



13

## Faça como eu fiz: consumindo uma API, criando classes e interfaces, e implementando métodos

Agora é com você! Faça o mesmo procedimento da aula para consumir a API e modelar os dados da série.



### Opinião do instrutor

Para iniciar, crie o projeto Spring utilizando o site do [Spring Initializr](https://start.spring.io/) (<https://start.spring.io/>), utilizando o Maven como gerenciador de dependências. Faça o download do projeto, descompacte-o e abra no IntelliJ. Aproveite para já criar os pacotes `model` e `service`.

No pacote `service`, crie a classe `ConsumoAPI`, com o método que irá retornar o json com o corpo da resposta obtida.

ConsumoApi.java :

```
package br.com.alura.scr
```

```
import java.io.IOExcepti
```

```
import java.net.URI;
```

```
import java.net.http.Htt
```

```
import java.net.http.Htt
```

```
import java.net.http.Htt
```

```
public class ConsumoApi
```

```
    public String obterD
```

```
        HttpClient clien
```

```
        HttpRequest requ
```

```
            .uri(URI
```

```
            .build())
```

```
        HttpResponse<Str
```

```
        try {
```

```
            response = c
```

```
            .sen
```

```
        } catch (IOExcep
```

```
            throw new Ru
```

```
        } catch (Interru
```

```
            throw new Ru
```

```
        }
```

```
        String json = re
```

```
        return json;
```

```
    }
```

}

[COPIAR CÓDIGO](#)

Como precisaremos  
desserializar o JSON, inclua a  
dependência do Jackson no  
arquivo `pom.xml` :

`pom.xml` :

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework</groupId>
    <artifactId>spring</artifactId>
    <version>3.1.1</version>
    <relativePath/>
  </parent>
  <groupId>br.com.alura</groupId>
  <artifactId>screenmatch</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>screenmatch</name>
  <description>Primeiro projeto</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org</groupId>
```

```
        <artifactId>
    </dependency>

    <dependency>
        <groupId>org
        <artifactId>
        <scope>test<
    </dependency>

    <dependency>
        <groupId>com
        <artifactId>
        <version>2.1
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>
            <artifac
        </plugin>
    </plugins>
</build>

</project>
```

[COPIAR CÓDIGO](#)

Agora com a dependência adicionada, dentro do pacote `model`, é necessário criar o

record DadosSerie para  
modelar os dados que  
desejamos representar na  
aplicação:

DadosSerie.java :

```
package br.com.alura.scr
```

```
import com.fasterxml.jac
```

```
import com.fasterxml.jac
```

```
@JsonIgnoreProperties(ig
```

```
public record DadosSerie
```

```
}
```

COPIAR CÓDIGO

Precisaremos desserializar os  
dados do JSON, representando-  
os como um objeto do tipo  
DadosSerie. Para isso, no  
pacote `service` vamos criar a  
interface `IConverteDados` e a  
classe `ConverteDados` que irá  
implementar o método  
descrito na mesma.

IConverteDados.java :

```
package br.com.alura.scr
```

```
public interface IConver  
    <T> T obterDados(St  
}
```

[COPIAR CÓDIGO](#)

ConverteDados.java :

```
package br.com.alura.scr
```

```
import com.fasterxml.jac  
import com.fasterxml.jac
```

```
public class ConverteDad  
    private ObjectMapper
```

```
@Override
```

```
public <T> T obterDa  
    try {  
        return mappe  
    } catch (JsonPro  
        throw new Ru  
    }  
}  
}
```

[COPIAR CÓDIGO](#)

Na classe principal do projeto, ScreenmatchApplication, faça a implementação da CommandLineRunner , sobrescrevendo o método run .

ScreenmatchApplication.java :

```
package br.com.alura.scr
```

```
import br.com.alura.scre
```

```
import br.com.alura.scre
```

```
import br.com.alura.scre
```

```
import org.springframework
```

```
import org.springframework
```

```
import org.springframework
```

```
@SpringBootApplication
```

```
public class Screenmatch
```

```
public static void m
```

```
SpringApplicatio
```

```
}
```

```
@Override
```

```
public void run(Stri
```

```
var consumoApi =
```

```
var json = consu
```

```
System.out.print
```

```
ConverteDados co
```

```
DadosSerie dados
```

```
System.out.print
```

```
}  
}
```

**COPIAR CÓDIGO**