

How to integrate Angular 4 with SpringBoot RestApi using SpringToolSuite

Angular 4 is the next version of Angular2 that comes with almost everything you need to build **Across All Platforms** frontend web or mobile apps with high speed & performance. So in the tutorial, **JavaSampleApproach** will guide you through the steps of integrating **Angular 4** with **SpringBoot Web App** and **SpringToolSuite** for development.

Related articles:

- [How to configure AngularJs with Spring MVC | SpringBoot](#)
- [How to integrate Http Angularjs with Spring MVC | Spring Boot](#)
- [How to setup Angular IDE with Spring Tool Suite](#)
- [How to use Angular Http Client to fetch Data from SpringBoot RestAPI – Angular 4](#)
- [How to use Angular HttpClient to POST, PUT, DELETE data on SpringBoot Rest APIs – Angular 4](#)
- [Angular 4 + Spring JPA + PostgreSQL example | Angular 4 Http Client – Spring Boot RestApi Server](#)
- [Angular 4 + Spring JPA + MySQL example | Angular 4 Http Client – Spring Boot RestApi Server](#)

Contents [\[hide\]](#)

[I. Technologies](#)

[II. Integrate Angular 4 with Spring Boot WebApp and SpringToolSuite](#)

- [1. Install Node.js for Angular](#)
- [2. Install Angular-CLI](#)
- [3. Setup SpringBoot Web server project](#)
- [4. Create Angular 4 client project](#)
- [5. Import Angular4 client project to SpringToolSuite](#)
- [6. Integrate SpringBoot server and Angular 4 client](#)
- [7. Deploy SpringBoot server with Angular4 client](#)

I. Technologies

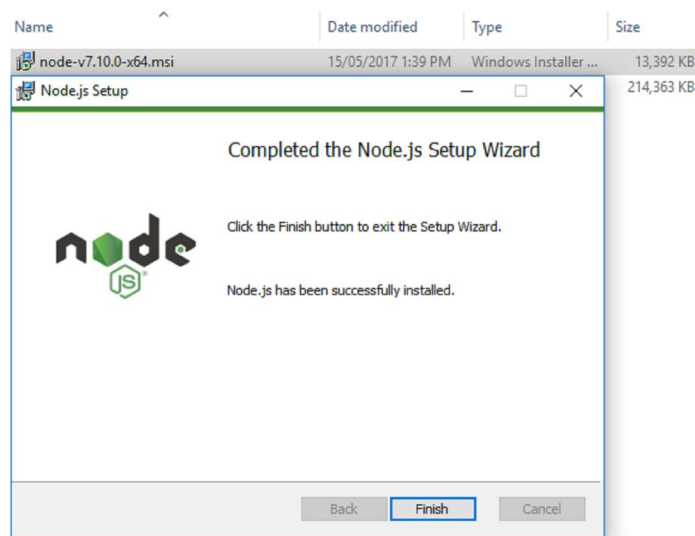
- Java 1.8
- Maven 3.3.9
- Spring Tool Suite - Version 3.8.1.RELEASE
- Spring Boot: RELEASE
- Angular 4
- Node.js

II. Integrate Angular 4 with Spring Boot WebApp and SpringToolSuite

1. Install Node.js for Angular

Firstly, checking whether or not you have Node.js installed, by command: `node -v` & `npm -v`. If the command goes unrecognized, we must install **Node.js**.

- Go to: nodejs.org. Download Node.js installer, the tutorial uses version: v7.10.0. We got a **node-v7.10.0-x64.msi** file, double click on it and follow the guides to setup Node.js, successfully result:



- Open **cmd**, checking **Node.js** by commandline: `node -v` & `npm -v`:

```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\User>node -v & npm -v
v7.10.0
4.2.0

C:\Users\User>
```

2. Install Angular-CLI

- Using commandline `npm install -g @angular/cli`:


```

C:\Users\User\workspace>ng new angular4-client
installing ng
  create .editorconfig
  create README.md
  create src\app\app.component.css
  create src\app\app.component.html
  create src\app\app.component.spec.ts
  create src\app\app.component.ts
  create src\app\app.module.ts
  create src\assets\.gitkeep
  create src\environments\environment.prod.ts
  create src\environments\environment.ts
  create src\favicon.ico
  create src\index.html
  create src\main.ts
  create src\polyfills.ts
  create src\styles.css
  create src\test.ts
  create src\tsconfig.app.json
  create src\tsconfig.spec.json
  create src\typings.d.ts
  create .angular-cli.json
  create e2e\app.e2e-spec.ts
  create e2e\app.po.ts
  create e2e\tsconfig.e2e.json
  create .gitignore
  create karma.conf.js
  create package.json
  create protractor.conf.js
  create tsconfig.json
  create tslint.json
Successfully initialized git.
Installing packages for tooling via npm.
Installed packages for tooling via npm.
Project 'angular4-client' successfully created.

```

To check angular version, go to **angular4-client** folder, type: `ng -v` :

```

C:\Users\User\workspace\angular4-client>ng -v

Angular CLI
@angular/cli: 1.0.3
node: 7.10.0
os: win32 x64
@angular/common: 4.1.2
@angular/compiler: 4.1.2
@angular/core: 4.1.2
@angular/forms: 4.1.2
@angular/http: 4.1.2
@angular/platform-browser: 4.1.2
@angular/platform-browser-dynamic: 4.1.2
@angular/router: 4.1.2
@angular/cli: 1.0.3
@angular/compiler-cli: 4.1.2

```

Start angular4-client project by cmd `npm start` , results:

```
C:\Users\User\workspace\angular4-client>npm start

> angular4-client@0.0.0 start C:\Users\User\workspace\angular4-client
> ng serve

** NG Live Development Server is listening on localhost:4200, open your browser
Hash: 4db290a2866c19c5938d
Time: 7895ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {4}
chunk    {1} main.bundle.js, main.bundle.js.map (main) 3.7 kB {3} [initial] [re
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 10.5 kB {4} [initia
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.4 MB [initial] [r
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [re
webpack: Compiled successfully.
```

← → ↻ 🏠 ⓘ localhost:4200

app works!

5. Import Angular4 client project to SpringToolSuite

Open **SpringToolSuite**, go to **Import** -> **General** -> **Projects from Folder or Archive**, press **Next**:

📁 Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source:	C:\Users\User\workspace\angular4-client\
type filter text	
Folder	Import as
<input checked="" type="checkbox"/> angular4-client	

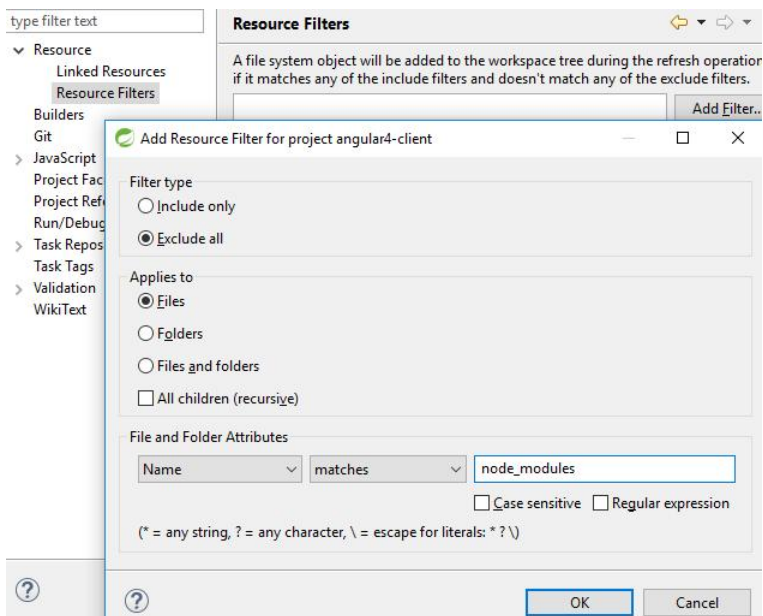
Press **Finish**, Angular4 client is imported:

```
angular4-client [angular4-client master]
├── e2e
├── node_modules
├── src
├── karma.conf.js
├── package.json
├── protractor.conf.js
├── README.md
├── tsconfig.json
└── tslint.json
```

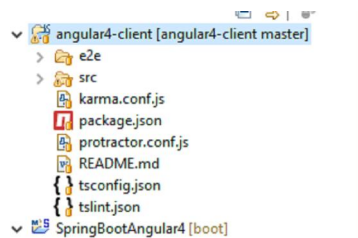
To clean the sourcecode in STS, we need to remove **node_modules** by following the steps:

– Right click on **angular4-client** project, choose **Properties**, then choose: **Resource** -> **Resource Filter**.

– Press **Add Filter...**, choose **Filter Type: Exclude all**, **Applies to: Files and folders**, and check **All children (recursive)**, with **File and Folder Attributes**, we specify **node_modules**:



Press **OK**. Then press **Apply**, result:



-> Now **node_modules** is already removed from the **SpringToolSuite**.

It's ready to modify something with Angular4-client project:

- Open **/src/app/app.component.ts**, edit:

```
export class AppComponent {  
  title = 'JavasampleApproach HelloWorld Angular4 App';  
}
```

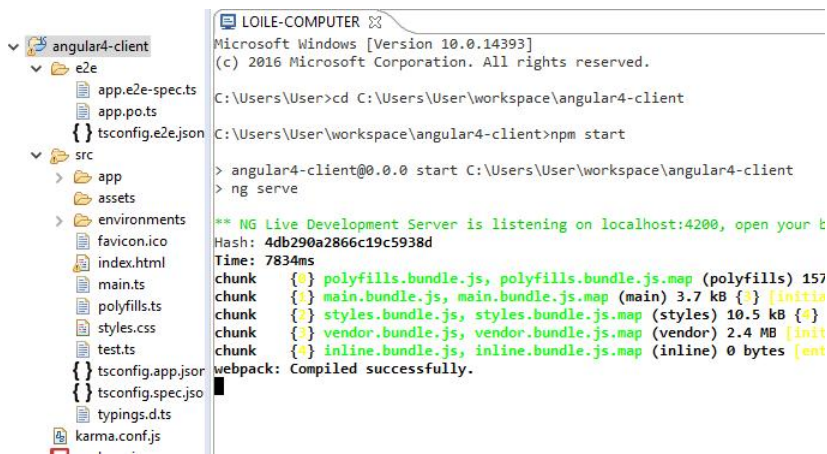
- Open **src/app/app.component.css**, add:

```
h1 {  
  color: blue;  
  font-size: 150%;  
}
```

Now, start **angular4-client** project with STS:

- Go to **Window -> Show View -> Other**, search and choose **Terminal**.

- Then launch a **Local Terminal**, cd to **C:\Users\User\workspace\angular4-client**. Press command **npm start** to launch the **angular4-client**, results:



JavaSampleApproach - Angular4 App!

6. Integrate SpringBoot server and Angular 4 client

Up to now, **Angular4-Client** and **SpringBoot server** work independently on ports 8080 and 4200.

Goal of below integration: the client at 4200 will proxy any **/api** requests to the server.

Step to do:

– Create a file **proxy.conf.json** under project **angular4-client** folder with content:

```
{
  "/api": {
    "target": "http://localhost:8080",
    "secure": false
  }
}
```

– Edit **package.json** file for "start" script:

```
...
"scripts": {
  "ng": "ng",
  "start": "ng serve --proxy-config proxy.conf.json",
  "build": "ng build",
  "test": "ng test",
  "lint": "ng lint",
  "e2e": "ng e2e"
},
...
```

– Build and run the **RestfulService** project with SpringBoot App mode at port 8080. And run **angular4-client** at port 4200.

Make a request **http://localhost:4200/api/hi**, result:



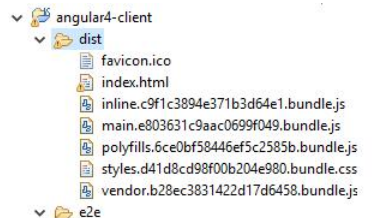
Hello World from Restful API

7. Deploy SpringBoot server with Angular4 client

Build **angular4 client** with command `ng build --env=prod`:

```
C:\Users\User\workspace\angular4-client>ng build -prod
Hash: 34b09595a87937a7cd6f
Time: 9947ms
chunk {0} polyfills.6ce0bf58446ef5c2585b.bundle.js (polyfills) 157 kB {4} [initial] [rendered]
chunk {1} main.e803631c9aac0699f049.bundle.js (main) 21.5 kB {1} [initial] [rendered]
chunk {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [initial] [rendered]
chunk {3} vendor.b28ec3831422d17d6458.bundle.js (vendor) 1.1 MB [initial] [rendered]
chunk {4} inline.c9f1c3894e371b3d64e1.bundle.js (inline) 0 bytes [entry] [rendered]
```

Result is a **dist** folder:



We have 2 approaches to deployment **Spring Boot server** with **angular4 client**:

- Manually copy all files from **dist** folder to **/src/main/resources/static** folder of **SpringBoot server** project.
- Using Maven plugin to copy all files from **dist** folder to **/src/main/resources/static** folder of **SpringBoot server** project.

```
<plugin>
<artifactId>maven-resources-plugin</artifactId>
<executions>
  <execution>
    <id>copy-resources</id>
    <phase>validate</phase>
    <goals><goal>copy-resources</goal></goals>
    <configuration>
      <outputDirectory>${basedir}/target/classes/static/</outputDirectory>
      <resources>
        <resource>
          <directory>${basedir}/../angular4-client/dist</directory>
        </resource>
      </resources>
    </configuration>
  </execution>
</executions>
</plugin>
```

Now build and run the **SpringBoot server** again with commands:

- Build: `mvn clean install`
- Run: `mvn spring-boot:run`

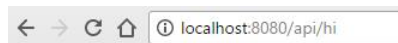
Then make some requests:

- `http://localhost:8080/`, result:



JavaSampleApproach - Angular4 App!

- `http://localhost:8080/api/hi`, result:



Hello World from Restful API

>>> Done! Now you can start to develop **Angular4** with **SpringBoot** and **SpringToolSuite!** 😊

By [JavaSampleApproach](#) | May 15, 2017.
Last updated on **October 20, 2017**.

Related Posts

- [Angular 4 ElasticSearch example – simple Full Text Search](#)
- [Angular 4 ElasticSearch example – Documents Pagination with Scroll](#)
- [Angular 4 ElasticSearch example – Get All Documents in Index](#)
- [Angular 4 ElasticSearch example – Add Document to Index](#)
- [Angular 4 ElasticSearch example – How to create an Index](#)
- [Angular 4 ElasticSearch – Quick Start – How to add Elasticsearch.js](#)
- [Angular 4 – Send Nested Object to Spring Boot Server](#)
- [Angular 4 – Upload/Get Images to/from Spring Boot Server](#)
- [Angular 4 – Upload/Get MultipartFile to/from Spring Boot Server](#)
- [Angular 4 Firebase – Upload Image & Display List Images from Storage](#)

Post Tags

[angular](#) [angular 4](#) [angularjs](#) [editor](#) [integration](#) [rest api](#) [spring boot](#) [web](#)

46 thoughts on “How to integrate Angular 4 with SpringBoot RestApi using SpringToolSuite”



enhshuld

May 15, 2017 at 4:45 pm

hello..., demo project?



Mohamed

June 20, 2017 at 5:08 pm

Hi I have been going through the steps and copied it but do not get loaded into angular project index page; get a white error page.



JavaSampleApproach

June 24, 2017 at 2:09 am

Hi,

Had you met the problem at **step 7**: *Deploy SpringBoot server with Angular4 client?*

Here, We have 2 approaches to deployment **Spring Boot server** with **angular4 client**:

1. Manually copy all files from dist folder to **/src/main/resources/static** folder of SpringBoot server project.
2. Using Maven plugin to copy all files from dist folder to **/src/main/resources/static** folder of SpringBoot server project.

I think you have mistake with approach **2**, So please double check the **\${basedir}**, or replace it with the absolute path for double checking,

```
<configuration>
  <outputDirectory>${basedir}/target/classes/static/</outputDirectory >
  <resources>
    <resource>
      <directory>${basedir}/../angular4-client/dist</directory >
    </resource>
  </resources>
</configuration>
```

Regards,
JSA

 **Jam**

July 10, 2017 at 2:21 pm

i got my project loaded into angular project index page, but no styling is applied. Otherwise the <http://localhost:8080/api/hi> gives me 404 error.

 **JavaSampleApproach** 

July 11, 2017 at 9:22 am

Hi Jam,

I see you had met the same mistakes with other guys!
Be careful when doing each steps on the tutorial!

See the above comment: mistakes at **step 7** or adding `../angular4-client/dist` ! I think it helps you!

Good Luck!

 **AVINASH FRANCIS CULAS**

August 1, 2017 at 6:37 am

Can we implement this approach in production environment, please help we are in the design phase of an angular 4 spring boot project

 **Santhi**

August 2, 2017 at 12:09 pm

This is greatly useful.. Thank you

 **JavaSampleApproach** 

August 4, 2017 at 2:20 pm

Please be noticed that we had made a supporting video guide: <https://youtu.be/Ps-7ZV8YpI0>.

 **ghostwriter9151**

September 11, 2017 at 11:35 am

Hello! I can't import the angular project in the latest STS.
The project is not shown in the import dialog... Any guess?

 **JavaSampleApproach** 

September 11, 2017 at 11:40 am

Please give me your STS version?
Or You can try using STS Version 3.8.1.RELEASE.

 **Lokesh**

September 16, 2017 at 3:16 pm

Hi There,

Hi There Angular 4 project is running on Spring boot now.
Can you advice me how can I integrate it with Spring Security?

If I give in configure method that authenticate (**), then it should go to login page which angular 4 is made and authenticate the credentials using spring security, how can I achieve this?

Regards
Lokesh

 **JavaSampleApproach** 

September 17, 2017 at 12:39 pm

Hello Lokesh,

For security, You can follow the guide of tutorial:
[Spring Security – Config Security for Web MVC by Spring Boot](#)

Details:

– Add security dependency:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

– Define a customize login.html page and use a @RequestMapping(value={"/login"}):

```
@RequestMapping(value={"/login"})
public String login(){
    return "login";
}
```

– Configure WebSecurityConfigurerAdapter:

```
@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
            .antMatchers("/", "/home").permitAll()
            .antMatchers("/api/**")
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .permitAll();
        http.exceptionHandling().accessDeniedPage("/403");
    }
}
```

Please let me know if you done!

Regards,

 **jeremy**

September 18, 2017 at 1:29 am

Hi

I speak French so sorry for my English.
I have some problems when I want to launch starter project with spring boot app.
I'm on mac os.

I have this error :

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Exception in thread "main" java.lang.NoClassDefFoundError: ch/qos/logback/classic/Level
    at org.springframework.boot.logging.logback.LogbackLoggingSystem.(LogbackLoggingSystem.java:66)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
    at org.springframework.boot.logging.LoggingSystem.get(LoggingSystem.java:170)
    at org.springframework.boot.logging.LoggingSystem.get(LoggingSystem.java:160)
    at org.springframework.boot.logging.LoggingApplicationListener.onApplicationStartingEvent(LoggingApplicationListener.java:229)
    at org.springframework.boot.logging.LoggingApplicationListener.onApplicationEvent(LoggingApplicationListener.java:209)
    at org.springframework.context.event.SimpleApplicationEventMulticaster.doInvokeListener(SimpleApplicationEventMulticaster.java:172)
    at org.springframework.context.event.SimpleApplicationEventMulticaster.invokeListener(SimpleApplicationEventMulticaster.java:165)
    at org.springframework.context.event.SimpleApplicationEventMulticaster.multicastEvent(SimpleApplicationEventMulticaster.java:139)
    at org.springframework.context.event.SimpleApplicationEventMulticaster.multicastEvent(SimpleApplicationEventMulticaster.java:122)
    at org.springframework.boot.context.event.EventPublishingRunListener.starting(EventPublishingRunListener.java:69)
    at org.springframework.boot.SpringApplicationRunListeners.starting(SpringApplicationRunListeners.java:48)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:292)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1118)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:1107)
    at com.example.demo.Angularstarterproject1Application.main(Angularstarterproject1Application.java:10)
Caused by: java.lang.ClassNotFoundException: ch.qos.logback.classic.Level
    at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:335)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    ... 19 more
```

JavaSampleApproach

September 18, 2017 at 11:49 am

What's your STS version?

Your error is clear: java.lang.ClassNotFoundException: ch.qos.logback.classic.Level

Please base on it for tracking!

 **jeremy**

September 18, 2017 at 5:43 pm

hi,
my version of STS is 3.9.0.RELEASE.

i just create the project and start with spring boot app.
i dont understand why log4j make some problems, i dont use it

 **hanz**

September 29, 2017 at 2:37 pm

hi ,
i just follow the tuto but im blocked in step 6 , the server turn with angular but when i try to run the SpringBootAngular i get this error
"" Exception in thread "main" java.lang.NoClassDefFoundError: org/springframework/util/ClassUtils ""
any help 😊

JavaSampleApproach

September 30, 2017 at 5:45 am

Hi,

The problem is related with your development environment for creating a SpringBoot project.

So you can double check your STS version, then re-create a new SpringBoot project with simple "Hello world" RestController.

 **Yekalyvan**

October 5, 2017 at 7:37 am

I think we don't need a proxy.conf.json file...it's working fine for me..

 **JavaSampleApproach** 

October 5, 2017 at 2:29 pm

Hi Rubysavach,

What step in the tutorial do you mention that suites for your case?

Please give me the way Angular's client can know what '/apis' of servers (ip + port) for calling without proxy.conf.json file, Rubysavach?

Regards,

 **hussem**

October 17, 2017 at 12:08 pm

Thank you very much.

I have learned a lot with your few valuable steps and line.

 **Justin**

October 31, 2017 at 5:13 am

I have the following error on the local terminal:

```
C:\Users\jwm19_000\Documents\STS\SpringBoot-angular>mvn
'mvn' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\Users\jwm19_000\Documents\STS\SpringBoot-angular>
```

—

And also:

```
C:\Users\jwm19_000\Documents\STS\angular4-client>npm start
'npm' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\Users\jwm19_000\Documents\STS\angular4-client>4
```

What I am doing wrong?

 **JavaSampleApproach** 

October 31, 2017 at 5:23 am

Hi Justin,

Your problem is related with your **mvn** setting and **npm** setting in **STS**.

Please double check it!

Regards,
JSA

 **Justin**

November 1, 2017 at 3:24 am

I see, do you know what the settings should be? I thought I followed your instructions, but I could have missed something. Where should I go in STS and what should I change? Thank you.

 **Justin**

November 2, 2017 at 3:52 am

What should the settings be?

 **JavaSampleApproach** 🌟

November 2, 2017 at 5:12 am

You got 2 messages: 'mvn' is not recognized and 'npm' is not recognized
It means you **did NOT** install **mvn** and **Node.js** successfully!

So open an **cmd** to check mvn by `mvn -v`
For npm, using cmd: `node -v` & `npm -v` to check it.

 **Justin**

November 3, 2017 at 12:54 am

So I need to download maven? Do you recommend a site/version? I tried a version, but my computer warned me it might have a virus. Just wanted your opinion. Thank you.

 **JavaSampleApproach** 🌟

November 3, 2017 at 5:33 am

You can download mvn at: [Download Apache Maven](#).
Then follow the install step by step at [Installing Apache Maven](#).

 **Justin**

November 5, 2017 at 5:20 pm

I have tried to follow the step by step instructions for installing on Windows, but they do not make sense to me. Is there a video I can follow? I have tried everything, but nothing I do works.

 **Justin**

November 5, 2017 at 5:43 pm

I finally got mvn to work properly, thank you for all of your help, but now I am getting the following message when I run the ng build command:

'ng' is not recognized as an internal or external command,
operable program or batch file.

Please keep in mind, I cannot run it in the local terminal on STS and I am running it on cmd terminal. What do I need to do to get the ng build command to run?

 **Scotte**

November 2, 2017 at 7:18 pm

Hey.. when I go in production environment with this setup.. will it work..Or I need to change something in proxy.conf.json?

 **JavaSampleApproach** 

November 3, 2017 at 5:24 am

Hi Scotte,

In production env, you just need copy all files in **dist** folder from cmd `ng build --env=prod`.

Regards,
JSA

 **Justin**

November 3, 2017 at 11:41 am

I am not sure what that means? Which files? And what's the production envelope? I am sorry if I seem confused.

 **Anonimo**

November 8, 2017 at 5:23 pm

Spring Boot + Jboss EAP 6.4

is possible?

 **JavaSampleApproach** 

November 12, 2017 at 1:07 pm

Hello Anonimo,

Please details for clearly your expected.

Regards,
JSA

 **Iorran**

December 13, 2017 at 2:33 pm

Yes, its possible, i using that.

 **Madhuri**

November 11, 2017 at 6:56 pm

I copied the dist folder as mentioned in step 7 but still I am getting error -"Whitelabel Error Page " when i load the pag e on local host.

Any help will be appreciated



JavaSampleApproach 👤

November 12, 2017 at 1:05 pm

Hi Madhuri,

Had you done with the step 6:

Integrate SpringBoot server and Angular 4 client

– If your app works well at step 6, please use cmd: `ng build --env=prod` to build a production image then:

We have 2 approaches to deployment Spring Boot server with angular4 client:

- Manually copy all files from dist folder to /src/main/resources/static folder of SpringBoot server project.
- Using Maven plugin to copy all files from dist folder to /src/main/resources/static folder of SpringBoot server project

See more at video <https://youtu.be/Ps-7ZV8YpI0>

Regards,
JSA



Sunil Pandey

December 3, 2017 at 5:31 am

Thanks for the awesome ! post.



Anton Romanov (FROM RUSSIA)

December 12, 2017 at 1:22 pm

It's a best example ive ever seen. Thank you!!!!



Sunil Roy

December 17, 2017 at 1:32 pm

```
chunk {vendor} vendor.bundle.js (vendor) 854 kB [initial] [rendered]

ERROR in Error: AppComponent cannot be used as an entry component.
    at syntaxError (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:528:34)
    at CompileMetadataResolver._getEntryComponentMetadata (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:15862:19)
    at C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:15334:110
    at Array.map (<anonymous>)
    at CompileMetadataResolver.getNgModuleMetadata (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:15334:73)
    at visitLazyRoute (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:31061:104)
    at AotCompiler.listLazyRoutes (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler\bundles\compiler.umd.js:31029:20)
    at AngularCompilerProgram.listLazyRoutes (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler-cli\src\transformers\program.js:156:30)
    at Function.NgTools_InternalApi_NG_2.listLazyRoutes (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@angular\compiler-cli\src\ngtools_api.js:44:36)
    at AngularCompilerPlugin._getLazyRoutesFromNgtools (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@ngtools\webpack\src\angular_compiler_plugin.js:248:66)
    at Promise.resolve.then.then (C:\Study\Development\Workspace_SpringBoot\angular4-client\node_modules\@ngtools\webpack\src\angular_compiler_plugin.js:562:50)
    at <anonymous>
    at process._tickCallback (internal/process/next_tick.js:160:7)
```

webpack: Failed to compile.



JavaSampleApproach 👤

December 17, 2017 at 2:01 pm

Hello Sunil Roy,

Please check the guide video for your setup: <https://youtu.be/Ps-7ZV8YpI0>

Regards,
JSA

 **Kiran**

December 20, 2017 at 3:44 pm

After implementing all the steps given in this tutorial, I thought of completing it by creating a WAR file, and the steps followed are,

- 1) Modified pom.xml by replacing jar to war
- 2) mvn spring-boot:run created a WAR file
- 3) Copied that WAR file to C:\Program Files\Apache Software Foundation\Tomcat 8.0\webapps and tried running
 - a) <http://localhost:8080/demo/api/hi>
 - b) <http://localhost:8080/demo>

But none of these two attempts (ie,a,b) worked for me. Is there any thing that I need to do as we do in JSP-Servlet web.xml? Where we create a and xml tags where in we place the jsp file to be loaded i,e

/jsp/login.jsp

Can any one please tell me the right procedure to create a WAR file and deploy it into a remote tomcat webapp folder ?

 **JavaSampleApproach** 

December 21, 2017 at 5:08 am

Hi Kiran,

Please visit this post:

[How to deploy Spring Boot Web App War file to Tomcat Server with Maven build](#)

Regards,
JSA.

 **Sajal (India)**

December 28, 2017 at 10:39 pm

It is the best video I have ever seen.

 **Wallace Howery**

January 3, 2018 at 6:20 pm

Just curious, in Step 5, why do you remove node_modules from STS?

 **JavaSampleApproach** 

January 5, 2018 at 3:48 am

Dear Wallace Howery,

Just to clean the sourcecode in STS on view. You can do nothing.

Regards,
JSA

FOLLOW US



ABOUT US

We are passionate engineers in software development by Java Technology & Spring Framework. We believe that creating little good thing with specific orientation everyday can make great influence on the world someday.