* **API**: um servidor que nos **serve dados**
* Para desenvolvedores web, é de suma importância entender como o **protocolo HTTP** funciona
* Ao fazer uma requisição a uma API, ela nos retorna um texto no formato **json** ou **xml**
* Funcionamento das **requisições**: é feita uma requisição à API pelo cliente, que busca a resposta no servidor e retorna para esse cliente
* Para montarmos uma requisição HTTP, temos::

- **URL (path)**: traz o **IP** e a **porta do servidor** junto com o **endereço do recurso** que desejamos. Composição da URL: **${Endereço}/${path = caminho de identificação do recurso}**

Exemplo de URL e breakdown:

URL: https://pokeapi.co/api/v2/pokemon

* IP: https://pokeapi.co **(os servidores de DNS convertem esse domínio para um IP ao fazer as requisições)**
* Path: /api/v2/pokemon **(solicitação do recurso pokemon da versão 2 dessa API)**
* Existem **vários tipos** de requisições. Ex.: quando digitamos uma URL no navegador e damos um enter, o tipo de requisição que é feita é o **GET** (Request Method: GET)
* Tipos de requisição:

- **GET**: o servidor entende que queremos buscar um elemento

- **POST** (caso o projeto tenha padrões REST): o servidor entende que queremos inserir um novo elemento

- **PUT**: o servidor entende que queremos atualizar um elemento

- **DELETE**: o servidor entende que queremos deletar um elemento

* Quando fazemos uma requisição GET em uma URL, o servidor entende que queremos buscar aquele recurso
* Toda requisição para um recurso, o método diz para o servidor o que queremos fazer sobre esse recurso (principalmente nas **APIs REST**)
* Para que façamos uma requisição e cumpramos uma determinada tarefa, em algum momento precisaremos passar dados para o servidor
* Existem algumas formas de transferir esses dados:

- através do path da URL: https://pokeapi.co/api/v2/pokemon**/${id}**

- através da **query string** (descrição de busca), bastante utilizada para **aplicar filtro/buscas** nas requisições (similar ao método WHERE do SELECT)

Sintaxe: https://pokeapi.co/api/v2/pokemon**?type=grass**

* No exemplo acima, type (chave) e grass (valor)
* Obs.: é tudo que vem depois do ponto de interrogação no fim do path e é uma estrutura de **chave e valor**
* Caso desejemos adicionar outro argumento a esse filtro, utilizamos o **&** para concatenar

Exemplo: https://pokeapi.co/api/v2/pokemon?type=grass**&name=i**

* Nos **Headers** da API, além do **General** onde vemos a URL que foi feita a requisição, o método e o status da requisição, também temos o **Response Headers** e o **Request Headers** (são áreas de dados que podemos transmitir e receber do servidor)
* Esses **Headers** normalmente são **metadados** da nossa requisição: descrevem ou complementam a nossa requisição
* Os **Headers** são as configurações da nossa requisição, então ao cria-los, nós definimos dados que fazem sentido para configurar a requisição

- **Request Headers**: feita pelo cliente

- **Response Headers**: feita pelo servidor

* **Response Headers e o Request Headers**: são a configuração da API
* Método do Responde Headers (breakdown):

accept-language: en-US,en;q=0.9,pt-PT;q=0.8,pt;q=0.7

1. en-US
2. en;q=0.9
3. pt-PT;q=0.8
4. pt;q=0.7

* Onde: en-US é prioridade máxima e as demais linguagens são definidas como prioridade secundárias a partir do **q=0.número**
* Normalmente, as **credenciais de autenticação** são passadas dentro do Headers, no método **Authorization**
* Dependendo do tipo de requisição, temos o **BODY** (corpo da requisição; dado que queremos trafegar nela)

Exemplo: o método GET não tem body na requisição, pois ele precisa somente do path, a query string, o método e os headers de request para funcionar. Porém, os métodos POST, PUT e PATCH podem ter um body em suas requisições

* Modelo de requisição POST com body:

Request Headers

content-type: application/json

Body

{

“name”: “algo”

}

* O que vai no body como requisição depende do **content-type**
* O body pode existir tanto na requisição quanto na resposta. **Detalhe:** na resposta, o GET tem body
* O body geralmente é um arquivo **JSON** e podemos acessá-lo no navegador pressionando as teclas **Ctrl + Shift + I** e indo na aba **Response**

Requisição

* **URL**: https://pokeapi.co/api/v2/pokemon
* **Request Method**: GET | POST | PUT | DELETE
* **Request Headers**

content-type: application/json

* **Request Body**

{

“name”: “teste”

}

Response

* **STATUS CODE** (número que indica o que aconteceu com nossa requisição: se foi ou não processada)
* Família 200 (200 – 299): sucesso
* Família 300 (300 – 399): redirect
* Família 400 (400 – 499): erro por parte do cliente
* Família 500 (500 – 599): erro interno do servidor
* **Response Headers**
* **Response Body**