★HTTP requests

We have multiple options for calling Http request from javascript like XMLHttpRequest, fetch etc. for the examples given below I used fetch API

- GET request
    - Mainly used to fetch data,
    - Don't have much effect on data in server.
    - Eg:

        -
          ```
          fetch(url)
              .then(function(){
                  //code for handling data
              });
              .catch(function(){
                  //error handling
              });
          ```
- POST request
    - Mainly for sending data to the server
    - It have body
    - Eg

        -
          ```
          fetch( url, {
          method: 'POST',
          body:  JSON.stringify(body)
          }).then(function(){
              //code for handling data
          })
          .catch(function(){
              //error handling
          });
          ```
- PUT request
    - Mainly for updating and replacing the data in server
    - It have body
    - Eg

        -
          ```
          fetch( url, {
          method: 'PUT',
          body:  JSON.stringify(body)
          }).then(function(){
          ```

```
            //code for handling data
    })
    .catch(function(){
            //error handling
    });
```

★Authentication

Most commonly used authentication for web services are **Token Based Authentication.** In token based authentication the server will generate a token and send to the client devices. In client devices the token will be saved and it will add this token in the http header of all the request after that. So there server will evaluate header of all request and ensure the validity and presence of token.

- Eg :

  -
    ```
    fetch( loginurl,{
            method:"POST",
            Body: JSON.stingify(body),
    })
    .then(finction(){
            //it will be returned with some access token

            //Authorization="eyJhbGciOiJSUzI1NiIsImtpZCI6IjIzNTBiNWY2ND
            M0Zjc2Y2NiM2IxMTliZGQ4OGQxMzhjOWFjNTVmY2UiLCJ0eXAiOiJK
            V1QifQ.eyJpc3MiOiJodHRwczovL3NlY3VyZXRva2VuLmdvb2dsZS5jb2
            0vc3R1ZGVudG1hbmFnZW1lbnQtM2FmZDAiLCJhdWQiOiJzdHVkZW5
            0bWFuYWdlbWVudC0zYWZkMCIsImF1dGhfdGltZSI6MTU0Mzk0MjQw
            MCwidXNlcl9pZCI6IlVNNWpQNnBiUGtSUXg1clRSQ2g0ZkpUQ1dZbDIi
            LCJzdWIiOiJVTTVqUDZwYlBrUlF4NXJUUkNoNGZKVENXWWwyIiwia
            WF0IjoxNTQzOTQyNDAwLCJleHAiOjE1NDM5NDYwMDAsImVtYWlsIj
            oidmlzaG51Y2hpcnVrYW5kYXRoQGdtYWlsLmNvbSIsImVtYWlsX3Zlc
            mlmaWVkIjp0cnVlLCJmaXJlYmFzZSI6eyJpZGVudGl0aWVzIjp7ImVtY
            WlsIjpbInZpc2hudWNoaXJ1a2FuZGF0aEBnbWFpbC5jb20iXX0sInNpZ
            25faW5fcHJvdmlkZXIiOiJwYXNzd29yZCJ9fQ.abINe7F9zjkI3G5Mez1t6
            PFIsEnzxky6iccbnhXX4UiwRWmgmS38xU8-CYkqLL9zjJF9AdhFER9s
            EGgScoXeBe_YsOQUTP-VrYW_cb8R-2BcDmO7KOFisVA841EFZWjn
            O4-pK-QrkR21OUvOq1uxnjvKfo6I96LGQoAmE5HK10_7QIIn32uaFnxY
            u0AcSCkM8MMNZkXwWCFrpM4xdGy78EHvMAqN7DRFzEhfUubmKx
    ```

```
                        MWpjJHDLd6INn-FK8KiWwxtVN3j8_zREkdrkvWFoP-up8JaB2-4Uown5
                        X8lusZv2p999fUDSdP22gH0SOgIVvAk_Fx2l_2Shs09DLPkOgIIA”
                })
```

For every request after this we need to append this token into header

```
Eg:     fetch(url,{
        Method: ‘GET/POST/…..’,
        Headers: {
                Authorization=”eyJhbGciOiJSUzI1NiIsImtpZCI6IjIzNTBiNWY2NDM0Zjc2Y2NiM2IxMTlmZGQ4
                OGQxMzhjOWFjNTVmY2UiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL3NlY3VyZXRva2VuL
                mdvb2dsZS5jb20vc3R1ZGVudG1hbmFnZW1lbnQtM2FmZDAiLCJhdWQiOiJzdHVkZW50bWFuY
                WdlbWVudC0zYWZkMCIsImF1dGhfdGltZSI6MTU0Mzk0MjQwMCwidXNlcl9pZCI6IlVNNWpQNnBi
                UGtSUXg1clRSQ2g0ZkpUQ1dZbDIiLCJzdWIiOiJVTTVqUDZwYlBrUlF4NXJUUkNoNGZKVENXW
                WwyIiwiaWF0IjoxNTQzOTQyNDAwLCJleHAiOjE1NDM5NDYwMDAsImVtYWlsIjoidmlzaG51Y2hpc
                nVrYW5kYXRoQGdtYWlsLmNvbSIsImVtYWlsX3ZlcmlmaWVkIjp0cnVlLCJmaXJlYmFzZSI6eyJpZ
                GVudGl0aWVzIjp7ImVtYWlsIjpbInZpc2hudWNoaXJ1a2FuZGF0aEBnbWFpbC5jb20iXX0sInNpZ25
                faW5fcHJvdmlkZXIiOiJwYXNzd29yZCJ9fQ.abINe7F9zjkI3G5Mez1t6PFIsEnzxky6iccbnhXX4UiwR
                WmgmS38xU8-CYkqLL9zjJF9AdhFER9sEGgScoXeBe_YsOQUTP-VrYW_cb8R-2BcDmO7KOFis
                VA841EFZWjnO4-pK-QrkR21OUvOq1uxnjvKfo6I96LGQoAmE5HK10_7QIIn32uaFnxYu0AcSCkM8
                MMNZkXwWCFrpM4xdGy78EHvMAqN7DRFzEhfUubmKxMWpjJHDLd6INn-FK8KiWwxtVN3j8_zR
                EkdrkvWFoP-up8JaB2-4Uown5X8lusZv2p999fUDSdP22gH0SOgIVvAk_Fx2l_2Shs09DLPkOgIIA”

                }
}) .then(()=>{
})
.catch(()=>){
}
```

★BEST JSON RESPONSE

It is mainly depends on the developer, but in my experience what i felt the beat format is given below

```
//for success response
{
   "status_code": 406,
   "error": false,
   "data": {}
}


//for error message
{
   "status_code": 406,
   "error": true,
   "error_code": "oauth_invalid_token",
   "message": "The OAuth token is invalid."
}
```

In this type of response we could easily categories the response, so the error handling and management of response will be easy. Error filed will ensure data availability.