

# 스프링(Spring) MVC : View

## (1) Outline

### View 까지의 흐름

- > 클라이언트로부터의 HTTP 요청을 [DispatcherServlet]이 받음
- > [DispatcherServlet]은 [HandlerMapping]을 통해 요청 URL을 처리할 [Controller]를 검색
- > [DispatcherServlet]은 요청 URL을 처리할 [Controller]에게 처리를 요청
- > [Controller]은 요청을 적절히 처리한 후 [ModelAndView]를 [DispatcherServlet]으로 전달
- > [DispatcherServlet]은 [ViewResolver]를 통해 처리 결과를 보여줄 [View]를 검색
- > [DispatcherServlet]은 처리 결과를 보여줄 [View]를 출력

### View

- DispatcherServlet은 ViewResolver를 사용하여 결과 출력 View 객체를 구함
- 해당 View 객체를 통해 출력 내용을 생성

## (2-1) ViewResolver

컨트롤러는 결과 생성 뷰의 이름을 지정

```
1 public class SomeController extends AbstractController{
2     ...
3
4     @Override
5     protected ModelAndView handleRequestInternal(HttpServletRequest request,
6                                                     HttpServletResponse response) throws
7     Exception{
8         ...
9
10        ModelAndView modelAndView = new ModelAndView();
11        ...
```

```

12     modelAndView.setViewName("someView");
13     ...
14     return modelAndView;
15 }
16}

```

## ViewResolver

- 컨트롤러에 지정한 뷰 이름으로 ViewResolver를 통해 View 객체를 구함

### ViewResolver 구현 클래스

#### - InternalResourceViewResolver

- > 뷰 이름으로 JSP, Tiles 연동을 위한 View 객체 리턴

#### - VelocityViewResolver

- > 뷰 이름으로 Velocity 연동을 위한 View 객체 리턴

#### - BeanNameViewResolver

- > 뷰 이름과 동일한 이름을 가지는 빈 객체를 View 객체로 사용

#### - ResourceBundleViewResolver

- > 뷰 이름과 View 객체 매핑 정보 저장을 위해 파일 사용

#### - XmlViewResolver

- > 뷰 이름과 View 객체 매핑 정보 저장을 위해 XML 파일 사용

### ViewResolver 인터페이스

- 뷰 이름과 지역화 설정을 위한 Locale을 파라미터로 전달받음
- 매핑된 View 객체를 리턴 (존재하지 않을 시 null)

```

1 package org.springframework.web.servlet;
2
3 import java.util.Locale;
4
5 public interface ViewResolver{
6     View resolveViewName(String viewName, Locale locale) throws Exception;
7
8 }

```

## (2-2) ViewResolver - View

## View 인터페이스

- ViewResolver를 통해 반환되는 뷰 객체는 모두 View 인터페이스를 구현
- `getContentType()` : "text/html" 같은 응답 결과 콘텐츠 타입을 리턴
- `render()` : 실제 응답 결과를 생성
- > Map model 파라미터에 컨트롤러가 리턴한 ModelAndView 객체의 모델 데이터 전달

```
1 package org.springframework.web.servlet;
2
3 import java.util.Map;
4
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7
8 public interface View{
9
10     String getContentType();
11
12     void render(Map model,
13                 HttpServletRequest request,
14                 HttpServletResponse response) throws Exception;
15 }
```

## (2-3) ViewResolver - InternalResourceViewResolver

### InternalResourceViewResolver

- AbstractUrlBasedView 타입의 뷰 객체를 리턴
  - <- AbstractUrlBasedView : JSP, HTML 등 웹 애플리케이션 내부 자원을 이용하여 뷰를 생성
- 기본 사용 View 클래스는 InternalResourceView
- 설정

```
1 <bean id="someViewResolver"
2     class="org.springframework.web.servlet.view.InternalResourceViewResolver"
```

```

3      p:prefix="/WEB-INF/viewPath/"
4      p:suffix=".jsp">
5
6      ...
7
8  </bean>

```

- InternalResourceViewResolver 는 컨트롤러가 지정한 뷰 이름으로 사용 뷰를 선택
- > 컨트롤러가 지정한 뷰 이름 앞뒤에 prefix, suffix 프로퍼티를 붙인 값이 뷰 파일이 됨

```

1 ModelAndView modelAndView = new ModelAndView("someView");
2 return modelAndView;

```

- > 이 경우 사용 뷰 경로는 [/WEB-INF/viewPath/someView.jsp]

## - 스프링 국제화 관련 정보를 JSTL에서 사용하고자 할 경우

- > JstlView 클래스를 viewClass 프로퍼티로 지정

```

1 <bean id="someViewResolver"
2     class="org.springframework.web.servlet.view.InternalResourceViewResolver"
3     p:viewClass="org.springframework.web.servlet.view.JstlView"
4     p:prefix="/WEB-INF/viewjsp/"
5     p:suffix=".jsp">
6
7     ...
8
9 </bean>

```

## (2-4) ViewResolver - BeanNameViewResolver

### BeanNameViewResolver

- 뷰 이름과 동일한 이름의 빈 객체를 뷰 객체로 사용
- 커스텀 View 클래스를 뷰로 사용해야 하는 경우 주로 사용
- 뷰 클래스 및 BeanNameViewResolver 등록

```
1 <bean id="viewResolver"
2 class="org.springframework.web.servlet.view.BeanNameViewResolver"/>
3
4 <bean id="someView" class="some.package.view.SomeView"/>
```

### ex) Controller

```
1 public class SomeController implements Controller{
2
3     ...
4
5     @Override
6     public ModelAndView handleRequest(HttpServletRequest request,
7                                     HttpServletResponse response) throws Exception{
8
9         ...
10
11         return new ModelAndView("someView");
12     }
13
14     ...
15 }
```

## (2-5) ViewResolver - XmlViewResolver

### XmlViewResolver

- 뷰 이름과 동일한 이름의 빈 객체를 뷰 객체로 사용
- 별도의 XML 설정 파일로부터 빈 객체를 검색
- location 프로퍼티에 외부 설정 파일 경로를 지정
  - > 지정하지 않을 경우 기본 값은 [/WEB-INF/views.xml]
  - > 지정 파일은 스프링 XML 설정 파일이어야 함

```
1 <bean id="viewResolver"
2     class="org.springframework.web.servlet.view.XmlViewResolver"
3     p:location="/WEB-INF/views.xml"/>
```

## (2-6) ViewResolver - ResourceBundleViewResolver

### ResourceBundleViewResolver

- 프로퍼티 파일으로 뷰 이름에 View 클래스를 매핑
  - > 프로퍼티 파일을 사용할 경우 Spring DI, AOP 사용 불가
- basename(s) 프로퍼티를 지정하여 뷰 클래스 정보가 저장된 경로를 명시

ex) views/someViewsProperty.properties, views/someViewsProperty.properties

```
1 <bean id="viewResolver"
2     class="org.springframework.web.servlet.view.ResourceBundleViewResolver"
3     p:basename="views/someViewsProperty"/>
```

- 프로퍼티 파일은 [뷰이름.class=뷰클래스] 와 같이 뷰 이름과 뷰 클래스를 매핑

```
1 someView.class = some.pakcage.view.SomeView
```

## (2-7) ViewResolver - Multiple ViewResolver

### ViewResolver를 여러개 설정하고자 할 경우

- order 프로퍼티 지정으로 ViewResolver의 우선 순위를 설정
- order 프로퍼티의 값이 낮을수록 높은 우선 순위
- 우선 순위를 명시하지 않을 경우, 가장 낮은 우선 순위로 지정됨
- 우선 순위가 높은 ViewResolver가 null 리턴시 다음 순위의 ViewResolver에 뷰를 요청

```
1 <bean class="org.springframework.web.servlet.view.XmlViewResolver"
2     p:location="/WEB-INF/someXmlViews.xml"
3     p:order="0"/>
4
5 <bean
6 class="org.springframework.web.servlet.view.InternalResourceViewResolver"
7     p:prefix="/WEB-INF/views/"
8     p:suffix=".jsp"
9     p:order="1" />
```

- 뷰 이름에 매핑되는 뷰 객체를 리턴하는 InternalResourceViewResolver는 마지막 우선 순위로 지정
- <- InternalResourceViewResolver는 null 리턴이 없으므로 우선순위가 높을시 다른 ViewResolver가 사용되지 않음
- VelocityViewResolver, VelocityLayoutViewResolver 경우도 낮은 우선순위
- <- 뷰 이름과 매핑되는 Velocity 템플릿이 없을 경우 null이 아닌 Exception 발생

### (3) HTML Special Character

출력값이 HTML 문법에 사용되는 특수문자("<", ">") 일 경우, "&lt;", "&gt;" 변환

#### 컨텍스트 파라미터 defaultHtmlEscape

- 스프링에서 특수문자 처리를 위해 지정하는 파라미터
- [true]로 지정될 경우, 특수문자를 변환
- [false]일 경우, 특수문자 그대로 출력
- 기본값은 [true]

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2
3  <web-app ... >
4
5  ...
6
7      <context-param>
8          <param-name>defaultHtmlEscape</param-name>
9          <param-value>false</param-value>
10     </context-param>
11
12 ...
13
14 </web-app>
```



## (4-1) JSP View

Spring에서 뷰를 JSP로 사용

- InternalTrsourceViewResolver 사용

```
1 <bean id="someViewResolver"
2     class="org.springframework.web.servlet.view.InternalResourceViewResolver"
3     p:prefix="/WEB-INF/viewPath/"
4     p:suffix=".jsp">
5
6 ...
7
8 </bean>
```

- JSP에 사용될 스프링의 커스텀 태그

- > <spring:message> : 메시지 국제화
- > <form:xxx> : 커맨드 객체와 HTML <form> 태그와의 연동
- > <form:errors> : 에러 처리