Universidade Federal do Ceará
Centro de Ciências/Departamento de Computação
Código da Disciplina: CK0236
Professor: Ismayle de Sousa Santos

Aula 15

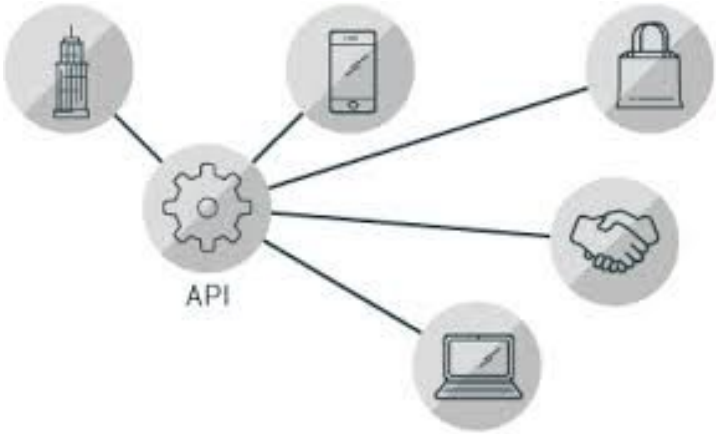
Técnica de Programação II

Testes de API

qpg4p5x ismaylesantos@great.ufc.br @IsmayleSantos

O que é uma API?

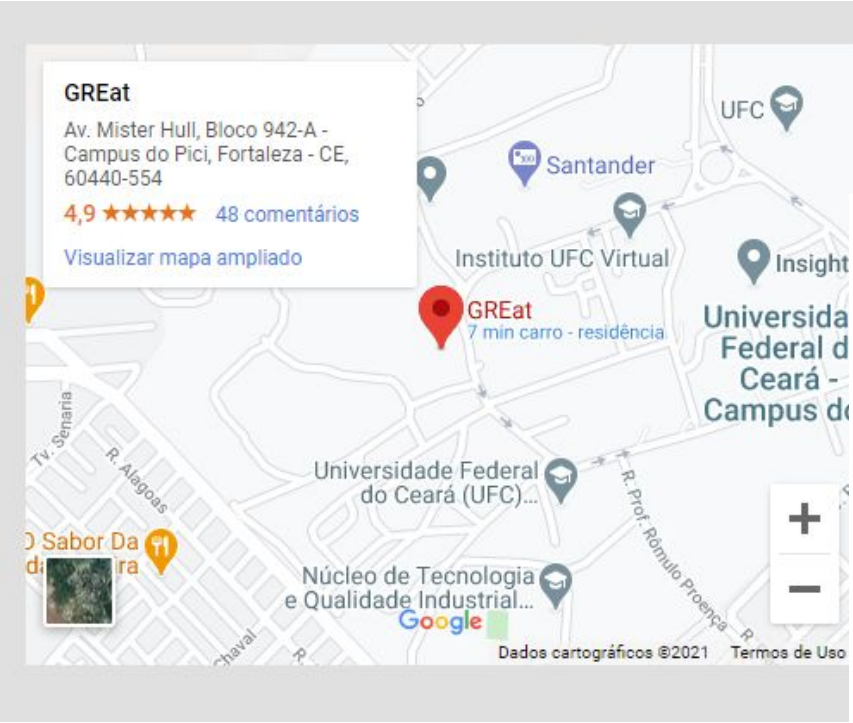
- Interface de Programação de Aplicações
  - API (**A**pplication **P**rogramming **I**nterface)
  - É um conjunto de assinaturas que são disponibilizadas para os usuários de uma biblioteca/framework desenvolverem os aplicativos deles [Swebok 2014]
- Uma API permite a interoperabilidade entre sistemas



Fonte: https://www.redhat.com/pt-br/topics/api

O que é uma API?

- Exemplo
  - API do Google Maps
    - No site do GReat é utilizada a API do Google Maps

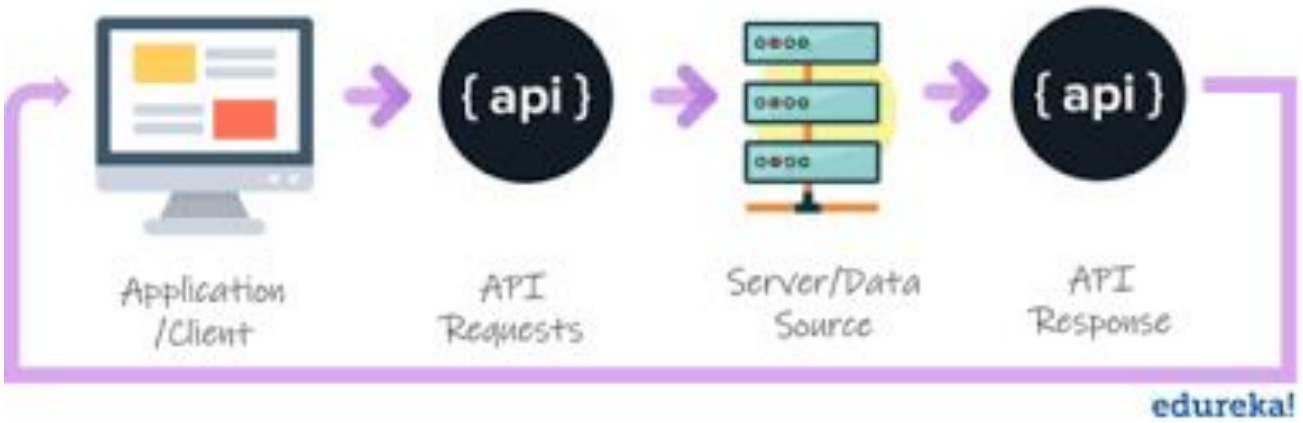


Testes de Interface

- Teste de interface
  - Objetiva verificar se interfaces de componentes fornecem a troca correta de dados e informações de controle [Swebok]
  - Usualmente casos de testes são gerados a partir da especificação de interfaces
  - Simula o uso de APIs por aplicações do usuário final
    - Envolve a geração de parâmetros para as chamadas de API, configuração de parâmetros ambientais e a definição de dados internos que afetam a API

Testes de Interface

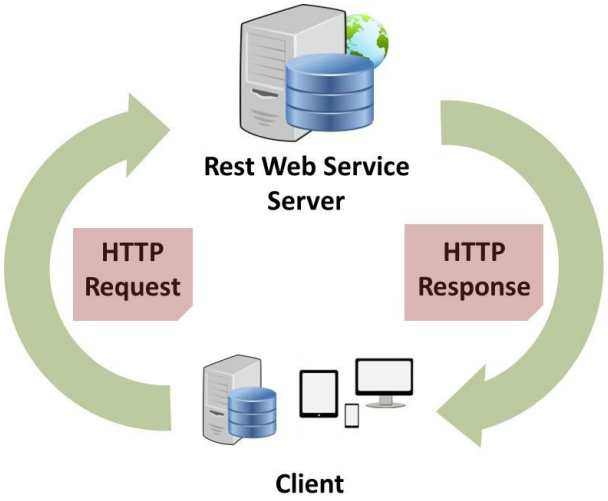
- No teste de API, você vai simular uma requisição e verificar a resposta



Fonte: https://www.edureka.co/

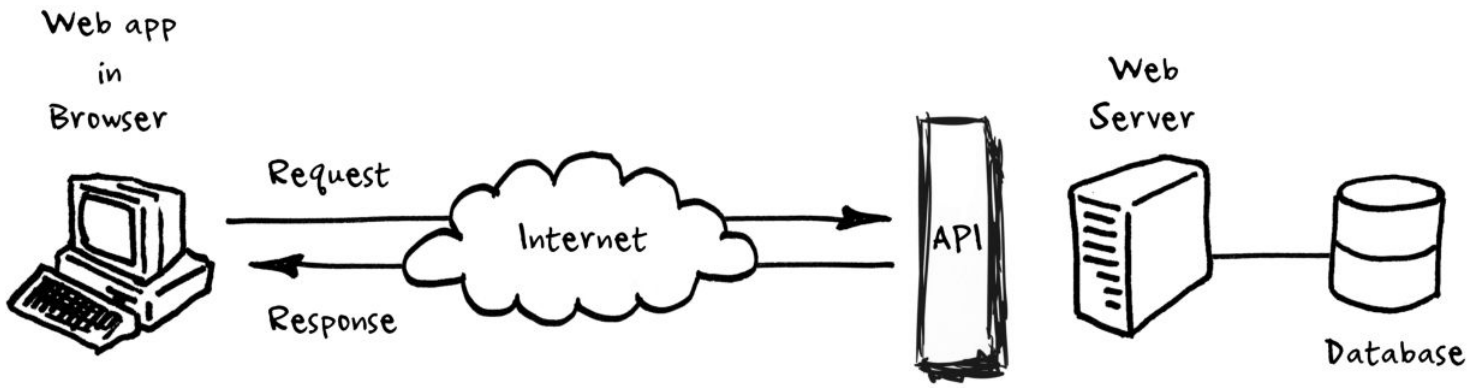
Rest

- Padrão REST (Representational State Transfer)
  - Transferência de Estado Representacional
  - Representa uma série de princípios (arquiteturais) visando a padronização de rotas, requisições e comunicações sem estado
- Restful é o termo atribuído a uma API que implementa o padrão REST



Requisição

- Requisição HTTP
  - Mensagens enviadas pelo cliente com alguns parâmetros
  - A resposta contém alguns dados e status



HTTP é um protocolo cliente-servidor

Fonte: https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f

Formato de uma requisição HTTP

- Exemplo de Requisição
  - Na resposta temos o cabeçalho e o corpo

Formato de uma requisição HTTP

- Métodos em REST
  - GET
    - Buscar
  - POST
    - Registrar
  - PUT
    - Alterar
  - DELETE
    - Remover

Task	Method	Path
Create a new customer	POST	/customers
Delete an existing customer	DELETE	/customers/{id}
Get a specific customer	GET	/customers/{id}
Search for customers	GET	/customers
Update an existing customer	PUT	/customers/{id}

Formato de uma requisição HTTP

- URI (Uniform Resource Identifier)
  - Indica o endereço e recurso a ser utilizado
  - Exemplo de api fictícia
    - Endereço: http://api.lojinha.com
    - Recurso: login
    - URI: http://api.lojinha.com/login

Formato de uma requisição HTTP

- Cabeçalho
  - Suporte do padrão REST
    - XML
    - JSON
    - HTML
  - Como vamos enviar JSON
    - Content-Type
      - application/json

JSON

- **Jav**a**S**cript **O**bject **N**otation
  - https://www.json.org/
  - É uma forma de representação dos dados baseado na sintaxe de object JavaScript

```
"restaurant": {
 "name": "Fish Witch",
 "address": "214 NE Broadway",
 "zipcode": "97232",
 "phone": "503-000-0000",
 "website": "http://fishwitch.com",
 "email": "hellofishy@fishwitch.com"
}
```

Fonte: https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f

Formato de uma requisição HTTP

- Corpo
  - Envia os dados via JSON
  - Supondo Atributos
    - Usuario
    - Senha

```
{
    “usuarioLogin”: klebinho
    “senhaSenha”: 123456
}
```

Formato de uma requisição HTTP

Envio da Requisição

- Ao receber a requisição, a API REST da Lojinha
  - extrai os dados do corpo da requisição
  - pesquisa no banco por um usuário klebinho com senha 123456
  - devolve um Cabeçalho e um Corpo
- No cabeçalho
  - Retorna o código de resposta (Status Code)
- No corpo
  - Poderia vir o Token do login do Kleber

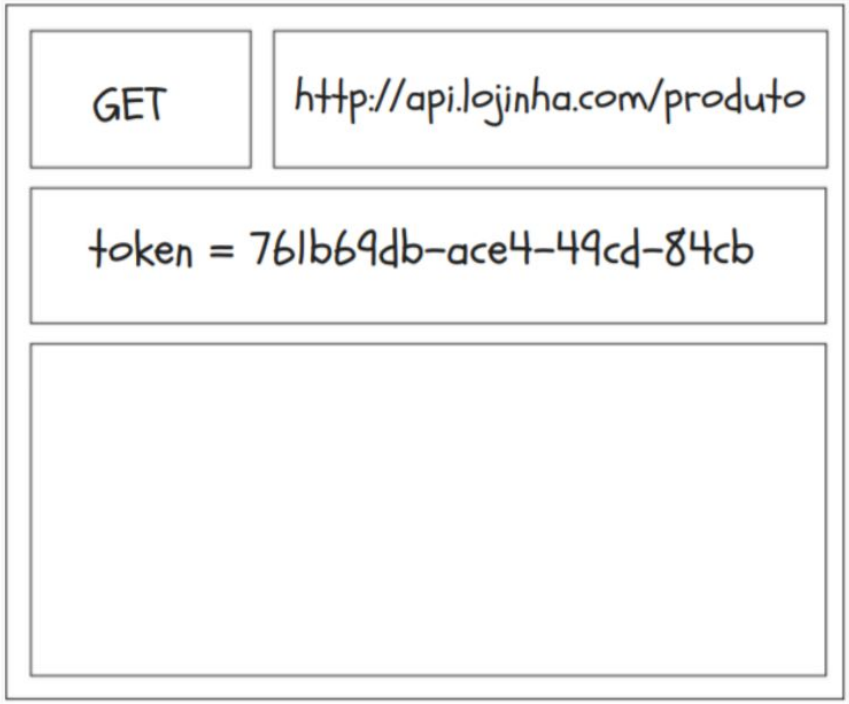
    {

    “token”: b467200db234509zo98

    }

Envio da Requisição

- E se agora quisermos pesquisar aquele usuário?
  - Basta fazer uma requisição GET

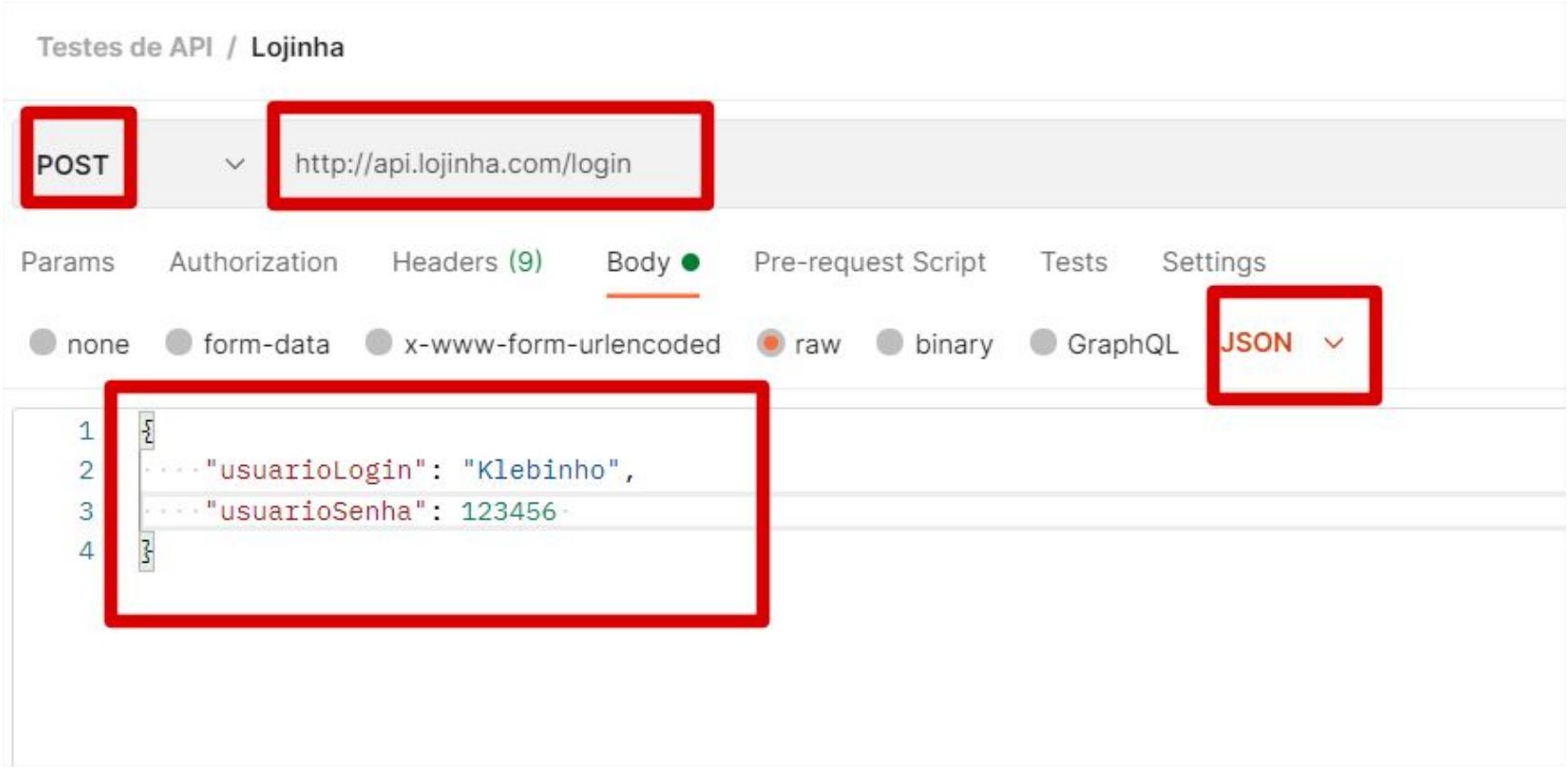


Ferramentas de Teste de API

- Ferramentas
  - Postman
  - Advanced REST Client
  - Soap UI

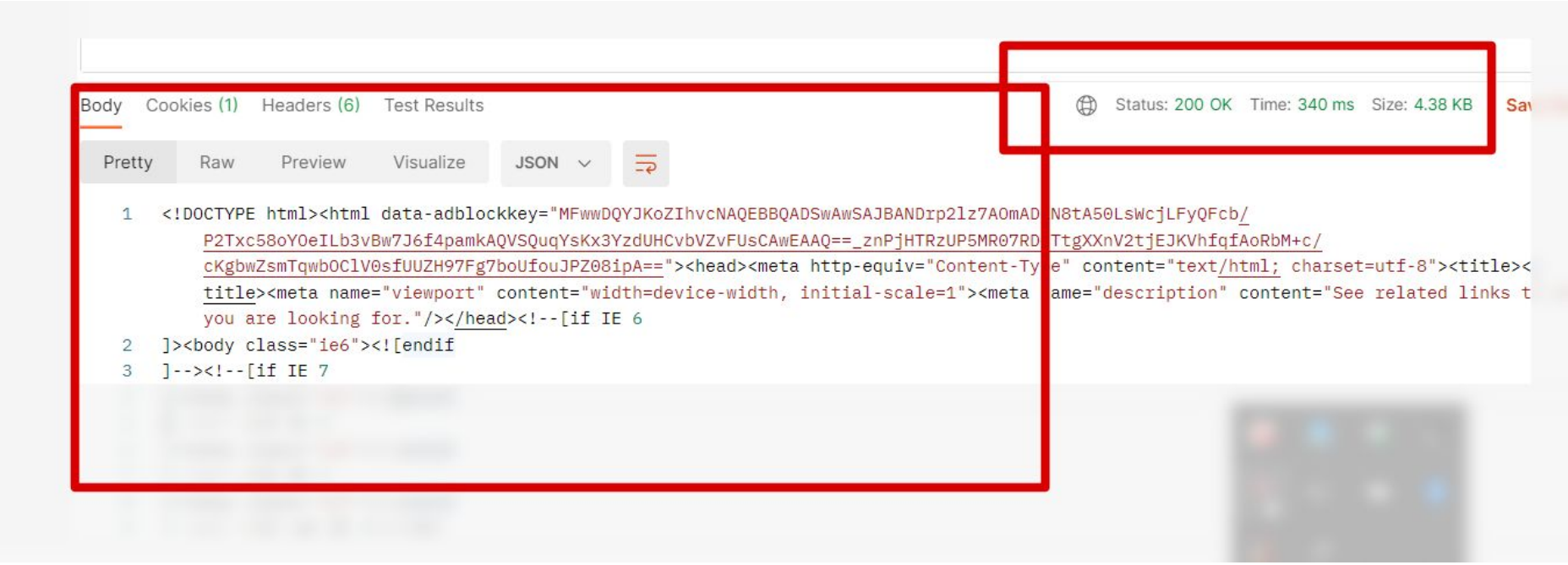
Postman

- Exemplo de Envio - API fictícia



Postman

- Exemplo de Resposta - API fictícia

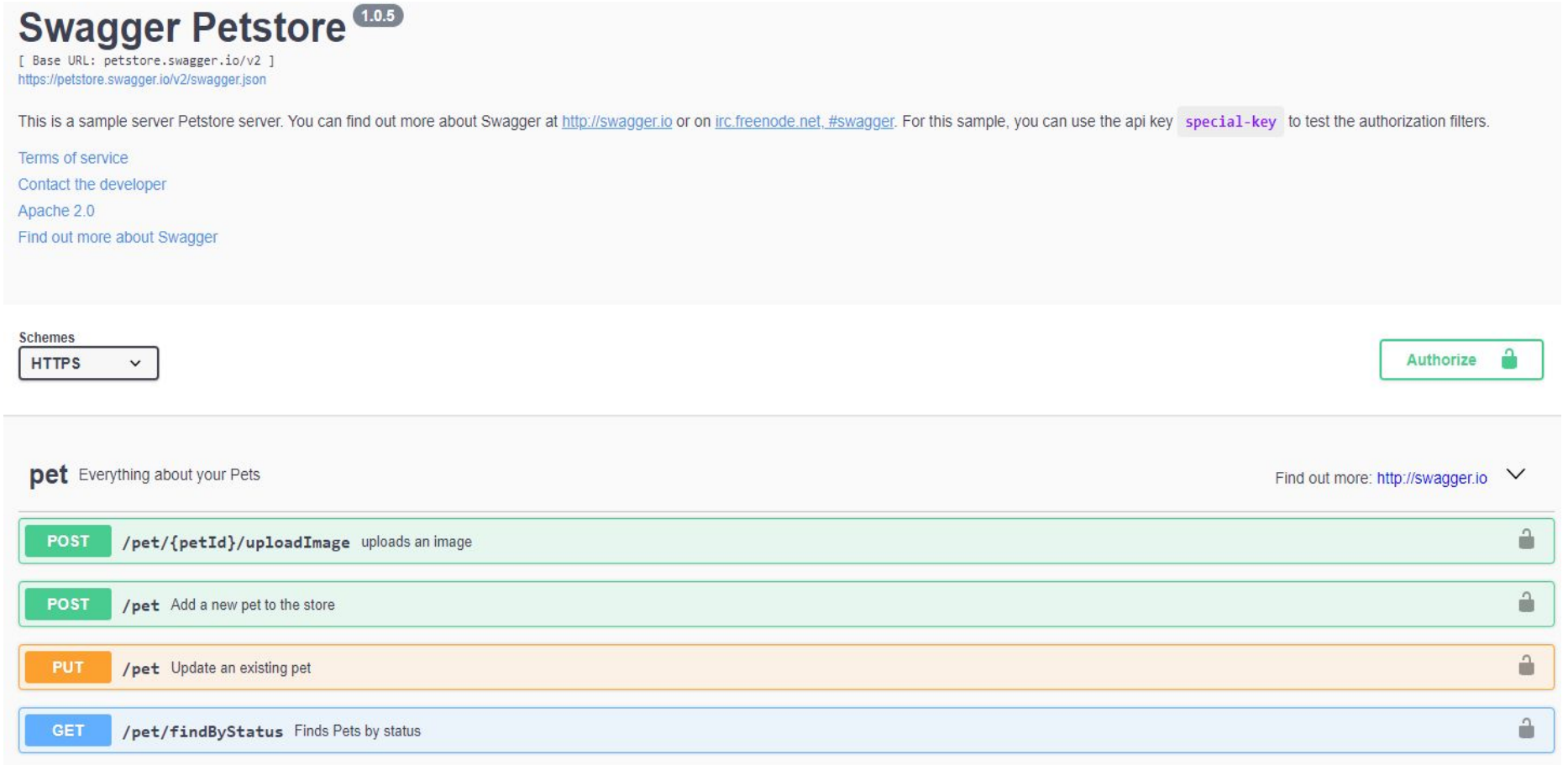


Swagger

- Modelo de documentação de APIs
  - Linguagem de Descrição de Interfaces para APIs RESTful
- Feito pelo desenvolvedor
- Mantém a lista das URIs, métodos, corpos de requisição e respostas

Exemplo - Swagger

- https://petstore.swagger.io/



Encontrando Falhas

- Você pode criar diferentes requisições para expor o sistema a situações que podem resultar em falhas
- Alguns pontos a serem testados
  - As regras de negócio foram implementadas corretamente?
  - Cada método faz o que deveria?
  - A estrutura de resposta segue o Swagger?
  - Os códigos de estados estão corretamente configurados?
  - O tempo de resposta está satisfatório?

Encontrando Falhas

- Exemplos de Testes

  {

  usuarioLogin: Klebinho ;

  usuárioSenha: 123456

  }

  {

  usuarioLogin: ,

  usuárioSenha: 123456

  }

Encontrando Falhas

- Exemplos de Testes

```
{
        usuarioLogin:  true,
        usuárioSenha: 123456
}

{
        usuarioLogin:  klebinho,
        usuárioSenha: @!
 }
```

Encontrando Falhas

- Exemplos de Testes - GET
  - Suponha que a API espere algo no formato
    - http://api.lojinha.com/usuarios?UserId=2
  - Exemplos de Testes
    - http://api.lojinha.com/usuarios?UserId=x
    - http://api.lojinha.com/usuarios?UserId=
    - http://api.lojinha.com/usuarios?UserId=8985612
      - *(id inexistente)*
    - http://api.lojinha.com/usuarios?UserId=""

Códigos de Retorno

- Existem Vários códigos
  - https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status
- Classes
  - Respostas de Informação (100–199)
  - Respostas de Sucesso (200–299)
  - Redirecionamentos (300–399)
  - Erros do cliente (400–499)
  - Erros do servidor (500–599)
- Exemplo

`201 Created`

A requisição foi bem sucedida e um novo recurso foi criado como resultado. Esta é uma tipica resposta enviada após uma requisição POST.

Códigos de Retorno

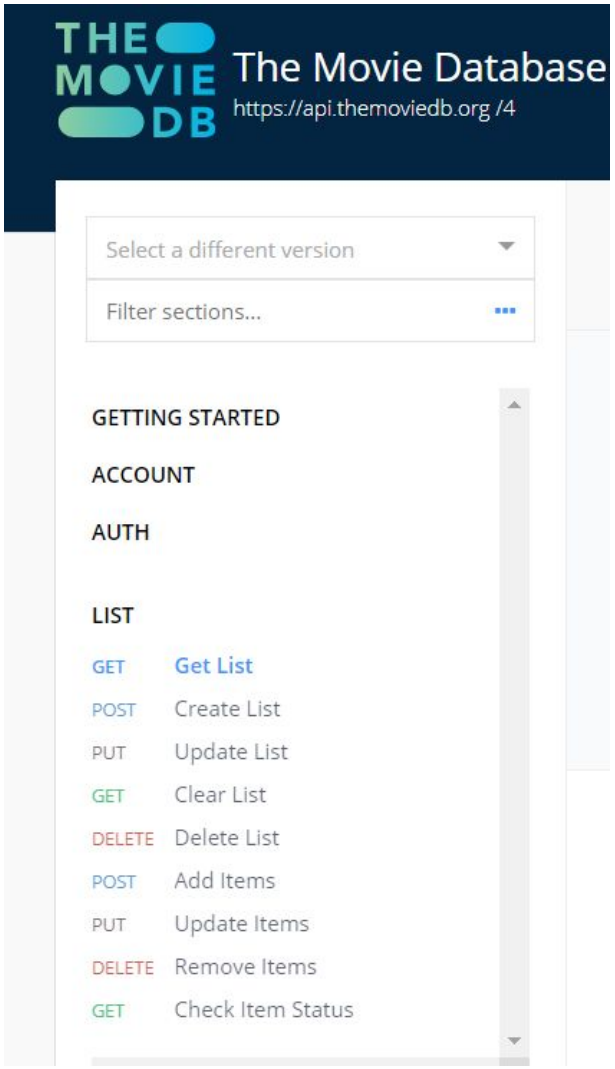
- Observação
  - Métodos GET retornam o código 200 mesmo que nenhum recurso tenha sido encontrado
    - O corpo da resposta é um array vazio
  - Métodos DELETE e PUT retornam o código 200 se forem processadas com sucesso

Exemplo Real

- The Movie DB
  - É um banco de dados de filmes e TV
    - Atores, ano de lançamento etc
    - www.themoviedb.org
  - API pública
    - precisa de cadastro para utilizar
  - Documentação
    - https://developers.themoviedb.org/3/getting-started/introduction

Exemplo Real

- The Movie DB
  - Documentação da API
    - https://developers.themoviedb.org/3/getting-started/introduction



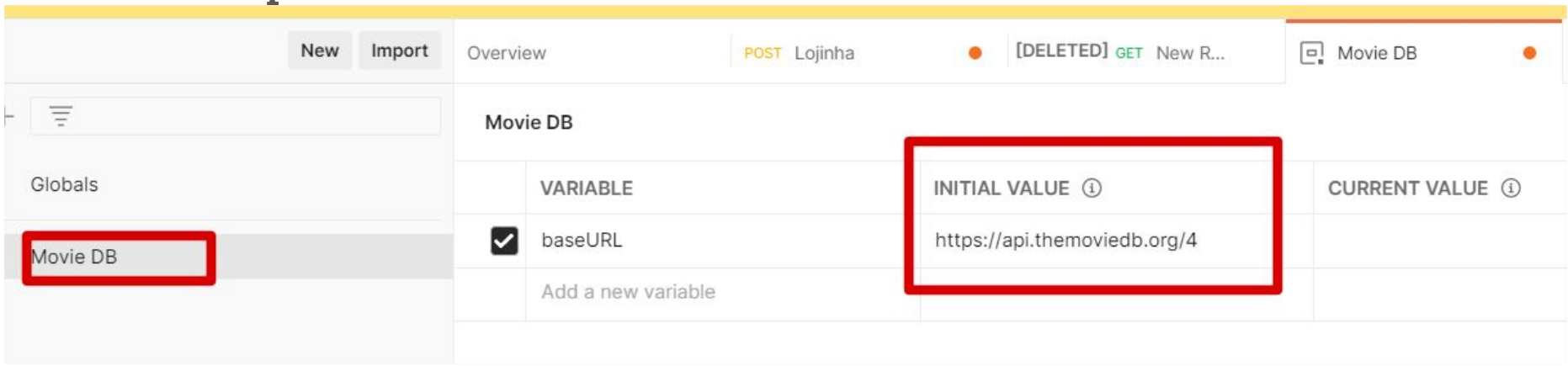
Nos exemplos vamos usar métodos da lista AUTH e LIST

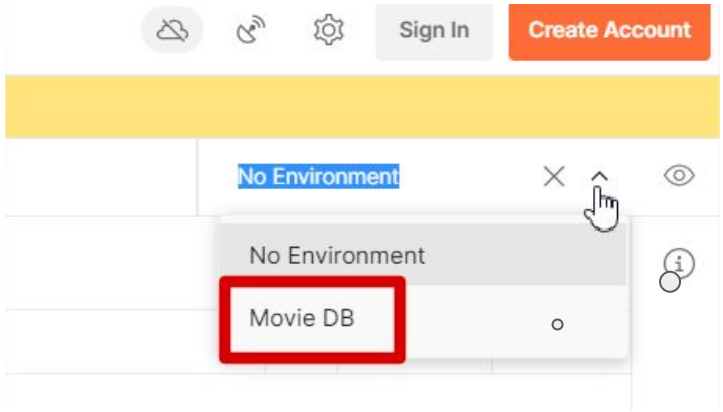
Exemplo Real

- No postman
  - Workspace
    - Serve para organizar os projetos, agrupar as Coleções
      - ** precisa criar conta no Postman*
  - Ambiente
    - Podemos criar variáveis para o ambiente
  - Coleções
    - Conjunto de Requisições
  - Pasta

Exemplo Real

- Exemplo Ambiente

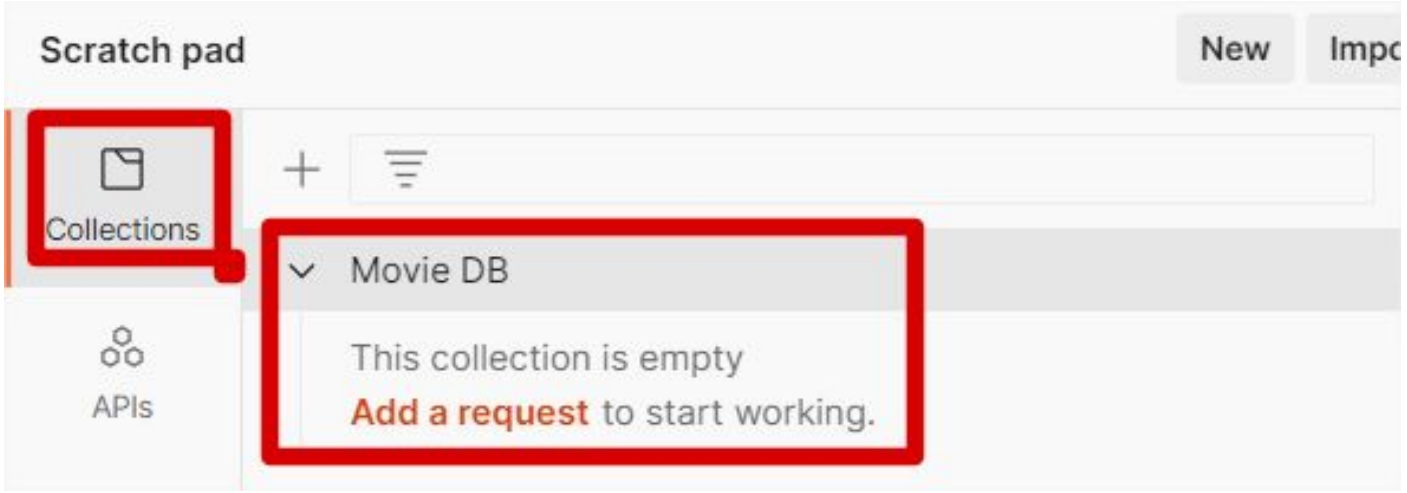




Depois tem que alterar o ambiente utilizado

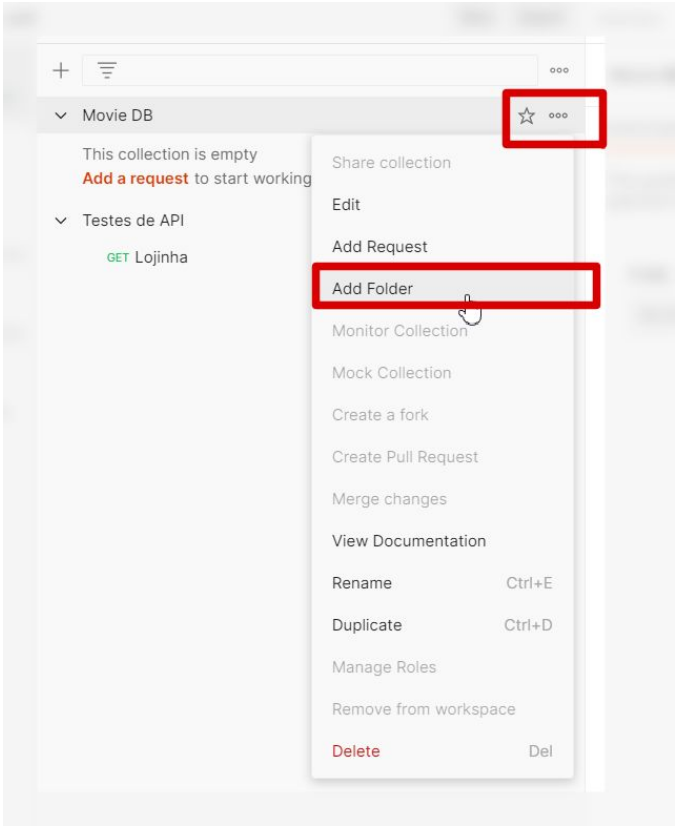
Exemplo Real

- Exemplo Coleção



Exemplo Real

- Exemplo de Pastas



Exemplo Real

- Como funciona a autorização no Movie DB
  a. [Postman] Gerar Token de Requisição
     - Cria Request-Token utilizando Token de leitura
  b. [Site] Autorizar Request-Token
     - *https://www.themoviedb.org/auth/access?request_token={request_token}*
     - *Autoriza Request-Token*
  c. [Postman] Gerar Token de Acesso
     - Cria Access_Token (que pode ser usado nas requisições)

Isso é uma particularidade desta API aberta

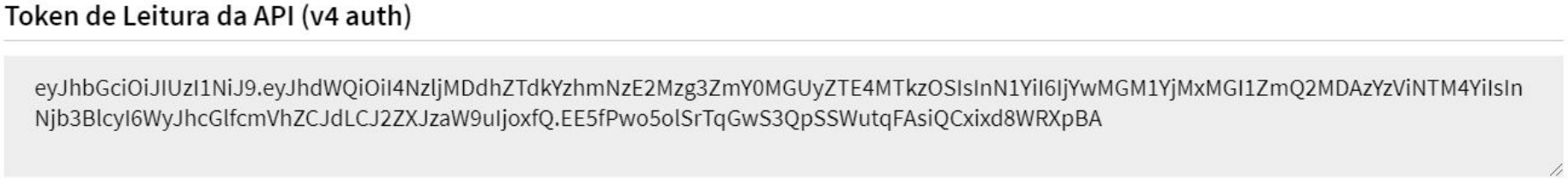
https://developers.themoviedb.org/4/auth/user-authorization-1

Exemplo Real

- Passo 1: Criando Request_Token
  a. Efetuar login no sistema
     - https://www.themoviedb.org/
  b. No menu de perfil clicar em configurações
     - https://www.themoviedb.org/settings/api
  c. No submenu API
     - Solicitar chave

Exemplo Real

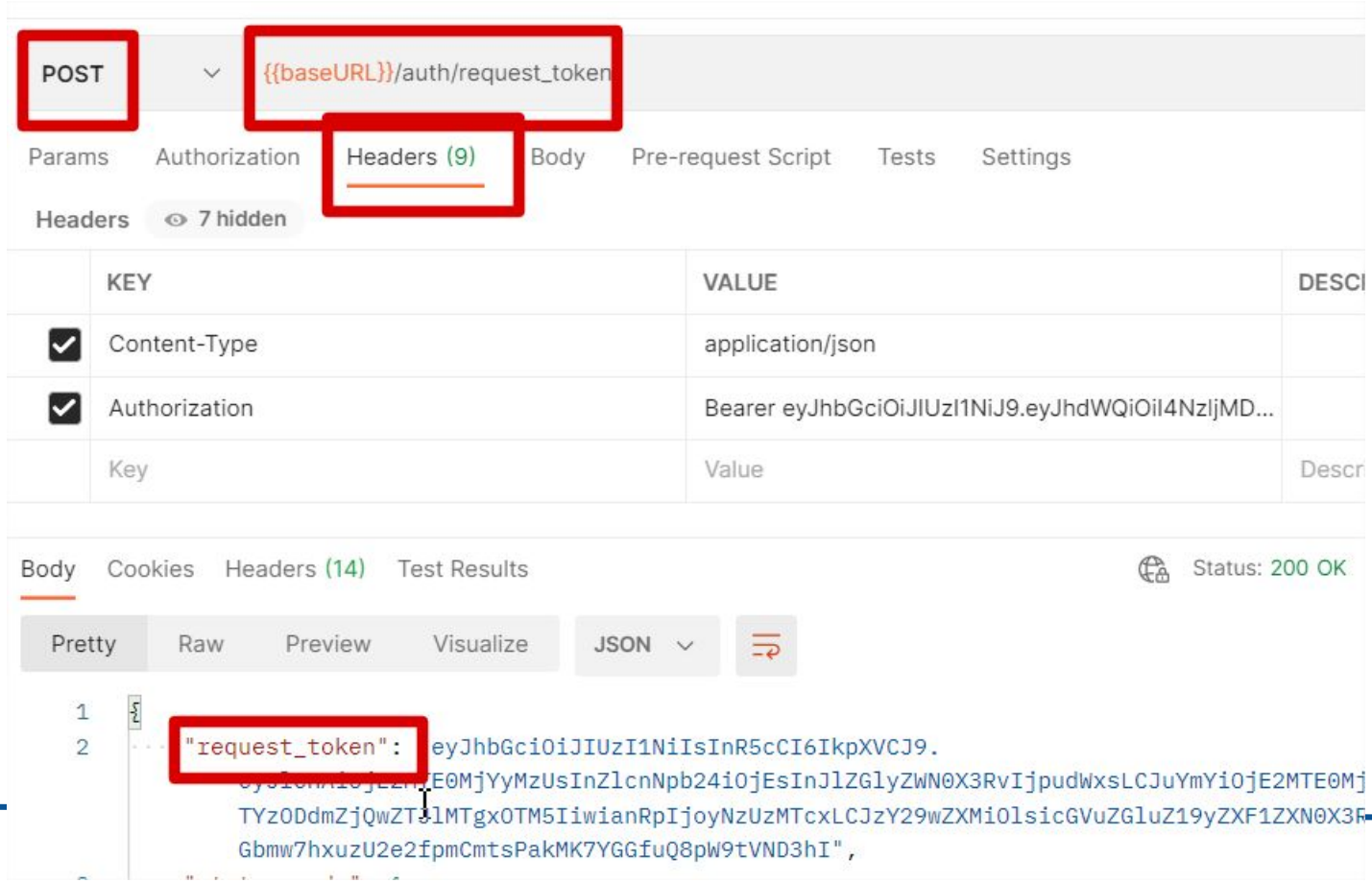
- Passo 1: Criando Request_Token
  a. Depois de preencher os dados solicitados copie o Token de Leitura da API



  b. Vamos mandar uma requisição POST
    - https://developers.themoviedb.org/4/auth/create-request-token

Exemplo Real

- Passo 1: Criando Request_Token
  a. Ao enviar a requisição vamos receber o request_token

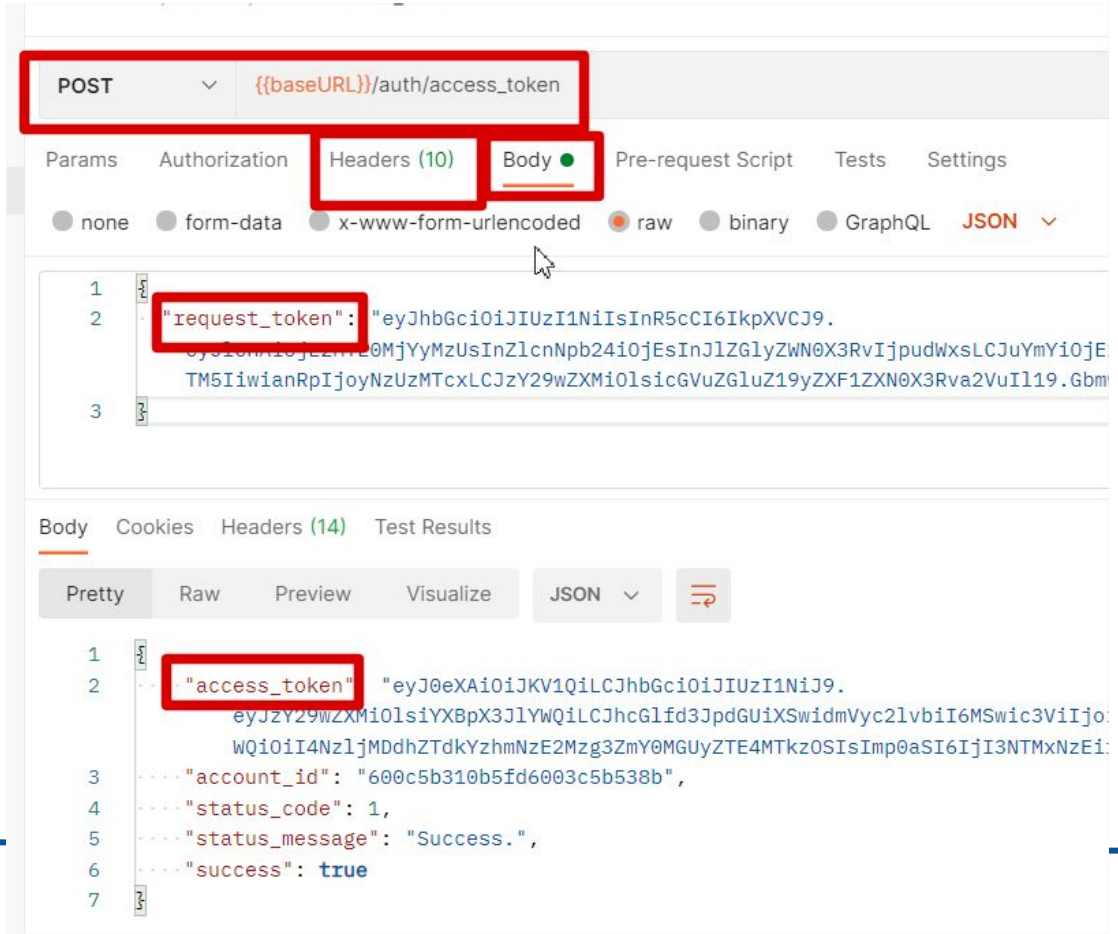


Exemplo Real

- Passo 2: agora vamos autorizar o request_token
  - *https://www.themoviedb.org/auth/access?request_token={request_token}*

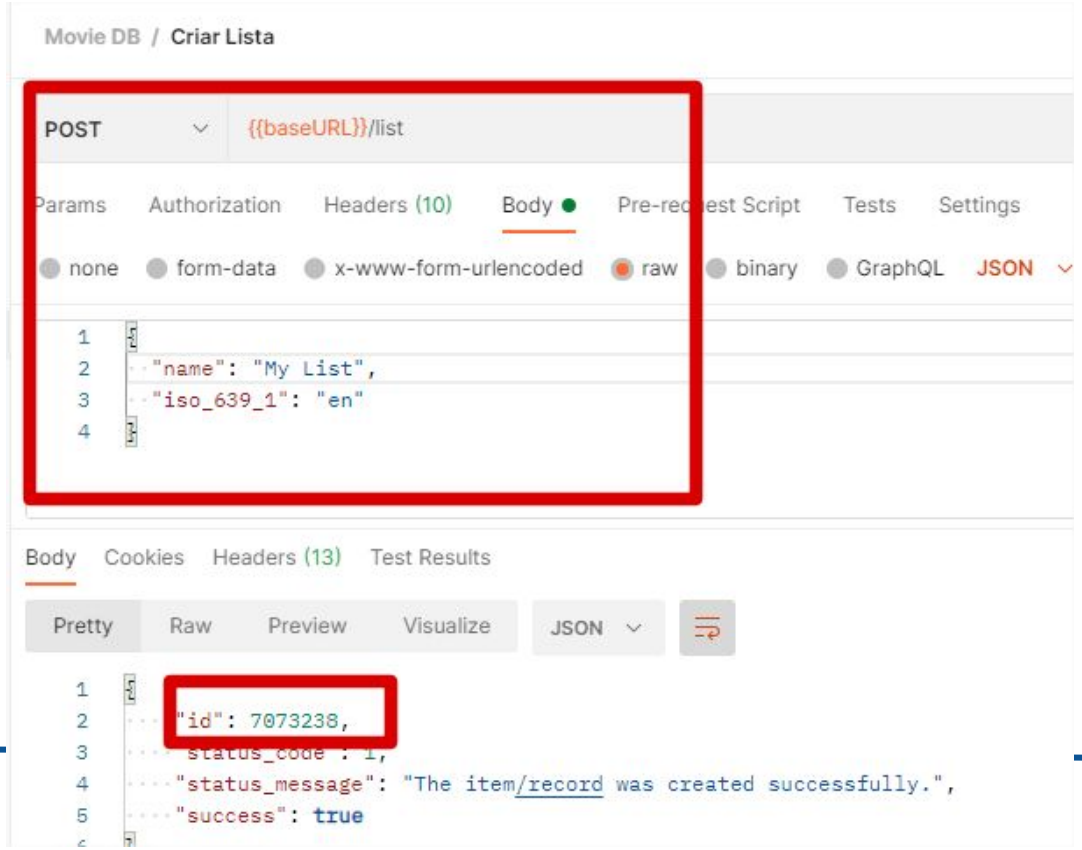
Exemplo Real

- Passo 3: Gerando Access_Token
  - https://developers.themoviedb.org/4/auth/create-access-token



Exemplo Real

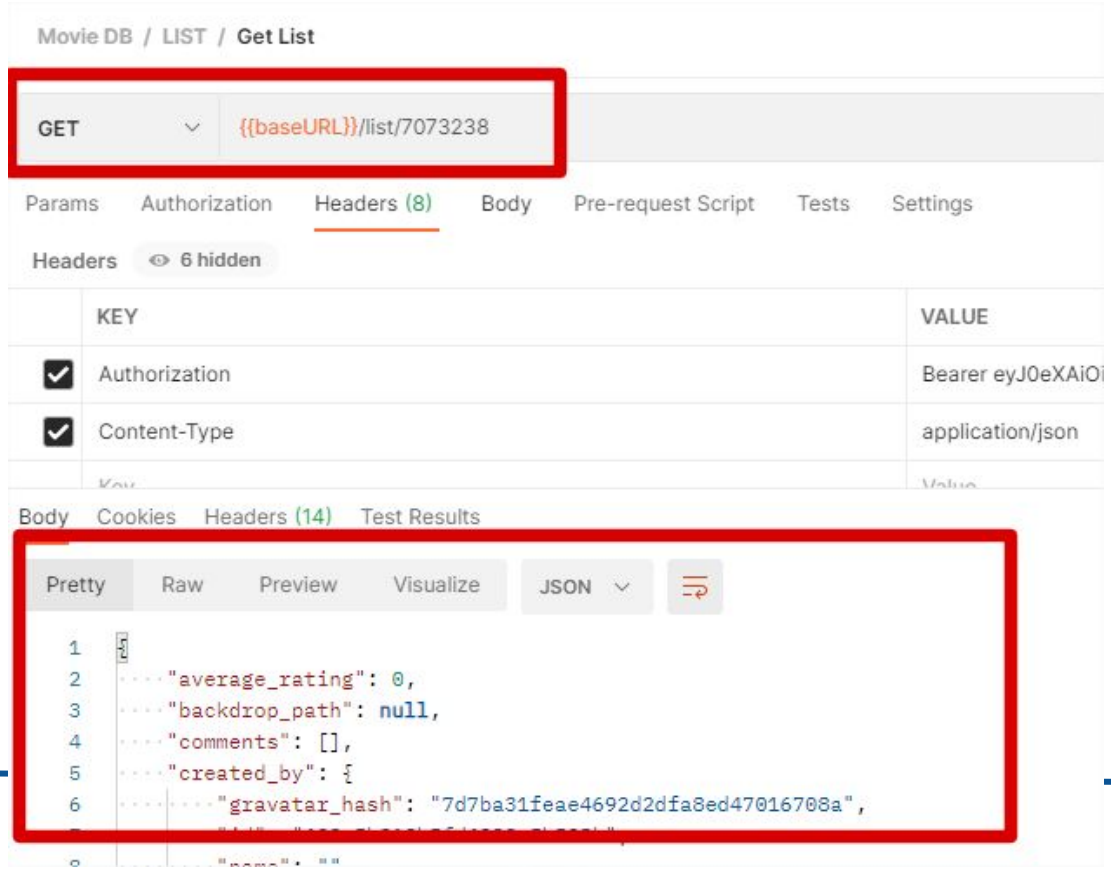
- Fazendo outras requisições
  - Criar Lista
    - https://developers.themoviedb.org/4/list/create-list



Idioma e Nome da Lista

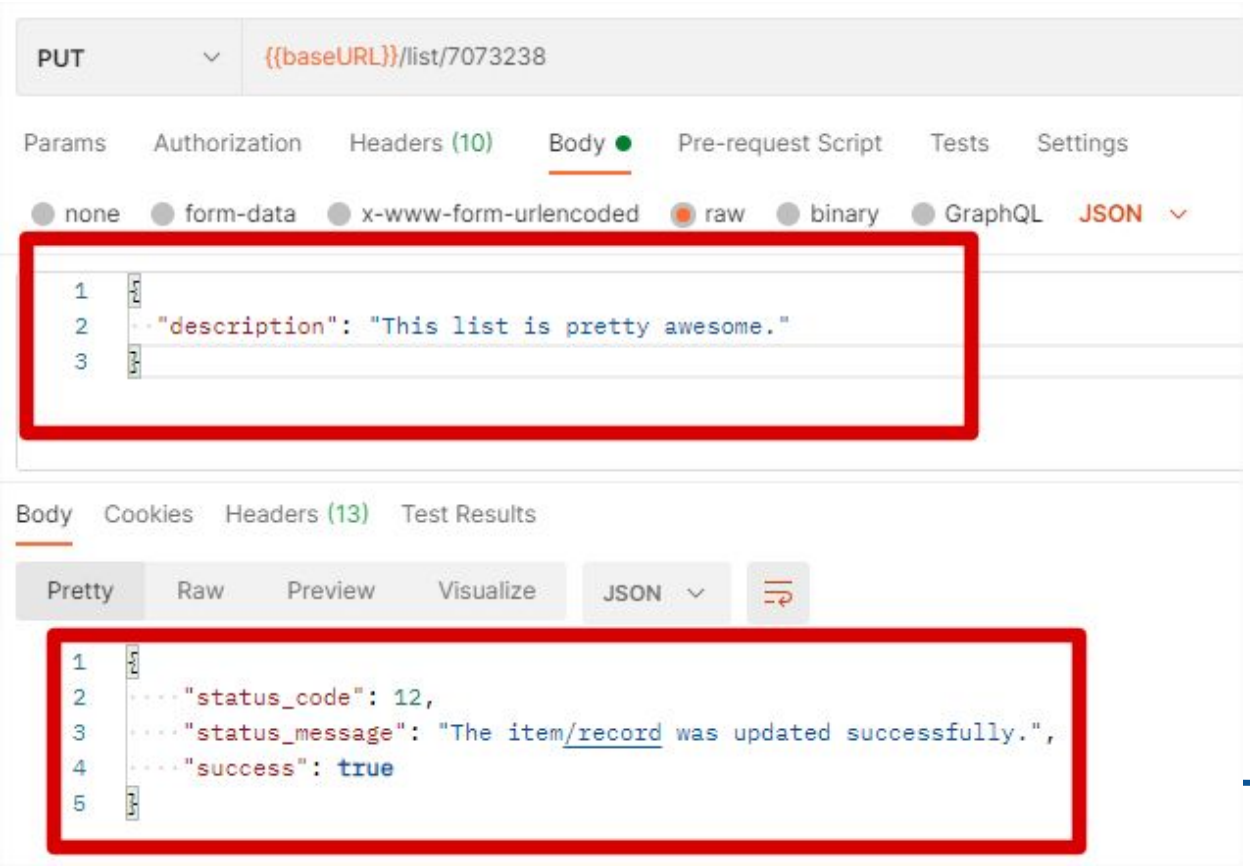
Exemplo Real

- Fazendo outras requisições
  - Get Lista
    - https://developers.themoviedb.org/4/list/get-list



Exemplo Real

- Fazendo outras requisições
  - Atualizar Lista
    - https://developers.themoviedb.org/4/list/update-list



Outras formas de automatizar testes de API

REST-assured

Trabalho Prático TP-7

- Criar Testes Automatizados de API
  - Pode ser no projeto final ou no movie DB
  - Se for no movie DB
    - Add Items
      - https://developers.themoviedb.org/4/list/add-items
    - Update Items
      - https://developers.themoviedb.org/4/list/update-items
    - Remove Items
      - https://developers.themoviedb.org/4/list/remove-items

Obrigado!

Por hoje é só pessoal...

Dúvidas?

qpg4p5x

ismaylesantos@great.ufc.br

@IsmayleSantos