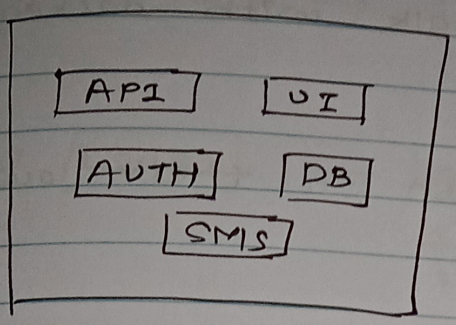
Episode 6 - Exploring the world!

apsara

Date: / /

↳ Monolith

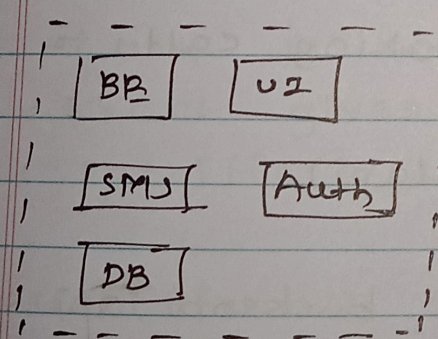
Traditional Apps developed with monolith architecture



- All code written in one single project
- both frontend & backend in 1 project
- complicated to maintain.

↳ Microservices

Different Services for different jobs



This all are microservices combine together forms a big app.

This is also known as separation of concern.

All the micro services talks together depending on the use cases.

Each & every service has its own job.

• How do this services interact with each other?

- Every services needs to talk each other
  e.g : UI needs to talk with backend, etc

- All the services run on their own ports

port	→	services	(domain / URL)
:1234	→	UI	/
:1000	→	BE	/api
:8000	→	SMS	/sms

After all this services map to the domain name
e.g domain name : mamastedev.com/api

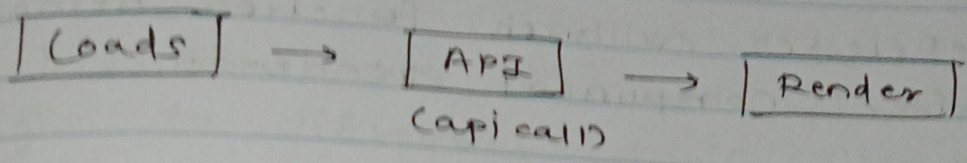
Services interact after making call to different URLs.
e.g UI call BE URL (/api)

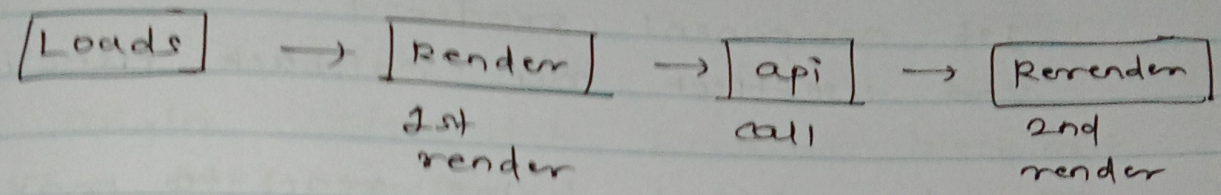
• How React app make a backend api call? How to fetch data dynamically & used in react app.

2 approaches (How UI applications fetch the data from backend)

- when to make an api call?

1) As soon as page loads



2) 

In react, 2nd approach get used. It gives better UX (user experience)

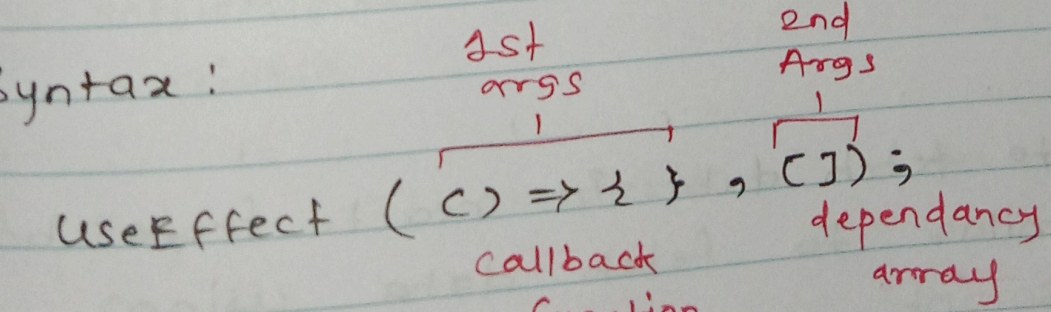
1st Render → Render Skeleton
end Render → Render data

↳ useEffect()

A hook is normal javascript function, which react gives to us and it has its own specific use case

useEffect() comes from react library and we can import it as named import

Syntax:



- when will this useEffect() callback function being called?

useEffect() callback function being called after component rendered.

1st component rendered
2nd useEffect run callback function

Imp: useEffect helps react to make api calls.
Fetchdata inside useEffect callback function.

- Logic for fetching data

1] useEffect (() => {
    fetchData();
});

2] const fetchData = async () => {

const data = await fetch ("URL");

const json = await data.json();
console.log (json);
}

Logic to fetch data in react from backend is similar to fetch data using javascript

Error: "Access to fetch Data at swiggy.com from 'localhost:1234' has been blocked by CORS policy".

- A browser block us to call api from one origin to different origin (localhost to api)

- If origin mismatch browser block call i.e CORS policy.

- way to bypass CORS
  Use or Install chrome extension
  Allow CORS: Access-control-Allow-origin

↳ Live API data

- Rerendering happens in 2nd approach when we update state variable

① Make 1st render
↓
② useEffect() to fetch Data
↓
③ after fetching data, put data into setListOfRestaurant() to update state variable
↓
④ Page will rerender, data will show on UI.

↳ Shimmer UI

Shimmer UI is like skeleton for our UI.

We can render it on 1st render of our app when page is load.

It makes UX more better

During the process of fetching data or making the api call we can show shimmer UI to user.

↳ Conditional Rendering

Rendering on the basis of some condition

Ternary operator

eg return listofRes.length === 0 ? <shimmer/>
: (JSX);

↳ useState()

- Why we need usestate() state variable?

1) updating button value with JS variable in react

eg let btnName = "Login";

```
return (
    <button onClick={() => {
        btnName = "Logout"
        console.log (btnName);
    }}>
    {btnName}
    </button>
```

- onClick event update the value of button to logout

- But UI did not get updated. It did not get render.

- To update UI we need to refresh or render component to take updated values show on UI

- In such cases, Javascript variables does not works

- If we want to make component dynamic, here we use state variable

- To render UI with updated value we use state variables

2] Updating button value with React variable

eg `const [btnNameReact, setBtnNameReact] = useState ("Login");`

```
return (
  <button onClick={() => {
    setBtnName("Logout");
  }}>{btnName}</button>
```

- when we update button value with state variable, react will triggers the render process once again.

- whenever state variable changes react refresh the component

- Is react render whole component or button value?
  ⇒ React render the whole component but only change button value with updated value

- How const variable is getting updated?

We can't modify const variable

Whenever react rerender the component with updated value, the variable with updated value is a new variable render with component

The state variable is changed with its function, react creat new variable with new value and render it with component

apsara

Date

Search filter

1] state variables

```
const [searchTxt, setSearchTxt] = useState("");
const [filteredRestaurant, setFilteredRestaurant]
  = useState([]);
```

2) Search-box

```
<input value = {searchTxt}
  onChange = {(e) => setSearchTxt(e.target.value);
  }/>

<button className = 'search-btn'
  onClick = {() => {

const filteredRes = listOfRestaurant.filter(
    (res) => res.info.name.toLowerCase()
  .includes(searchTxt.toLowerCase()));
setFilteredRestaurant(filteredRes);
}}> search </button>
```

3] Rendering cards

```
<div className = "res-container">
  {
    filteredRestaurant.map((res) =>
  <RestaurantCard key = {res.info.id}
  resData = {res} />)
  }
```

4] While fetching Data

```
const fetchData = async () => {
setFilteredRestaurant (json?.data?.cards[4].
?.card?.card?.gridElements?.infoWithStyle?.
                                restaurants);
```

apsara

Episode 6 (6.1) Swiggy API Issues Resolved
(6.2) CORS Plugin Issue Solved.

- Fetch / XHR

Go to Inspect element → Network tab → Fetch / XHR (apply filter)

It will give us api URLs only. and we can able to fetch backend api data from those URL.

[All] → filter give us all the URLs fetched for our page.

- JSON Viewer chrome Extension

When we open api in a new tab. It will hard to read the data

Install JSON Viewer pulgin to chrome the data will in readable format.

- CORS Plugin (Get rid from CORS plugin)

- corsproxy.io

- Helps us to bypass CORS Error