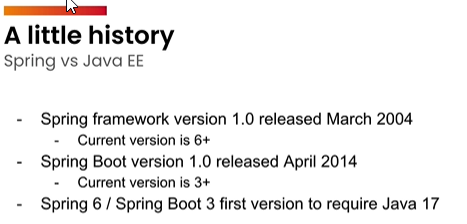


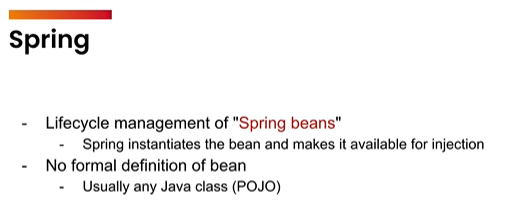
A close-up of a white background

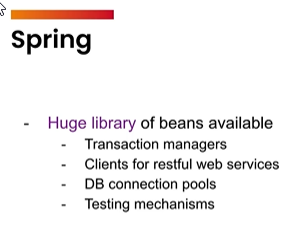
Description automatically generated

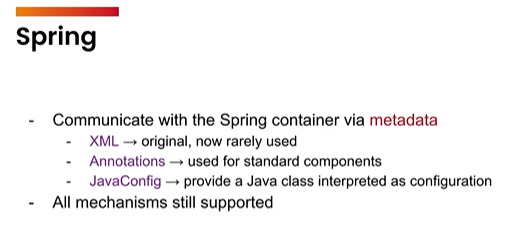
A close-up of a computer history

Description automatically generated



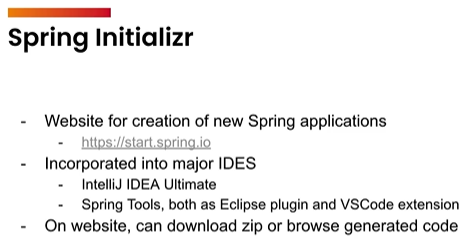


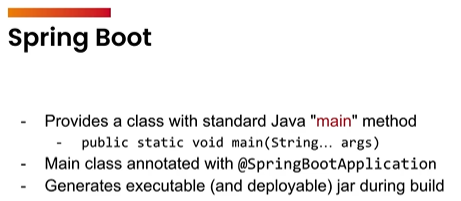




A close-up of a computer screen

Description automatically generated





A blue and white background with text

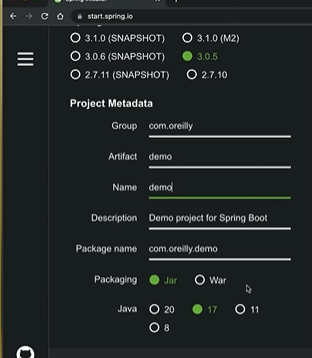
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

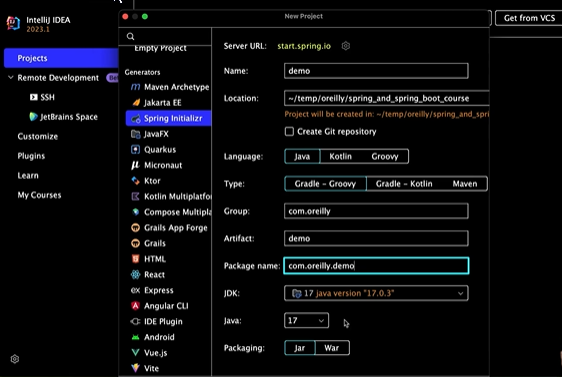
Description automatically generated

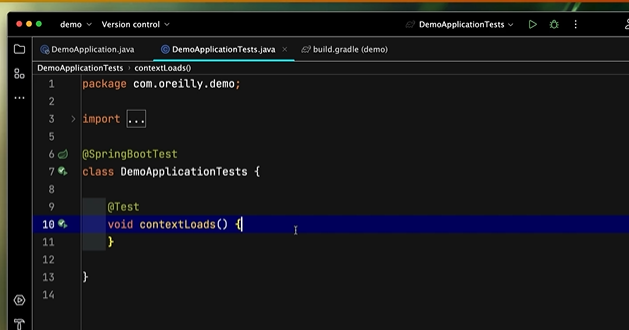
A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

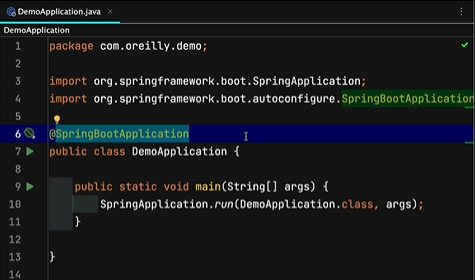
Description automatically generated



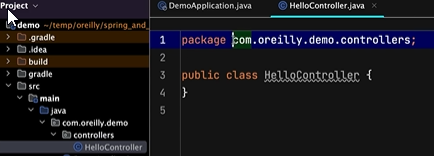


A screen shot of a computer

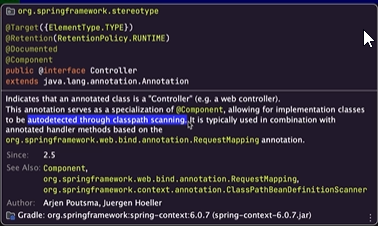
Description automatically generated



Component Scan starts here, since SpringBootApplication is defined here. Important we define the controller under this root package



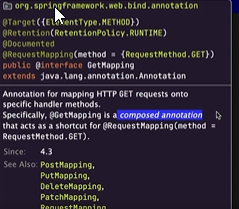
Controller: Receives a HTTP request and they either return a HTTP response or return you to a view somewhere.

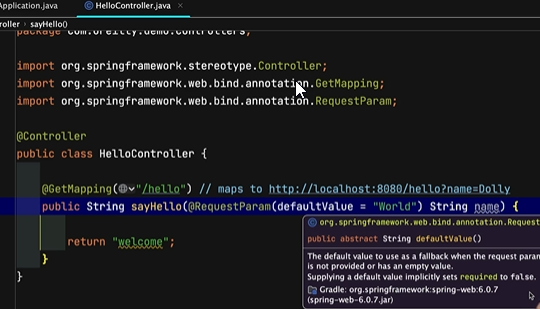


A computer screen with text and images

Description automatically generated

Need a way of connecting a particular URL(HTTP request) to a method.





Model: Basically a Map, collection of keys and values, Spring will take it, add it to the current HTTP request and forward it to some view.

When url goes to /hello after the default localhost:8080, then it should invoke the sayHello function.

Name is an optional argument, since default is given.

A screen shot of a computer program

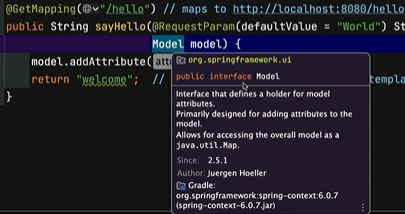
Description automatically generated

A computer screen shot of a program

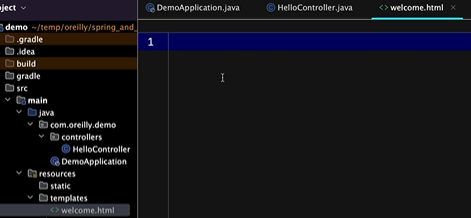
Description automatically generated

A screen shot of a computer screen

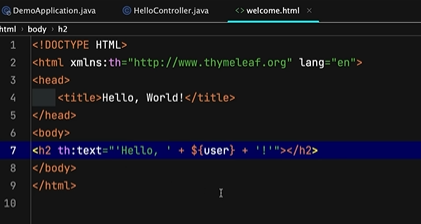
Description automatically generated



Model is an interface, yet we are calling addAttribute, so there must be some class which implements it.



Add the file in templates folder.



The text here, comes from Thymeleaf.

User variable here, is the exact same thing as the user variable inside the model.



Now in Controller, welcome no longer shows a warning.



A computer screen shot of a program

Description automatically generated

A black background with white text

Description automatically generated

A screenshot of a phone

Description automatically generated

A screen shot of a computer

Description automatically generated

Thymeleaf syntax gone. Template Engine.

A screenshot of a computer

Description automatically generated



Another way to do this.

A screen shot of a computer

Description automatically generated

A black background with white text

Description automatically generated



The one that is more size, 19M, is our deployable .jar file.





Another way to run the same app.





Special command



But if we just go to localhost: 8080, we get a error message.

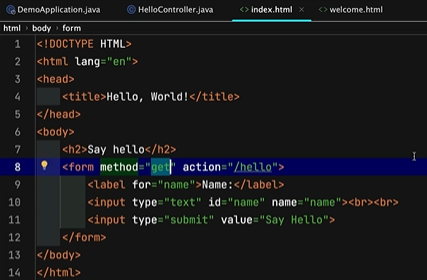


A screenshot of a computer

Description automatically generated

We create the index.html page on static folder which isn’t going to be changed on a request to request basis.

In templates, it might be different on every single request, might be different name each time.



Form, GET request

A screenshot of a computer

Description automatically generated

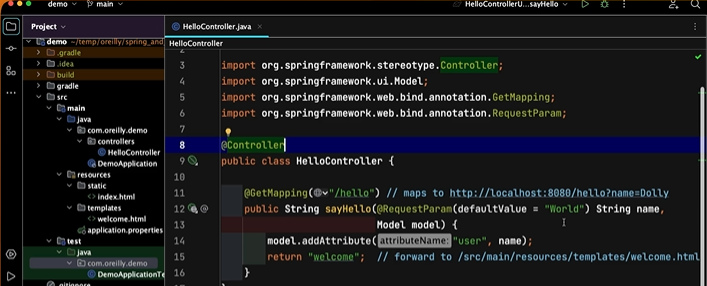
A screenshot of a computer

Description automatically generated

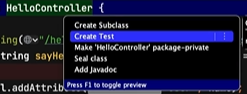
A screenshot of a phone

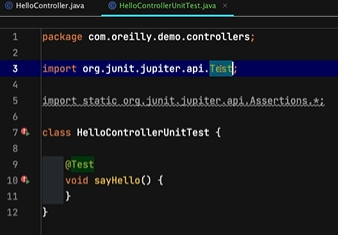
Description automatically generated



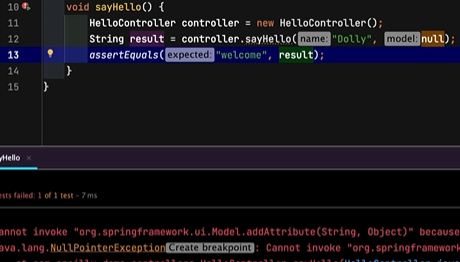


Alt+Enter: Create a Test





Same Package where the controller exists.



Null Pointer Exception

AI suggested code.



BindingAwareModelMap() implements the Model

Check returned and expected values.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

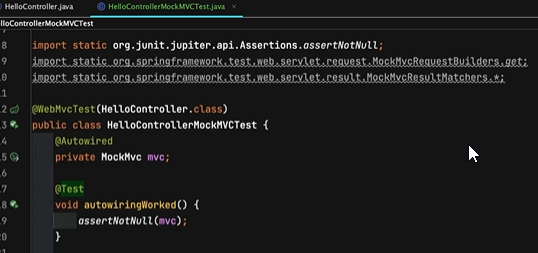
Description automatically generated

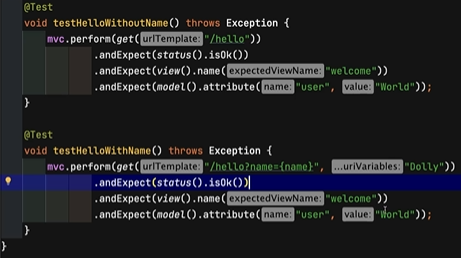


A screenshot of a phone

Description automatically generated

In the same Controller package.





A close up of a computer screen

Description automatically generated

Put together a trivial, restful web service, only going to respond to GET requests, but it will give us back an object and we can serialize that object to JSON data, JavaScript Object Notation data.

A screen shot of a computer

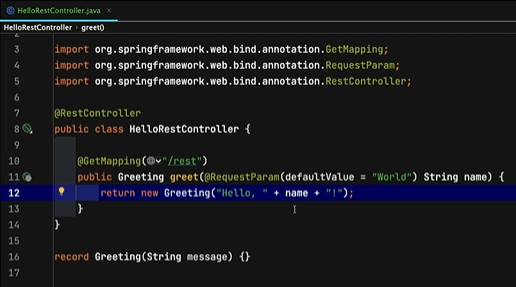
Description automatically generated

Controller package, Hello REST Controller

With regular Controller, we redirected it or forwarded it to a view, with REST controller, we will return a object, which will automatically get serialized into JSON Data and dumped into the output stream automatically.

Record: from java 17, data holder, allows us to put data in an object that we normally call a POJO, but it has got a ton of auto-generated code in it.

Immutable: Once you have instantiated a record, you can never change a message.



A computer screen with a cursor

Description automatically generated

A red and white sign with white text

Description automatically generated

