

# API REST PLANETS

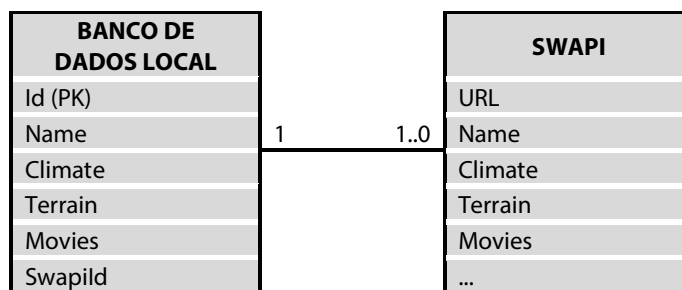
Nelson Frick, maio de 2019

## 1 Estrutura de Dados

Segundo a especificação do desafio:

- Para cada planeta, os seguintes dados devem ser obtidos do banco de dados da aplicação, sendo inserido manualmente:
  - Nome
  - Clima
  - Terreno
- Para cada planeta também devemos ter a quantidade de aparições em filmes, que podem ser obtidas pela API pública do Star Wars: <https://swapi.co/>

Me pareceu estranho permitir cadastramento de clima e terreno livremente mas buscar as aparições em filmes no swapi. Inicialmente, imaginei assim a tabela Planetas, permitindo cadastrar, além dos planetas Star Wars, outros planetas.

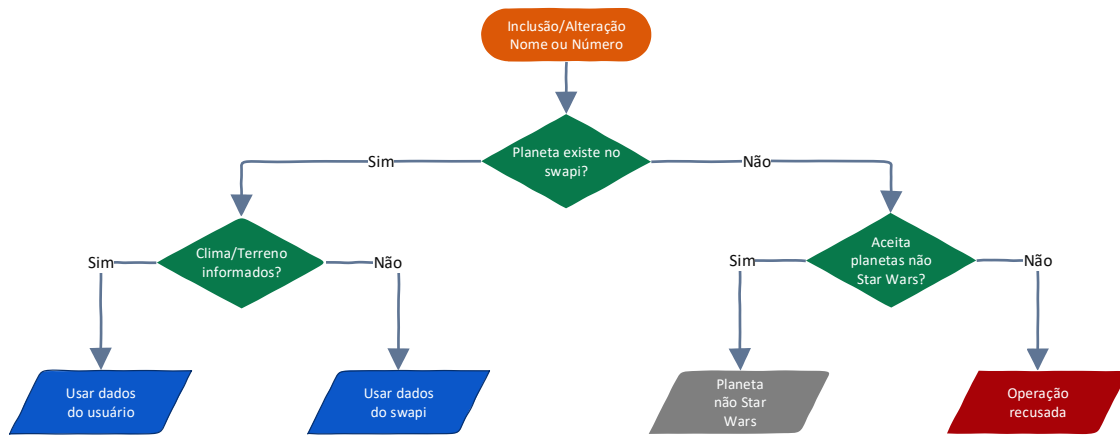


- Usuário pode cadastrar planetas fornecendo um nome ou um número, clima (opcional) e terreno (opcional).
- Se for informado um nome, o planeta é pesquisado por nome no swapi através de um ou mais GETs. O swapi responde a um GET com um objeto contendo uma “página” com 10 planetas e URLs para as páginas anterior e seguinte. Se o planeta não estiver na página, lê-se a página seguinte. Encontrado o planeta, obtém-se a quantidade de filmes e a busca é encerrada. Se, após a última página, o planeta não for encontrado, ele será considerado “não Star Wars”.
- Se for informado um número, o planeta é pesquisado no swapi diretamente usando <https://swapi.co/planets/número>. Se o planeta for encontrado (número entre 1 e 61), obtém-se a quantidade de filmes. Caso contrário, o planeta será “não Star Wars” e o número será o nome do planeta. O cadastramento por número agiliza muito a entrada de dados.
- Se o planeta for encontrado no swapi e clima e/ou terreno estiverem em branco, clima e/ou terreno do swapi serão usados. Se o clima e/ou terreno forem fornecidos pelo usuário, terão precedência. Em outras palavras, o usuário pode fazer *override* dos dados do swapi e fazer reset, bastando executar um PUT com clima e/ou terreno em branco.
- Se o planeta não for encontrado no swapi, a quantidade de filmes é zero.

Em um segundo momento, resolvi deixar opcional o cadastramento de planetas não Star Wars. Para isso, bastou alterar o método que valida o planeta, executado antes de adicionar ou alterar planetas. Dependendo das linhas de código habilitadas, a API aceitará ou não planetas não Star Wars:

```
public bool IsPlanetInvalid(PlanetM planet) {
    // Para permitir apenas planetas Star Wars, comentar as linhas abaixo
    /*return string.IsNullOrEmpty(planet.Name) || // Nome em branco
       _planets.Find(p => p.Id != planet.Id && // Não existir outro planeta com mesmo nome
          p.Name.ToLower() == planet.Name.ToLower()).Any(); */

    // Para permitir qualquer planeta, comentar as linhas abaixo
    return string.IsNullOrEmpty(planet.Name) || // Nome em branco
       !planet.IsStarWars || // Planeta não é Star Wars
       _planets.Find(p => p.Id != planet.Id && // Não existir outro planeta com mesmo nome
          p.Name.ToLower() == planet.Name.ToLower()).Any();
}
```



Há um problema ao se permitir planetas não Star Wars: com a liberdade de alterações (nome, clima, terreno), é possível, a partir de um planeta Star Wars, criar um planeta completamente diferente, editando nome, clima e terreno.

## 2 Desenvolvimento

Para desenvolvimento, normalmente escolho as melhores ferramentas disponíveis, sem preferências pessoais. Para este projeto, optei por C# por não haver tempo hábil para me familiarizar com as ferramentas e bibliotecas Java para APIs REST. Para facilitar os testes, iniciei o projeto usando um banco de dados “in memory”. Após os testes iniciais, não foi difícil portar a API para MongoDB. A diferença está no Id: enquanto o banco “in memory” usa *auto-increment integers* (1, 2, 3...), o MongoDB usa chaves *string* com 24 caracteres, por exemplo 5ceacdec22326e179430ca13.

Para não haver redundância de código, criei a *abstract class* PlanetBase, da qual derivam Planet e PlanetM, cada uma com sua definição de Id.

Além do *driver* para MongoDB, a única biblioteca externa utilizada foi a Newtonsoft.Json, que simplificou muito a leitura do *json* retornado pelo swapi. Para testes, usei Postman e uma página HTML.

Apesar do código ser bastante simples, inclui comentários em todas as situações.

```

if (IsPlanetInvalid(planet)) {
    return false;
}
  
```

Valida o planeta Nome não pode estar em branco e não pode existir outro planeta com mesmo nome Opcional: invalidar se não for Star Wars

Finalmente, optei por deixar as duas APIs funcionando: **api/planet** para o banco “in memory” e **api/planetm** para o MongoDB.

## 3 Testes da API PlanetM (para banco de dados MongoDB)

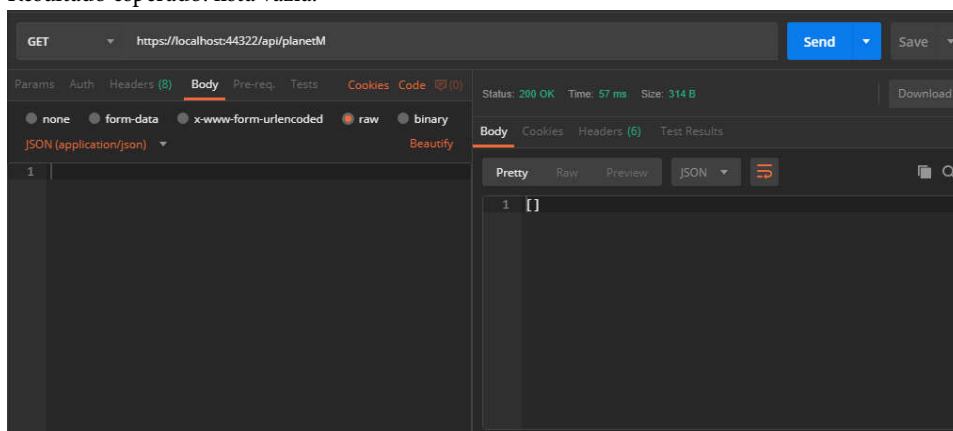
Os mesmos testes foram realizados para a API Planet (“banco de dados in memory”). Como não é o objetivo do desafio, estes testes foram omitidos desta documentação.

### 3.1 Testes com Postman

#### 3.1.1 GET: com banco de dados vazio

Cenário: nenhum planeta ainda cadastrado, tenta-se listar planetas.

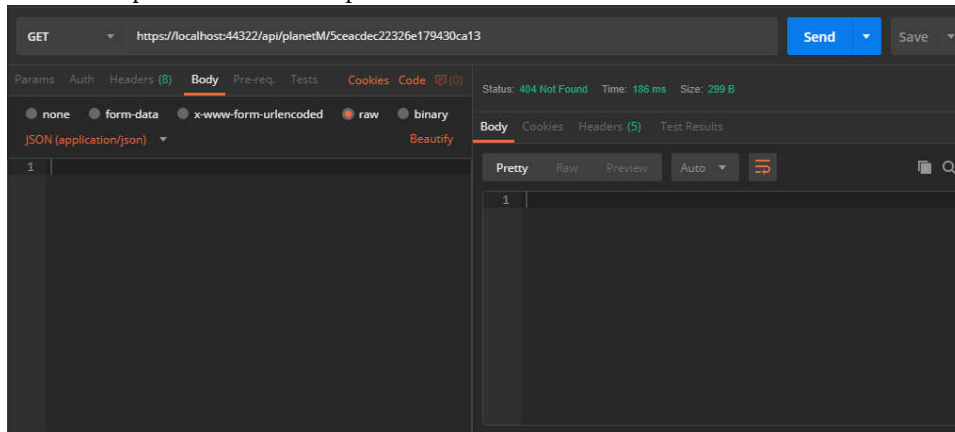
Resultado esperado: lista vazia.



### 3.1.2 GET: um planeta, com banco de dados vazio

Cenário: nenhum planeta ainda cadastrado, tenta-se obter dados de um planeta.

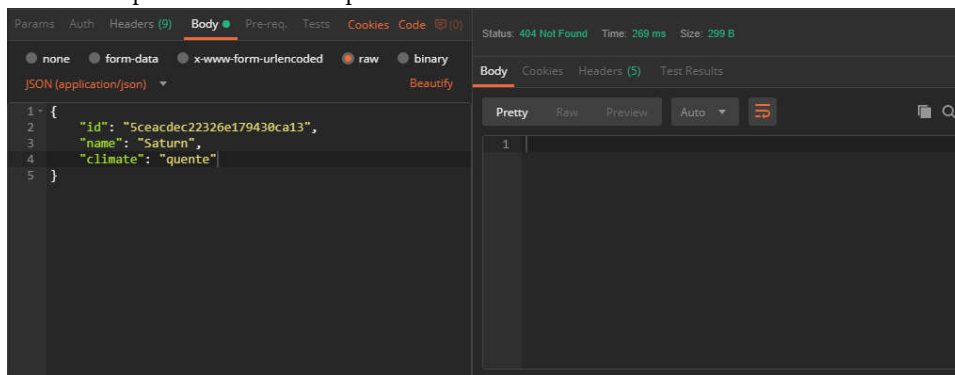
Resultado esperado: nenhuma resposta.



### 3.1.3 PUT: com banco de dados vazio

Cenário: nenhum planeta ainda cadastrado, tenta-se alterar dados de um planeta.

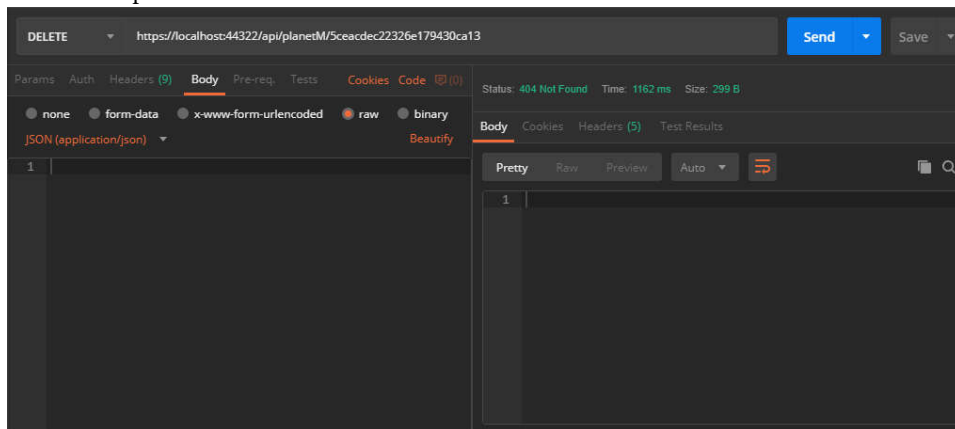
Resultado esperado: nenhuma resposta.



### 3.1.4 DELETE: com banco de dados vazio

Cenário: nenhum planeta ainda cadastrado, tenta-se deletar planeta.

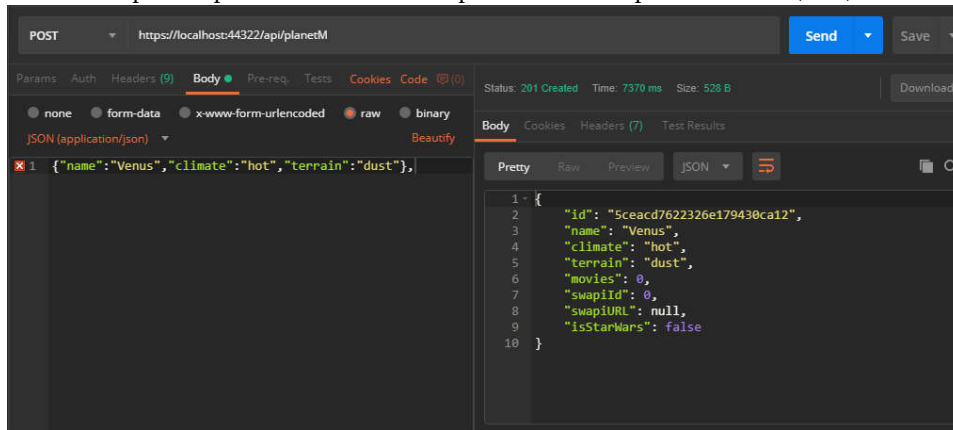
Resultado esperado: nenhum resultado



### 3.1.5 POST: Adicionar planeta não Star Wars (Star Wars-only mode OFF)

Cenário: tenta-se cadastrar planeta não Star Wars com sistema aceitando planetas não Star Wars.

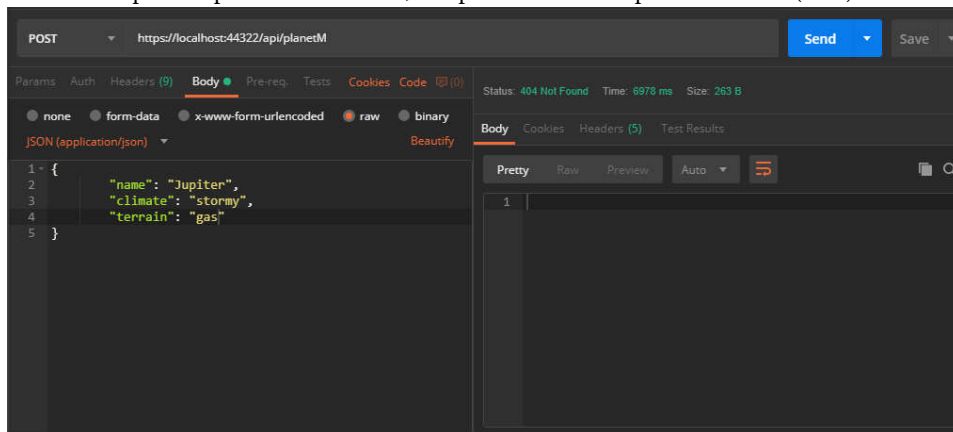
Resultado esperado: planeta adicionado, campos Movies e SwapId contendo 0 (zero).



### 3.1.6 POST: Adicionar planeta não Star Wars (Star Wars-only mode ON)

Cenário: tenta-se cadastrar planeta não Star Wars com sistema rejeitando planetas não Star Wars.

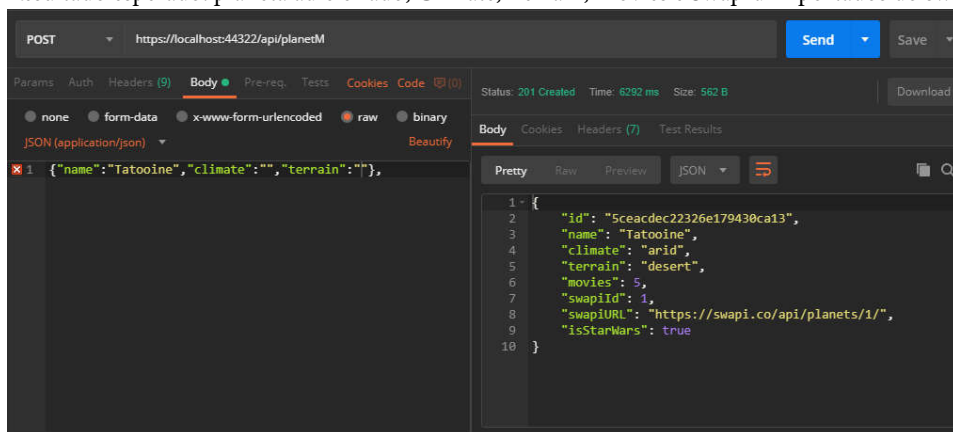
Resultado esperado: planeta adicionado, campos Movies e SwapId contendo 0 (zero).



### 3.1.7 POST: Adicionar planeta Star Wars por nome, sem clima e terreno

Cenário: tenta-se cadastrar planeta Star Wars sem informar Climate e Terrain.

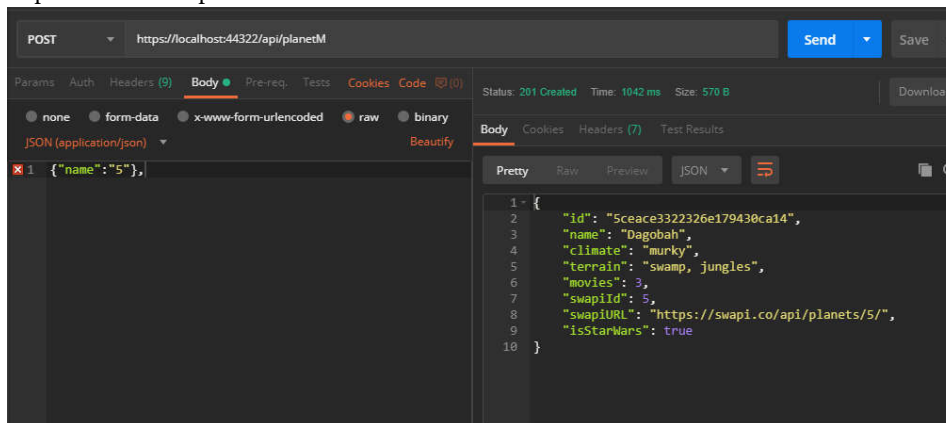
Resultado esperado: planeta adicionado, Climate, Terrain, Movies e SwapId importados do swapi.



### 3.1.8 POST: Adicionar planeta Star Wars por número, sem clima e terreno

Cenário: tenta-se cadastrar planeta Star Wars informando apenas o número (entre 1 e 61).

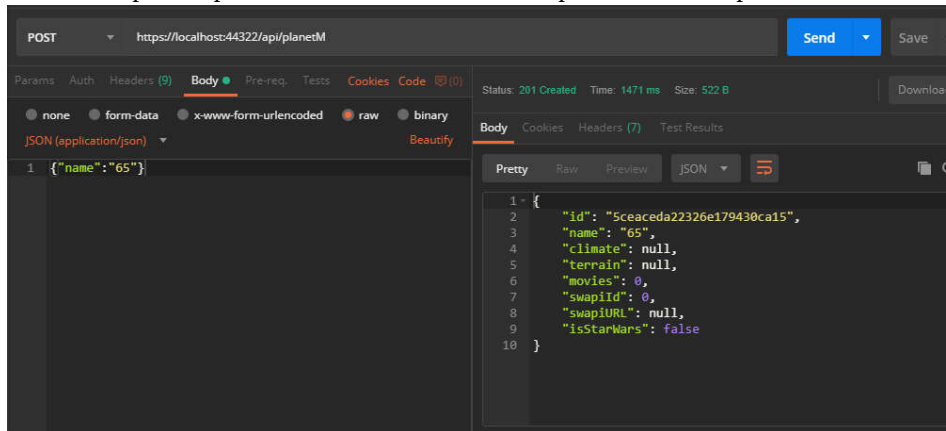
Resultado esperado: planeta adicionado, SwapId contendo Id do planeta no swapi, Name, Climate, Terrain e Movies importados do swapi.



### 3.1.9 POST: Adicionar planeta Star Wars por número inexistente, sem clima e terreno

Cenário: tenta-se cadastrar planeta Star Wars informando um número maior que 61.

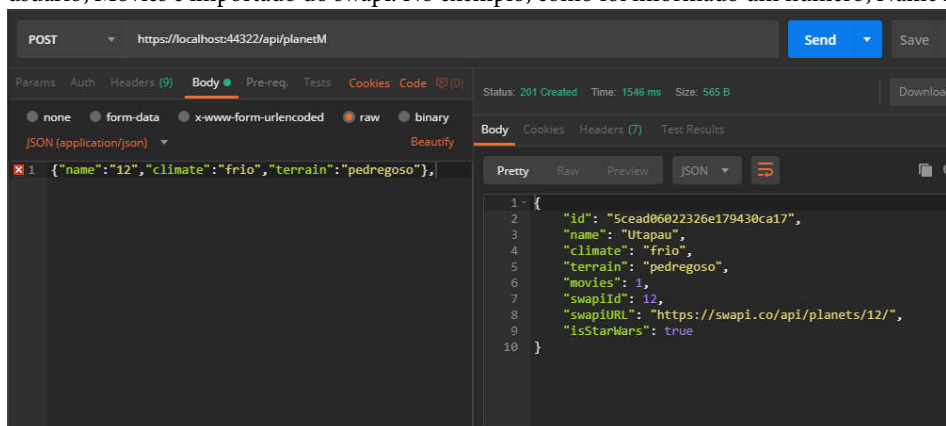
Resultado esperado: planeta adicionado, Como não há planeta 65 no swapi, Name recebe "65".



### 3.1.10 POST: Adicionar planeta Star Wars por número, sobrescrevendo clima e terreno

Cenário: tenta-se cadastrar planeta Star Wars informando nome ou número, clima e terreno.

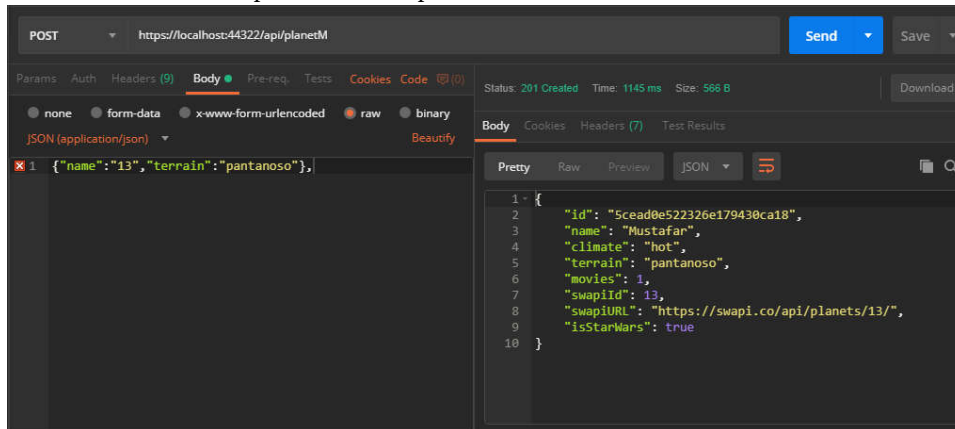
Resultado esperado: planeta adicionado, SwapId contendo Id do planeta no swapi, Climate, Terrain são os fornecidos pelo usuário, Movies é importado do swapi. No exemplo, como foi informado um número, Name foi importado do swapi.



### 3.1.11 POST: Adicionar planeta Star Wars por nome ou número, sobrescrevendo terreno

Cenário: tenta-se cadastrar planeta Star Wars informando nome ou número e terreno.

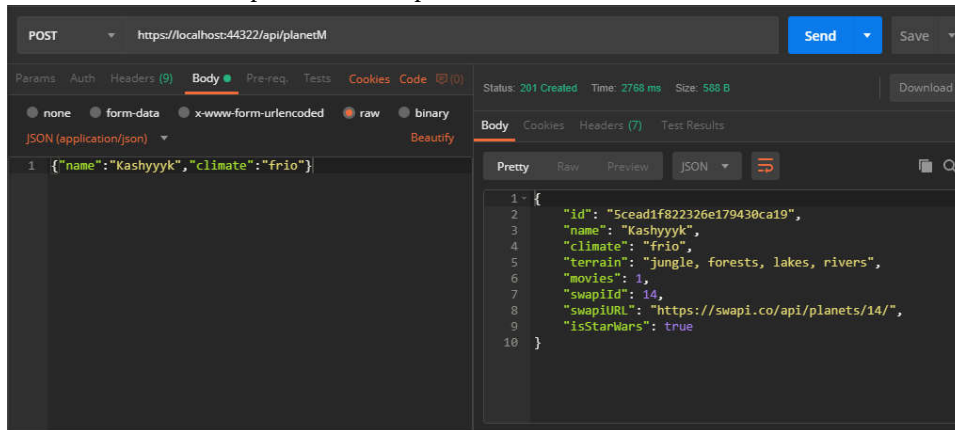
Resultado esperado: planeta adicionado, SwapId contendo Id do planeta no swapi, Terrain é o fornecido pelo usuário, Climate e Movies são importados do swapi.



### 3.1.12 POST: Adicionar planeta Star Wars por nome ou número, sobrescrevendo clima

Cenário: tenta-se cadastrar planeta Star Wars informando nome ou número e clima.

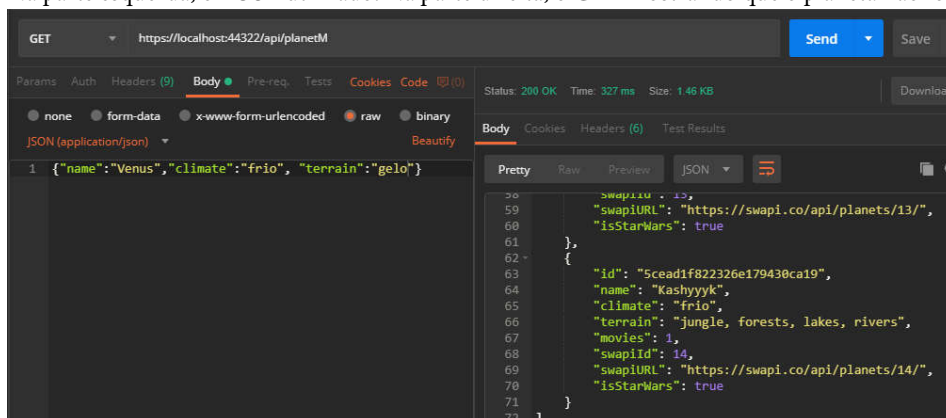
Resultado esperado: planeta adicionado, SwapId contendo Id do planeta no swapi, Climate é o fornecido pelo usuário, Terrain e Movies são importados do swapi.



### 3.1.13 POST: Adicionar planeta não Star Wars já existente

Cenário: tenta-se cadastrar planeta não Star Wars já cadastrado.

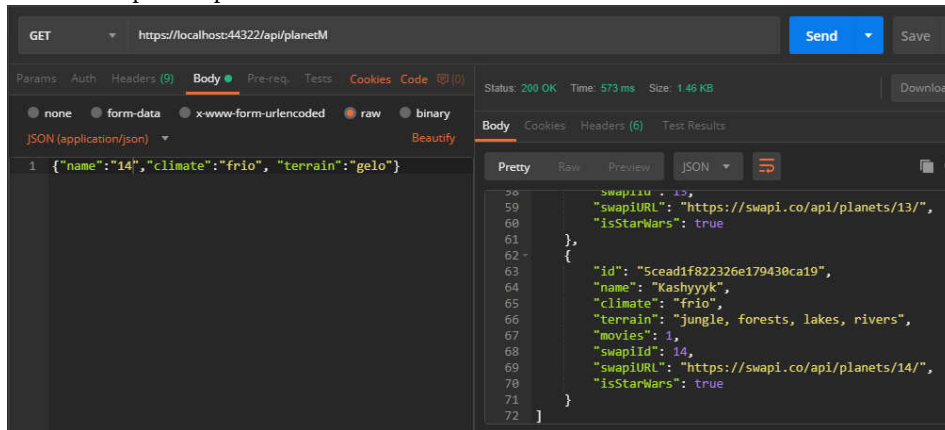
Resultado esperado: planeta não adicionado, mesmo que Climate e Terrain sejam diferentes do planeta já cadastrado. Na parte esquerda, o POST utilizado. Na parte direita, o GET mostrando que o planeta não foi cadastrado.



### 3.1.14 POST: Adicionar planeta Star Wars já existente

Cenário: tenta-se cadastrar planeta Star Wars já cadastrado.

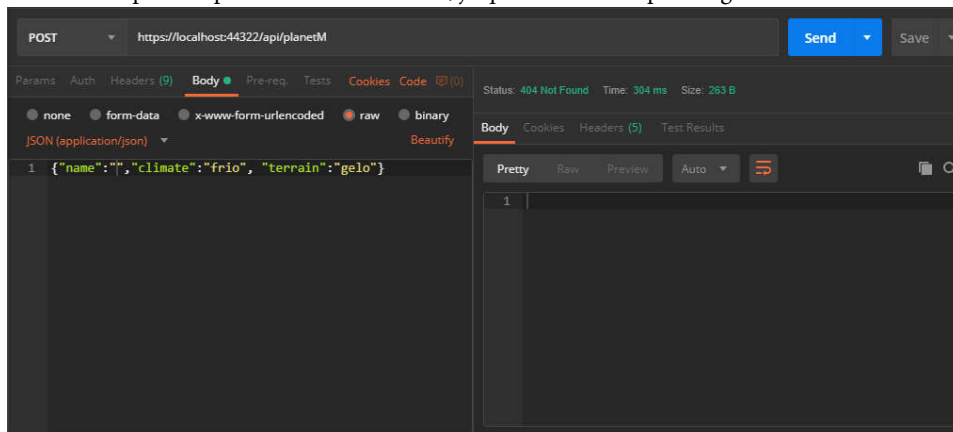
Resultado esperado: planeta não adicionado.



### 3.1.15 POST: Adicionar planeta com nome em branco

Cenário: tenta-se cadastrar planeta sem informar o nome.

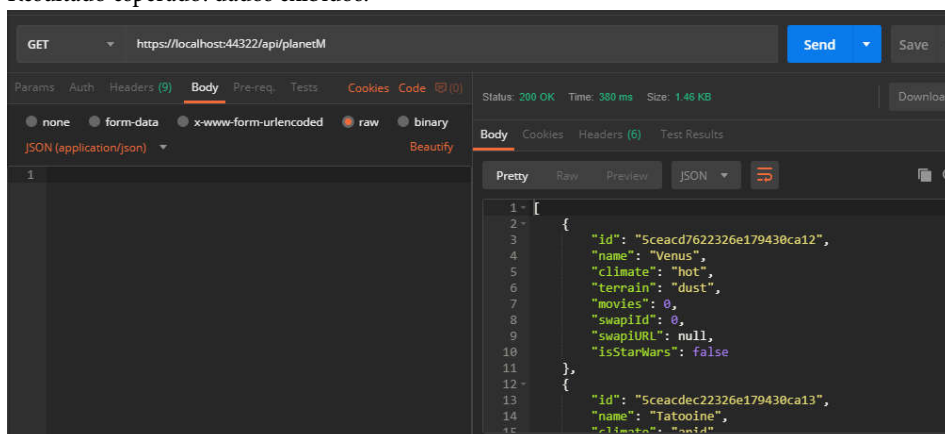
Resultado esperado: planeta não adicionado, já que nome é campo obrigatório.



### 3.1.16 GET: Todos os planetas

Cenário: obter dados de um planeta já cadastrado.

Resultado esperado: dados exibidos.

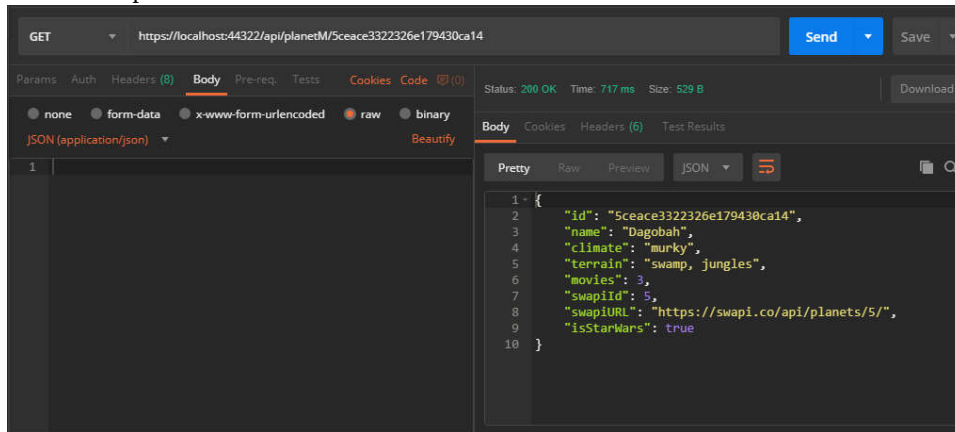




### 3.1.17 GET: Um planeta

Cenário: obter dados de um planeta já cadastrado.

Resultado esperado: dados exibidos.

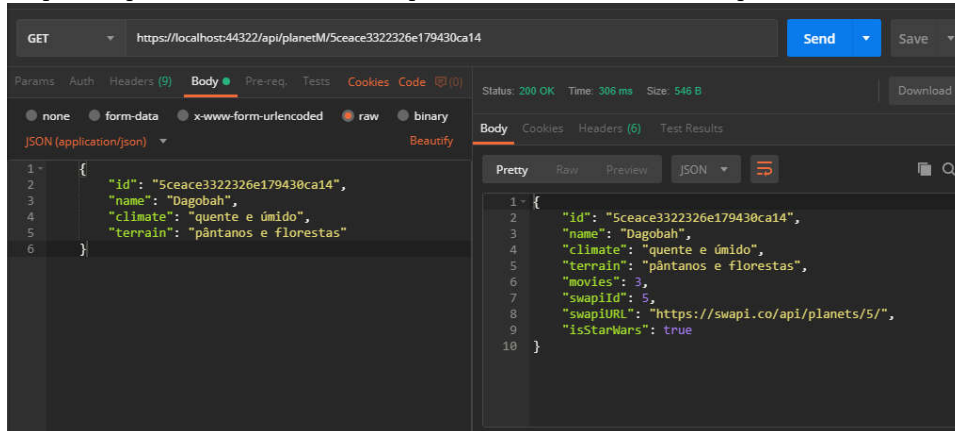


### 3.1.18 PUT: alterando um planeta

Cenário: tenta-se alterar dados de um planeta, independentemente de ser Star Wars ou não.

Resultado esperado: dados alterados.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que os dados foram alterados.

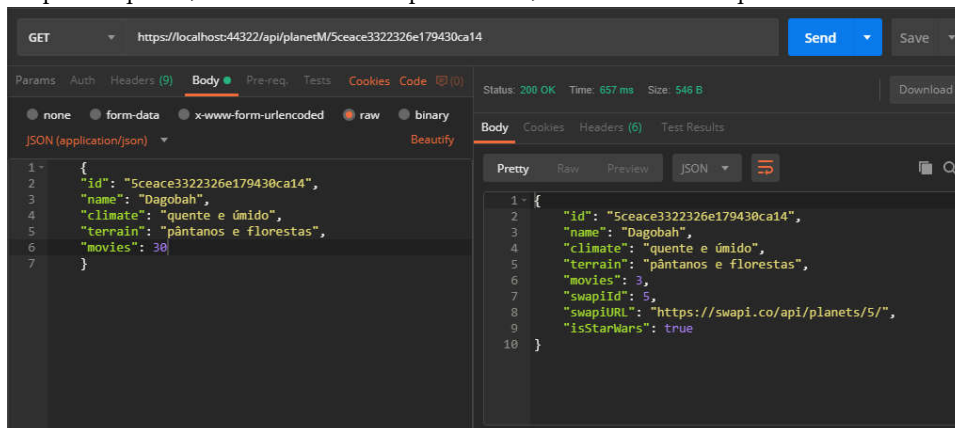


### 3.1.19 PUT: alterando um planeta, sobrescrevendo campos do swapi

Cenário: tentativa de alterar o valor do campo Movies para 30.

Resultado esperado: valor não alterado, já que Movies sempre é obtido do swapi, não sendo editável.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que Movies não foi alterado.





### 3.1.20 PUT: alterando um planeta com reset dos campos do swapi

Cenário: para voltar os valores de Climate e Terrain, executa-se um PUT com os valores em branco.

Resultado esperado: campos Climate e Terrain alterados para valores provenientes do swapi.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que os dados foram alterados.

The screenshot shows a Postman interface with a PUT request on the left and a GET response on the right. The PUT request is to `https://localhost:44322/api/planetM/5ceace3322326e179430ca14` with a JSON body: `{ "id": "5ceace3322326e179430ca14", "name": "Dagobah", "climate": "", "terrain": "" }`. The GET response is from `https://localhost:44322/api/planetM/5ceace3322326e179430ca14` with a status of 200 OK and a JSON body: `{ "id": "5ceace3322326e179430ca14", "name": "Dagobah", "climate": "murky", "terrain": "swamp, jungles", "movies": 3, "swapiId": 5, "swapiURL": "https://swapi.co/api/planets/5/", "isStarWars": true }`.

### 3.1.21 PUT: em um planeta, alterar o nome para nome já existente

Cenário: tentativa de alterar o nome de um planeta para outro nome já cadastrado.

Resultado esperado: Nome não atualizado.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que o nome não foi alterado.

The screenshot shows a Postman interface with a PUT request on the left and a GET response on the right. The PUT request is to `https://localhost:44322/api/planetM/5ceacd7622326e179430ca12` with a JSON body: `{ "id": "5ceacd7622326e179430ca12", "name": "Dagobah", "climate": "hot", "terrain": "dust" }`. The GET response is from `https://localhost:44322/api/planetM/5ceacd7622326e179430ca12` with a status of 200 OK and a JSON body: `{ "id": "5ceacd7622326e179430ca12", "name": "Venus", "climate": "hot", "terrain": "dust", "movies": 0, "swapiId": 0, "swapiURL": null, "isStarWars": false }`.

O fato do nome já existir invalida toda a operação. Abaixo, tentativa de alterar Name e Climate. Como o nome é duplicado, toda a operação é invalidada.

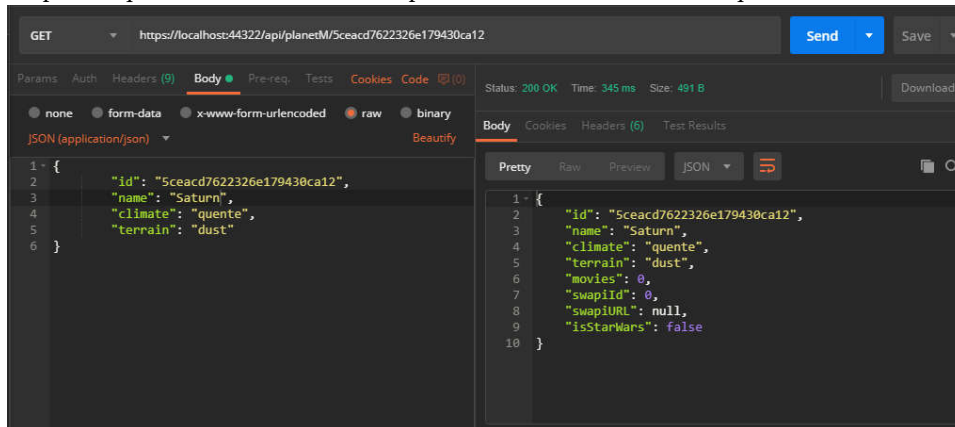
The screenshot shows a Postman interface with a PUT request on the left and a GET response on the right. The PUT request is to `https://localhost:44322/api/planetM/5ceacd7622326e179430ca12` with a JSON body: `{ "id": "5ceacd7622326e179430ca12", "name": "Dagobah", "climate": "quente", "terrain": "dust" }`. The GET response is from `https://localhost:44322/api/planetM/5ceacd7622326e179430ca12` with a status of 200 OK and a JSON body: `{ "id": "5ceacd7622326e179430ca12", "name": "Venus", "climate": "hot", "terrain": "dust", "movies": 0, "swapiId": 0, "swapiURL": null, "isStarWars": false }`.

### 3.1.22 PUT: em planeta não Star Wars, alterar o nome para nome não existente

Cenário: tentativa de alterar o nome de um planeta não Star Wars para nome não cadastrado.

Resultado esperado: Nome atualizado.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que o Name e Climate foram alterados.

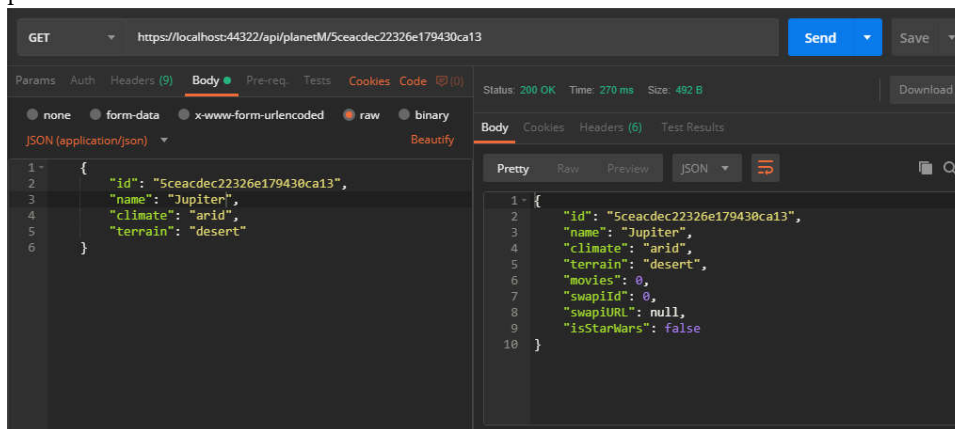


### 3.1.23 PUT: em planeta Star Wars, alterar o nome para nome não existente (Star Wars-only mode OFF)

Cenário: Tentativa de alterar o nome de um planeta Star Wars para nome não cadastrado.

Resultado esperado: Nome atualizado, vínculo com swapi perdido.

Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que Name foi alterado e o vínculo com o swapi foi perdido.

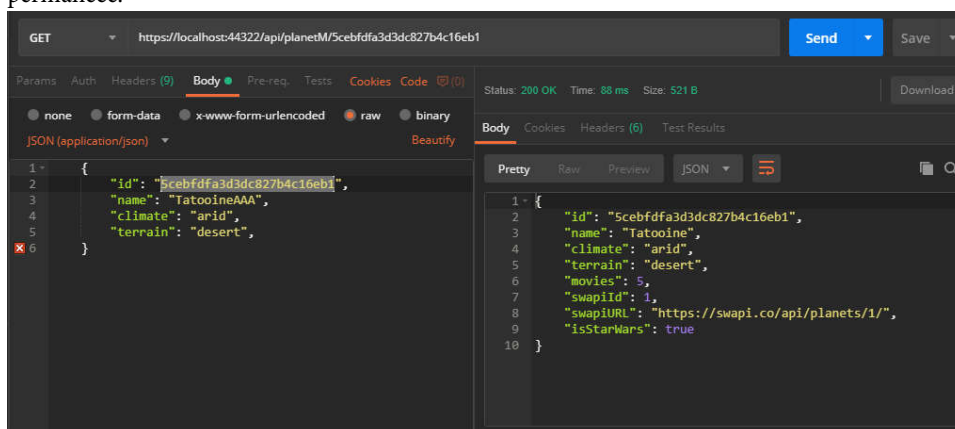


### 3.1.24 PUT: em planeta Star Wars, alterar o nome para nome não existente (Star Wars-only mode ON)

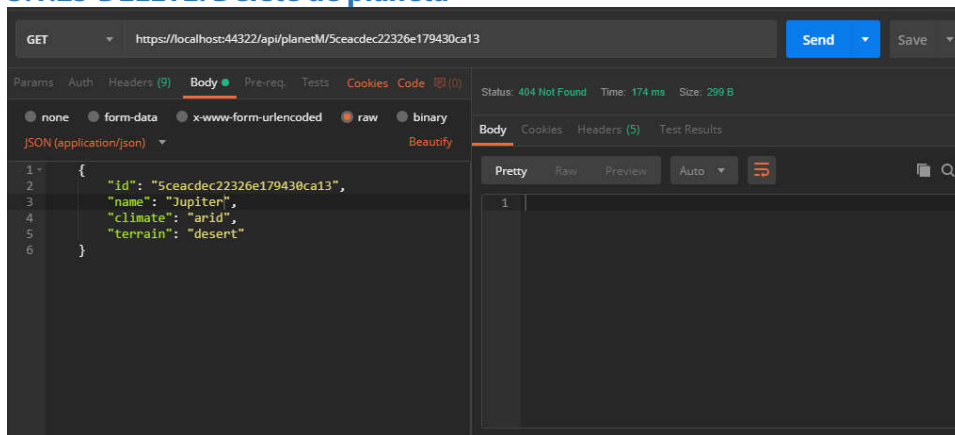
Cenário: Tentativa de alterar o nome de um planeta Star Wars para nome não cadastrado.

Resultado esperado: Nome não alterado.

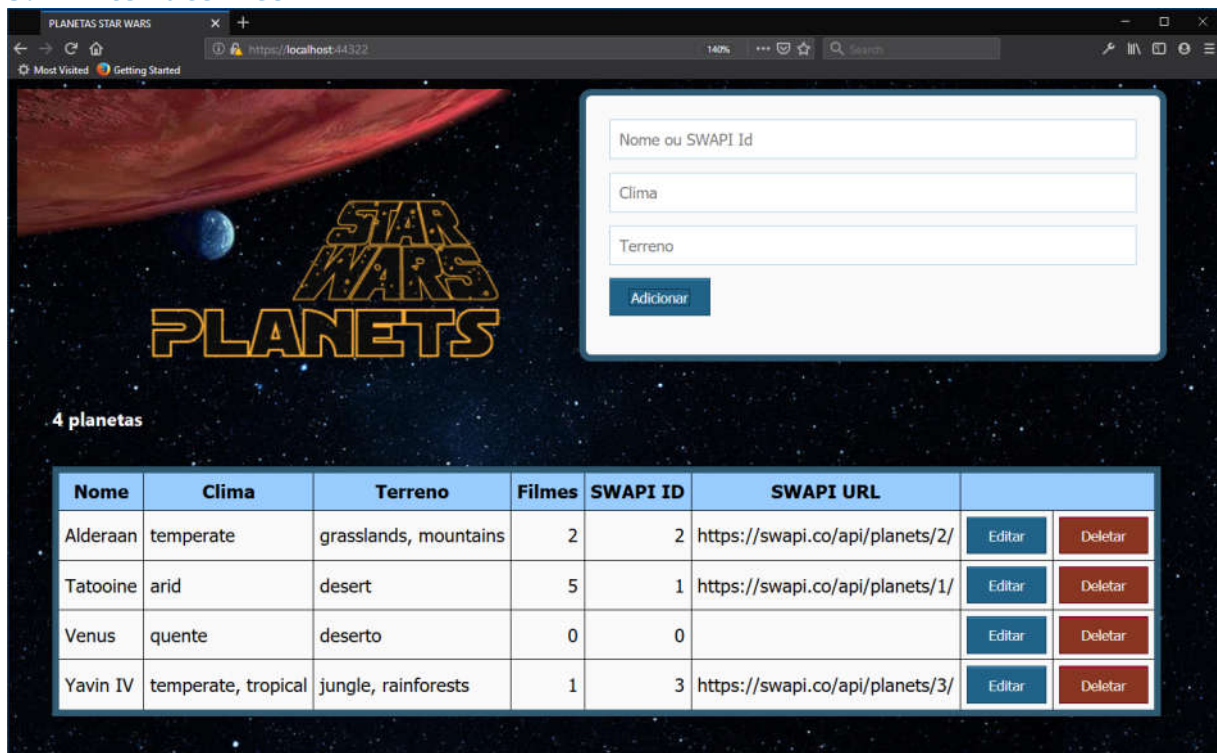
Na parte esquerda, o PUT utilizado. Na parte direita, o GET mostrando que Name não foi alterado e o vínculo com o swapi permanece.



### 3.1.25 DELETE: Delete de planeta



## 3.2 Interface Web



### 3.2.1 Adicionando planetas por nome (Star Wars-only mode OFF)

Não Star Wars

Venus

quente

deserto

Adicionar

Star Wars

Tatooine

Clima

Terreno

Adicionar

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar

### 3.2.2 Adicionando planeta não Star Wars por nome (Star Wars-only mode ON)

Planeta Venus já existente ou não Star Wars!

OK

### 3.2.3 Adicionando planeta Star Wars por número

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar

### 3.2.4 Adicionando planeta Star Wars por número com override de clima e terreno

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar
Yavin IV	frio	gelo	1	3	https://swapi.co/api/planets/3/	Editar	Deletar

### 3.2.5 Reset em planeta Star Wars

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar
Yavin IV	temperate, tropical	jungle, rainforests	1	3	https://swapi.co/api/planets/3/	Editar	Deletar

### 3.2.6 Alterando nome de planeta Star Wars (Star Wars-only mode OFF)

De Yavin IV para Jupiter.

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Jupiter	temperate, tropical	jungle, rainforests	0	0		Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar

### 3.2.7 Alterando nome de planeta Star Wars (Star Wars-only mode ON)

Alteração é recusada sem mensagem de erro.

### 3.2.8 Alterando nome de planeta não Star Wars para nome já existente

De Jupiter para Venus. Como já existe planeta com este nome, a alteração é ignorada.

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Jupiter	temperate, tropical	jungle, rainforests	0	0		Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar

### 3.2.9 Adicionando planeta com nome já cadastrado

Planeta Tatooine já existente ou não Star Wars!

OK

### 3.2.10 Adicionando planeta sem nome

Nome em branco!

OK

### 3.2.11 Deletando planeta

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	https://swapi.co/api/planets/2/	Editar	Deletar
Tatooine	arid	desert	5	1	https://swapi.co/api/planets/1/	Editar	Deletar
Venus	quente	deserto	0	0		Editar	Deletar
Yavin IV	temperate, tropical	jungle, rainforests	1	3	https://swapi.co/api/planets/3/	Editar	Deletar

Nome	Clima	Terreno	Filmes	SWAPI ID	SWAPI URL		
Alderaan	temperate	grasslands, mountains	2	2	<a href="https://swapi.co/api/planets/2/">https://swapi.co/api/planets/2/</a>	Editar	Deletar
Tatooine	arid	desert	5	1	<a href="https://swapi.co/api/planets/1/">https://swapi.co/api/planets/1/</a>	Editar	Deletar
Yavin IV	temperate, tropical	jungle, rainforests	1	3	<a href="https://swapi.co/api/planets/3/">https://swapi.co/api/planets/3/</a>	Editar	Deletar