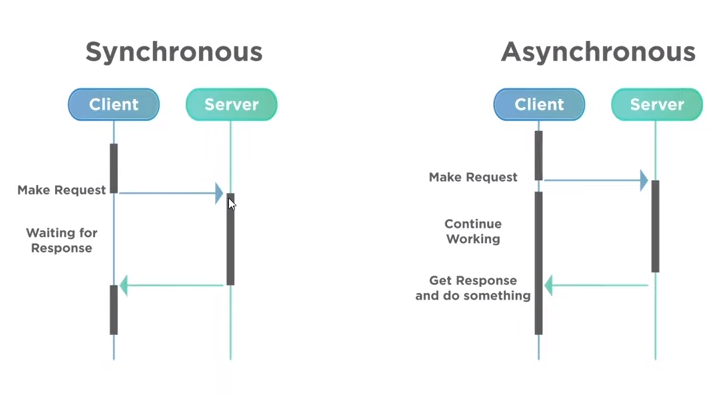
# Assincronicidade (JavaScript Assincrono)

Assíncrono é algo que não ocorre ou não se efetiva ao mesmo tempo

O JavaScript roda de maneira **síncrona**.



## O que são Promises e como manipulá-las

Objeto de processamento assíncrono.

Inicialmente o valor de uma promises é desconhecido. Ela pode então, ser **resolvida** ou **rejeitada**

Estados da promise:

1. Pending (Pendente)
2. Fulfilled (Completado)
3. Rejected (Rejeitado)

Exemplo:

Const myPromise = new Promise((resolve, reject) => {

window.setTimeout(() => {

resolve(console.log(‘Resolvida’));

}, 2000);

});

## Palavras-chave “async” e “await”

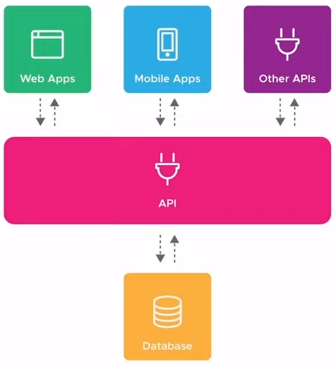
Funções assíncronas precisam destas duas palavras chaves.

## O que são API’s

Application Programming Interface

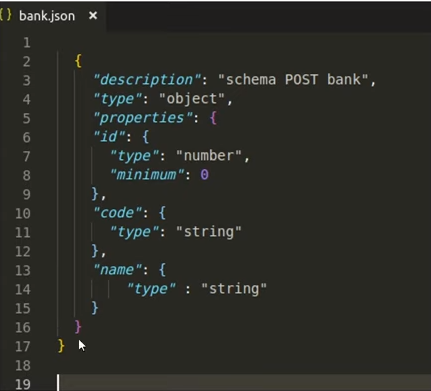
Uma API é uma forma de intermediar os resultados do back-end com o que é apresentado no front-end.

Você consegue acessá-la por meio de URL’s.



JSON – JavaScript Object Notation

É muito comum que API’s retornem seus resultados no formato .json, portanto precisamos tratar esses dados



## Apresentando o método fetch()

Consumindo API’s

fetch (url, options)

.then( response => response.json())

.then( json => console.log(json))

// retorna uma Promise

Quando for utilizar o fetch() temos que utilizar obrigatoriamente o aways

Operações no banco com fetch() (POST, GET, PUT, DELETE, etc)



O fetch() retorna uma promise