



Curso completo de Python!





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite:

<http://vmzsolutions.com.br>



REST API com Python

O python tem alguns módulos excelentes para consumir uma API's. Iremos utilizar um módulo e chamar de requests. Para instalar o módulo, execute esse comando abaixo:

```
# pip install requests
```



REST API com Python

Depois disto, quando vamos começar um arquivo python, precisamos importar o módulo necessário

```
import requests
```

Agora, quando você fizer uma requisição, você deverá seguir essa sintaxe:

```
requests.<METODO>
```



REST API com Python

Exemplo:

```
print requests.get("http://localhost:3000/usuarios/")
```



REST API com Python

Temos um site para você fazer os seus testes de todos os métodos existentes, o <http://httpbin.org/> ex:

Método POST:

```
print requests.post("http://httpbin.org/post", data={'key':'value'})
```

Método PUT:

```
print requests.put("http://httpbin.org/put", data={'key':'value'})
```



Método DELETE:

```
print requests.delete("http://httpbin.org/post", data={'key':'value'})
```

Método HEAD:

```
print requests.head("http://httpbin.org/post", data={'key':'value'})
```

Método OPTIONS:

```
print requests.options("http://httpbin.org/post", data={'key':'value'})
```



Caso queira buscar um usuário de forma específica, aí terá que seguir a mesma lógica usado no RestClient:

```
print requests.head("http://127.0.0.1/usuarios/1/")
```

O seu retorno será de apenas o código 200, pois somente mostra o seu Status. E 200 é uma requisição com sucesso!



Se você quiser ver o status code de forma específica,
poe usar o método `status_code`

```
print requests.get("http://127.0.0.1:3000/usuarios/1/").status_code
```



Para saber o seu conteúdo, precisamos colocar o método `._content`

```
print requests.get("http://127.0.0.1:3000/usuarios/1/")._content
```



Vamos testar todas essas aplicações de operações usando o python junto com o API em Json. Vamos criar um arquivo que contenha todas as operações CRUD.



O primeiro será importar todos os módulos necessários

```
import requests  
import json  
import sys
```



Agora, vamos criar as nossas funções, o primeiro, será para criar o caminho da URL.

```
def _url(path):  
    return 'http://127.0.0.1:3000' + path
```



Depois teremos outras funções de operações CRUD.

Listar todos os itens no Json

```
def get_tasks()
```

Adicionar um item no Json

```
def add_task()
```

Apagar um item no Json

```
def del_task()
```

Atualizar um item no Json

```
def update_task()
```



E temos o `if..main` para fazer as operações que desejar

```
if __name__ == '__main__':
```

* Veja mais no arquivo `rest_cli.py`