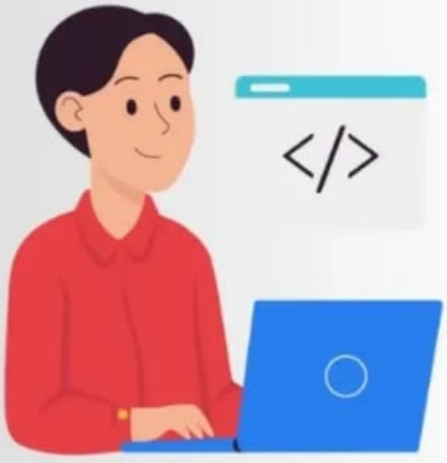
TYPES OF APIs



@frontend_in_depth

The most important and widely used types of APIs are:

- REST APIs
- SOAP APIs
- GraphQL APIs
- WebSocket APIs

REST APIs

- Most popular and commonly used API type. REST APIs adhere to a set of principles and are based on standard HTTP methods like GET, POST, PUT, and DELETE.

- Ideal for web services and mobile applications due to their simplicity, Scalability, and use of ISON for data exchange.

```
const url = 'https://api.github.com/repos/nodejs/node';

fetch(url)
    .then(response => response.json())
    .then(data => {
        console.log('Repository name:', data.name);
        console.log('Description:', data.description);
        console.log('Stars:', data.stargazers_count);
    })
    .catch(error => {
        console.error('Failed to retrieve data:', error);
    });
```

SOAP APIs

- Older but still important, especially in enterprise applications where strong security standards and formal contracts are required.

- Uses XML messaging and is suitable for banking, payment processing, and other industries with high security requirements.

SOAP API

```
const soap = require('soap');

// SOAP API WSDL URL
const url = 'http://www.dneonline.com/calculator.asmx?WSDL';

soap.createClient(url, (err, client) => {
    if (err) {
        console.error('Error creating SOAP client:', err);
        return;
    }

    // Call the Add method
    client.Add({ intA: 5, intB: 3 }, (err, result) => {
        if (err) {
            console.error('Error calling SOAP API:', err);
            return;
        }

        console.log('Result of Addition:', result.AddResult);
    });
});
```

GRAPH-QL APIs

- Gaining popularity due to their flexibility in allowing clients to specify exactly what data they need. It reduces the amount of data transferred and can improve performance.

- Especially useful in complex applications where multiple queries can be combined into one request, like social media platforms.

GRAPH-QL APIS

```
const url = 'https://api.spacex.land/graphql/';
const query = `
{
  launchesPast(limit: 2) {
    mission_name
    launch_date_local
    launch_site {
      site_name_long
    }
  }
}
`;

fetch(url, {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify({ query })
})
    .then(response => response.json())
    .then(data => {
        data.data.launchesPast.forEach(launch => {
            console.log('Mission:', launch.mission_name);
            console.log('Launch Date:', launch.launch_date_local);
            console.log('Launch Site:', launch.launch_site.site_name_long);
            console.log('---');
        });
    })
    .catch(error => {
        console.error('Failed to retrieve data:', error);
    });
```

WEBSOCKET APIs

- Essential for real-time communication needs. They Support full-duplex communication, allowing for live updates, such as chat apps, online gaming, and live financial data.

- WebSocket APIs enable applications to push data to clients instantly without requiring repeated requests.

WEBSOCKET APIs

```
// Create a WebSocket connection
const socket = new WebSocket('wss://echo.websocket.org');

// Connection opened
socket.addEventListener('open', () => {
    console.log('Connected to WebSocket server');

    // Send a message
    socket.send('Hello WebSocket!');
    console.log('Message sent!');
});

// Listen for messages
socket.addEventListener('message', event => {
    console.log('Message received:', event.data);
});

// Connection closed
socket.addEventListener('close', () => {
    console.log('WebSocket connection closed');
});

// Handle errors
socket.addEventListener('error', error => {
    console.error('WebSocket error:', error);
});
```