Stack Overflow

1. About
2. Products
3. OverflowAI

1. Stack Overflow Public questions & answers
2. Stack Overflow for Teams Where developers & technologists share private knowledge with coworkers
3. Talent Build your employer brand
4. Advertising Reach developers & technologists worldwide
5. Labs The future of collective knowledge sharing
6. About the company

Loading…

current community

Stack Overflow

help chat

- Meta Stack Overflow

your communities

Sign up or log in to customize your list.

more stack exchange communities

company blog

3. Log in
4. Sign up

2024 Developer survey is here and we would like to hear from you!

Home
Questions
Tags
Users
Companies
Labs
Jobs
New
Discussions
Collectives

Communities for your favorite technologies. Explore all Collectives

Teams

Ask questions, find answers and collaborate at work with Stack Overflow for Teams.

Explore Teams Create a free Team

Teams

Ask questions, find answers and collaborate at work with Stack Overflow for Teams. Explore Teams

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

Learn more about Collectives

Teams

Q&A for work

Connect and share knowledge within a single location that is structured and easy to search.

Learn more about Teams

How to configure port for a Spring Boot application

Ask Question

Asked 10 years, 4 months ago

Modified 1 month ago

Viewed 1.1m times

1075

How do I configure the TCP/IP port listened on by a Spring Boot application, so it does not use the default port of 8080.

- java
- spring
- spring-boot
- server
- port

Share

Follow

edited Mar 20, 2019 at 7:32

Tung

1,631 44 gold badges 17 17 silver badges 33 33 bronze badges

asked Jan 13, 2014 at 2:59

Paul Verest Paul Verest

62.4k 52 52 gold badges 219 219 silver badges 345 345 bronze badges

4

> 12
>
> If someone interested, here is shown how to have multiple ports - stackoverflow.com/questions/36357135/…
>
> – Betlista
>
> Feb 8, 2018 at 13:21

> if you use "yml" file for configuration then you can use this server: port: 8081 Also annotate you main class as "@SpringBootApplication" and remove @ EnableAutoConfiguration
>
> – Keaz
>
> Jun 23, 2018 at 7:52

> your project [application.properties] for add the server.port=8080
>
> – Lahiru Samishka
>
> Oct 8, 2019 at 11:07

> set `server.port=8080` in application properties. this configuration is in `ServerProperties.class` class under `org.springframework.boot.autoconfigure.web.`
>
> – Atif
>
> Mar 11, 2020 at 10:12

Add a comment |

65 Answers

Sorted by: Reset to default

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

1

2 3 Next

1581

As said in docs either set `server.port` as system property using command line option to jvm `-Dserver.port=8090` or add `application.properties` in `/src/main/resources/` with

```
server.port=8090
```

For a random port use:

```
server.port=0
```

Similarly add `application.yml` in `/src/main/resources/` with:

```
server:
 port: 8090
```

17

77

When random port is used, port info can get with `@Value("${local.server.port}")`

– azizunsal
Jul 23, 2015 at 12:46

52

Actually command line option is --server.port=8090 not -Dserver.port=8090. docs.spring.io/spring-boot/docs/current/reference/html/…

– alpert
Aug 19, 2015 at 6:39

1

As a compliment to this answer: According to the spring docs there are other paths you can put `application.properties` on. In my case that helped a lot.

– sargas
Oct 2, 2015 at 19:37

17

-Dserver.port=XXXX did not work for me. I used OS environment variable mode: `$ SERVER_PORT=8090 java -jar <path/to/my/jar>`

– Soumya Kanti
Oct 8, 2015 at 7:38

17

Both (1) `java -Dserver.port=XXXX -jar <path/to/my/jar>` and (2) `java -jar <path/to/my/jar> --server.port=YYYY` works. The first command defines `server.port` system property, and the second command pass the property through the command line arguments (`String... args` in the `main` method). Moreover, if you run with `java -Dserver.port=XXXX -jar <path/to/my/jar> --server.port=YYYY`, YYYY takes precedence over XXXX, this is why Spring Boot Externalized Configuration is so charming.

– tan9
Nov 29, 2016 at 16:28

| Show **12** more comments

274

There are two main ways to change the port in the Embedded Tomcat in a Spring Boot Application.

Modify application.properties

First you can try the application.properties file in the /resources folder:

```
server.port = 8090
```

Modify a VM option

The second way, if you want to avoid modifying any files and checking in something that you only need on your local, you can use a vm arg:

Go to Run -> Edit Configurations -> VM options

```
-Dserver.port=8090
```

Additionally, if you need more information you can view the following blog post here: Changing the port on a Spring Boot Application

1

In STS 4 it is at run -> run configurations -> main, then scroll down to Table with Parameter Name and Value

[Add a comment](#) |

214

Since Spring Boot provides [various configuration externalization mechanism](#) (through various `PropertySource` implementations and/or processors wired into `Environment` object in order), you can set any property **outside** of your jar archive through following methods:

Pass property through command line argument as **application argument**

```
java -jar <path/to/my/jar> --server.port=7788
```

From property in `SPRING_APPLICATION_JSON` (Spring Boot 1.3.0+)

Define environment variable in U*IX shell:

```
SPRING_APPLICATION_JSON='{"server.port":7788}' java -jar <path/to/my/jar>
```

By using Java system property:

```
java -Dspring.application.json='{"server.port":7788}' -jar <path/to/my/jar>
```

Pass through command line argument:

```
java -jar <path/to/my/jar> --spring.application.json='{"server.port":7788}'
```

Define **JVM system property**

```
java -Dserver.port=7788 -jar <path/to/my/jar>
```

Define **OS environment variable**

U*IX Shell

```
SERVER_PORT=7788 java -jar <path/to/my/jar>
```

Windows

```
SET SERVER_PORT=7788
java -jar <path/to/my/jar>
```

Place property in `./config/application.properties` **configuration file**

```
server.port=7788
```

and run:

```
 java -jar <path/to/my/jar>
```

Place property in `./config/application.yaml`

```
server:
   port: 7788
```

and run:

```
 java -jar <path/to/my/jar>
```

Place property in `./application.properties`

```
server.port=7788
```

and run:

```
 java -jar <path/to/my/jar>
```

Place property in `./application.yaml`

```
server:
   port: 7788
```

and run:

```
java -jar <path/to/my/jar>
```

You can combine above methods all together, and the former configuration in the list take precedence over the latter one.

For example:

```
SERVER_PORT=2266 java -Dserver.port=5566 -jar <path/to/my/jar> --server.port=7788
```

The server will start and listen on port 7788.

This is very useful providing default properties in PropertySources with lower precedence (and usually packaged in the archive or coded in the source), and then override it in the runtime environment. And it is the design philosophy of Spring Boot:

Be opinionated out of the box, but get out of the way quickly as requirements start to diverge from the defaults.

`SERVER_NAME` to `server.name` conversion was done by [Relaxed Binding](#).

[Share](#)
Follow
[edited Dec 4, 2017 at 15:53](#)
answered Nov 29, 2016 at 17:39
[tan9](#)tan9
3,57022 gold badges1919 silver badges2121 bronze badges
0
[Add a comment](#) |
141

Also, you can configure the port programmatically.

For Spring Boot 2.x.x:

```
@Configuration
public class CustomContainer implements WebServerFactoryCustomizer<ConfigurableServletWebServerFactory> {
 public void customize(ConfigurableServletWebServerFactory factory){
   factory.setPort(8042);
 }
}
```

For older versions:

```
@Configuration
public class ServletConfig {
    @Bean
    public EmbeddedServletContainerCustomizer containerCustomizer() {
        return (container -> {
            container.setPort(8012);
        });
    }
}
```

[Share](#)
Follow
[edited Nov 30, 2020 at 11:48](#)
[Positive Navid](#)
2,67122 gold badges2929 silver badges4242 bronze badges
answered Nov 3, 2015 at 18:29
[makerj](#)makerj
2,22922 gold badges2020 silver badges2727 bronze badges
8

  2

  This is working and very useful, when you have port in your own config file and want to set it during runtime.

  – [Xdg](#)

  Nov 8, 2015 at 9:48

  6

This was helpful when i needed to deploy an application to a AWS Elastic Beanstalk service, to get the port from an environment variable.

– [Martin Hansen](#)

Nov 19, 2015 at 12:51

This is super useful when all you want is a self-contained unit or integration test, +1.

– [Priidu Neemre](#)

Apr 28, 2016 at 10:34

Very useful when the env variable for port is already defined under a different name.

– [higuaro](#)

Aug 18, 2016 at 8:27

2

Is'nt it the `@Configuration` instead of `@Controller`? Please update if so.

– [Lucky](#)

Sep 13, 2016 at 11:52

| [Show **3** more comments](#)

113

If you would like to run it locally, use this -

```
mvn spring-boot:run -Drun.jvmArguments='-Dserver.port=8085'
```

As of **Spring Boot 2.0**, here's the command that works:

```
mvn spring-boot:run -Dspring-boot.run.arguments=--server.port=8085
```

clues were at:

* [Maven spring boot run debug with arguments](#)

[Share](#)

Follow

[edited Jul 16, 2023 at 16:39](#)

[Manuel Jordan](#)

15.9k2323 gold badges100100 silver badges171171 bronze badges

answered Apr 15, 2016 at 23:00

[itwarilal](#)itwarilal

1,37411 gold badge1010 silver badges1717 bronze badges

3

5

Starting from Spring Boot 2, you should use `spring-boot.run.jvmArguments`.

– [mapm](#)

Apr 12, 2018 at 2:57

1

`-Dspring-boot.run.arguments` works with Spring Boot 3

– [Marco Sulla](#)

Apr 3, 2023 at 19:13

For the first command is valid wihout `''` too - therefore `mvn spring-boot:run -Drun.jvmArguments=-Dserver.port=8085` works. Tested on SB 3.1.1

– [Manuel Jordan](#)

Jul 16, 2023 at 16:41

[Add a comment](#) |

99

You can set port in java code:

```
HashMap<String, Object> props = new HashMap<>();
props.put("server.port", 9999);

new SpringApplicationBuilder()
   .sources(SampleController.class)
   .properties(props)
   .run(args);
```

Or in application.yml:

```
server:
    port: 9999
```

Or in application.properties:

```
server.port=9999
```

Or as a command line parameter:

```
-Dserver.port=9999
```

2

Using the HashMap will work only if no port is set in applications.properties or .yml.

– Milgo
Apr 25, 2019 at 11:28

I have a 2nd "application" class as a utility, and I wanted to only set a different port in that one. Therefore all the config file based approaches were not helpful for me. Close to giving up I found your programatic approach. Thank you!

– Thomas Schütt
Oct 8, 2021 at 12:43

Add a comment |

63

In case you are using `application.yml` add the Following lines to it

```
server:
    port: 9000
```

and of course 0 for random port.

2

1

this didn't seem to work. I used server.port in the application.yml and it worked

– yathirigan
Mar 5, 2015 at 13:11

Also, make sure to keep a space between the : and the value.

– Upulie Handalage
Jul 20, 2022 at 16:28

Add a comment |

54

As explained in Spring documentation, there are several ways to do that:

Either you set the port in the command line (for example 8888)

```
-Dserver.port=8888 or --server.port=8888
```

Example : `java -jar -Dserver.port=8888 test.jar`

Or you set the port in the application.properties

```
server.port=${port:4588}
```

or (in application.yml with yaml syntax)

```
server:
  port: ${port:4588}
```

If the port passed by -Dport (or -Dserver.port) is set in command line then this port will be taken into account. If not, then the port will be 4588 by default.

If you want to enforce the port in properties file whatever the environment variable, you just have to write:

```
server.port=8888
```

Share

Follow

edited Nov 11, 2017 at 22:56

answered Jan 21, 2017 at 13:21

OlivierTerrienOlivierTerrien

2,55111 gold badge2020 silver badges3232 bronze badges

Add a comment |

43

Include below property in `application.properties`

```
server.port=8080
```

Share

Follow

edited Aug 8, 2018 at 11:23

Derrick

3,91555 gold badges3737 silver badges5252 bronze badges

answered Oct 18, 2016 at 18:22

ChandramouliChandramouli

57244 silver badges66 bronze badges

1

This doesn't work. What application.properties? Which properties can overwrite this one? How can I be sure Spring is picking it up?

– Philip Rego

Dec 10, 2021 at 16:24

Add a comment |

31

When you need a programatically way of doing it, you can set it during startup:

```
System.getProperties().put( "server.port", 80 );
SpringApplication.run(App.class, args);
```

This might help for things like environment dependent port. Have a nice day

Share

Follow

answered Apr 3, 2016 at 16:21

Luis MauricioLuis Mauricio

41655 silver badges88 bronze badges

2

2

`System.setProperty("server.port", 80);` is another way to achieve the same.

– hd1

Feb 11, 2017 at 20:34

@hd1, I added our answers to the main answer, check it out and modify as you se fit please

– Luis Mauricio

Dec 30, 2019 at 14:40

Add a comment |

21

if you are using gradle as the build tool, you can set the server port in your **application.yml** file as:

```
server:
 port: 8291
```

If you are using maven then the port can be set in your **application.properties** file as:

```
server.port: 8291
```

[Share](#)
Follow
[edited Oct 24, 2019 at 9:37](#)
[Gaurav Jeswani](#)
4,49266 gold badges2929 silver badges4949 bronze badges
answered Oct 24, 2019 at 9:20
[VSharma](#)VSharma
48344 silver badges88 bronze badges

2

> for application.properties its `server.port = 8291`
>
> – [iQuestProgrammer](#)
> Dec 5, 2019 at 9:52

> What do Maven and Gradle have to do with whether you use a properties or YAML file? The build process (Maven) is completely disparate from the application framework (Spring Boot).
>
> – [SeverityOne](#)
> Jun 3, 2021 at 9:01

[Add a comment](#) |

20

To extend other answers:

There is a section in the docs for testing which explains how to configure the port on integration tests:

- [41.3 Testing Spring Boot applications](#)
- [41.3.3 Working with random ports](#)

At integration tests, the port configuration is made using the annotation `@SpringBootTest` and the `webEnvironment` values.

Random port:

```
@SpringBootTest(webEnvironment=WebEnvironment.RANDOM_PORT)
```

You can inject the value using `@LocalServerPort` which is the same as `@Value("${local.server.port}")`.

- Example:

Random port test configuration:

```
@RunWith(SpringRunner.class
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
public class ExampleTest {
  ...
  @LocalServerPort //to inject port value
  int port;
}
```

Defined port:

```
@SpringBootTest(webEnvironment=WebEnvironment.DEFINED_PORT)
```

It takes the value from `server.port` if is defined.

- If is defined using `@TestPropertySource(properties = "server.port=9192")`, it overrides other defined values.
- If not, it takes the value from `src/test/resources/application.properties` (if exists).
- And finally, if it is not defined it starts with the default `8080`.

Example:

Defined port test configuration:

```
@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.DEFINED_PORT)
```

```
@TestPropertySource(properties = "server.port=9192")
public class DemoApplicationTests {

    @Test
    public void contextLoads() {
    }

}
```

Share
Follow
edited Aug 6, 2017 at 13:04
andi
99122 gold badges1111 silver badges2121 bronze badges
answered May 29, 2017 at 18:40
PauPau
15.6k1616 gold badges6868 silver badges9494 bronze badges
Add a comment |

19

You can specify port by overriding `EmbeddedServletContainerFactory` bean within your configuration (java based or xml). There you can specify port for used embedded servlet container. Please, see Spring Boot - Core "Embedded Servlet Container Support" paragraph and example there. Hope this helps.

Share
Follow
edited Jun 15, 2014 at 8:39
Will
6,61133 gold badges3131 silver badges4242 bronze badges
answered Jan 13, 2014 at 10:20
nndrunndru
2,0872222 silver badges1616 bronze badges

2

> 1
>
> Direct link: docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/…
>
> – Erik Martino
> Nov 13, 2014 at 10:24
>
> Here's a current link to the programmatic override section: docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/…
>
> – jocull
> Jun 14, 2018 at 15:33

Add a comment |

19

In `application.properties` file present in resources:

```
server.port=8082
```

Share
Follow
edited Apr 24, 2017 at 8:15
FelixSFD
6,0721010 gold badges4444 silver badges121121 bronze badges
answered Apr 24, 2017 at 7:57
Amit GujarathiAmit Gujarathi
1,10011 gold badge1212 silver badges2525 bronze badges
Add a comment |

19

There are three ways to do it depending on the application configuration file you are using

a) If you are using **application.properties** file set

```
server.port = 8090
```

b) If you are using **application.yml** file set server port property in YAML format as given below

```
server:
    port: 8090
```

c) You can also Set the property as the System property in the **main method**

```
System.setProperty("server.port","8090");
```

15

There are many other stuffs you can alter in server configuration by changing application.properties. Like session time out, address and port etc. Refer below post

ref: http://docs.spring.io/spring-boot/docs/1.4.x/reference/html/common-application-properties.html

I used few of them as below.

```
server.session.timeout=1
server.port = 3029
server.address= deepesh
```

15

Add this in your `application.properties` file

```
server.port= 8080
```

2

2

Welcome to SO :-) please look at How to Answer

Why to add same asnwer one year later?!? and `server.port 8080` is wrong syntax for Java property file...

14

As everyone said, you can specify in application.properties
server.port = 9000 (could be any other value)

If you are using spring actuator in your project, by default it points to
8080, and if you want to change it, then in application.properties mention
management.port = 9001 (could be any other value)

14

In the `application.properties` file, add this line:

```
server.port = 65535
```

where to place that fie:

> 24.3 Application Property Files
>
> SpringApplication loads properties from application.properties files in the following locations and adds them to the Spring Environment:
>
> ```
> A /config subdirectory of the current directory
> The current directory
> A classpath /config package
> The classpath root
> ```
>
> The list is ordered by precedence (properties defined in locations higher in the list override those defined in lower locations).

In my case I put it in the directory where the `jar` file stands.

From:

https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#boot-features-external-config-application-property-files

14

By default spring boot app start with embedded tomcat server start at default port 8080. spring provides you with following different customization you can choose one of them.

> **NOTE** – you can use **server.port=0** spring boot will find any unassigned http random port for us.

1) application.properties

```
server.port=2020
```

2) application.yml

```
server:
   port : 2020
```

3) Change the server port programatically

3.1) By implementing WebServerFactoryCustomizer interface - Spring 2.x

```
@Component
public class MyTomcatWebServerCustomizer implements WebServerFactoryCustomizer<TomcatServletWebServerFactory> {

   @Override
   public void customize(TomcatServletWebServerFactory factory) {
      // customize the factory here
      factory.setPort(2020);
   }
}
```

3.2) By Implementing EmbeddedServletContainerCustomizer interface - Spring 1.x

```
@Component
public class CustomizationBean implements EmbeddedServletContainerCustomizer {

    @Override
    public void customize(ConfigurableEmbeddedServletContainer container) {
        // customize here
        container.setPort(2020);
    }
}
```

4) By using command line option

```
 java -jar spring-boot-app.jar -Dserver.port=2020
```

Share
Follow
edited Apr 26, 2019 at 14:25
answered Apr 26, 2019 at 14:09
anandchauguleanandchaugule
1,0411313 silver badges2121 bronze badges
Add a comment |

12

Indeed, the easiest way is to set the *server.port* property.

If you are using STS as IDE, from version 3.6.7 you actually have *Spring Properties Editor* for opening the properties file.

This editor provides autocomplete for all Spring Boot properties. If you write *port* and hit CTRL + SPACE, *server.port* will be the first option.

Share
Follow
answered Dec 15, 2015 at 9:11
Adrian CosmaAdrian Cosma
48688 silver badges1010 bronze badges

1

    1
    Or Also if you are using Intellij IDEA the autocomplete works too. ;)
    – Lucky
    Sep 13, 2016 at 11:53
Add a comment |

12

By default, spring-web module provides an embedded tomcat server that is running under the port number 8080. If you need to change the port number of the
application then go to `application.properties` file and configure the port number by using `server.port` property.

```
   server.port= 9876
```

then your application is running under the port 9876.

Share
Follow
edited Oct 28, 2020 at 11:34
answered May 14, 2020 at 9:46
Nafaz M N MNafaz M N M
1,63844 gold badges2828 silver badges4343 bronze badges
Add a comment |

11

Hope this one help

```
application.properties=>
```

```
server.port=8090
```

```
application.yml=>
```

```
server
```

```
port:8090
```

1

application.yml => server: port: 8090

– [Vijay Mohan](#)

May 23, 2018 at 17:36

Add a comment |

11

Using property server.port=8080 for instance like mentioned in other answers is definitely a way to go. Just wanted to mention that you could also expose an environment property:

```
SERVER_PORT=8080
```

Since spring boot is able to replace "." for "_" and lower to UPPER case for environment variables in recent versions. This is specially useful in containers where all you gotta do is define that environment variable without adding/editing `application.properties` or passing system properties through command line (i.e `-Dserver.port=$PORT`)

1

`SERVER_PORT=8081 mvn spring-boot:run`

– [Brent Bradburn](#)

Sep 17, 2018 at 4:11

Add a comment |

9

You can add the port in below methods.

Run -> Configurations section

In `application.xml` add `server.port=XXXX`

1

1

Do you mean `application.yml` and what IDE are you using? Please be specific.

– [Lucky](#)

Sep 13, 2016 at 11:56

Add a comment |

9

Just have a `application.properties` in `src/main/resources` of the project and give there

```
server.port=****
```

where **** refers to the port number.

9

1.1 Update via a properties file.

/src/main/resources/application.properties

server.port=8888

Update via a yaml file.

```
    server:

     port: 8888
```

EmbeddedServletContainerCustomizer

```
@Component
public class CustomContainer implements EmbeddedServletContainerCustomizer {

    @Override
    public void customize(ConfigurableEmbeddedServletContainer container) {

        container.setPort(8888);

    }

}
```

9

Providing the port number in application.properties file will resolve the issue

```
 server.port = 8080
```

"port depends on your choice, where you want to host the application"

8

You can also use SERVER_PORT environment variable to configure Spring Boot port. Just set the environment variable and restart the app:

```
set SERVER_PORT=9999 // on windows machine
export SERVER_PORT=9999 // on linux
```

Please note that if you do not set those environment variables system wide, you should run the boot app on the same session.

[Share](#)
Follow
answered Apr 26, 2016 at 13:03
[Ali Dehghani](#)Ali Dehghani
47.4k1616 gold badges168168 silver badges152152 bronze badges
[Add a comment](#) |
8

Mostly springboot runs on `port:8080` because of embedded Tomcat. In some it may throw an error `port 8080 already in use`. To avoid this kind of issues we can config the server port.

Using application.properties

add `server.port=9898`

On runtime config

run your application with below arguments.

```
spring-boot:run -Drun.jvmArguments='-Dserver.port=8081'
```

[Share](#)
Follow
answered Aug 16, 2017 at 4:10
[Mohanraj](#)Mohanraj
40633 gold badges99 silver badges2222 bronze badges

2

> 1
>
> This answer is outdated. You need to use the following arguments `mvn spring-boot:run -Dspring-boot.run.jvmArguments='-Dserver.port=8081'`, see [stackoverflow.com/a/66370832/179014](#)
>
> – [asmaier](#)
> Nov 25, 2022 at 13:43
>
> I've confirmed it works in SB 3.1.1 - BTW is not necessary use `''`, therefore is `mvn spring-boot:run -Dspring-boot.run.jvmArguments=-Dserver.port=8081` works
>
> – [Manuel Jordan](#)
> Jul 16, 2023 at 16:24

[Add a comment](#) |
1

[Highly active question](#). Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.

Not the answer you're looking for? Browse other questions tagged

- **[java](#)**

- **[spring](#)**

- **[spring-boot](#)**

- **[server](#)**

- **[port](#)**

or **[ask your own question](#).**

- The Overflow Blog

- Featured on Meta

Linked

Related

Hot Network Questions

Subscribe to RSS

Question feed

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

- [Business](#)
- [API](#)
- [Data](#)
- [Blog](#)
- [Facebook](#)
- [Twitter](#)
- [LinkedIn](#)
- [Instagram](#)