

Spring MVC2 03

[타임리프 - 기본 기능 3]

스프링 MVC 2편 - 백엔드 웹 개발 활용 기술

김영한

2022.10.17

타임리프 자바스크립트 인라인

타임리프는 자바스크립트를 편하게 사용하게 해준다.

<script th:inline="javascript">

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<!-- 자바스크립트 인라인 사용 전 -->
<script>
  var username = [[${user.username}]];
  var age = [[${user.age}]];
  //자바스크립트 내추럴 템플릿
  var username2 = /*[[${user.username}]]*/ "test username";
  //객체
  var user = [[${user}]];
</script>
<!-- 자바스크립트 인라인 사용 후 -->
<script th:inline="javascript">
  var username = [[${user.username}]];
  var age = [[${user.age}]];
  //자바스크립트 내추럴 템플릿
  var username2 = /*[[${user.username}]]*/ "test username";
  //객체
  var user = [[${user}]];
</script>
</body>
</html>
```

페이지 소스보기

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<!-- 자바스크립트 인라인 사용 전 -->
<script>
  var username = UserA;
  var age = 10;
  var username2 = /*UserA*/ "test username";
  var user = BasicController.User(username=UserA, age=10);
</script>
<!-- 자바스크립트 인라인 사용 후 -->
<script>
  var username = "UserA";
  var age = 10;
  //자바스크립트 내추럴 템플릿
  var username2 = "UserA";
  var user = {"username":"UserA", "age":10};
</script>
</body>
</html>

```

장점

문자열 이라면 "" 를 추가해준다.

내추럴 템플릿을 지원하게 /* */주석처리로 사용되는 것을 정하고 정적페이지도 열 수 있다.

객체를 JSON으로 자동으로 형변환 해준다.

자바 스크립트 인라인 **each**

```

<script th:inline="javascript">
  [# th:each = "user, stat : ${users}"]
  var user[`${stat.count}`] = [`${user}`];
  [/]
</script>

```

```
var user1 = {"username":"UserA","age":10};
```

```
var user2 = {"username":"UserB","age":20};
```

```
var user3 = {"username":"UserC","age":30};
```

형식으로 렌더링된다.

템플릿 조각

웹페이지의 공통조각이 많다. 이것을 처리하는 것에 대해 타임 리프 템플릿 조각과 레이아웃 기능을 지원한다.

footer.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<body>
<footer th:fragment="copy">
    푸터 자리 입니다.
</footer>
<footer th:fragment="copyParam (param1, param2)">
    <p>파라미터 자리 입니다.</p>
    <p th:text="${param1}"></p>
    <p th:text="${param2}"></p>
</footer>
</body>
</html>
```

main에서 사용시

```
<div th:insert=~{template/fragment/footer :: copy}></div>
```

형식으로 삽입가능하다. 푸터html 에 copy

```
<h2>부분 포함 insert</h2>
<div th:insert=~{template/fragment/footer :: copy}></div>
<h2>부분 포함 replace</h2>
<div th:replace=~{template/fragment/footer :: copy}></div>
<h2>부분 포함 단순 표현식</h2>
```

```
<div th:replace="template/fragment/footer :: copy"></div>
<h1>파라미터 사용</h1>
<div th:replace="~{template/fragment/footer :: copyParam ('데이터1', '데이터
2')}"></div>
```

파라미터를 사용해서 동적으로 랜더링도 가능하다.

템플릿 레이아웃

코드 조각을 넘겨서 사용하는 방법

base.html

```
<html xmlns:th="http://www.thymeleaf.org">
<head th:fragment="common_header(title,links)">
  <title th:replace="${title}">레이아웃 타이틀</title>
  <!-- 공통 -->
  <link rel="stylesheet" type="text/css" media="all"
th:href="@{/css/awesomeapp.css}">
  <link rel="shortcut icon" th:href="@{/images/favicon.ico}">
  <script type="text/javascript" th:src="@{/sh/scripts/codebase.js}"></script>
  <!-- 추가 -->
  <th:block th:replace="${links}" />
</head>
```

title과 link를 변경하거나 추가하고 싶으면

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head th:replace="template/layout/base :: common_header(~{::title},~{::link})">
  <title>메인 타이틀</title>
  <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
  <link rel="stylesheet" th:href="@{/themes/smoothness/jquery-ui.css}">
</head>
<body>
메인 컨텐츠
</body>
</html>
```

공통부분이 아닌 다른 부분들을 레이아웃을 사용하게 된다.

헤더 뿐만 아니라 다른 영역에도 레이아웃을 넣을 수 있다. 뿐만 아니라 **html** 전체에 적용가능

레이아웃 틀. **layout**

```
<!DOCTYPE html>
<html th:fragment="layout (title, content)" xmlns:th="http://www.thymeleaf.org">
<head>
  <title th:replace="${title}">레이아웃 타이틀</title>
</head>
<body>
<h1>레이아웃 H1</h1>
<div th:replace="${content}">
  <p>레이아웃 콘텐츠</p>
</div>
<footer>
  레이아웃 푸터
</footer>
</body>
</html>
```

적용 시킬 콘텐츠를 넣은것 main

```
<!DOCTYPE html>
<html th:replace="~{template/layoutExtend/layoutFile ::
layout(~{::title},~{::section})}"
  xmlns:th="http://www.thymeleaf.org">
<head>
  <title>메인 페이지 타이틀</title>
</head>
<body>
<section>
  <p>메인 페이지 콘텐츠</p>
  <div>메인 페이지 포함 내용</div>
</section>
</body>
</html>
```

~{::title} 의 형식으로 자신이 가진 것을 파라미터 넘기는 것 처럼 보내는 것.

```
<title th:replace="${title}">레이아웃 타이틀</title>
```

이 부분이 넘어온 부분으로 랜더링 되어 보여주게 된다.