

## RESTful Web Service com Spring

Depois da atividade com serviços REST usando Maven, apresenta-se agora um exemplo com Gradle e Spring.

### 1- Construir um RESTful Web Service com Spring:

- Resource Representation Class
- [Resource Controller](#)

Seguir o exemplo em:

<https://spring.io/guides/gs/rest-service/>

(ou ir diretamente para a subpasta com o projeto completo, em

`gs-rest-service-main/complete`

)

Passos:

1. *Create a Resource Representation Class*
2. *Create a Resource Controller*
3. Aplicação Spring
4. Testar o serviço, nos endereços:
  - `http://localhost:8080/greeting`
  - `http://localhost:8080/greeting?name=User`

---

### 2- Consuming a RESTful Web Service

Suponhamos que o serviço já existe... e pretendemos apenas um cliente.

Seguir o exemplo em:

<https://spring.io/guides/gs/consuming-rest/>

---

## REST Service, Java Records e JDK *preview features*

3.1- obtenha o código:

- [download link](#) direto

**Notas:**

- se usa uma versão de Java que não suporta records, este exemplo não funciona; mas pode trocar os records por classes normais em Java
- se usa Java 15, deve ajustar `sourceCompatibility = '15'` , no `build.gradle`.

3.2- Veja a descrição deste exemplo em

<https://dev.to/psideris89/java-14-records-in-spring-boot-rest-api-n29>

**3.3- Executar na consola:**

```
gradle clean
gradle build
gradle test
```

# execução com o que o gradle gerou (com enable-preview)

\$ gradle bootJar # para gerar o jar executável, seguido de:

\$ java --enable-preview -jar build/libs/java-records-demo-0.0.1-SNAPSHOT.jar

ou então (binário gradle de sistema ou o wrapper):

\$ ./gradlew bootRun

**3.4- Teste no browser... URL:**

<http://localhost:8080/movies>

Curiosidade sobre a estrutura de ficheiros no projeto

```
$ tree src/
src/
├── main
│   ├── java
│   │   └── com
│   │       ├── psideris
│   │       └── moviesapi
│   │           ├── MoviesApiApplication.java
│   │           ├── MoviesController.java
│   │           ├── records
│   │           ├── Director.java
│   │           └── Movie.java
│   ├── resources
│   └── application.properties
└── test
    ├── java
    ├── com
    ├── psideris
    ├── moviesapi
    └── MoviesApiApplicationTests.java
```

Contudo, o tamanho da pasta onde estamos é

```
$ du . -sh
27M .
```

Para além da pasta build (.class, jars)

Há a pasta "escondida" com downloads do gradle

```
$ du -sh .gradle/
9,7M
```

 [Contactar suporte do site](#) 

Nome de utilizador: [Rodrigo Alves](#) (Sair)

[Resumo da retenção de dados](#)

[Obter a Aplicação móvel](#)

---

Fornecido por [Moodle](#)