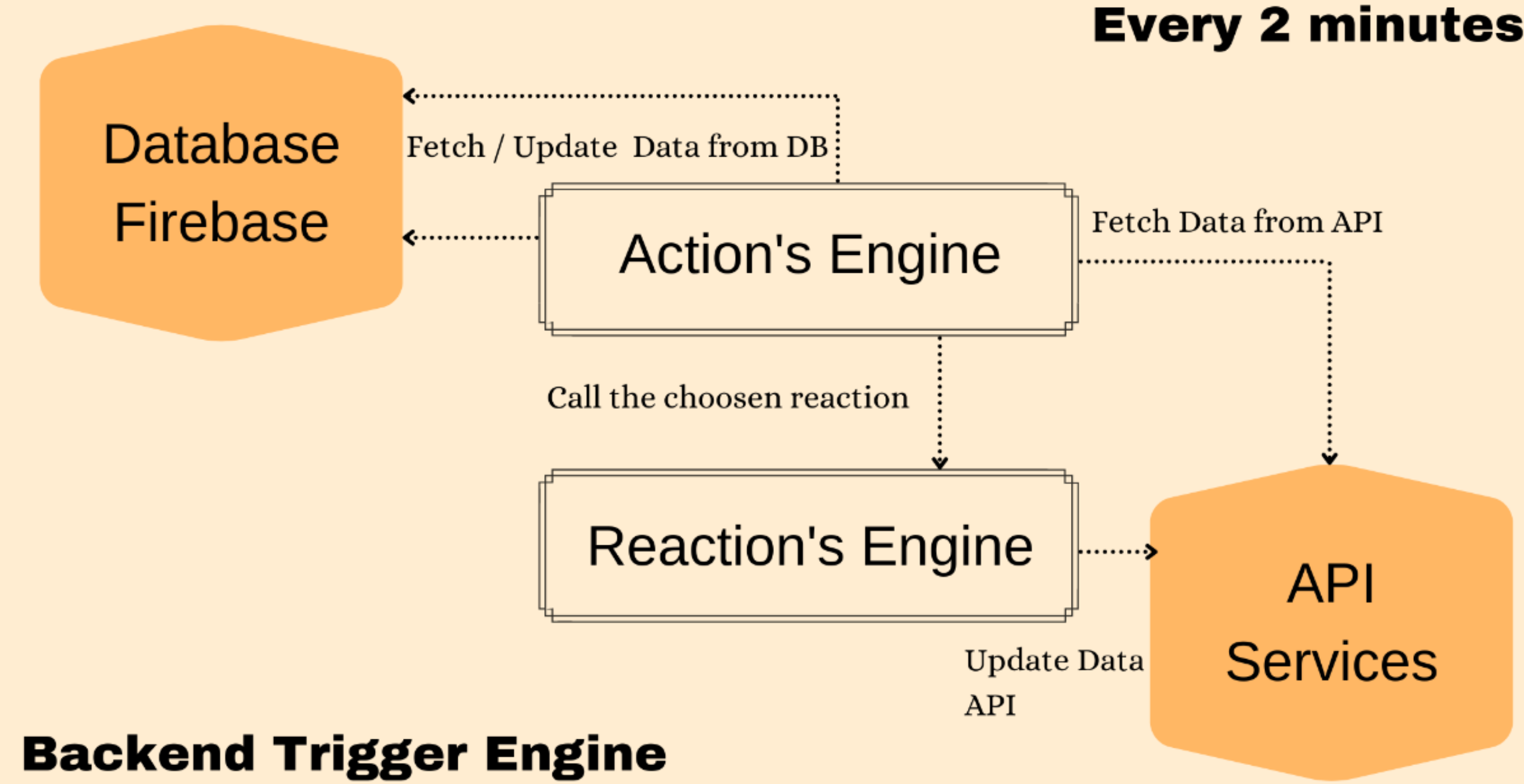
```
1   export class Action {
2       constructor(service, name, description) {
3         this.service = service;
4         this.name = name;
5         this.description = description;
6         this.args = [];
7       }
8   }
9
10  export class Reaction {
11    constructor(service, name, description) {
12      this.service = service;
13      this.name = name;
14      this.description = description;
15      this.args = [];
16    }
17  }
```

Function async getAllParams(userData, page)

- UserData is the doc in Firestore Bdd of the user
- Each action and reaction are represented by a class

- Get all action of a service:
  - If page == service name
  - If field "service name" + "Token" is here in UserData ("spotifyToken", discordToken", …)
  - Or if service isNotOauth is set to true

- Get all reaction of a service:
  - If field "service name" + "Token" is here in UserData ("spotifyToken", discordToken", …)
  - Or if service isNotOauth is set to true



Backend Trigger Engine