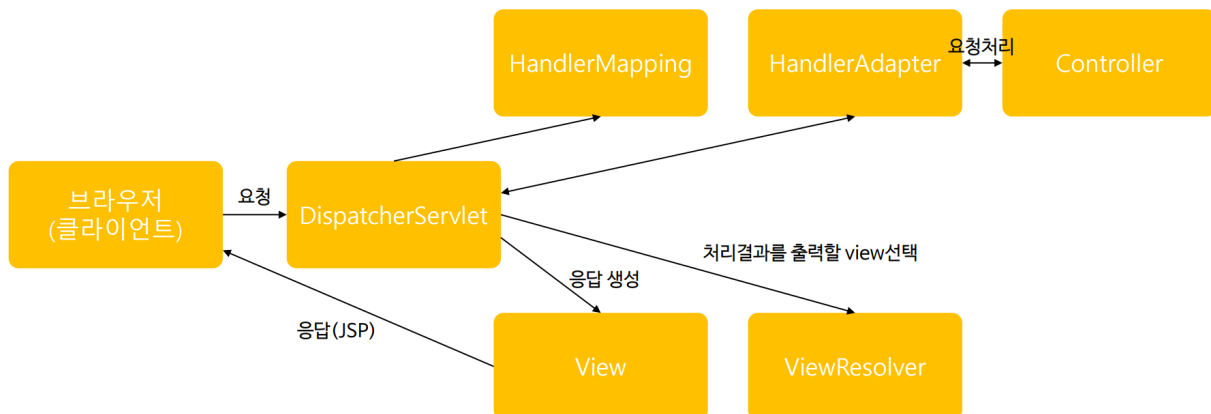


# 스프링 - Controller 01

## Front-Controller 패턴

req -(요청 ,URL ) -> DispatcherServlet -> 요청-> [HandlerMapping] 찾음 -> DispatcherServlet  
-> HandlerAdapter (DispatcherServlet이 전달한 컨트롤러를 찾음) -> Controller로 이동 ->  
DBMS등 이용 -> HandlerAdapter -> DispatcherServlet (돌아감) -> ViewResolver (jsp확장자  
페이지를 만들어주는 것)-> DispatcherServlet -> view 경로로 이동 -> (html, jsp등) -> view ->  
DispatcherServlet -> Response 마무리



## 정리

1. 사용자의 Request는 Front-Controller인 DispatcherServlet을 통해 처리된다.
2. HandlerMapping은 Request의 처리를 담당하는 컨트롤러를 찾기 위해서 존재한다. 여러 객체 중 @RequestMapping 어노테이션이 적용된 것을 기준으로 판단하며, 적절한 컨트롤러가 찾아졌다면 HandlerAdapter를 이용해서 해당 컨트롤러를 동작시킨다.
3. Controller는 Request를 처리하는 비즈니스 로직을 작성하며, View에 전달해야 하는 데이터를 RequestScope에 담아서 전달한다. 응답 페이지에 대한 경로 처리는 ViewResolver를 이용하게 된다.
4. ViewResolver는 Controller가 리턴한 결과 값을 전체 경로로 완성시켜준다.

- 
5. View는 실제 응답을 보내야하는 데이터를 HTMT, JSP등을 이용해서 생성하는 역할을 한다.
  6. 만들어진 응답 페이지를 DispatcherServlet을 통해 전송한다.

## Front-Controller 패턴의 특징

- HttpServletRequest, HttpServletResponse를 거의 사용할 필요 없이 기능 구현
- 다양한 타입의 파라미터 처리, 다양한 타입의 리턴 타입 사용 가능
- GET 방식, POST 방식 등 전송 방식에 대한 처리를 어노테이션으로 처리 가능
- 상속/인터페이스 방식 대신 어노테이션만으로도 설정 가능

## HomeController 이해

```
@RequestMapping(value = "/", method = RequestMethod.GET)
```

value 해당 위치에 따른 맵핑, method 속성

```
public String home(Locale locale, Model model) {
```

locale -> 시간

model!: data 전달자 -request도 Model 객체속에 있다.

```
logger.info("Welcome home! The client locale is {}.", locale);
```

포맷형 {}

```
model.addAttribute("serverTime", formattedDate );
```

request.setAttribute방식.

```
return "home";
```

이것을 servlet-context.xml에서

---

```
<beans:bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>
```

"home" => /WEB-INF/views/home.jsp 로 ViewResolver가 변경해준다.

## SampleController 생성

```
@Controller
@RequestMapping("/ex/*")
@Log4j
public class SampleController {

    @RequestMapping(value = "/basic", method = {RequestMethod.GET,
RequestMethod.POST})
    public void basic() {

    }

}
@RequestMapping("/ex/*")
```

/ 루트 경로 아래 ex 아래 모든 폴더 맵핑 => 클래스에도 사용가능하다.

```
@RequestMapping(value = "/basic", method = {RequestMethod.GET,
RequestMethod.POST})
/ex02/ex/basic =>이고 GET, POST 메소드일때 도달.
```