ORIGAMID

JAVASCRIPT ASSÍNCRONO

API e HTTP

API

- ## Application

  Um servidor, aplicativo, objeto JavaScript ou qualquer outra coisa que você interaja através de comandos. Ao digitar uma URL, estamos utilizando a API do browser para se comunicar com a API do servidor.

- ## Programming

  Programação, isso significa que um comando irá encadear uma cadeia de eventos pré-definidos. O resultado esperado é geralmente o mesmo.

- ## Interface

  A interface são os comandos criados para permitir a interação com a aplicação. Ex: `'VIOLAO'.toLowerCase()` é um método que faz parte da interface do objeto String. A interação com a interface retorna um efeito / resposta.

Exemplos de API's

- GitHub

  https://api.github.com/users/origamid

  https://api.github.com/users/origamid/followers

- Array / Element

  [].map();

  [].filter();

  Element.classList;

  Element.attributes;

- Tempo

  https://www.metaweather.com/api/location/455825/

  https://github.com/toddmotto/public-apis

HTTP

Hypertext Transfer Protocol é o protocolo utilizando para enviarmos/recebermos arquivos e dados na Web.

```
fetch('https://pokeapi.co/api/v2/pokemon/')
.then(r => r.json())
.then(pokemon => {
  console.log(pokemon);
});
```

url e method

Uma requisição HTTP é feita através de uma URL. O método padrão é o GET, mas existem outros como POST, UPDATE, DELETE, HEADER.

```
const url = 'https://jsonplaceholder.typicode.com/posts/';
const options = {
  method: 'POST',
  headers: {
    "Content-Type": "application/json; charset=utf-8",
  },
  body: '"aula": "JavaScript"';
}

fetch(url, options);
.then(response => response.json())
.then(json => {
  console.log(json);
});
```

method

- **GET**

  Puxa informação, utilizado para pegar posts, usuários e etc.

- **POST**

  Utilizado para criar posts, usuários e etc.

- **PUT**

  Geralmente utilizado para atualizar informações.

- **DELETE**

  Deleta uma informação.

- **HEAD**

  Puxa apenas os headers

GET

GET irá puxar as informações da URL. Não é necessário informar que o método é GET, pois este é o padrão.

```
const url = 'https://jsonplaceholder.typicode.com/posts/';
fetch(url, {
  method: 'GET'
})
.then(r => r.json())
.then(r => console.log(r))
```

POST

POST irá criar uma nova postagem, utilizando o tipo de conteúdo especificado no headers e utilizando o conteúdo do body.

```
const url = 'https://jsonplaceholder.typicode.com/posts/';

fetch(url, {
  method: 'POST',
  headers: {
    "Content-Type": "application/json; charset=utf-8",
  },
  body: '{"titulo": "JavaScript"}'
})
.then(r => r.json())
.then(r => console.log(r))
```

PUT

PUT irá atualizar o conteúdo do URL com o que for informado no conteúdo do body.

```
const url = 'https://jsonplaceholder.typicode.com/posts/1/';

fetch(url, {
  method: 'PUT',
  headers: {
    "Content-Type": "application/json; charset=utf-8",
  },
  body: '{"titulo": "JavaScript"}'
})
.then(r => r.json())
.then(r => console.log(r))
```

HEAD

HEAD puxa apenas os headers. É uma requisição mais leve pois não puxa o body.

```
const url = 'https://jsonplaceholder.typicode.com/posts/1/';

fetch(url, {
  method: 'HEAD',
})
.then(response => {
  response.headers.forEach(console.log);
  console.log(response.headers.get('Content-Type'));
});
```

Headers

- ### Cache-Control

  Tempo que o arquivo deve ficar em cache em segundos. Ex: public, max-age=3600

- ### Content-Type

  Tipo de conteúdo. Ex: text/html; charset=utf-8. Indicar o tipo de arquivo principalmente em métodos POST e PUT.

- ### Lista de Headers

  https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

CORS

Cross-Origin Resource Sharing, gerencia como deve ser o compartilhamento de recursos entre diferente origens.

É definido no servidor se é permitido ou não o acesso dos recursos através de scripts por outros sites. Utilizando o Access-Control-Allow-Origin.

Se o servidor não permitir o acesso, este será bloqueado. É possível passar por cima do bloqueio utilizando um proxy.

CORS é um acordo entre browser / servidor ou servidor / servidor. Ele serve para dar certa proteção ao browser, mas não é inviolável.

```
const url = 'https://cors-
anywhere.herokuapp.com/https://www.google.com/';
const div = document.createElement('div');

fetch(url)
.then(r => r.text())
.then(r => {
  div.innerHTML = r;
  console.log(div);
});
```