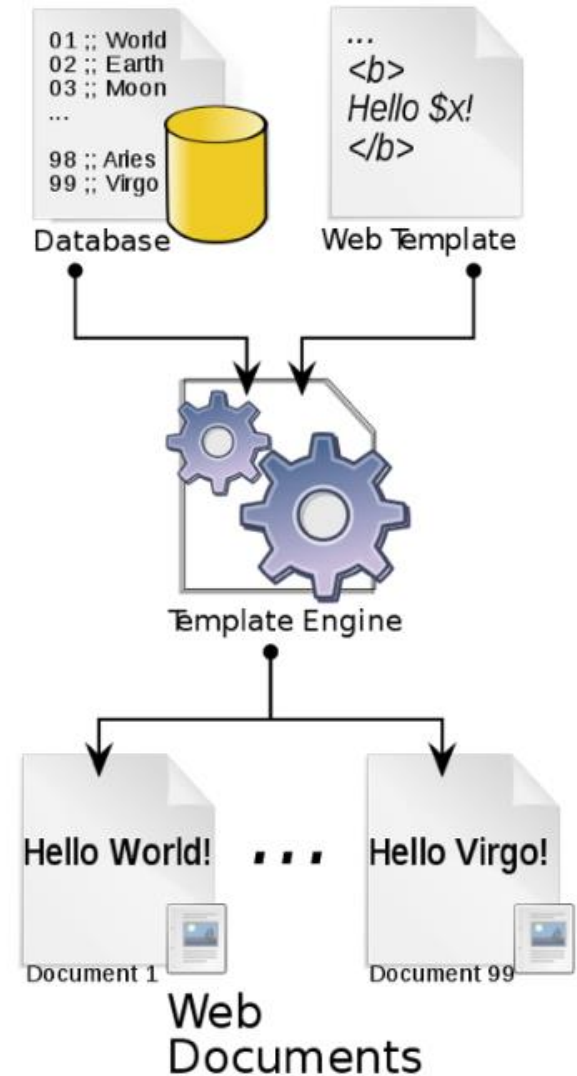


Thymeleaf

Spring boot

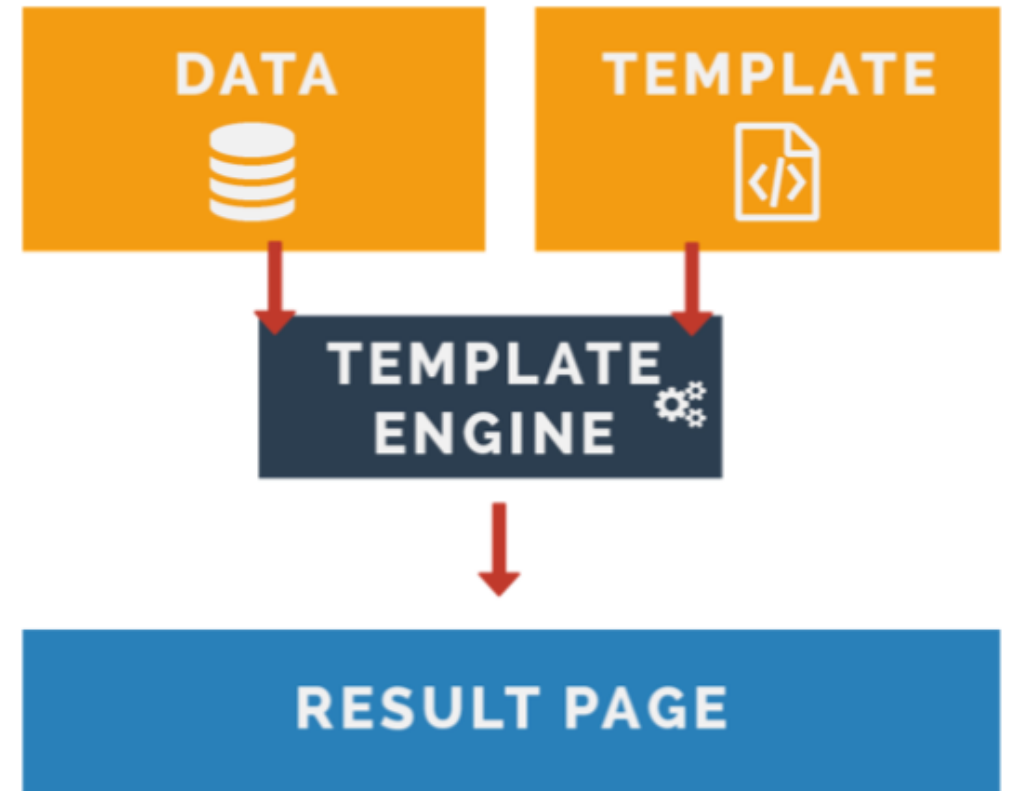
Template Engine이란

- 템플릿 양식과 특정 데이터 모델에 따른 입력 자료를 합성하여 결과 문서를 출력하는 소프트웨어
- Web Template Engine
 - 웹 문서가 출력되는 템플릿 엔진
 - 웹 템플릿들(web templates)과 웹 콘텐츠 정보(content information)를 처리하기 위해 설계된 소프트웨어
 - view code(html)와 data logic code(db connection)를 분리해주는 기능



Template Engine 종류

- Thymeleaf
- Freemarker
- Mustache
- Groovy



Template Engine 장점

많은 코드를 줄일 수 있다

- 기존의 HTML에 비해서 간단한 문법을 사용

재사용성이 높다

- 동일한 디자인의 페이지에 데이터만 변경하여 사용

유지보수에 용이하다

- 하나의 TEMPLATE 만들어 여러 페이지를 렌더링하는 작업

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-thymeleaf</artifactId>  
</dependency>
```

Thymeleaf

- <https://www.thymeleaf.org/>

```
#Thymeleaf Cache Setting  
spring.thymeleaf.cache=false
```

Thymeleaf 기본설정

- 파일의 확장자는 '.html'
- 인코딩은 'UTF-8'
- MINE 타입은 'text/html'
- 서버의 내부 cache는 'true'

```
hello.html ×
1 <!DOCTYPE html>
2 <html xmlns:th="http://thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>Test</title>
6   </head>
7   <body>
8     <h3>Hello~ Spring Boot!</h3>
9   </body>
10 </html>
```

```
@RequestMapping("/hello")
public void hello() {
    LOGGER.info("hello()가 호출되었습니다");
}
```

템플릿페이지 작성

- 기본 폴더 : src/main/resource/templates
- 생성 : [New] → [File]
- <html xmlns:th="http://thymeleaf.org">

Thymeleaf 문법

- <https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html#introducing-thymeleaf>
- HTML 태그의 속성으로 지정
 - `<html xmlns:th="http://thymeleaf.org">`
- DTO 객체 값 출력
 - `<td th:text="${board.seq}" />`
- 반복문 처리
 - `<tr th:each="board : ${boardList}">`
- 링크 처리
 - `<a th:href="@{/insertBoard}">새글 등록`
- Thymeleaf 내부 객체
 - `<td th:text="${#temporals.format(board.regDate, 'yyyy-MM-dd')}" />`

Controller

```
package com.edu.controller;
```

~ 생략 ~

```
@Controller
```

```
@RequiredArgsConstructor
```

```
public class BoardController2 {
```

```
    private final BoardService boardService;
```

```
    @GetMapping(value = "/")
```

```
    public String main() {
```

```
        return "redirect:/getBoardList";
```

```
    }
```

```
    @GetMapping(value = "/insertBoard")
```

```
    public String boardForm(Model model) {
```

```
        model.addAttribute("boardDto", new BoardDto());
```

```
        return "boardForm";
```

```
    }
```

```
    @PostMapping(value = "/insertBoard")
```

```
    public String insertBoard(BoardDto dto) {
```

```
        boardService.insertBoard(dto);
```

```
        return "redirect:/getBoardList";
```

```
    }
```

```

@PostMapping(value = "/updateBoard")
public String updateBoard(BoardDto dto) {
    boardService.updateBoard(dto);
    return "redirect:/getBoardList";
}

@GetMapping(value = "/deleteBoard/{seq}")
public String deleteBoard(@PathVariable Long seq) {
    boardService.deleteBoard(seq);
    return "redirect:/getBoardList";
}

@GetMapping(value = "/getBoard/{seq}")
public String getBoard(@PathVariable Long seq, Model model) {
    BoardDto board = boardService.getBoard(seq);
    model.addAttribute("board", board);
    return "getBoard";
}

@GetMapping(value = "/getBoardList")
public String getBoardList(Model model) {
    List<BoardDto> boardList = boardService.getBoardList();
    model.addAttribute("boardList", boardList);
    return "getBoardList";
}
}

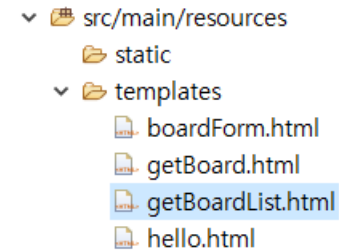
```

View

```

<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
  <head>
    <meta charset="UTF-8">
    <title>글목록</title>
    <style type="text/css">
      table {width: 700px; border: 1px solid black; }
      th    {background-color: orange; width: 200px; }
      th, td{border: 1px solid black;}
    </style>
  </head>
  <body th:align="center">
    <h1>게시글 목록</h1>
    <table th:align="center">
      <tr><th>번호</th><th>제목</th><th>작성자</th><th>등록일</th></tr>
      <tr th:each="board : ${boardList}">
        <td th:text="${board.seq}" />
        <td><a th:href="@{/getBoard/}+${board.seq}" th:text="${board.title}"></a></td>
        <td th:text="${board.writer}" />
        <td th:text="${#temporals.format(board.regDate, 'yyyy-MM-dd')}" />
      </tr>
    </table>
    <p><a th:href="@{/insertBoard}">새글 등록</a></p>
  </body>
</html>

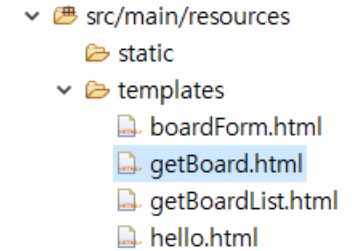
```



```

<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
  <head>
    <meta charset="UTF-8">
    <title>글보기</title>
    <style type="text/css">
      table { width: 700px;border: 1px solid black;}
      .rh {background-color: orange;width: 80px;}
      th,td {border: 1px solid black;}
    </style>
  </head>
  <body th:align="center">
    <h1>게시글 상세 보기</h1>
    <form action="/updateBoard" method="post">
      <input type="hidden" name="seq" th:value="${board.seq}" />
      <table th:align="center">
        <tr><td class="rh">제목</td>
          <td><input type="text" name="title" th:value="${board.title}" /></td>
        </tr>
        <tr><td class="rh">작성자</td>
          <td><input type="text" name="writer" th:value="${board.writer}" /></td>
        </tr>
        <tr><td class="rh">내용</td>
          <td><textarea name="content" th:text="${board.content}" cols="40" rows="10"></textarea></td></tr>
        <tr><td class="rh">등록일</td>
          <td><input type="text" th:text="${#temporals.format(board.regDate, 'yyyy-MM-dd')}" /></td>
        </tr>
        <tr><td colspan="2"><input type="submit" value="글수정" /></td></tr>
      </table>
    </form>
    <p>
      <a th:href="@{/getBoardList}">글목록</a>
      <a th:href="@{/deleteBoard/{seq}}">글삭제</a>
    </p>
  </body>
</html>

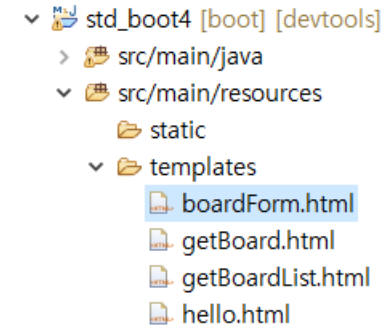
```



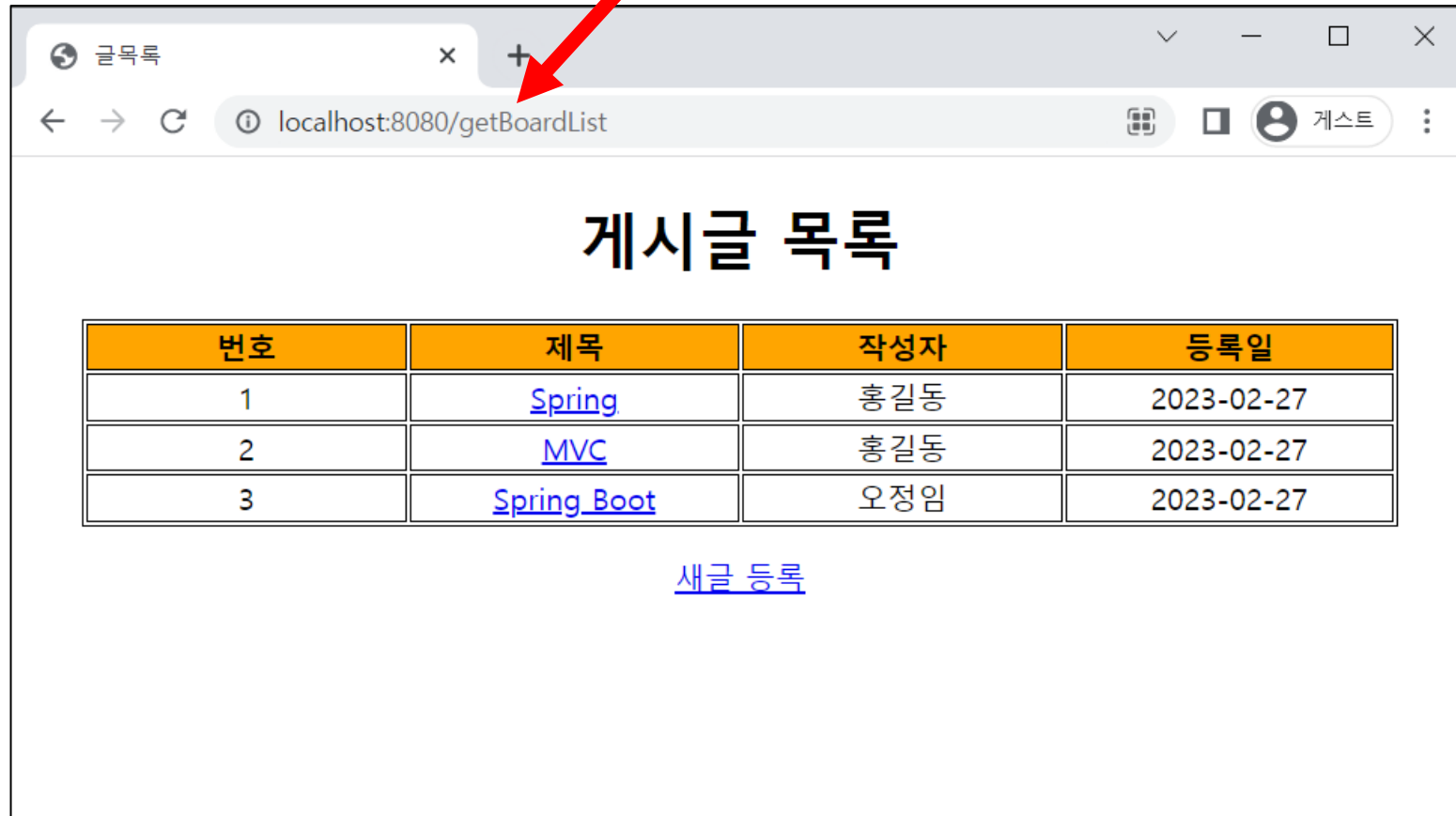
```

<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
  <head>
    <meta charset="UTF-8">
    <title>새글 등록</title>
    <style type="text/css">
      table { width: 700px;border: 1px solid black;}
      .rh {background-color: orange;width: 80px;}
      th, td {border: 1px solid black;}
    </style>
  </head>
  <body th:align="center">
    <h1>게시글 등록</h1>
    <form action="insertBoard" method="post">
      <table th:align="center">
        <tr><td class="rh">제목</td> <td><input type="text" name="title" /></td></tr>
        <tr><td class="rh">작성자</td><td><input type="text" name="writer" /></td></tr>
        <tr><td class="rh">내용</td> <td><textarea name="content" cols="40" rows="10"></textarea></td></tr>
        <tr><td colspan="2"><input type="submit" value="게시글 등록" /></td></tr>
      </table>
    </form>
    <p><a th:href="@{/getBoardList}">글목록</a></p>
  </body>
</html>

```



http://localhost:8080/ 입력



게시글 목록

번호	제목	작성자	등록일
1	Spring	홍길동	2023-02-27
2	MVC	홍길동	2023-02-27
3	Spring_Boot	오정임	2023-02-27

[새글 등록](#)

글 목록

localhost:8080/getBoardList

게스트

게시글 목록

번호	제목	작성자	등록일
1	Spring	홍길동	2023-02-27
2	MVC	홍길동	2023-02-27
3	Spring_Boot	오정임	2023-02-27

[새글 등록](#)

새글 등록

localhost:8080/insertBoard

게스트

게시글 등록

제목

작성자가

내용

게시글 등록

[글 목록](#)

새글 등록

localhost:8080/insertBoard

게시글 등록

제목	JPA
작성자	김푸름
내용	자바 ORM 기술 표준
	<div>게시글 등록</div>

[글목록](#)

글목록

localhost:8080/getBoardList

게시글 목록

번호	제목	작성자	등록일
1	Spring	홍길동	2023-02-27
2	MVC	홍길동	2023-02-27
3	Spring_Boot	오정임	2023-02-27
4	JPA	김푸름	2023-02-27

[새글 등록](#)

글목록

localhost:8080/getBoardList

게스트

게시글 목록

번호	제목	작성자	등록일
1	Spring	홍길동	2023-02-27
2	MVC	홍길동	2023-02-27
3	Spring Boot	오정임	2023-02-27
4	JPA	김푸름	2023-02-27

[새글 등록](#)

글보기

localhost:8080/getBoard/4

게스트

게시글 상세 보기

제목	JPA
작성자	김푸름
내용	자바 ORM 기술 표준
등록일	2023-02-27

글수정

[글목록](#) [글삭제](#)

Validation

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>

```

어노테이션	설명
@NotEmpty	NULL 체크 및 문자열의 경우 길이 0인지 검사
@NotBlank	NULL 체크 및 문자열의 경우 길이 0 또는 빈 문자열(" ") 검사
@Length(min=, max=)	최소, 최대 길이 검사
@Email	이메일 형식인지 검사
@Max(숫자)	지정한 값보다 작은지 검사
@Min(숫자)	지정한 값보다 큰지 검사
@Null	값이 NULL인지 검사
@NotNull	값이 NULL이 아닌지 검사

<Sample Code>

```
import javax.validation.constraints.Email;
import javax.validation.constraints.NotBlank;
import javax.validation.constraints.NotEmpty;

@Getter @Setter
public class MemberFormDto {

    @NotBlank(message = "이름은 필수 입력 값입니다.")
    private String name;

    @NotEmpty(message = "이메일은 필수 입력 값입니다.")
    @Email(message = "이메일 형식으로 입력해주세요.")
    private String email;

    @NotEmpty(message = "비밀번호는 필수 입력 값입니다.")
    @Length(min=8, max=16, message = "비밀번호는 8자 이상, 16자 이하로 입력해주세요")
    private String password;

    @NotEmpty(message = "주소는 필수 입력 값입니다.")
    private String address;
}
```

```

package com.edu.dto;

import javax.validation.constraints.NotBlank;
import javax.validation.constraints.NotEmpty;
import org.hibernate.validator.constraints.Length;

~ 생략 ~
@Getter
@Setter
@ToString
public class BoardDto {
    private Long seq;

    @NotBlank(message = "제목을 입력해 주세요.")
    private String title;

    private String writer;

    @NotEmpty(message = "내용을 입력해 주세요.")
    @Length(min = 5, max = 50, message = "내용을 5 글자 이상, 50 글자 이하로 입력해주세요.")
    private String content;

    ~ 생략 ~
}

```

```
package com.edu.controller;
```

```
import javax.validation.Valid;
```

```
~ 생략 ~
```

```
@Controller
```

```
@RequiredArgsConstructor
```

```
public class BoardController2 {
```

```
    ~ 생략 ~
```

```
    @GetMapping(value = "/insertBoard")
```

```
    public String boardForm(BoardDto boardDtoModel) {
```

```
        return "boardForm";
```

```
    }
```

```
    @PostMapping(value = "/insertBoard")
```

```
    public String insertBoard(@Valid BoardDto dto, BindingResult bindResult ) {
```

```
        if(bindResult.hasErrors()) {
```

```
            return "boardForm";
```

```
        }
```

```
        boardService.insertBoard(dto);
```

```
        return "redirect:/getBoardList";
```

```
    }
```



```

<!DOCTYPE html>
<html xmlns:th="http://thymeleaf.org">
  <head>
    ~ 생략 ~
  </head>
  <body th:align="center">
    <h1>게시글 등록</h1>
    <form action="insertBoard" method="post" th:object="${boardDto}">
      <table th:align="center">

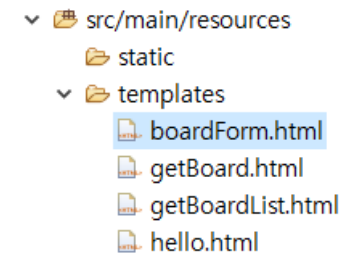
        <tr><td class="rh">제목</td>
          <td><input type="text" th:field="*{title}" />
            <p th:if="${#fields.hasErrors('title')}" th:errors="*{title}"></p></td></tr>

        <tr><td class="rh">작성자</td><td><input type="text" th:field="*{writer}" /></td></tr>

        <tr>
          <td class="rh">내용</td>
          <td><textarea th:field="*{content}" cols="40" rows="10"></textarea>
            <p th:if="${#fields.hasErrors('content')}" th:errors="*{content}"></p></td></tr>

        <tr><td colspan="2"><input type="submit" value="게시글 등록" /></td></tr>
      </table>
    </form>
    <p><a th:href="@{/getBoardList}">글목록</a></p>
  </body>
</html>

```



새글 등록

localhost:8080/insertBoard

게스트

게시글 등록

제목	<input type="text"/> 제목을 입력해 주세요.
작성자	<input type="text"/>
내용	<div><div></div><div>내용을 입력해 주세요. 내용을 5 글자 이상, 50 글자 이하로 입력해주세요.</div></div>
<input type="button" value="게시글 등록"/>	

[글목록](#)