4 Different ways we can call API in JavaScript

1. Axios
2. Fetch
3. XML HTTP Request
4. jQuery AJAX

@coders_notes

1] Axios

- It is an open source library for making HTTP requests.
- It works on both Browsers and Node js.
- It can be included in an HTML file by using an external CDN.
- It also returns promises like fetch API.

```
<script
src = "https://cdn.jsdelivr.net/npm/axios/dist/
        axios.min.js"></script>

axios.get ("https://jsonplaceholder.typicode.com/
                todos")
.then (response => {
        console.log (response.data)
})
```

2] Fetch

- The Fetch API provides an interface for fetching resources (including across the network) in an asynchronous manner.

- It returns a Promise

- It is an object which contains a single value either a Response or an Error that occured.

- .then() tells the program what to do once promise is completed.

```
fetch('https://jsonplaceholder.typicode.com/
        todos').then(response=> {
            return response.json();
  }).then(data => {
        console.log(data);
  })
```

@coders_notes

3] XML HTTP Request

- All modern browsers support the XMLHttp Request object to request data from a server.

- It works on the oldest browsers as well as on new ones.

- It was deprecated in ES6 but is still widely used.

Scanned by CamScanner

```
var request = new XMLHttpRequest();
request.open('GET',
'https://jsonplaceholder.typicode.com/todos')
 request.send();
 request.onload = () => {

console.log(JSON.parse(request.response));
  }
```

@coders_notes

4] jQuery AJAX

- It performs asynchronous HTTP requests.
- Uses $.ajax() method to make the requests.

```
<script
src = "https://ajax.googleapis.com/ajax/libs
/jquery/3.5.1/jquery.min.js"></script>

$(document).ready(function() {
        $.ajax({
              url:
"https://jsonplaceholder.typicode.com/todos",
        type : "GET",
        Sucess : function (result) {
         console.log(result);
             }
     })
})
```