**SIMPLE-TEST**

1. How to get started……………………………………………………………………………………………….
2. API structure………………………………………………………………………………………………………….
3. **GETTING STARTED**
4. Step own clone or download project from github,
5. Open folder in cmd, powershell or bash.
6. Download and install nodejs

* Goto <https://nodejs.org/>
* If OS is linux, run this command “sudo apt install nodejs”

1. Download a mysql demon from <https://www.mysql.com/downloads/>

OR download XAMPP from <https://www.apachefriends.org/index.html>

1. Start mysql demon, or XAMPP
2. Run this command “npm install” from cmd, powershell or bash.
3. Create a mysql Database with the name “fastPaceTransfer”
4. Verify all account details
5. Start server by running “node app.js”
6. Locate app from url “http://localhost:3000
7. Enjoy!
8. **APIs**

**POST(Create/Read)**

All POST requests send data in a JSON format, hence specify “Content-type:application/json”.

Example: {

  method: 'POST',

  headers: {

Content-Type:”application/json”

},

  body: {

  "answer": "This is ook",

  "questionRef": "OTGR%G"

}

}

**GET(Read)**

Most GET requests will likely return a rendered file, otherwise, will take a query in the URL.

Example with the JS fetch API

fetch(`${baseURL}/question/answer?question=OTGR%G`,{

  method: 'GET',

  })

**PUT(Update)**

PUT request acts almost the same as a POST request, the only difference being the **request** **method**

Example: {

  method: 'PUT',

  headers: {

Content-Type:”application/json”

},

  body: {

  "question": "What is code?",

  "questionRef": "OTGR%G"

}

}

**Delete**

Delete requests are handled similar ways as that of **POST** and **PUT,** with the significant difference being the **request method**

Example: {

  method: 'DELETE',

  headers: {

Content-Type:”application/json”

},

  body: {

  "questionRef": "OTGR%G"

}

}

**Authentication**

* The API authenticates each **User** sending a request via any method type.
* When a **User** logs in, a cookie is sent along the responds, this **cookie** contains an encrypted form of the **User.**
* When ever a request is sent, most APIs will look for the **cookie** named **user**
* Hence attach to the headers a **cookie** for authentication

**Example**

Example: {

  method: 'PUT',

  headers: {

Content-Type:”application/json”,

Cookie:"user=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Ik1hbmN1bmlhLUl4bUsiLCJpYXQiOjE2NDM xMTQ1MTIsImV4cCI6MTY0MzM3MzcxMn0.w\_UlFn17Paih56Ih5ktwgpFfjpG1UAOxKN0ncdqgJrM"

},

  body: {

  "question": "What is code?",

  "questionRef": "OTGR%G"

}

}