Can be accessed thru URLs

UI

CRUD

Frontend Application

html
CSS

JS

BE URL

DB

Backend Application
Spring/Springboot app

Data
Source

MySQL DB
MongoDB

CRUD

http://localhost:5500/home.html

Backend app URLs
http://x.x.x.x:yyyy/getdata
http://x.x.x.x:yyyy/savedata
http://x.x.x.x:yyyy/updatedata
http://x.x.x.x:yyyy/deletedata
http://x.x.x.x:yyyy/authenticate
http://x.x.x.x:yyyy/signup

JSON-server

Frontend Application

html
CSS

JS

BE URL

Backend Application
Req handling+Data source

req
handler

data
board.json

CRUD

http://localhost:5500/home.html

BE URL
http://localhost:3000/books
insert/retreive/update/delete
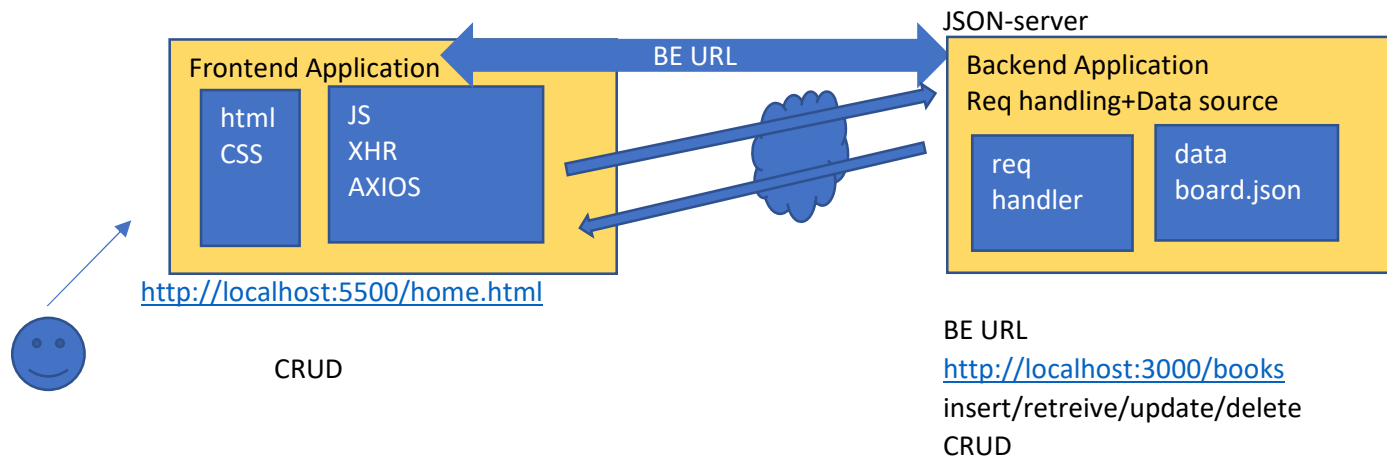CRUD

http   header
                        Information about traveling
Request   data+header Request header : information about request
Reponse   data+header Response header : information about response

Types of request

GET *   retreiving data
POST   sending data(insert)   <form method="POST"
PUT   updating data
DELETE   delete data
..
..

JSON-server

Frontend Application	BE URL	Backend Application

Frontend Application
html
CSS
JS
XHR
AXIOS

Backend Application
Req handling+Data source
req handler
data board.json

http://localhost:5500/home.html

CRUD

BE URL
http://localhost:3000/books
insert/retreive/update/delete
CRUD

XHR   Synchronous   t1
XmlHttpRequest   Asynchronous   t2
                                                    t3
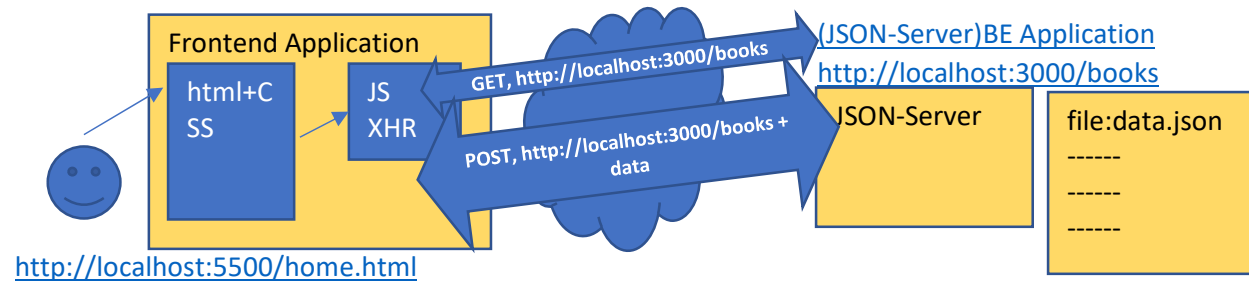   t4

post   req+data
get   req
delete   req
put   req+data
..
..
..

XHR   Ready state   Response status

UNSENT=0; // initial state   Response codes
OPENED=1; // called open   100-199   Informational Response
HEADERS_RECEIVED=2; // response headers received   200-299   Successful response
LOADING=3; // response is loading (data packets received) 300-399   Redirection message
DONE=4; // request complete   400-499   Client error response
                                                     500-599   Server error response

Demo

Frontend Application

html+C
SS

JS
XHR

GET, http://localhost:3000/books

POST, http://localhost:3000/books + data

(JSON-Server)BE Application
http://localhost:3000/books

JSON-Server

file:data.json

------

http://localhost:5500/home.html

Step 1   Make sure JSON server is running and URL is active

```
C:\Wave-31\Exers-Demos\js\demos\sprint5>json-server -w data.json

\{^_^}/ hi!

Loading data.json
Done

Resources
http://localhost:3000/books

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...

GET /books 304 23.357 ms - -
```

localhost:3000/books

```
[
  {
    "id": "B001",
    "name": "Spirit of C",
    "subject": "C Language",
    "author": "M Cooper",
    "price": 230
  },
  {
    "id": "B002",
    "name": "Servlets and JSP",
    "subject": "JSP",
    "author": "MC-G-Hil",
    "price": 435
  },
```

Step 2   Create js file
              create XHR object
              set properties to XHR (type,URL)
              send the request

Step 3   Receive response
              display response data in console.log

Step 4   Render response on webpage

Step 1   Make sure JSON server is running and URL is active

```
C:\Wave-31\Exers-Demos\js\demos\sprint5>json-server -w data.json

\{^_^}/ hi!

Loading data.json
Done

Resources
http://localhost:3000/books

Home
http://localhost:3000

Type s + enter at any time to create a snapshot of the database
Watching...

GET /books 304 23.357 ms - -
```

← → C ① localhost:3000/books

```
[
  {
    "id": "B001",
    "name": "Spirit of C",
    "subject": "C Language",
    "author": "M Cooper",
    "price": 230
  },
  {
    "id": "B002",
    "name": "Servlets and JSP",
    "subject": "JSP",
    "author": "MC-G-Hil",
    "price": 435
  },
```

Step 2   in html page
                create form to capture book data

Step 3   in js file
                write logic to capture form data when submitted

                                                                                        b001
                create XHR object (POST,URL)   C
                                                                                        abcd

                                                            Delete

        b001   C   ..   .   .   .   ■   update
        B002   JAV   ..   .   .   .   ■   update
        B003   .NET   .   .   .   .   ■   update

B01	delete

update

js
hardcoded logic to delete a particular record

hardcoded logic to update particular records with particular details

id	pwd	role	status
myid	mypwd	USER	Active