

Base url

localhost:3000

Our basic server

Request

Get

/PG13

Get all the cards with
categorie PG-13

Response

```
{
  "PG-13": {
    "answers": ["My liness ",
                "The auto-immune deficiency syndrome", ... ],
    "questions": ["This quarantine really got me horny for [[BLANK]].",
                  "Ew, I just stepped in [[BLANK]].", ... ]
  }
}
```

Request

Get

/PG

Get all the cards with
categorie PG

Response

```
{
  "PG": {
    "answers": ["Copious amounts of milk", "Shrek 2", ...],
    "questions": ["It's the newest Mario Kart item, [[BLANK]].",
                  "I sprayed my cat with [[BLANK]].", ...]
  }
}
```

Request

Get

/R

Get all the cards with categorie R

Response

```
{
  "R": {
    "answers": ["Nut sock", "Self-mutalation", ...],
    "questions": ["Stepbro, stop [[BLANK]].",
                  "[[BLANK]] really turns me on.", ...]
  }
}
```

Request

Get

/community

Get all the cards with categorie community

Response

```
{
  "community": {
    "answers": ["Rule 34", "Four little", ...],
    "questions": ["It's the newest Mario Kart item, [[BLANK]].",
                  "[[BLANK]] really turns me on.", ...]
  }
}
```

Request

Post

/community

Create a Room

Body

```
{
  "answer": "kids",
  "question": "These [[BLANK]] are so annoying"
}
```

(This is data that will be posted.)

Response

```
{
  "success": 1,
  "data": {
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 24,
    "serverStatus": 2,
    "warningCount": 0,
    "message": "",
    "protocol41": true,
    "changedRows": 0
  }
}
```

Request

Get

/R/id

Get the cards with an id

Response

```
{
  "success": 1,
  "R": {
    "answers": [
      {
        "id": 1,
        "value": "The purple crayon I used as a dildo when I was homeless"
      }
    ],
    "questions": [
      {
        "id": 1,
        "value": "Breaking news: A sex tape has surfaced where President Obama is seen having sex with [[BLANK]]."
      }
    ]
  }
}
```

All of the categories work the same way you can all call them using the /id parameter. I didn't do this for every category to remove repetition.

So all this endpoints will work:

pg/id

pg13/id

R/id

community/id

Request

Get

/Rooms

Get all the Rooms

Body

```
{
  "id_user": 2,
  "code": 125505,
  "open": 0
}
```

(This is data that will be posted. It needs to be json format since our api only accepts that.)

Response

```
{
  "success": 1,
  "Rooms": [
    {
      "id": 20,
      "id_user": 3,
      "code": 5000280,
      "open": 0
    },
    {
      "id": 23,
      "id_user": 2,
      "code": 12552005,
      "open": 1
    }
  ]
}
```

Request

Get

/Rooms/id

Get all the Rooms

Response

```
{
  "success": 1,
  "Room": {
    "room_id": 5,
    "id_user": 0,
    "code": 24,
    "cat": 0,
    "pub": 0,
    "open": 0
  }
}
```

(This is all the data with id=1)

Request

Post

/Rooms

Create a Room

Body

```
{
  "id_user": 2,
  "code": 125505,
  "open": 0
}
```

(This is data that will be posted)

Response

```
{
  "success": 1,
  "data": {
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 24,
    "serverStatus": 2,
    "warningCount": 0,
    "message": "",
    "protocol41": true,
    "changedRows": 0
  }
}
```

Request

Get

/Users

Get all the users and their info

Response

```
{
  "success": 1,
  "Users": [
    {
      "id_user": 1,
      "username": "test",
      "level": 0
    },
    {
      "id_user": 2,
      "username": "Ryan",
      "level": 0
    }
  ]
}
```


Request

Get

/Users/id

Get all the users on their id and their info

Response

```
{
  "success": 1,
  "User": {
    "id_user": 1,
    "username": "test",
    "level": 0
  }
}
```

Request

Get

/Stats

Get all the stats of the users

Response

```
{
  "success": 1,
  "Stats": [
    {
      "id": 1,
      "id_user": 1,
      "games_played": 5,
      "games_lost": 2,
      "games_won": 3
    },
    {
      "id": 2,
      "id_user": 2,
      "games_played": 4,
      "games_lost": 3,
      "games_won": 1
    }
  ]
}
```

Request

Get

/Stats/id

Get all the stats of the users

Response

```
{
  "success": 1,
  "Stat": [
    {
      "id": 1,
      "id_user": 1,
      "games_played": 5,
      "games_lost": 2,
      "games_won": 3
    }
  ]
}
```

(This doesn't look at the id of the stat but at the user id. This made more sense to me.)

Request

Get

/RoomUsers

Get all the stats of the users

Response

```
{
  "success": 1,
  "RoomsUsers": [
    {
      "room_id": 5,
      "user_id": 1
    }
  ]
}
```

Request

Get

/RoomUsers/id

Get all the stats of the users

Response

```
{
  "success": 1,
  "RoomsUsers": [
    {
      "room_id": 5,
      "user_id": 1
    }
  ]
}
```

(This looks at the room_id)

Request

Post

/RoomsUsers

Create a RoomUser a room can have multiple users

Body

```
{  
  "room_id": 5,  
  "user_id": 2,  
}
```

(This is data that will be posted)

Response

```
{  
  "success": 1,  
  "data": {  
    "fieldCount": 0,  
    "affectedRows": 1,  
    "insertId": 24,  
    "serverStatus": 2,  
    "warningCount": 0,  
    "message": "",  
    "protocol41": true,  
    "changedRows": 0  
  }  
}
```

Summary:

get(PG&PG/id) db: PGAnswers, PGQuestions

get(PG13&PG13/id) db: PG13Answers, PG13Questions

get(R&R/id) db: RAnswers, RQuestions

get(community&community/id) db:CommunityAnswers,
CommunityQuestions

post(community&community/id)
db:CommunityAnswers,
CommunityQuestions

get(rooms&rooms/id) db: Rooms

post(rooms) db: Rooms

get(roomUsers&roomUsers/id) db:RoomUsers

post(roomUsers) db: Users

get(users&users/id) db: Users

get(stats&stats/id) db: Stats

post(stats) db: Stats