

The image shows the 'New Project' dialog in IntelliJ IDEA. On the left, the 'Generators' list has 'Spring Boot' selected. The main area contains the following configuration: 'Server URL' is 'start.spring.io'; 'Name' is 'SpringBootDemo2'; 'Location' is '~\IdeaProjects'; 'Project will be created in:' is '~\IdeaProjects\SpringBootDemo2'; 'Create Git repository' is checked; 'Language' is 'Java'; 'Type' is 'Gradle - Groovy'; 'Group' is 'org.example'; 'Artifact' is 'SpringBootDemo2'; 'Package name' is 'org.example.springbootdemo2'; 'JDK' is '21 Oracle OpenJDK 21.0.4'; 'Java' version is '17'; and 'Packaging' is 'Jar'. At the bottom right, the 'Next' button is highlighted in blue.

The image shows the Spring Initializr web application interface. At the top, there's a 'New Project' button. Below it, the 'Spring Boot' version is set to '3.4.1'. The 'Dependencies' section is expanded, showing a search bar and a list of dependencies. Under the 'Web' category, 'Spring Web' is selected with a blue checkmark. To the right, the 'Spring Web' dependency is described: 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.' Below this, there are three links: 'Building a RESTful Web Service', 'Serving Web Content with Spring MVC', and 'Building REST services with Spring'. At the bottom, the 'Added dependencies' list shows 'Spring Web' has been added.

Nos crea la clase main por defecto

```
SpringBootDemoApplication.java x HolaMundo.java
1 package org.example.springbootdemo;
2
3 > import ...
4
5
6 @SpringBootApplication new *
7 public class SpringBootDemoApplication {
8
9     public static void main(String[] args) { new *
10         SpringApplication.run(SpringBootDemoApplication.class, args);
11     }
12 }
13
```

Creamos una clase HolaMundo y le ponemos `@RestController` para conectarlo. Hacemos un `@GetMapping("/hola")` donde hola será la ruta y debajo donde imprimimos un String que pone "Hola Mundo!!!".

```
SpringBootDemoApplication.java x HolaMundo.java x
1 package org.example.springbootdemo;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HolaMundo {
8     @GetMapping("/hola")
9     public String hola() {
10         return "Hola Mundo!!!";
11     }
12 }
13
```

Aquí vemos el resultado de nuestro programa en el PostMan y en un buscador.

The image shows two side-by-side screenshots. The left screenshot is from the Postman application, displaying a GET request to `http://localhost:8080/hola`. The 'Params' tab is selected, showing a table with one query parameter: 'Key' with a value of 'Value'. The 'Body' tab is also visible, showing the response body: 'Hola Mundo!!!'. The right screenshot is from a web browser, showing the same URL `localhost:8080/hola` in the address bar. The browser's content area displays the text 'Hola Mundo!!!'.

Postman interface showing a GET request to `http://localhost:8080/hola`. The response body is `Hola Mundo!!!`.

Browser interface showing the URL `localhost:8080/hola` and the response `Hola Mundo!!!`.