

API Flow

1. After Successfull login either by facebook or google.
 - a. Need to hit an api to register user in our database,
 - b. API Details

POST: <http://192.168.2.29:8000/registerUser>

Body:

```
{
  "User_Name": "g1",
  "Unique_User_Id": "gk4",
  "User_Type": "facebook"
}
```

Desired Response:

```
{
  "status": true,
  "result": {
    "User_Id": 6,
    "Round_Id": "def786b0-eea7-11e5-8403-d55abbc89005"
  }
}
```

- c. Response Detail
 - i. In response handle for "status": false; though it will not occur but need a handling from client end also.
 - ii. You will get User_Id and Round_Id in every registration request, which need to save and used in whole game; sending others request.
 - iii. As soon as round is complete on each restart or login you need to follow same process. *Note: don't use previous round Round_Id, as it will get changed in each round. As a good practice also replace User_Id on each round.
2. Each Level has a Question to show with options.
 - a. To get question and options to be shown on each level.

- b. API Details

GET: <http://192.168.2.29:8000/questions?level=1>

Desired Response:

```
{
  "status": true,
```

```

"result": [{
    "Question_Id": 1,
    "Option_Id": 1,
    "Level": 1,
    "Option_String": "option String1",
    "Question_String": "Questins1"
  }, {
    "Question_Id": 1,
    "Option_Id": 2,
    "Level": 1,
    "Option_String": "option String2",
    "Question_String": "Questins1"
  }, {
    "Question_Id": 1,
    "Option_Id": 3,
    "Level": 1,
    "Option_String": "option String3",
    "Question_String": "Questins1"
  }, {
    "Question_Id": 1,
    "Option_Id": 4,
    "Level": 1,
    "Option_String": "option String4",
    "Question_String": "Questins1"
  }
}]

```

c. Response Detail

- i. In response handle for "status": false; though it will not occur but need a handling from client end also.
- ii. You can see 1 at the end of HTTP Get request, which says get me a question and option list for level 1.
- iii. In result array you will have
 1. Question_Id - Id of a question shown, will be same in all element of array
 2. Option_Id - Id of option shown, **will not be same** in all element of array
 3. Level - level number, will be same in all element of array
 4. Option_String - Option string to be shown, **will not be same** in all element of array
 5. Question_String - Question String to be shown, will be same in all element of array

3. Register Successful Answer

- a. Need to hit an api on each successful answer submitted,

b. API Details

POST: <http://192.168.2.29:8000/registerAnswers>

```
Body: [{
    "User_Id": 1,
    "Round_Id": "rd1",
    "Question_Id": 1,
    "Option_Id": 1,
    "Option_Value": "value1",
    "Is_Final": 0,
    "Time_Spend": 500
}, {
    "User_Id": 1,
    "Round_Id": "rd1",
    "Question_Id": 1,
    "Option_Id": 2,
    "Option_Value": "value2",
    "Is_Final": 0,
    "Time_Spend": 500
}, {
    "User_Id": 1,
    "Round_Id": "rd1",
    "Question_Id": 1,
    "Option_Id": 3,
    "Option_Value": "value3",
    "Is_Final": 0,
    "Time_Spend": 500
}, {
    "User_Id": 1,
    "Round_Id": "rd1",
    "Question_Id": 1,
    "Option_Id": 4,
    "Option_Value": "value4",
    "Is_Final": 0,
    "Time_Spend": 500
}]
```

c. Request Detail

i. In request array you will have

1. User_Id - Get from session you created after login/round start
2. Round_Id - Get from session you created after login/round start
3. Question_Id - Id of a question shown, will be same in all element of array

4. Option_Id - Id of option shown, **will not be same** in all element of array
5. Option_Value - User input value
6. Is_Final - 0 or 1 ; 0 - not final submit, 1- final submit (will explain in detail)
7. Time_Spend - Time Spend on answering.

4. Register Incomplete Answer

- a. When user is trying to submit answer even though incomplete
- b. API Detail

POST: <http://192.168.2.29:8000/registerIncompleteAnswer>

Body:

```
{
    "User_Id": 1,
    "Round_Id": "rd1",
    "Question_Id": 1
}
```

c. Request Detail

- i. In request array you will have
 1. User_Id - Get from session you created after login/round start
 2. Round_Id - Get from session you created after login/round start
 3. Question_Id - Id of a question shown

5. Register Log/Event

- a. Register Event
- b. API Detail

POST: <http://192.168.2.29:8000/registerLog>

Body:

```
{
    "User_Id": 1,
    "Round_Id": "rd1",
    "Log_Type": "Replay",
    "Comment": "Level 1"
}
```

c. Request Detail

- i. In request array you will have
 1. User_Id - Get from session you created after login/round start
 2. Round_Id - Get from session you created after login/round start
 3. Log_Type - Event name triggered.

6. Register Loading Time

- a. Register Event
- b. API Detail

POST: <http://192.168.2.29:8000/registerLoadingTime>

Body:

```
{
    "User_Id": 1,
    "Round_Id": "rd1",
    "Time_Taken": 300
}
```

c. Request Detail

i. In request array you will have

1. User_Id - Get from session you created after login/round start
2. Round_Id - Get from session you created after login/round start
3. Time_Taken - Time in seconds in app loading.

7. Get Screen Event List

a. Get Screen Event List

b. API Details

GET: <http://192.168.2.29:8000/getScreenEvent>

Response:

```
{
    "status": true,
    "result": [
        {
            "Screen_Id": 1,
            "Screen_Name": "Screen1",
            "Created_On": 124564,
            "Updated_On": 464348
        },
        {
            "Screen_Id": 2,
            "Screen_Name": "Screen2",
            "Created_On": 124564,
            "Updated_On": 464348
        }
    ]
}
```

8. Record Screen Time

a. Record Screen Time

b. API Details

POST: <http://192.168.2.29:8000/recordScreen>

Body:

```
{  
    "User_Id": 1,  
    "Round_Id": "rd1",  
    "Screen_Id": 1,  
    "Time_Spend": 300  
}
```

c. Request Detail

i. In request array you will have

1. User_Id - Get from session you created after login/round start
2. Round_Id - Get from session you created after login/round start
3. Time_Spend - Time in seconds spent in each screen.