Interceptor

Interceptor

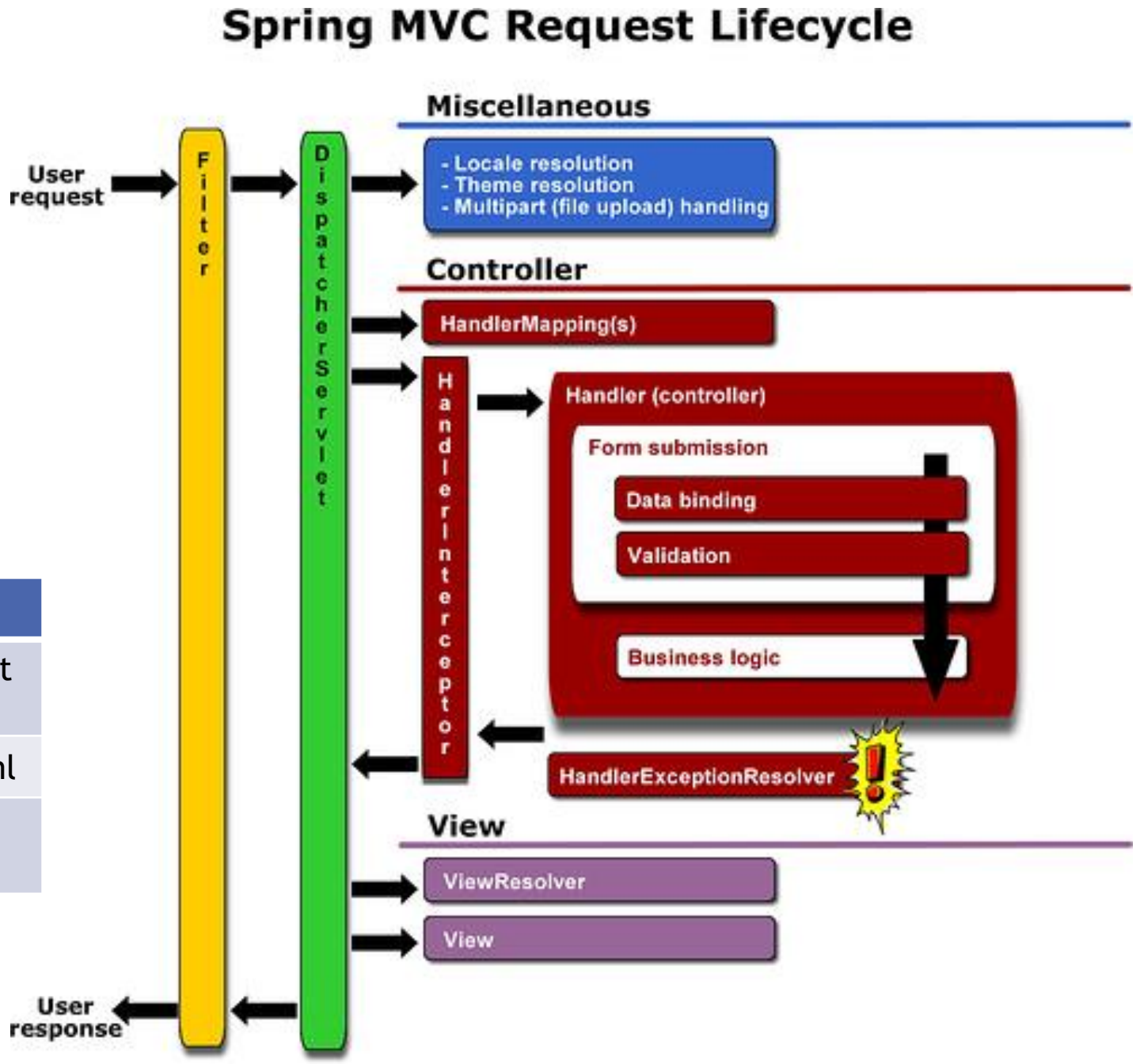
- 인터셉터란?
  - Spring에서 HTTP Request와 HTTP Response를 Controller 앞과 뒤에서 가로채는 역할을 수행
  - Servlet의 앞과 뒤에서 HTTP Request와 HTTP Response를 가로채는 필터와 유사함
  - Interceptor를 구현하기 위해서는 HandleInterceptor 인터페이스를 구현해야 한다

Interceptor

- 서블릿 필터와의 차이

1. 호출되는 시점
2. 설정 위치
3. 구현방식

	필터	인터셉터
호출 시점	DispatcherServlet 실행 이전	DispatcherServlet 실행 이후
설정 위치	web.xml	spring-servlet.xml
구현 방식	web.xml에 설정	설정 후 메서드 구현 필요



Interceptor
: 구현 예제

- Custom Interceptor 구현

  HandlerInterceptor 인터페이스의 3개의 메서드를 구현한다

```
public interface HandlerInterceptor {
    boolean preHandle( HttpServletRequest request, HttpServletResponse response,
        Object handler) throws Exception;

    void postHandle( HttpServletRequest request, HttpServletResponse response,
        Object handler, ModelAndView modelAndView) throws Exception;

    void afterCompletion( HttpServletRequest request, HttpServletResponse response,
        Object handler, Exception ex) throws Exception;
}
```

1. preHandle() 메서드는 컨트롤러가 호출되기 전에 실행된다. handler 파라미터는 HandlerMapping이 찾아준 컨트롤러의 메서드를 참조할 수 있는 HandlerMethod 오브젝트이다
   반환값이 true이면 핸들러의 다음 체인이 실행되지만 false이면 중단하고 남은 Interceptor와 컨트롤러가 실행되지 않는다
2. postHandler() 메서드는 컨트롤러가 실행된 후에 호출된다
3. afterCompletion()은 뷰에서 최종 결과가 생성되는 일을 포함한 모든 일이 완료되었을 때 실행된다

Interceptor
: 구현 예제

- Custom Interceptor 등록 (in spring-servlet.xml)

```
<!-- Interceptors -->
<mvc:interceptors>
    <mvc:interceptor>
    <mvc:mapping path="/board/**" />
    <bean class="com.example.mysite.interceptor.MyInterceptor" />
    </mvc:interceptor>
</mvc:interceptors>
```

  - 핸들러 인터셉터는 하나 이상 등록할 수 있으며 등록 순서대로 preHandle() 이 실행된다
  - postHandle()과 afterCompletion()은 그 반대로 실행된다

- 테스트
  1. mapping path 변경해 보기
  2. mapping path 여러 개 추가해 보기
  3. MyInterceptor의 preHandle 리턴 값 변경해보기

Interceptor
: 구현 예제

- HandlerInterceptorAdapter 추상 클래스를 상속하여 구현할 수 있다
  - HandlerInterceptorAdapter는 HandlerInterceptor 인터페이스 기본 메서드를 구현하여 놓았기 때문에 상속한 인터셉터가 필요한 메서드만 오버라이딩하여 다시 구현하면 된다

- HandlerInterceptorAdapter를 상속하여 MyInterceptor2를 작성한 후, 설정하고 로그를 확인해 봅시다

AuthInterceptor 구현하기

- 인증 여부에 따른 특정 URL 접근 제한
- /user/login URL 처리
  - Interceptor에서는 ApplicationContext를 구해서 직접 저장된 Bean을 가져와야 한다

```
ApplicationContext applicationContext =
    WebApplicationContextUtils.getWebApplicationContext( request.getServletContext() );
UserService userService = applicationContext.getBean( UserService.class );
```

- /user/logout URL 처리