**HowToDoInJava**

# Spring Boot – Configure Jetty Server

📅 Last Updated: October 25, 2019     👤 By: Lokesh Gupta     📁 Spring Boot     🏷️ Jetty

By default, Spring boot uses embedded tomcat server to run the application. At times, you may need to use jetty server in place of tomcat server. Spring Boot provides Tomcat and Jetty dependencies bundled together as separate starters to help make this process as easy as possible. You can use jetty with following simple steps.

## Add spring-boot-starter-jetty dependency

You will need to update `pom.xml` and add dependency for `spring-boot-starter-jetty`. Also, you will need to exclude default added `spring-boot-starter-tomcat` dependency.

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <exclusions>
        <exclusion>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

In gradle, able change can be achieved by this:

```gradle
configurations {
    compile.exclude module: "spring-boot-starter-tomcat"
}

dependencies {
    compile("org.springframework.boot:spring-boot-starter-web:2.0.0.BUILD-SNAPSHOT")
    compile("org.springframework.boot:spring-boot-starter-jetty:2.0.0.BUILD-SNAPSHOT")
}
```

## Configure Jetty Options

To override, default jetty runtime configuration – you can configure them in `application.properties` file.

```
application.properties
```

```
server.port=8080
server.servlet.context-path=/home

####Jetty specific properties########

server.jetty.acceptors= # Number of acceptor threads to use.
server.jetty.max-http-post-size=0 # Maximum size in bytes of the HTTP post or put content.
server.jetty.selectors= # Number of selector threads to use.
```

Also, you may configure these options programatically using `JettyEmbeddedServletContainerFactory` bean.

```java
@Bean
public JettyEmbeddedServletContainerFactory  jettyEmbeddedServletContainerFactory() {
    JettyEmbeddedServletContainerFactory jettyContainer =
      new JettyEmbeddedServletContainerFactory();

    jettyContainer.setPort(9000);
    jettyContainer.setContextPath("/home");
    return jettyContainer;
}
```

**Update for Spring boot 2.0.0.RELEASE**

Above code snippet was valid for spring boot spanshot version. After Spring boot 2.0.0.RELEASE is available, you shall be using ConfigurableServletWebServerFactory and JettyServletWebServerFactory classes.

```java
Create bean of type ConfigurableServletWebServerFactory

@Bean
public ConfigurableServletWebServerFactory webServerFactory()
{
  JettyServletWebServerFactory factory = new JettyServletWebServerFactory();
  factory.setPort(9000);
  factory.setContextPath("/myapp");
  factory.addErrorPages(new ErrorPage(HttpStatus.NOT_FOUND, "/notfound.html"));
  return factory;
}
```

Drop me your questions in comments section related to using jetty in stead of tomcat server in any spring boot application.

Happy Learning !!

Ref: Configure Jetty

## Was this post helpful?

Let us know if you liked the post. That's the only way we can improve.

> Yes

> No

# Recommended Reading:

1. [Spring boot change default port of embedded server](#)

2. [Spring boot embedded server logs](#)

3. [Spring Cloud Config Server with Git Integration](#)

4. [JavaMail API – Java program to send email – Gmail SMTP server example](#)

5. [HornetQ Stand Alone Server Example using Maven](#)

6. [Running multiple instances of Tomcat with single server installation](#)

7. [Tomcat – Architecture and server.xml configuration](#)

8. [Angular – Fake REST APIs using Mock Server](#)

9. [Configure Hibernate with Spring Boot](#)

0. [Configure In-memory DB to Unit Test Hibernate](#)

## Join 7000+ Awesome Developers

Get the latest updates from industry, awesome resources, blog updates and much more.

**Email Address**

Subscribe

*\* We do not spam !!*

## 6 thoughts on "Spring Boot – Configure Jetty Server"

**Rudolf Flütsch**

June 1, 2018 at 12:29 pm

I am looking for a way to configure Jetty only to listen on 127.0.0.1.
I found the option to be used within the "normal" jetty configuration, i.e. using jetty.http.host.

Unfortunately I have not found a way to do this using the spring configuration file or programatically – any idea how to do this ?

Reply

**Karl**

April 3, 2018 at 10:23 pm

Doesn't seem to work with version 2.0.0.RELEASE. Gives error:

Caused by: org.springframework.context.ApplicationContextException: Unable to start ServletWebServerApplicationContext due to missing ServletWebServerFactory bean.

Reply

**Lokesh Gupta**

April 4, 2018 at 12:14 pm

This works in spring boot 2.0.0.RELEASE.

```
@Bean
public ConfigurableServletWebServerFactory webServerFactory()
{
  JettyServletWebServerFactory factory = new JettyServletWebServerFactory();
  factory.setPort(9000);
  factory.setContextPath(&quot;/myapp&quot;);
  factory.addErrorPages(new ErrorPage(HttpStatus.NOT_FOUND, &quot;/notfound.html&quot;));
  return factory;
}
```

Reply

**vikas**

October 26, 2017 at 12:40 pm

I have a spring boot application (version 2.0) reactive web one , i by defaultt have the tomcat connected , what should i do if need to connect to tomcat and another server say jetty at the same time by modifying the application.properties , say like if i choose tomcat it should start up with tomcat and other way around .?

Reply

**Markus Schulte**

May 17, 2017 at 3:54 pm

I linked your example, how to configure Jetty programmatically, at

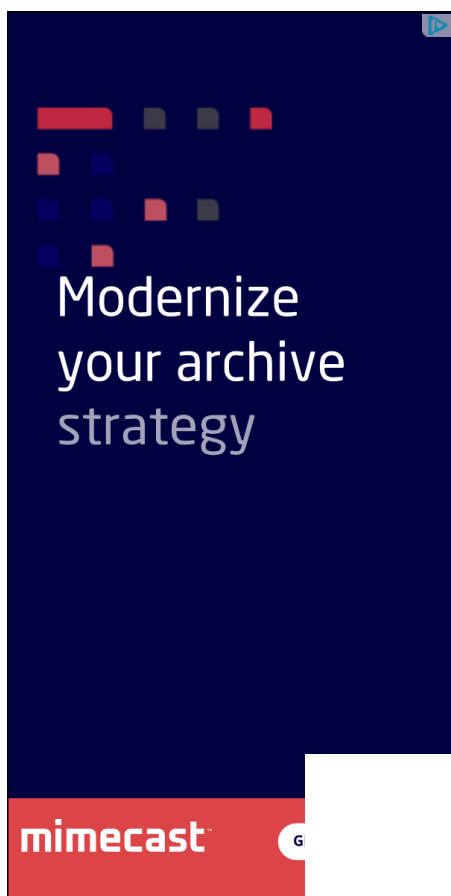https://stackoverflow.com/questions/20454441/how-to-configure-jetty-in-spring-boot-easily/44021864#44021864

Reply

**Lokesh Gupta**

May 17, 2017 at 5:55 pm

Thanks Markus !!

Reply

## Leave a Comment

Name *

Email *

Website

☐   Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment

Search …    🔍

## HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.
**Meta Links**

> About Me

> Contact Us

> Privacy policy

> Advertise

> Guest Posts

**Blogs**

REST API Tutorial

Copyright © 2022 · Hosted on Cloudways · Sitemap