

Spring week 9

유채린

chaerin.du.ub@gmail.com



해당 내용은 인프런 강의를 정리한 내용입니다.
잘못된 내용에 대하여 피드백주시면 감사하겠습니다!

(인프런 아이콘 클릭 시, 참고한 인프런 강의 사이트로 이동합니다.)

Index

- ThymeLeaf 타임리프
- 리다이렉트

ThymeLeaf 타임리프

ThymeLeaf (타임리프)



- View 템플릿
- JSP와 다르게 HTML 확장자를 갖고 있어 서버 없이도 동작 가능하다.
- 서버에서 받아온 데이터를 \${}을 이용하여 표기한다.
- th:XXXX 가 붙은 부분은 서버 사이드에서 렌더링 되고, 기존 것을 대체한다.
th:XXX가 없으면 기존 HTML의 XXXX 속성이 그대로 사용된다.
 - HTML을 파일로 직접 열었을 때 th:XXXX가 있어도 웹 브라우저는 th: 속성을 알지 못하므로 무시한다.

HTML에서 타임리프 사용 선언하는 방법

```
<!DOCTYPE HTML>  
<html xmlns:th="http://thymeleaf.org">  
<head>  
  
...
```

Thymeleaf 문법

- 리터럴 대체 |...|
- 타임 리프에서 문자와 표현식은 분리되어 있어 더해서 사용해야 한다.

예시 1)

- ~~~~
-

예시 2)

- ~~th:onclick="'location.href=' + 'W' + '@{/basic/items/add}' + 'W'"~~
- th:onclick="|location.href='@{/basic/items/add}'|"

Thymeleaf 문법

- 반복 출력 th:each
- `<tr th:each="item : ${items}">`
 - 모델에 포함된 items 컬렉션 데이터가 item 변수에 하나씩 포함되고 반복문 안에서 item 변수를 사용할 수 있다.
 - 컬렉션의 수 만큼 `<tr>`이 하위 태그를 포함해 생성된다.
- 변수 표현식 `${...}`, 내용 변경 th:text
- `<td th:text="${item.price}">1000</td>`
 - 모델에 포함된 값이나 타임리프 변수로 선언한 값 조회 가능하다.
 - 프로퍼티 접근법을 사용한다. 예) `item.getPrice()`
- 내용 값을 th:text 값으로 변경한다. 여기서는 1000을 `${item.price}`의 값으로 변경한다.

Thymeleaf 문법

- URL 링크 표현식 @{...}
- th:href="@{/css/bootstrap.min.css}"
- th:href="@{/basic/items/{itemId}(itemId=\${item.id})}"
- 경로 변수({itemId}) 뿐만 아니라 쿼리 파라미터도 생성한다.
 - 예시) th:href="@{/basic/items/{itemId}(itemId=\${item.id}, query='test')}"
 - 생성 링크: <http://localhost:8080/basic/items/1?query=test>
- th:href="@{/basic/items/\${item.id]}"
- 리터럴 대체 문법을 활용해서 간단히 사용할 수도 있다.
- 속성 변경 th:action
- HTML form에서 action 에 값이 없으면 현재 URL에 데이터를 전송한다.

리다이렉트

리다이렉트

- 스프링은 redirect:/... 으로 편리하게 리다이렉트를 지원한다.

```
@PostMapping("/{itemId}/edit")
public String edit(@PathVariable Long itemId, @ModelAttribute Item item) {
    itemRepository.update(itemId, item);
    // redirect
    return "redirect:/basic/items/{itemId}";
}
```

컨트롤러에 매핑된 @PathVariable 의 값은 redirect 에도 사용 할 수 있다.

리다이렉트

- 웹 브라우저 새로 고침은 마지막에 서버에 전송한 데이터를 다시 전송한다.
- 만약 데이터 저장 후, 새로 고침을 하게 되면 여러 번 저장하는 중복 문제가 발생한다.
- 이러한 문제를 해결하기 위해서, 우리는 마지막 요청으로 GET을 보내면 된다.
(예시) 저장 후 상세화면 조회
→ RPG Post/Redirect/Get

RedirectAttributes

- RedirectAttributes 를 사용하면 URL 인코딩도 해주고, pathVariable, 쿼리 파라미터까지 처리해준다.
 - redirect:/basic/items/{itemId}
 - pathVariable 바인딩: {itemId}
 - 나머지는 쿼리 파라미터로 처리: ?status=true

```
@PostMapping("/add")
public String addItemV6(Item item, RedirectAttributes redirectAttributes) {
    Item savedItem = itemRepository.save(item);
    redirectAttributes.addAttribute(attributeName: "itemId", savedItem.getId());
    redirectAttributes.addAttribute(attributeName: "status", attributeValue: true);

    return "redirect:/basic/items/{itemId}";
}
```

PathVariable

Query parameter

감사합니다.