

# Hello SpringBoot - ein erstes Projekt

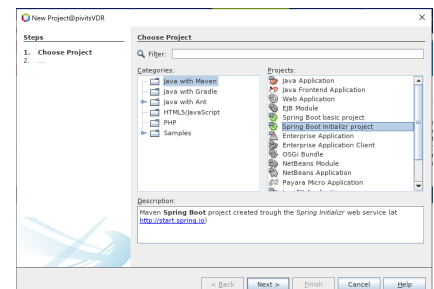
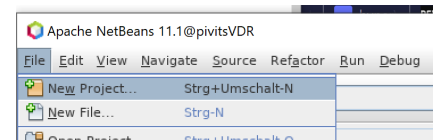
```
$ curl localhost:8085
Hello World!
```

Mit einem installierten SpringBoot-Framework ein "HelloWorld"-Projekt starten.

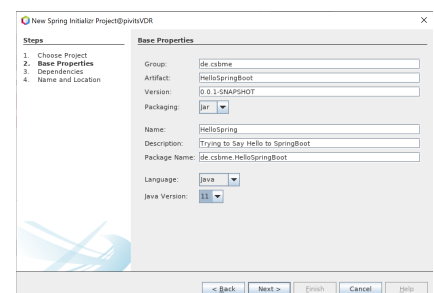
## 1 Ein erstes Projekt erstellen

Nach der Installation von *NB SpringBoot* findet sich unter File / New Project eine neue Rubrik

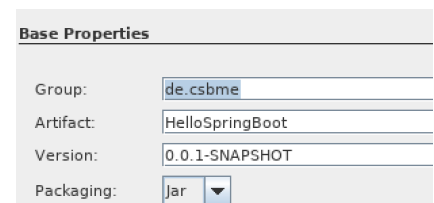
"Java with Maven" / "Spring Boot Initializr Project"



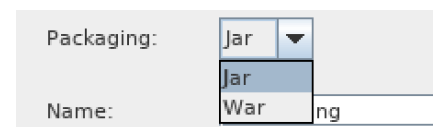
Im Folgenden werden müssen eine Reihe von Projektdaten angegeben werden. Wer nicht Netbeans nutzt kann sein Projekt auch unter <https://start.spring.io/> mit einer identischen Maske erstellen.



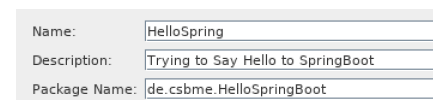
Group:  
Artifact:  
Version:



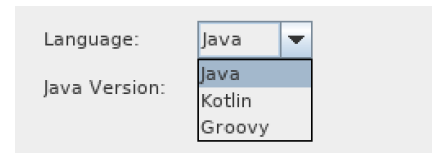
JAR: Ein mit Maven oder per Konsole ausführbares JAR-Archiv (auswählen)  
WAR: Ein Java EE-Container, der z.B. mit Tomcat genutzt wird  
Hintergrund z.B.: <https://www.baeldung.com/java-jar-war-packaging>



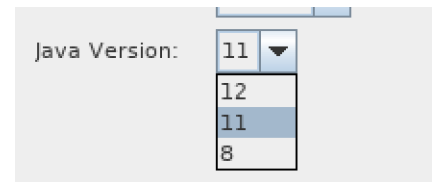
Name und Package Name sollten sich automatisch angepasst haben - eine Beschreibung kann noch ergänzt werden.



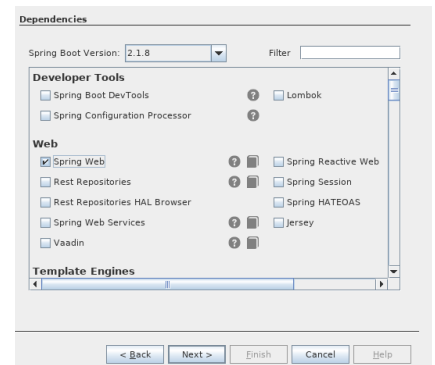
Es stehen prinzipiell verschiedene JVM-Sprachen zur Auswahl



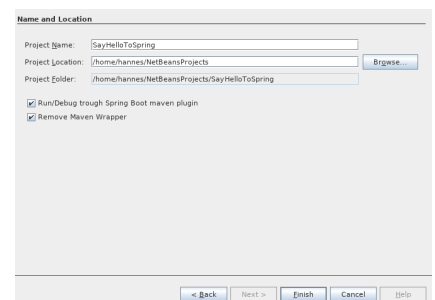
Theoretisch ist SpringBoot 2 mit Java 8 möglich - sinnvoll ist es jetzt die aktuelle JDK11 LTS zu verwenden.



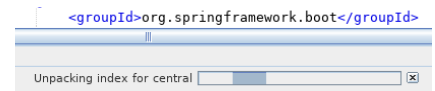
Das Startprojekt soll als einzige Abhängigkeit "Spring Web" erhalten.



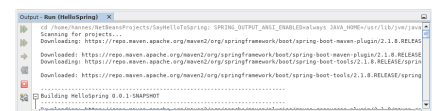
Hier einfach weiterklicken oder Namen anpassen...



Es müssen erst allerlei Abhängigkeiten geladen werden, was man unten am Progressbar sieht



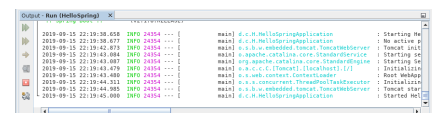
Wenn das Projekt jetzt startet holt er die Dateien...



Zeigt ein bisschen ASCII-Art...

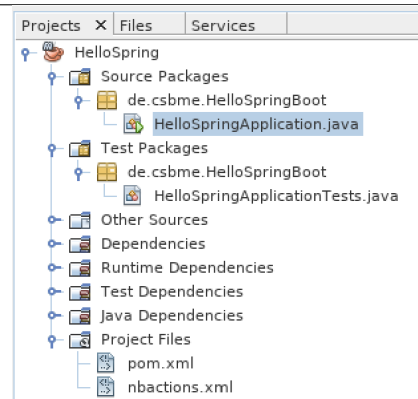
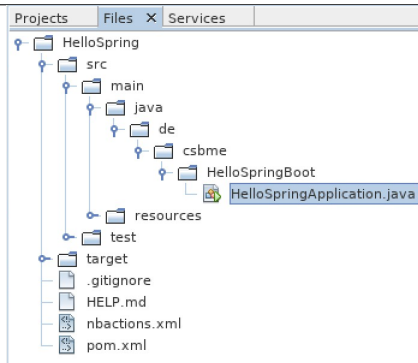


Und ist fertig...



## 2 Was wurde erstellt?

Im Wesentlichen wurden zwei Dateien erstellt: eine Application-Class (HelloSpringApplication.java) und eine Konfigurations-Datei (pom.xml). In Netbeans kann man entweder über den Projects oder den Files-Tab navigieren, um sie zu finden:



Die erstellten Dateien sehen etwa so aus:

```
1 package de.csbme.HelloSpringBoot;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class HelloSpringApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(HelloSpringApplication.class, args);
11     }
12
13 }
```

Das Programm ist im Prinzip startbar - bleibt aber ohne jede Funktion. Für ein erstes "Hello World!" fehlen noch drei Anpassungen:

Ein Import, der die nötigen Bibliotheken lädt:

```
14 import org.springframework.web.bind.annotation.*;
```

Die Annotation `text@RestController`, die dem Framework mitteilt, dass diese Klasse regelt, wie auf HTTP-Request reagiert werden soll:

```
15 @RestController
16 @SpringBootApplication
17 public class HelloSpringApplication {
```

Und eine Methode, die das Verhalten für den Fall festlegt, dass die Ressource "/" aufgerufen wird (also "localhost:8080/")

```
18 @RequestMapping("/")
19 String home() {
20     return "Hello World! \n";
21 }
```

Ein HTTP-GET-Request auf der Ressource "localhost:8080/" sollte jetzt ein "Hello World!" als Antwort erhalten. In einem Linux/MacOSX-Terminal mit installiertem `curl` getestet:

```
22 $ curl localhost:8080
23 Hello World!
```

Oder unter Windows über die PowerShell (curl ist hier ein Alias des CmdLets Invoke-WebRequest)

```
24 > Invoke-WebRequest http://localhost:8080/
25
26
27 StatusCode      : 200
28 StatusDescription :
29 Content         : Hello World!
```

```
30
31 RawContent      : HTTP/1.1 200
32 Content-Length: 14
33 Content-Type: text/plain; charset=UTF-8
34 Date: Sun, 22 Sep 2019 16:14:40 GMT
35
36 Hello World!
37
38 Forms           : {}
39 Headers         : {[Content-Length, 14], [Content-Type, text/plain; charset=UTF-8], [Date,
40   Sun, 22 Sep 2019 16:14:40
41   GMT]}
42 Images          : {}
43 InputFields     : {}
44 Links           : {}
45 ParsedHtml      : mshtml.HTMLDocumentClass
46 RawContentLength : 14
```

Sofern der Port 8080 bereits für andere Dienste benötigt wird lässt sich dieser Default-Port überschreiben, z.B. in der Datei `src/main/resources/application.properties`

```
46 server.port=8085
```

Check:

```
47 $ curl localhost:8085
48 Hello World!
```

```
49 package de.csbme.HelloSpringBoot;
50
51 import org.junit.Test;
52 import org.junit.runner.RunWith;
53 import org.springframework.boot.test.context.SpringBootTest;
54 import org.springframework.test.context.junit4.SpringRunner;
55
56 @RunWith(SpringRunner.class)
57 @SpringBootTest
58 public class HelloSpringApplicationTests {
59
60     @Test
61     public void contextLoads() {
62
63     }
64 }
65
66 }
67
68 package de.csbme.HelloSpringBoot;
69
70 import org.junit.Test;
71 import org.junit.runner.RunWith;
72 import org.springframework.boot.test.context.SpringBootTest;
73 import org.springframework.test.context.junit4.SpringRunner;
74
75 import org.springframework.beans.factory.annotation.Autowired;
76 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
77 import org.springframework.http.MediaType;
78 import org.springframework.test.web.servlet.MockMvc;
79 import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
80
81 @AutoConfigureMockMvc
82 @RunWith(SpringRunner.class)
83 @SpringBootTest
84 public class HelloSpringApplicationTests {
```

```
84
85     @Autowired
86     private MockMvc mvc;
87
88     @Test
89     public void contextLoads() {
90         mvc.perform(MockMvcRequestBuilders.get("/"))
91             .andExpect(status().isOk())
92             .andExpect(content().string("Hello World! \n"));
93     }
94
95 }
96
97 }
```

---

```
98 <?xml version="1.0" encoding="UTF-8"?>
99 <project xmlns="http://maven.apache.org/POM/4.0.0"
100     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
101     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
102         https://maven.apache.org/xsd/maven-4.0.0.xsd">
103     <modelVersion>4.0.0</modelVersion>
104     <parent>
105         <groupId>org.springframework.boot</groupId>
106         <artifactId>spring-boot-starter-parent</artifactId>
107         <version>2.1.8.RELEASE</version>
108         <relativePath/> <!-- lookup parent from repository -->
109     </parent>
110     <groupId>de.csbme</groupId>
111     <artifactId>HelloSpringBoot</artifactId>
112     <version>0.0.1-SNAPSHOT</version>
113     <name>HelloSpring</name>
114     <description>Trying to Say Hello to SpringBoot</description>
115
116     <properties>
117         <java.version>11</java.version>
118     </properties>
119
120     <dependencies>
121         <dependency>
122             <groupId>org.springframework.boot</groupId>
123             <artifactId>spring-boot-starter-web</artifactId>
124         </dependency>
125
126         <dependency>
127             <groupId>org.springframework.boot</groupId>
128             <artifactId>spring-boot-starter-test</artifactId>
129             <scope>test</scope>
130         </dependency>
131     </dependencies>
132
133     <build>
134         <plugins>
135             <plugin>
136                 <groupId>org.springframework.boot</groupId>
137                 <artifactId>spring-boot-maven-plugin</artifactId>
138             </plugin>
139         </plugins>
140     </build>
141 </project>
```

---