Introdução ao AngularJS

Professor Anderson Henrique

TECH
soluções em ti

Configurando o jersey no arquivo pom.xml

Precisamos configurar o Maven com Dependências do Jersey, como podemos saber quais dependências jersey devemos utilizar no Maven?

Acessando a documentação Jersey no seguinte endereço:

https://eclipse-ee4j.github.io/jersey.github.io/documentation/latest/index.html

No item **2.3.2**

Links: Table of Contents | Single HTML

Jersey 2.31 User Guide

Table of Contents

```xml
<dependency>
    <groupId>org.glassfish.jersey.containers</groupId>
    <!-- if your container implements Servlet API older than 3.0, use "jersey-container-servlet-core"  -->
    <artifactId>jersey-container-servlet</artifactId>
    <version>2.31</version>
</dependency>
```

Na seção de dependências vamos adicionar essa dependência, nosso arquivo pom.xml ficará assim

```xml
        <dependency>
            <groupId>org.glassfish.jersey.containers</groupId>
            <artifactId>jersey-container-servlet</artifactId>
            <version>2.0</version>
        </dependency>
    </dependencies>
</project>
```

Vamos limpar nosso arquivo pom.xml, deletando toda a seção <build> do nosso projeto.

Vamos alterar o <groupId> para com.crud.back e <version> para 1.0

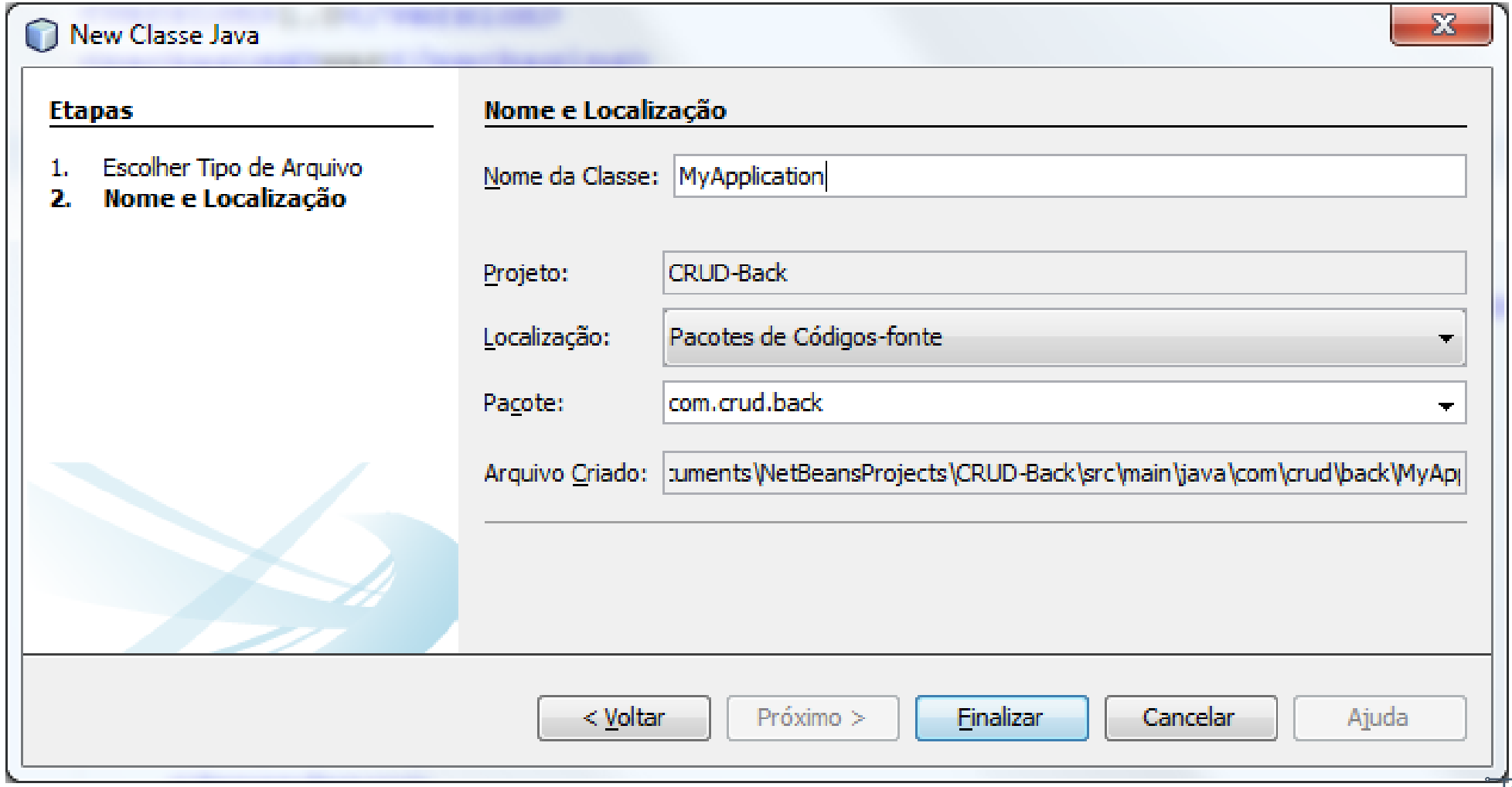
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.crud.back</groupId>
    <artifactId>CRUD-Back</artifactId>
    <version>1.0</version>
    <packaging>war</packaging>

    <name>CRUD-Back</name>
```

Como configurar o jersey, como falar para o Java que estamos utilizando o jersey?

TECH
soluções em ti

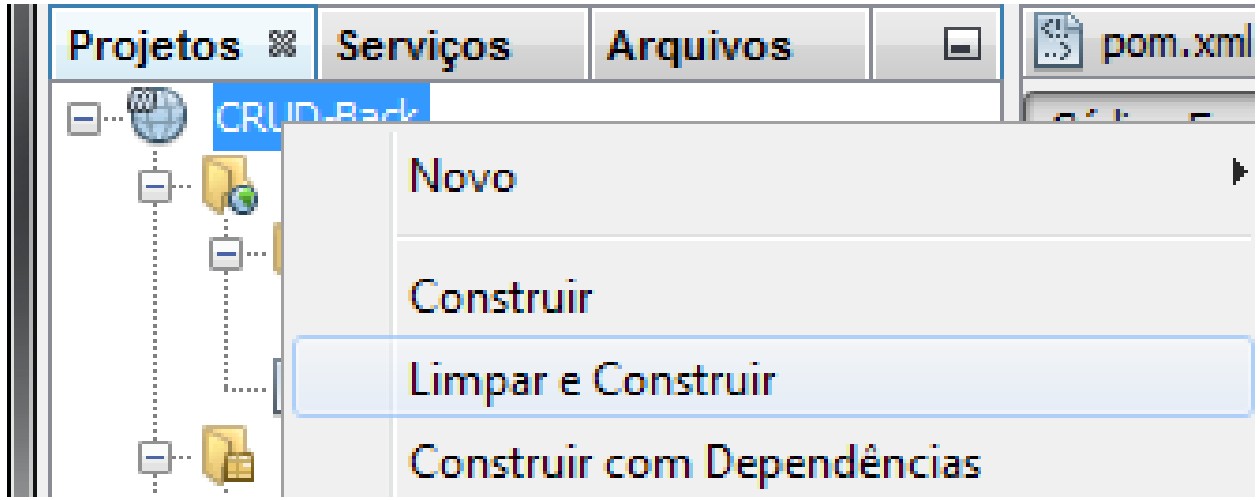
Vamos criar uma classe, um **startup** que inicia junto com o nosso projeto, essa classe será criada em um pacote que será informado no construtor dessa classe, crie um pacote na pasta Pacotes de Código-fonte chamado: **com.crud.back.controllers**

A classe será criada no pacote chamado: **com.crud.back** e o nome da classe é **MyApplication**

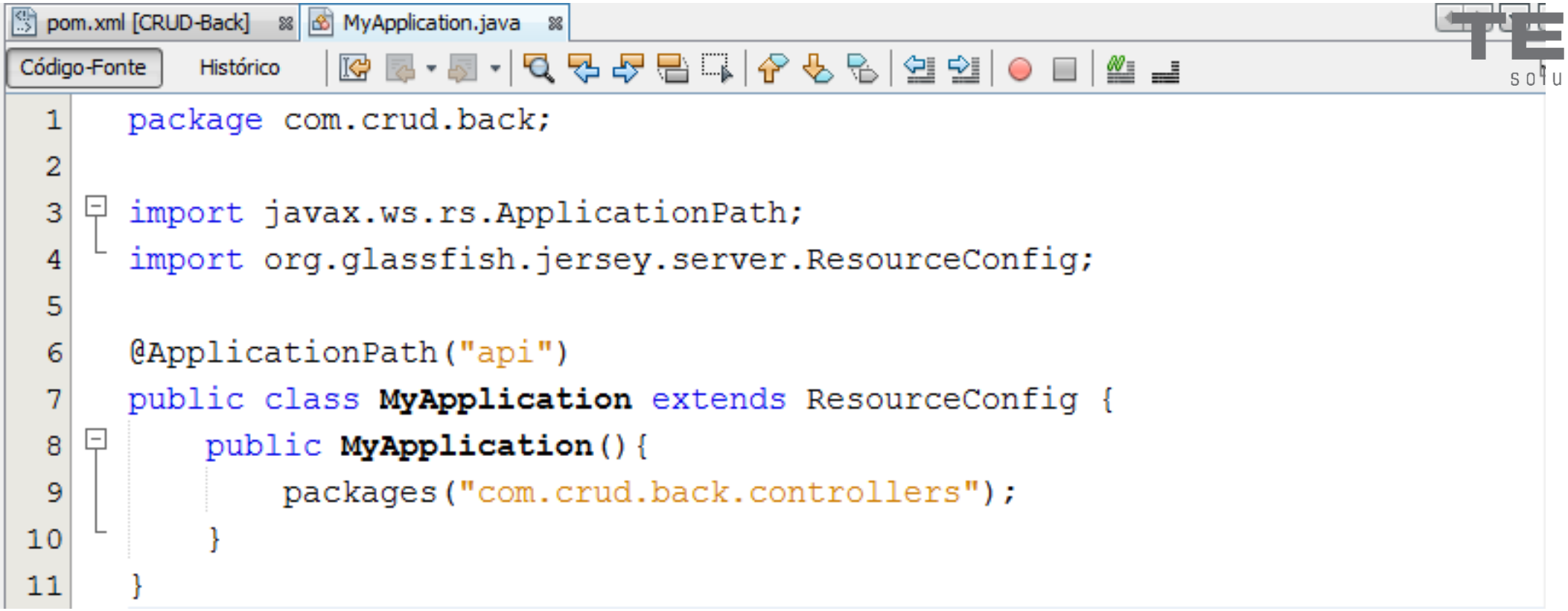


Limpar e Construir

Para que o Netbeans consiga fazer o download de todas as bibliotecas Jersey precisamos (Limpar e Construir nosso Projeto Java)



TECH
soluções em ti



Como sabemos que precisamos apenas dessa classe para configurar o jersey?

Acessando a documentação Jersey no seguinte endereço:

https://eclipse-ee4j.github.io/jersey.github.io/documentation/latest/index.html

No item **4.2**

TECH
soluções em ti

Example 4.2. Reusing Jersey implementation in your custom application model

```
1  public class MyApplication extends ResourceConfig {
2      public MyApplication() {
3          packages("org.foo.rest;org.bar.rest");
4      }
5  }
```

Dentro do pacote **com.crud.back.controllers** vamos criar uma classe java chamada **HelloController**

TECH
soluções em ti

Nesta classe teremos dois métodos, a nossa classe recebe uma anotação @Path que utilizamos para indicar o caminho da nossa classe

O primeiro método recebe a anotação @GET indicando que retornará um valor na requisição da URL, a anotação @Produces indica que o método produzirá um valor que é do tipo texto e vinculamos o valor do parâmetro na requisição da URL por meio da anotação @QueryParam

```
10  @Path("hello")
    public class HelloController {
12      @GET
13      @Produces(MediaType.TEXT_PLAIN)
        public String getMensagem(@QueryParam("usuario") String usuario){
15          return "Bem vindo: " + usuario;
16      }
```

TECH
soluções em ti

O segundo método recebe a anotação @GET indicando que retornará um valor na requisição da URL, a anotação @Produces indica que o método produzirá um valor que é do tipo texto e a anotação @Path onde definimos um caminho para passagem do parâmetro.

```
18      @GET
19      @Path("usuario/{id}")
20      @Produces(MediaType.TEXT_PLAIN)
        public String getUsuario(@PathParam("id") long id){
22          return "Recuperando usuário com ID: " + id;
23      }
24  }
```

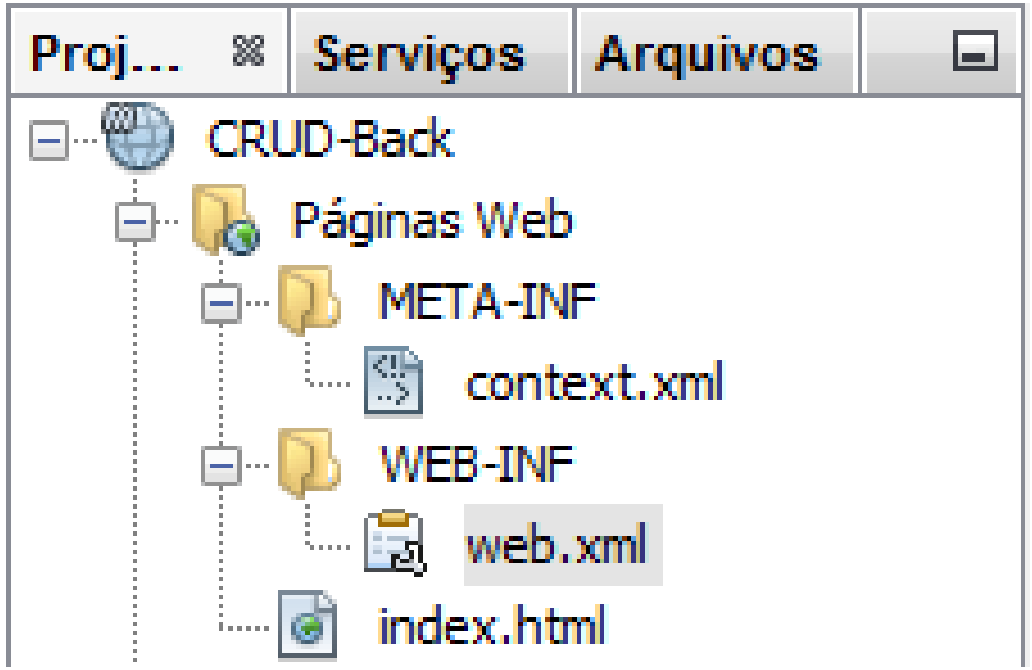
Como vinculamos o parâmetro do método com o caminho será passado na URL? Por meio da anotação @PathParam

TECH
soluções em ti

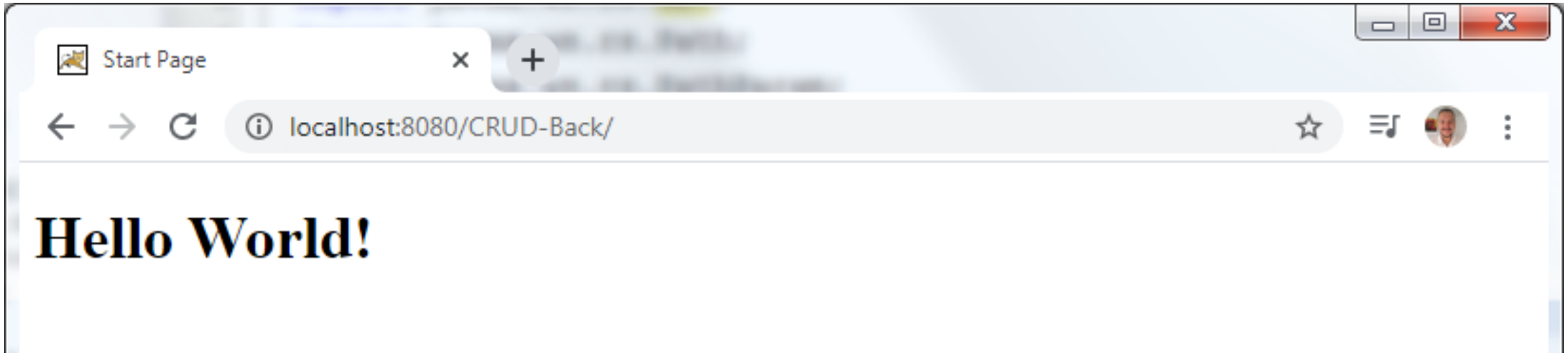
Ao Limpar e Construir o projeto, ocorrerá um erro na compilação informando que ainda não temos a pasta WEB-INF e o arquivo web.xml

Vamos criar no diretório Páginas Web uma pasta chamada **WEB-INF** e dentro desta pasta vamos criar um arquivo do tipo Descritor **de Implantação Padrão** (web.xml)

Nossa estrutura de pastas ficará assim:



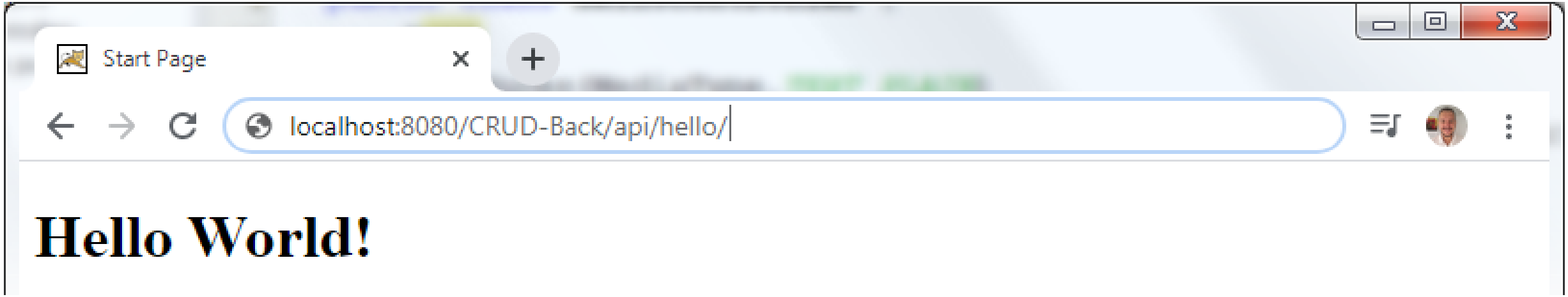
Vamos Limpar e Construir o projeto e executá-lo:



Na URL ele conseguiu executar nosso projeto no servidor Apache Tomcat **localhost:8080** e em seguida pegando o caminho do contexto da aplicação **CRUD-Back**

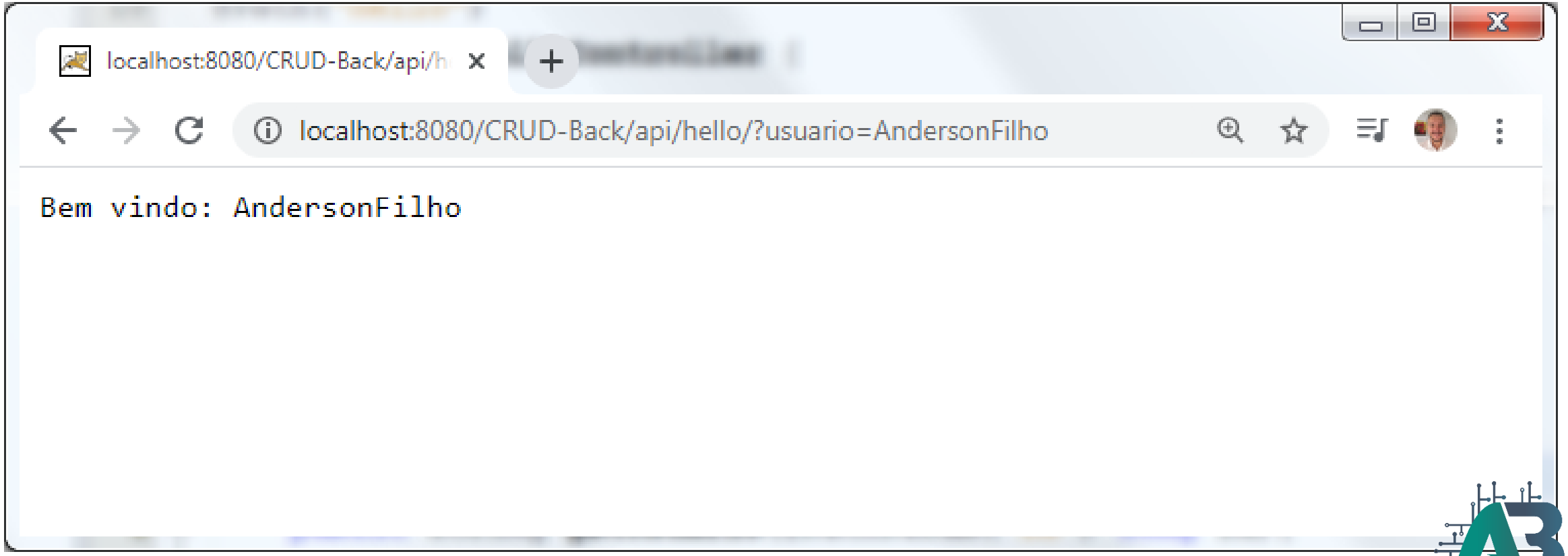
TECH
soluções em ti

Para acessarmos a nossa aplicação precisamos seguir o caminho que foi definido na classe startup **MyApplication** e o caminho que foi definido na classe **HelloController**



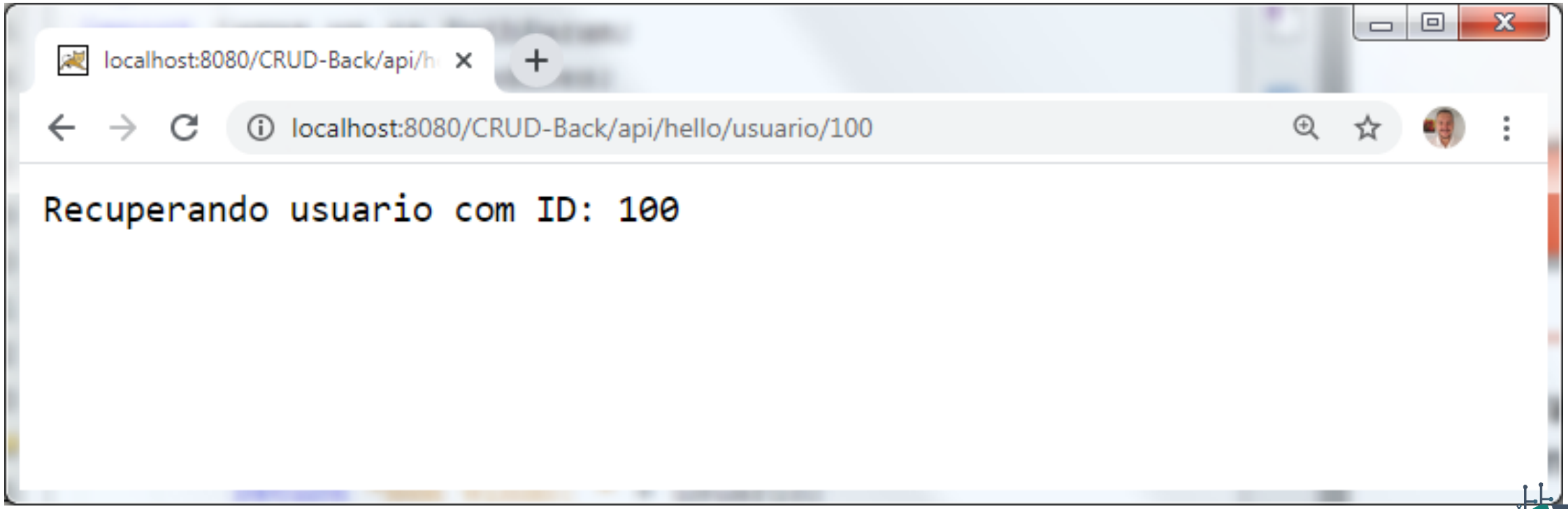
TECH
soluções em ti

Agora, quais dos dois métodos eu quero acessar? Acessando o **primeiro método** preciso passar um parâmetro na URL com o sinal **?** Seguido do **nome do parâmetro** definido no método passando um valor com o sinal de **=**



TECH
soluções em ti

Agora, quais dos dois métodos eu quero acessar? Acessando o **segundo método** preciso passar o caminho que foi definido no método e o valor do id do usuário



TECH
soluções em ti

Dúvidas?

Professor Anderson Henrique

Para a próxima aula

01 – Aplicação Back-End em JAVA

02 – Criando nosso projeto CRUD (Create, Read, Update e Delete) em um Controller Restful API

Professor Anderson Henrique