



# Desenvolvimento de Serviços com Spring Boot

Etapa/Aula 03.1

Professor(a): Flávio Neves

E-mail: [flavio.neves@prof.infnet.edu.br](mailto:flavio.neves@prof.infnet.edu.br)

# Roteiro da Aula

- Funcionamento do método GET
- Funcionamento do método POST
- Exemplo de uso dos métodos GET e POST

# APIs REST- método GET



# APIs REST- método GET

- É usado para obter algum recurso hospedado no servidor.
  - Solicitação de Método GET

```
#Request
GET /CASEREST/v1/case/9E45A997-0E42-406E-AAC4-EE55D8BCF2ED/status
?TargetObjectStore=MyExampleObjectStore
HTTP/1.1
Host: www.CaseMgmtExample.net
```

- Resposta do Método GET

GET



<http://localhost:8080/users>

```
#Response
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "Status": "Working",
    "CaseIdentifier": "MY_Case_000000100105",
    "DateCreated": "2010-07-16T21:50:36Z",
    "DateLastModified": "2010-07-16T21:50:36Z"
}
```

# APIs REST- método GET

- Códigos de resposta para o método GET

Código	Descrição
200 OK	O método foi concluído com êxito. A resposta que é retornada pelo método GET inclui o status do caso especificado.
400 Bad Request	O parâmetro <b>TargetObjectStore</b> obrigatório não foi especificado ou o valor do parâmetro era inválido.
404 Not Found	O caso especificado na solicitação não foi encontrado.
500 Internal Server Error	Ocorreu um erro do servidor. Para obter informações sobre o erro, consulte o elemento userMessage na resposta JSON.

# APIs REST- método POST

- É utilizado para enviar dados para o servidor.

POST

<http://localhost:8080/users>

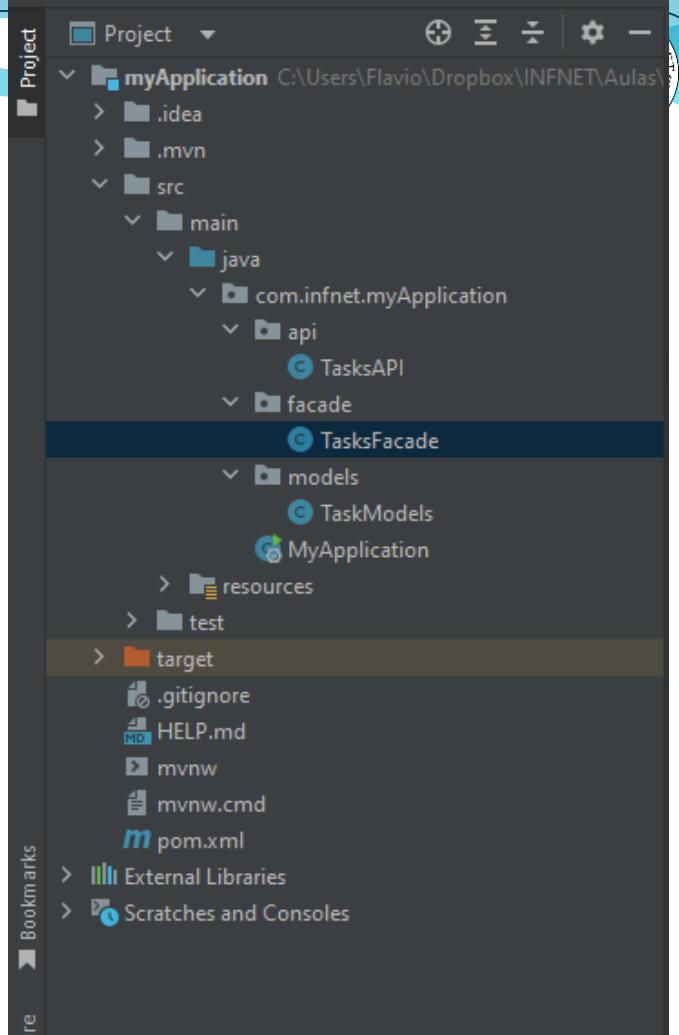
```
{  
    "Status": "Working",  
    "CaseIdentifier": "MY_Case_000000100105",  
    "DateCreated": "2010-07-16T21:50:36Z",  
    "DateLastModified": "2010-07-16T21:50:36Z"  
}
```

# Exemplo

- Lista de algumas anotações:

<https://domineospring.wordpress.com/2016/07/13/guia-das-annotations-do-spring/>

- Esta aplicação é uma **API REST simples** desenvolvida em **Spring Boot** para gerenciar tarefas (tasks).
- Ela implementa um sistema básico de CRUD
  - (Create, Read, Update, Delete) para objetos do tipo TaskModels,
- Possue atributos como id, name e apelido.





# Desenvolvimento de Serviços com Spring Boot

Etapa/Aula 03.1

Professor(a): Flávio Neves

E-mail: [flavio.neves@prof.infnet.edu.br](mailto:flavio.neves@prof.infnet.edu.br)