

- 먼저 Spring Legacy Project를 생성해줍니다.

New Spring Legacy Project

Spring Legacy Project

Enter a project name.

Project name:


☒ Use default location

Location:

Select Spring version:

Templates:

- ▼
 -
 -
 -
- >
- >
- >
- >
- >
-

 requires downloading [Configure templates...](#)

Description:
A new Spring MVC web application development project

URL: <https://dist.springsource.com/release/STS/help/org.springframework.templates.mvc-3.2.2.zip>

Working sets

☐ Add project to working sets

Working sets:

- 프로젝트 이름을 입력해주고 Spring MVC Project를 선택하고 Finish

```

SpringAjax
├── src/main/java
│   ├── com.campus.myapp
│   │   ├── AjaxController.java
│   │   └── HomeController.java
│   └── src/main/resources
├── src/test/java
├── src/test/resources
├── JRE System Library [JavaSE-1.6]
├── Maven Dependencies
├── src
│   ├── main
│   │   ├── webapp
│   │   │   ├── resources
│   │   │   ├── WEB-INF
│   │   │   │   ├── classes
│   │   │   │   ├── spring
│   │   │   │   └── views
│   │   │       ├── ajax
│   │   │       │   ├── ajaxView.jsp
│   │   │       │   ├── home.jsp
│   │   │       └── web.xml
│   │   └── test
│   └── target
└── pom.xml
  
```

- AjaxController와 ajaxView파일을 만들어 주었습니다.

- 그 다음 제이쿼리 다운로드 페이지 안에 원하는 버전의 CDNJS CDN를 복사해줍니다. 저는 3.6.0 버전을 사용했습니다.
- <https://jquery.com/download/>

Other CDNs




The following CDNs also host compressed and uncompressed versions of jQuery releases. Starting with jQuery 1.9 they may also host [sourcemap files](#); check the site's documentation.

Note that there may be delays between a jQuery release and its availability there. Please be patient, they receive the files at the same time the blog post is made public. Beta and release candidates are not hosted by these CDNs.

- [Google CDN](#)
- [Microsoft CDN](#)
- [CDNJS CDN](#)
- [jsDelivr CDN](#)

jquery

JavaScript library for DOM operations

★ 55k  GitHub  package  0 vulnerabilities

MIT licensed <http://jquery.com/>

Tags: jquery, library, ajax, framework, toolkit, popular

Version

3.6.0




▼

Asset Type

All

▼

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js>

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
    <title>Home</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
```

- jsp 파일 안에 넣어주면 됩니다.
- 그 다음 메이븐 저장소에서 'Gson'을 검색후 원하는 버전을 선택후 메이븐을 복사해줍니다. 저는 2.9.0을 사용했습니다.
- <https://mvnrepository.com/>

Gson

Search

Home » [com.google.code.gson](#) » gson**Gson**

Gson

License	Apache 2.0
Categories	JSON Libraries
Tags	google json
Used By	17,242 artifacts

Central (33)

Atlassian 3rd-P Old (4)

Spring Plugins (4)

WSO2 Dist (1)

Redhat GA (4)

Redhat EA (1)

ICM (2)

	Version	Vulnerabilities	Repository	Usages	Date
2.9.x	2.9.0		Central	304	Feb, 2022
	2.8.9		Central	1,130	Oct, 2021
	2.8.8		Central	718	Aug, 2021
	2.8.7		Central	674	May, 2021
	2.8.6		Central	3,405	Oct, 2019
	2.8.5		Central	4,427	May, 2018
	2.8.4		Central	281	May, 2018
	2.8.3		Central	28	Apr, 2018
	2.8.2		Central	1,779	Sep, 2017
	2.8.1		Central	785	May, 2017

**Gson » 2.9.0**

Gson

License	Apache 2.0
Categories	JSON Libraries
Date	(Feb 11, 2022)
Files	pom (7 KB) jar (243 KB) View All
Repositories	Central
Used By	17,242 artifacts

Maven

Gradle

Gradle (Short)

Gradle (Kotlin)

SBT

Ivy

Grape

Leiningen

Buildr

```

<!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.9.0</version>
</dependency>

```

- ▼ SpringAjax
 - > src/main/java
 - > src/main/resources
 - > src/test/java
 - > src/test/resources
 - > JRE System Library [JavaSE-1.6]
 - > Maven Dependencies
 - > src
 - > target
 - pom.xml

- 프로젝트 내의 pom.xml에서

```

home.jsp web.xml *SpringAjax/pom.xml ×
111
112     <!-- Test -->
113     <dependency>
114         <groupId>junit</groupId>
115         <artifactId>junit</artifactId>
116         <version>4.7</version>
117         <scope>test</scope>
118     </dependency>
119     <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
120     <dependency>
121         <groupId>com.google.code.gson</groupId>
122         <artifactId>gson</artifactId>
123         <version>2.9.0</version>
124     </dependency>
125 </dependencies>
126 <build>
127     <plugins>
128     <plugin>
129         <artifactId>maven-eclipse-plugin</artifactId>
130         <version>2.9</version>
131         <configuration>
132             <additionalProjectnatures>
133                 <projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>
134             </additionalProjectnatures>

```

- dependencies 태그 안에 넣어주면 사용할 준비가 다 되었습니다.

home.jsp

```

home.jsp × ajaxView.jsp AjaxController.java
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
3 <html>
4 <head>
5     <title>Home</title>
6     <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7     <script>
8         $(function(){
9             $("h1").click(function(){
10                 location.href = '/myapp/ajaxView';
11             });
12         });
13     </script>
14 </head>
15 <body>
16 <h1>
17     ajax(비동기식 통신)페이지로 이동하기
18 </h1>
19 </body>
20 </html>

```

ajaxView.jsp

```

home.jsp ajaxView.jsp × AjaxController.java
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
9 </head>
10 <body>
11 화면에 뜹니다
12 </body>
13 </html>

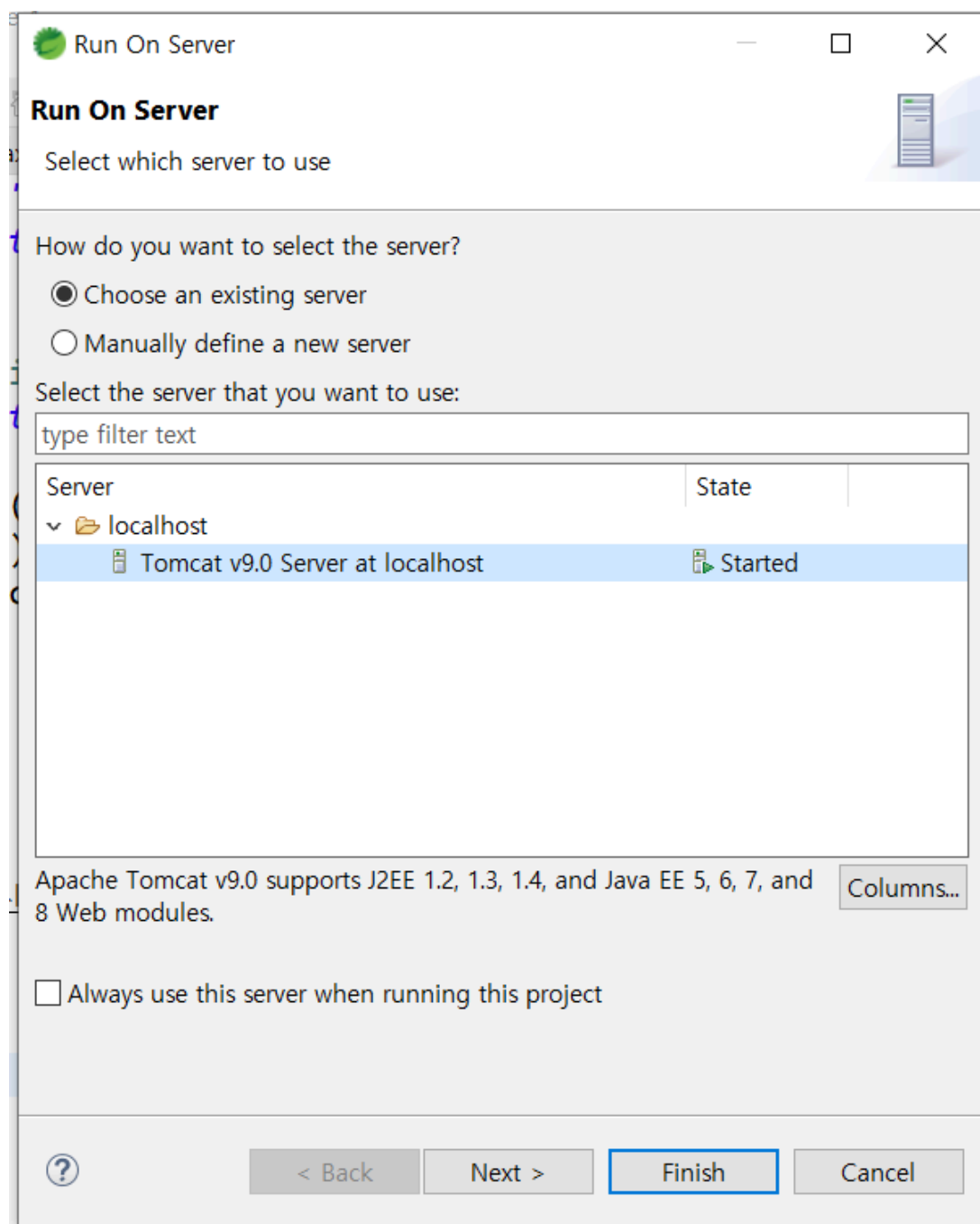
```

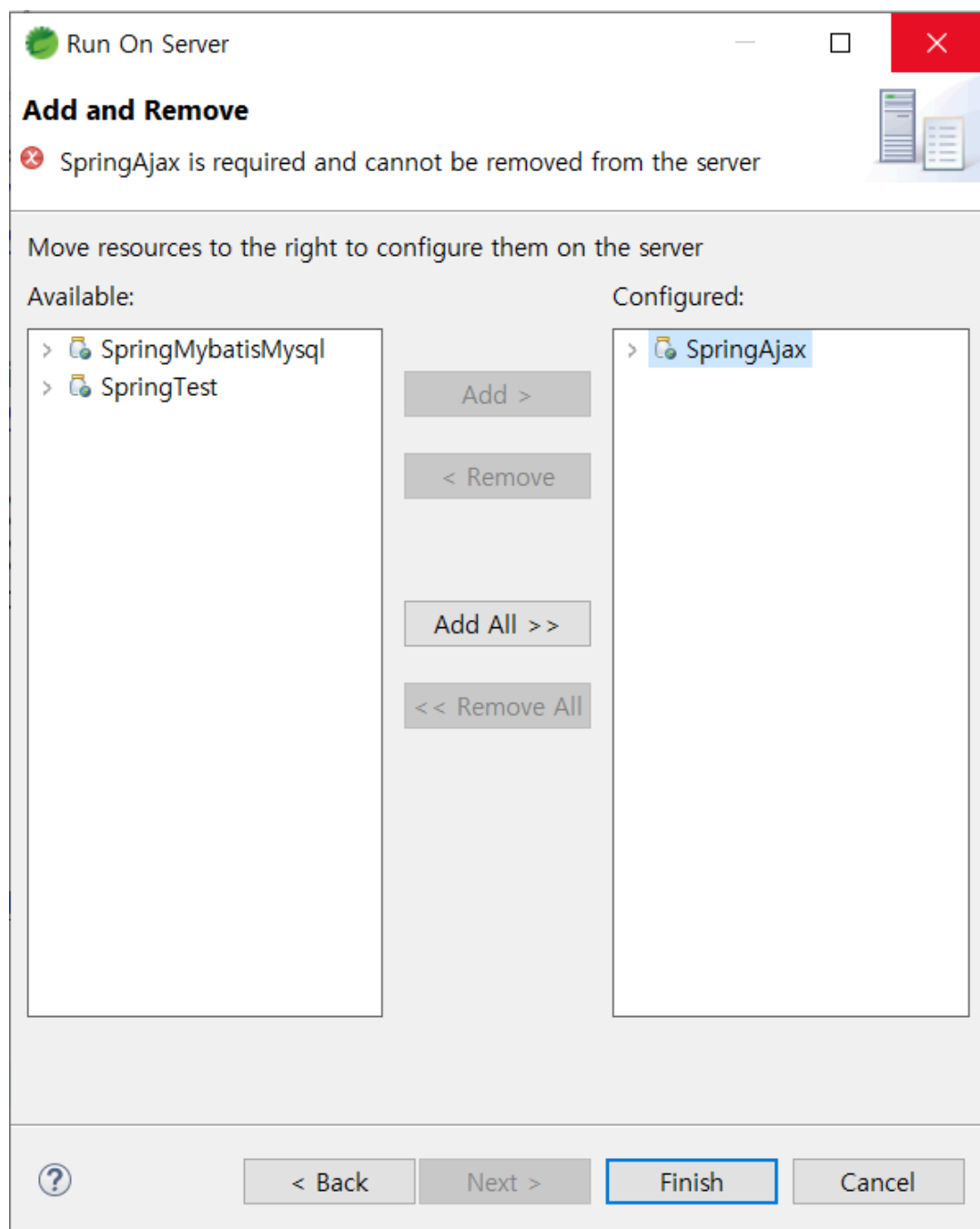
```

1 package com.campus.myapp;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5
6 @Controller
7 public class AjaxController {
8     @RequestMapping("/ajaxView")
9     public String ajaxViwe() {
10         return "ajax/ajaxView";
11     }
12 }

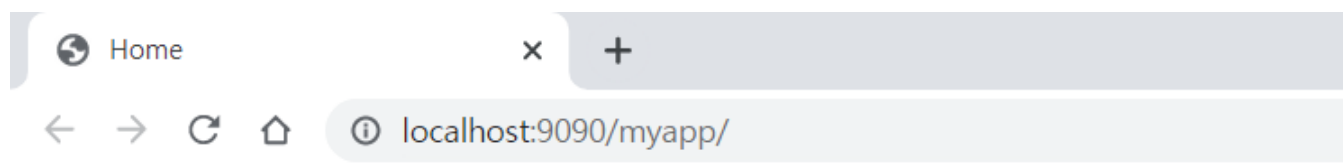
```

- AjaxController로 ajaxView페이지를 매핑해줍니다.



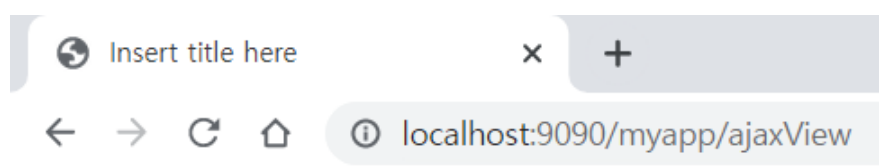


- 완료 되었으면 톰캣 서버를 구동시켜줍니다.



ajax(비동기식 통신)페이지로 이동하기

- ajax(비동기식 통신)페이지로 이동하기를 클릭하면



화면에 뜹니다

- ajaxView페이지로 잘 매핑된 것을 확인할 수 있습니다.

```
$.ajax({  
    url: 요청이 보내지는 곳,  
    type: http요청 방식 get/post,  
    success: http 요청 성공의 경우 동작할 함수,  
    error: http 요청 실패의 경우 동작할 함수,  
    complete: http 요청 완료 후 동작할 함수,  
    data: 서버로 넘어갈 값,  
    dataType: data의 타입,  
    global: 전역 함수 여부 true/false,  
    async: 동기 여부 true/false,  
});
```