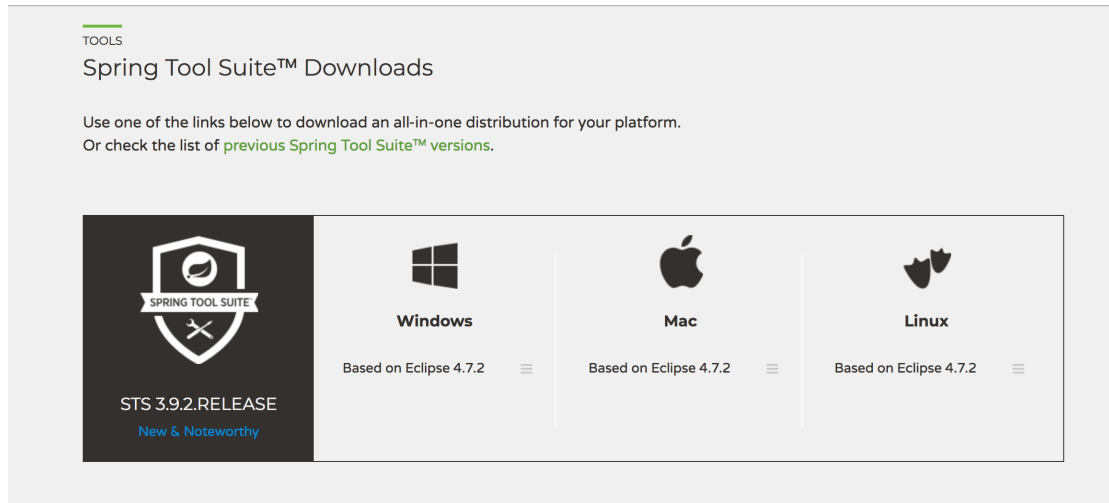


一. STS 下载:

在如下地址下载 : <https://spring.io/tools/sts/all>

选择自己喜欢的版本就好,如图:



我这里选择的是 3.7.3 版本

二:Spring 源码导入

1 下载的 源码版本是 4.3.9,地址如下:

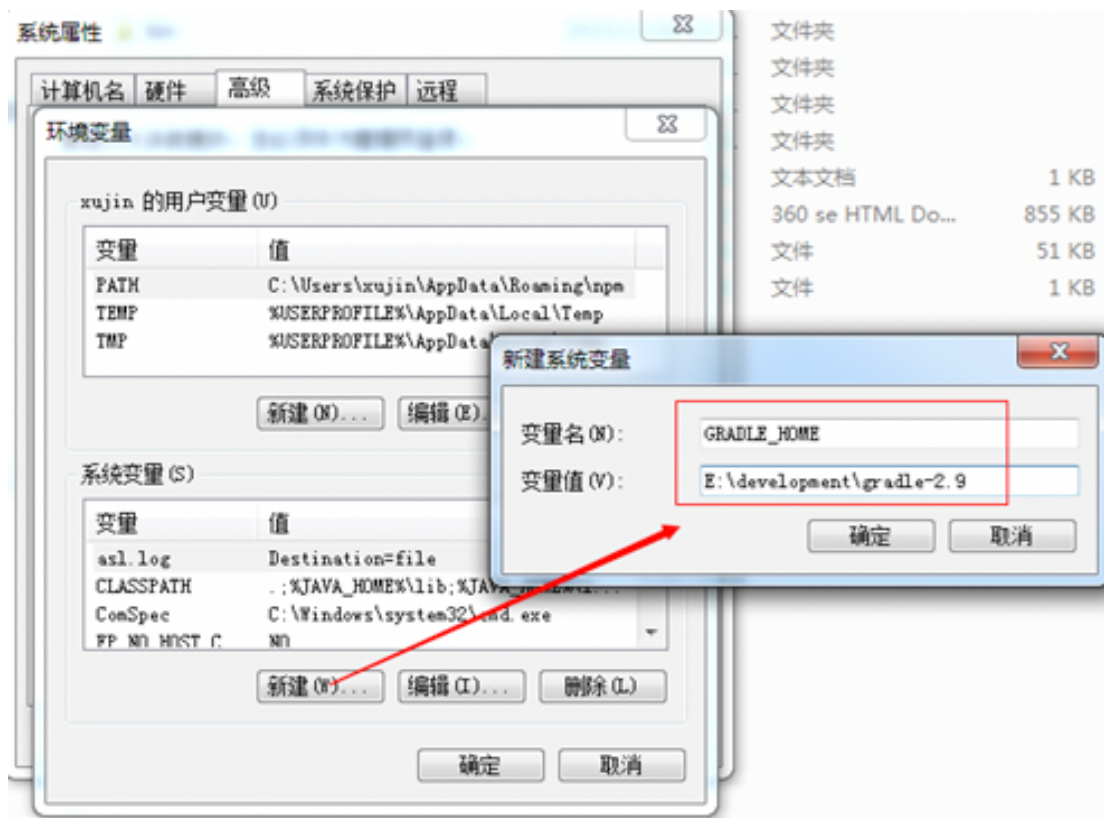
<https://github.com/spring-projects/spring-framework>

2. 下载 gradle

1. 下载地址如下:

<http://www.gradle.org/downloads>

2. 配置环境变量



3. 检查

配置完成之后，当完成系统变量的配置后打开命令窗口输入命令 " gradle - version ",如果出现 gradle 对应的版本信息，表示安装成功。

3. 编译:

1. 修改 import-into-eclipse.sh.内容如下:

```
cd `dirname $0`  
clear  
cat <<EOM
```


Spring Framework – Eclipse/STS Project Import Guide

This script will guide you through the process of importing the Spring Framework projects into Eclipse or the Spring Tool Suite (STS). It is recommended that you have a recent version of Eclipse or STS. As a bare minimum you will need Eclipse with full Java 8 support, the AspectJ Development Tools (AJDT), and the Groovy Compiler.

This script has been tested against:

- STS: 3.6.3.RELEASE (Eclipse Luna SR1 4.4.1)
- AJDT: 2.2.4.e44x-20141118-0700 (Luna 4.4)

If you need to download and install Eclipse or STS, please do that now

by visiting one of the following sites:

- Eclipse downloads:

<http://download.eclipse.org/eclipse/downloads>

- STS downloads: <http://spring.io/tools/sts/all>

- STS nightly builds:

<http://dist.springsource.com/snapshot/STS/nightly-distributions.html>

If you need to install a recent CI build for AJDT (i.e., so that the spring-aspects module properly compiles in Eclipse/STS), click on the "Artifacts" tab of a successful CI build and copy the link to the

"update site" which you can then use to install or update AJDT within Eclipse/STS.

- AJDT CI builds: <https://build.spring.io/browse/AJDT-AE44>

Once Eclipse/STS is installed, press enter, and we'll begin.
EOM

read

this command:

- wipes out any existing Eclipse metadata

- generates OXM test classes to avoid errors on import into Eclipse

- generates metadata for all subprojects

- skips metadata gen for the root project (-x :eclipse) to work

around Eclipse's inability to import hierarchical project structures

COMMAND="/Users/hejia Rui/gradle-3.5.1/bin/gradle --no-daemon
cleanEclipse :spring-oxm:compileTestJava eclipse -x :eclipse"

cat <<EOM

STEP 1: Generate subproject Eclipse metadata

The first step will be to generate Eclipse project metadata for each of

the spring-* subprojects. This happens via the built-in "Gradle wrapper"

script (./gradlew in this directory). If this is your first time using the Gradle wrapper, this step may take a few minutes while a Gradle distribution is downloaded for you.

The command run will be:

```
$COMMAND
```

Press enter when ready.

```
EOM
```

```
read
```

```
$COMMAND || exit
```

```
cat <<EOM
```

```
-----  
-----
```

STEP 2: Import subprojects into Eclipse/STS

Within Eclipse/STS, do the following:

- File > Import... > Existing Projects into Workspace
- When prompted for the 'root directory', provide \$PWD.
- Press enter. You will see the modules show up under "Projects".
- All projects should be selected/checked. Click Finish.
- When the project import is complete, you should have no errors.

When the above is complete, return here and press the enter key.

EOM

read

COMMAND="/Users/hejiarui/gradle-3.5.1/bin/gradle --no-daemon :eclipse"

cat <<EOM

STEP 3: Generate root project Eclipse metadata

Unfortunately, Eclipse does not support importing project hierarchies,
so we had to skip root project metadata generation during step 1. In
this step we simply generate root project metadata so that you
can
import it in the next step.

The command run will be:

\$COMMAND

Press the enter key when ready.

EOM

read

\$COMMAND || exit

```
cat <<EOM
```

```
-----  
-----
```

STEP 4: Import root project into Eclipse/STS

Follow the project import steps listed in step 2 above to import the root "spring" project.

Press enter when complete, and move on to the final step.

```
EOM
```

```
read
```

```
cat <<EOM
```

```
-----  
-----
```

STEP 5: Enable Git support for all projects

- In the Eclipse/STS Package Explorer, select all spring* projects.
- Right-click to open the context menu and select Team > Share Project...
- In the Share Project dialog that appears, select Git and press Next.
- Check "Use or create repository in parent folder of project".
- Click Finish.

When complete, you'll have Git support enabled for all projects.




You're ready to code! Goodbye!

EOM

4. 编译后的问题

1. 安装 eclipse 的 groovy 插件

clean 完之后发现项目还报错（关于 groovy 的，所以 eclipse 需要集成 groovy 插件）

 GroovyDynamicElementReader cannot be resolved to a type	GroovyBean...	/spring-...	line 391	Java Problem
 GroovyDynamicElementReader cannot be resolved to a type	GroovyBean...	/spring-...	line 689	Java Problem
 GroovyDynamicElementReader cannot be resolved to a type	GroovyBean...	/spring-...	line 697	Java Problem

解决：eclipse 菜单栏：help -- Install new software：地址输入

<http://dist.springsource.org/snapshot/GRECLIPSE/e4.5/>, 下载完即可。

具体 groovy 版本和 eclipse 版本，参考：<https://github.com/groovy/groovy-eclipse/wiki>


ps: 这里只需下载如下几个就可以:


name	version
 Groovy Compiler 2.2	2.9.2.xx-201801052009-e45-RELEASE
 Groovy Compiler 2.3	2.9.2.xx-201801052009-e45-RELEASE
 Groovy Compiler 2.4	2.9.2.xx-201801052009-e45-RELEASE
 Groovy Compiler 2.5 (early access)	2.9.2.xx-201801052009-e45-RELEASE
 Groovy-Eclipse (Required)	2.9.2.xx-201801052009-e45-RELEASE
 Groovy-Eclipse Feature	2.9.2.xx-201801052009-e45-RELEASE
 Maven Support (Optional)	
 Unrecognized	

2. 编译错误:

1. ValidatorFactoryTests.java 报错:

如图:

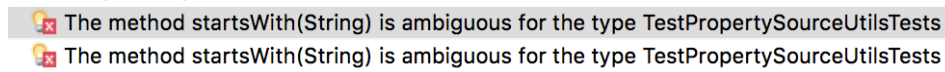
 The method addPropertyNode(String) is undefined for the type ConstraintValidatorContext.ConstraintViolationBuilder

 The method addPropertyNode(String) is undefined for the type ConstraintValidatorContext.ConstraintViolationBuilder

改成如下即可:

```
if (bean.getValue() == null) {  
    context.buildConstraintViolationWithTemplate("NULL").addNode("value").addConstraintViolation();  
    return false;  
}
```

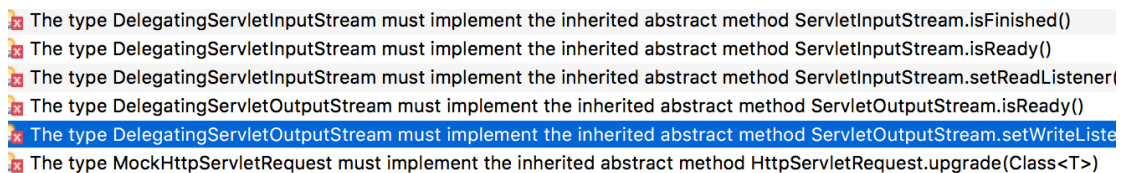
2. TestPropertySourceUtilsTests.java 报错:



改成如下即可:




























```
public void extendedEmptyAnnotation() {  
    expectedException.expect(IllegalStateException.class);  
    expectedException.expectMessage(CoreMatchers.startsWith("Could not detect default properties file for test"));  
    expectedException.expectMessage(containsString("ExtendedEmptyPropertySources.properties"));  
    buildMergedTestPropertySources(ExtendedEmptyPropertySources.class);  
}
```

3. DelegatingServletInputStream 项目报错:



现在还没有解决方法,等后续解决了,在告诉大家

导入后如图所示:

- ▶  spring
- ▶  spring-aop
- ▶  spring-aspects
- ▶  spring-beans
- ▶  spring-beans-groovy
- ▶  spring-build-src
- ▶  spring-context
- ▶  spring-context-support
- ▶  spring-core
- ▶  spring-expression
- ▶  spring-framework-bom
- ▶  spring-instrument
- ▶  **spring-instrument-tomcat**
- ▶  spring-jdbc
- ▶  spring-jms
- ▶  spring-messaging
- ▶  spring-orm
- ▶  spring-orm-hibernate4
- ▶  spring-orm-hibernate5
- ▶  spring-oxm
- ▶  spring-test
- ▶  spring-tx
- ▶  spring-web
- ▶  spring-webmvc
- ▶  spring-webmvc-portlet
- ▶  spring-webmvc-tiles2
- ▶  spring-websocket

三:Spring boot 源码导入

1. 在如下地址下载:







































<https://github.com/spring-projects/spring-boot>

我这里使用的是 1.5.9.RELEASE

2. 下载后,执行如下命令: `mvn clean install eclipse:eclipse` 即可

3. 导入

注意:需要 jdk1.8 maven 3.2.1 以上才可以 导入后如图:

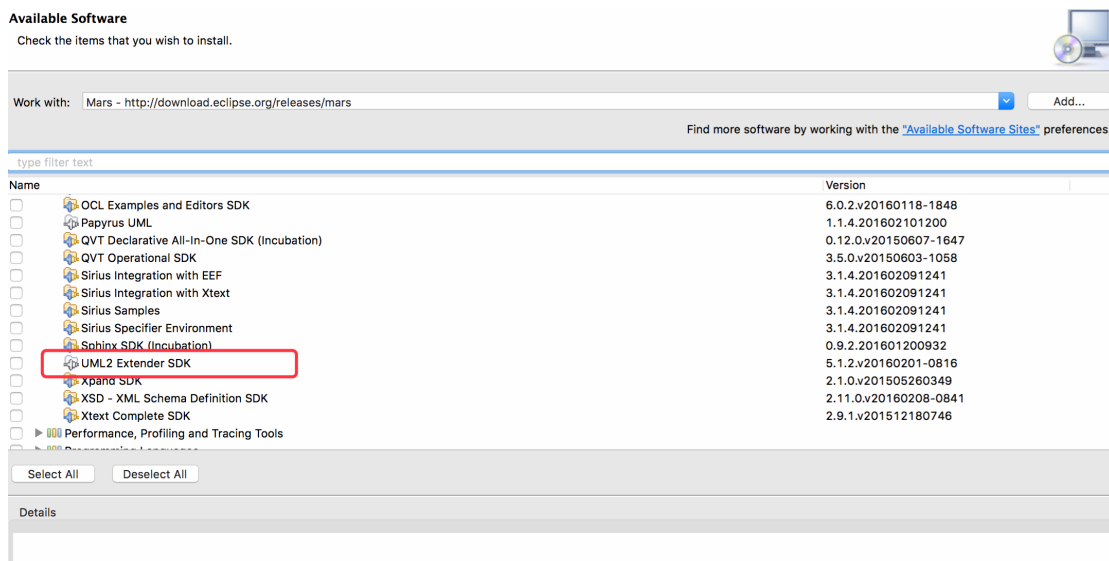
- ▶  spring-boot [boot]
- ▶  spring-boot-actuator [boot]
- ▶  spring-boot-actuator-docs [boot]
- ▶  spring-boot-antlib [boot]
- ▶  spring-boot-autoconfigure [boot]
- ▶  spring-boot-autoconfigure-processor [boot]
- ▶  spring-boot-build
- ▶  spring-boot-cli [boot]
- ▶  spring-boot-configuration-metadata
- ▶  spring-boot-configuration-processor [boot]
- ▶  spring-boot-dependencies
- ▶  spring-boot-devtools [boot]
- ▶  spring-boot-devtools-tests [boot]
- ▶  spring-boot-docs [boot]
- ▶  spring-boot-gradle-plugin [boot]
- ▶  spring-boot-gradle-tests [boot]
- ▶  spring-boot-integration-tests
- ▶  spring-boot-integration-tests-embedded-servlet-container [k
- ▶  spring-boot-launch-script-tests [boot]
- ▶  spring-boot-loader
- ▶  spring-boot-loader-tools [boot]
- ▶  spring-boot-maven-plugin [boot]
- ▶  spring-boot-parent
- ▶  spring-boot-security-test-web-helloworld [boot]
- ▶  spring-boot-security-tests
- ▶  spring-boot-starter [boot]
- ▶  spring-boot-starter-activemq [boot]
- ▶  spring-boot-starter-actuator [boot]
- ▶  spring-boot-starter-amqp [boot]
- ▶  spring-boot-starter-aop [boot]
- ▶  spring-boot-starter-artemis [boot]
- ▶  spring-boot-starter-batch [boot]
- ▶  spring-boot-starter-cache [boot]
- ▶  spring-boot-starter-cloud-connectors [boot]
- ▶  spring-boot-starter-data-cassandra [boot]
- ▶  spring-boot-starter-data-couchbase [boot]
- ▶  spring-boot-starter-data-elasticsearch [boot]
- ▶  spring-boot-starter-data-gemfire [boot]

四. 常用插件安装:

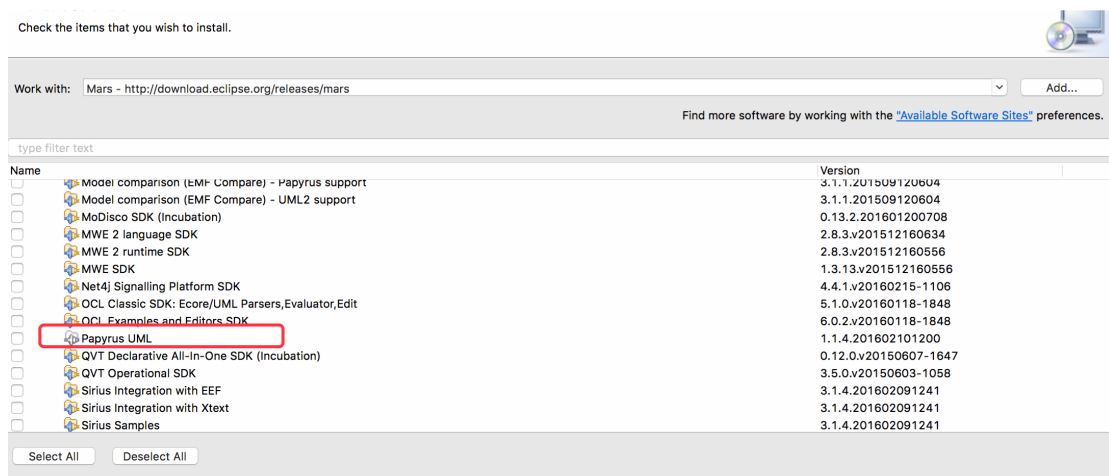
1. Papyrus uml

1. <http://download.eclipse.org/releases/mars/modeling/> 安装 uml2

如图:



2. <http://download.eclipse.org/releases/mars/modeling/> 选择 papyrus



3. <http://download.eclipse.org/modeling/mdt/papyrus/updates/releases/mars>

下选择 java

