

Mustache 템플릿 문법 가이드

Mustache는 로직 없는 템플릿 문법입니다. 모든 태그는 `{{`와 `}}`로 감싸서 사용합니다.

기본 문법

1. 변수 출력 (Variables)

변수 값을 출력합니다.

```
{{name}}
```

- 변수가 존재하면 값이 출력됩니다
- 변수가 없거나 null이면 아무것도 출력되지 않습니다
- HTML 이스케이프가 자동으로 처리됩니다

예시:

```
<h1>{{title}}</h1>
<p>가격: {{price}}원</p>
```

2. HTML 이스케이프 없이 출력 (Unescaped)

HTML 태그를 그대로 출력하려면 `{{{` 또는 `{{&`를 사용합니다.

```
{{{html_content}}}
{{&html_content}}
```

예시:

```
<div>{{{description}}}</div>
```

3. 섹션 (Sections)

조건부 렌더링과 반복문에 사용됩니다.

3.1. Truthy 섹션

변수가 존재하고 truthy 값이면 섹션 내용을 렌더링합니다.

```
{{#section}}
내용
{{/section}}
```

예시:

```
{{#user}}
<p>사용자: {{name}}</p>
{{/user}}
```

3.2. 리스트 반복

배열이나 리스트를 반복합니다.

```
{{#items}}
<li>{{name}}</li>
{{/items}}
```

예시:

```
{{#models}}
<tr>
  <td>{{id}}</td>
  <td>{{name}}</td>
</tr>
{{/models}}
```

3.3. False/Empty 섹션 (Inverted Section)

변수가 없거나 false, 빈 배열이면 섹션 내용을 렌더링합니다.

```
{{^section}}
내용이 없을 때 표시
{{/section}}
```

예시:

```
{{^products}}
<p>상품이 없습니다.</p>
{{/products}}
```

4. 주석 (Comments)

주석은 렌더링되지 않습니다.

```
{{! 이것은 주석입니다 }}
```

예시:

```
{{! 이 부분은 출력되지 않습니다 }}  
<h1>제목</h1>
```

5. 부분 템플릿 (Partials)

다른 템플릿 파일을 포함합니다.

```
{{> partial_name}}
```

예시:

```
{{> header}}  
<main>내용</main>  
{{> footer}}
```

6. 설정 변경자 (Set Delimiter)

구분자를 변경할 수 있습니다 (거의 사용하지 않음).

```
{{=<% %>=}}  
<%변수%>  
{{={{{ }}=}}}
```

실제 사용 예시

리스트 반복

```
<table>  
  <tr>  
    <th>ID</th>  
    <th>이름</th>  
  </tr>  
  {{#models}}
```

```
<tr>
  <td>{{id}}</td>
  <td><a href="/product/{{id}}">{{name}}</a></td>
</tr>
{{/models}}
</table>
```

조건부 렌더링

```
{{#user}}
  <p>안녕하세요, {{name}}님!</p>
{{/user}}

{{^user}}
  <p>로그인이 필요합니다.</p>
{{/user}}
```

중첩된 객체 접근

```
{{#product}}
  <h2>{{name}}</h2>
  <p>가격: {{price}}원</p>
  <p>재고: {{qty}}개</p>

  {{#category}}
    <p>카테고리: {{name}}</p>
  {{/category}}
{{/product}}
```

주의사항

1. **로직 없음**: Mustache는 if/else, for 루프 같은 로직이 없습니다. 데이터를 미리 준비해야 합니다.
2. **스페이스 민감**: `{{#section}}`과 `{{/section}}` 사이의 들여쓰기와 공백이 그대로 유지됩니다.
3. **변수명 규칙**:
 - 대소문자 구분
 - 영문, 숫자, 언더스코어(_) 사용 가능
 - 점(.)으로 중첩 객체 접근 가능
4. **HTML 이스케이프**: 기본적으로 HTML 특수문자(<, >, & 등)가 자동으로 이스케이프됩니다.

Spring Boot에서의 사용

Spring Boot에서 Mustache를 사용할 때:

```
@GetMapping("/product")
public String list(HttpServletRequest req) {
    List<Product> models = service.상품목록();
    req.setAttribute("models", models);
    return "list"; // list.mustache 템플릿 사용
}
```

템플릿에서:

```
{{#models}}
  <div>{{name}}</div>
{{/models}}
```

참고 자료

- [Mustache 공식 문서](#)
- [Mustache Manual](#)