Data Structure

Example :

URL =www.workshop-people.com.

JSON file : [
{ ‘id’ : 1930224,
‘Name ‘ : Valerian,
‘Gender’: 1
},
{‘id’ : 2344546,
‘Name ‘ : ‘Jeremy’,
‘Gender’: 1
}]

=> The server add an random ID to an object if it is not specified

Basics : POST, GET, DELETE, UPDATE (query, APIkey)

- /// GET INFO.

GET = Ask Data to the server (just asking but never save info)
-> parameter (URL/?var)

Example : See Valerian info

```
Fetch (URL/?id=1930224, {                    /// this is the ID of valerian
Method : ‘GET’
})
.then ….   // Transform data into JSON file
.then …    // Do what you want with this JSON file
.catch …   // Do that .. if something went wrong
```

- /// POST INFO.

POST = Save info in the server, send data to the server
No parameters but data from the Body
-> body : { we send img, title, text, …}

Example : Add Lily info

```
Fetch ( URL, {
Method : ‘POST’,
```

```
// Headers to see later...
Body :  {
        'id' : 304595,
        'Name ' : 'Lily',
        'Gender': 0
        }
} )

=>    Now JSON file is :
        [     {       'id' : 1930224,
                      'Name ' : Valerian,
                      'Gender': 1
              },
              {       'id' : 2344546,
                      'Name ' : 'Jeremy',
                      'Gender': 1
              },
              {       'id' : 304595,
                      'Name ' : Lily,
                      'Gender': 0
              }]
```

- /// MAKE AN UPDATE

```
UPDATE = change some stuff from your post data
->parameter (URL/?ID)
-> body : {title,  ...}

Fetch ( URL/?id=304595 , {                          /// change something in Lily
info with her ID
Method : 'UPDATE',
// Headers to see later...
Body :  { gender : 1                                /// Write only what you wanna
change.
} )

=>    Now JSON file is :
        [     {       'id' : 1930224,
                      'Name ' : Valerian,
                      'Gender': 1
              },
              {       'id' : 2344546,
                      'Name ' : 'Jeremy',
                      'Gender': 1
              },
              {       'id' : 304595,
```

```
        'Name ' : Lily,
        'Gender': 1                                  ///  ( Now I'm a man!)
    }]
```

- /// DELETE objects

DELETE = delete objects from your JSON file
-> parameter (URL/?id).
!! Delete ALL elements linked to the given ID

Example : Delete valerian info from the data base

```
Fetch ( URL/?id=1930224 , {                          /// ID of valerian
Method : 'DELETE'
} )
```

```
=>   Now JSON file is :
     [   {    'id' : 2344546,
              'Name ' : 'Jeremy',
              'Gender': 1
         },
         {    'id' : 304595,
              'Name ' : Lily,
              'Gender': 1
         }]
```

How to use Fetch

Fetch sit between the Front-end & Back-end (create in the JS file, Respond in Html)

```
<form>
    <input SEARCH BAR />          // search the title movie
    <input type= 'submit' />      // submit your search
</form>
```

```
input.value = John                // look for a movie title with John in it
```

```
url = apiMovieSearch/?title=input.value
```

DOM In JS
addEventListener('submit', function)

Function =

```
fetch (URL/?title=input.value {
```

Method: ‘GET’,
	// If everything ok OK ===200 :

.then (receivedData => receivedData.JSON()) // transform stringData into parsedData
.then (parsedData => function { //Do what you wanna do with that Data }

})

	// Else !== 200
.catch(error => display the error to the user // THIS MOVIE DOESN’T EXIST)

Headers

Mostly use for POST and UPDATE
Headers : {
	// say what type of info you give
‘Content-type’ : ‘application/json’ // we are now working with a son file…
}
Body : DataYouWantToSend.stringify() OR ‘’DataYouWantToSend’’. /// retransform your JSON data in STRING data

.MAP

.map is a LOOP system for arrays

You must do it in the second .then (-> when your data has been parsed).

parsedData = [{movie1}, {movie2}, {movie3}]

parsedData.map(movie => { ulTag.innerHTML =` <li> ${movie.title} </li>`)
	// for each movie in the parsedData create a <li> with the movie title in it