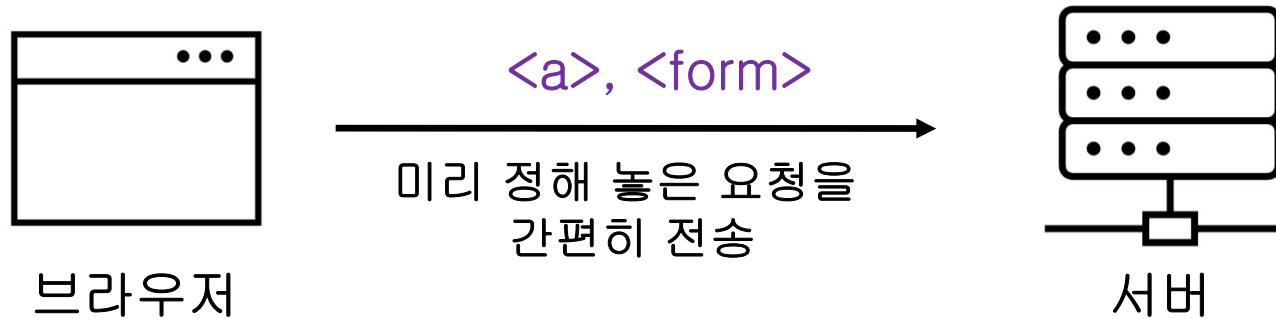


# 6장 게시판 내 페이지 이동하기

출처: 코딩 자율학습 스프링부트3 자바 백엔드 개발 입문, 홍팍, 길벗, 2023

# 링크(link)

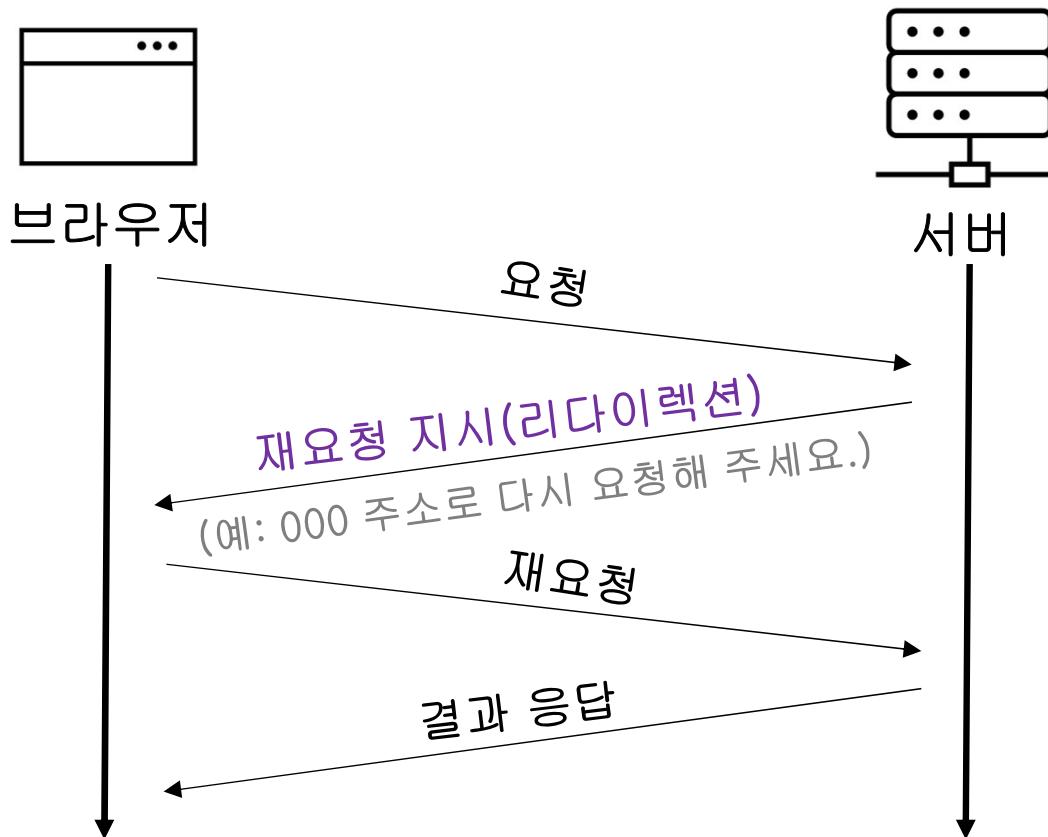
- 페이지 이동을 위해 사용, 미리 정해 놓은 요청을 간편히 전송
- HTML의 <a>, <form> 태그로 작성



- 뷰 파일에 링크 걸기
  - <a href="URL\_주소">링크를 걸 대상</a>

# 리다이렉트 (Redirect)

- 클라이언트가 보낸 요청을 마친 후, 계속해서 처리할 다음 요청 주소를 클라이언트에게 재지시
- 분리된 기능을 하나의 연속적인 흐름으로 연결
- 리다이렉트(재요청)를 지시 받은 클라이언트는 해당 주소로 다시 요청을 보내고 서버는 이에 대한 결과를 응답



# 리다이렉트 (Redirect)

- 리다이렉트 정의하기
  - 뷰 파일의 return문 사용해 클라이언트에게 새로운 URL 주소로 재요청을 하라고 지시
  - return “**redirect:URL\_주소**”

# 링크와 리다이렉션을 이용해 페이지 연결하기

## 게시판을 구성하는 페이지들의 연결

- 게시판 목록보기, 새 글 작성, 게시물 보기, 수정 , 삭제 페이지들을 연결
- <입력 페이지> → <목록 페이지> 돌아가기
- <입력 페이지> → <상세 페이지> 이동하기
- <상세 페이지> → <목록 페이지> 돌아가기
- <목록 페이지> → <상세 페이지> 이동하기

# 〈입력 페이지〉 → 〈목록 페이지〉 돌아가기

- 게시판 〈입력 페이지〉에서 〈목록 페이지〉로 돌아가는 링크 추가하기
  - 입력 페이지의 뷰 파일인 new.mustache 를 열기
  - new.mustache 파일 편집: 〈a〉 태그로 목록 페이지 (“/articles”) 링크 추가

```
<button type="submit" class="btn btn-primary">Submit</button>
<a href="/articles">Back</a>
```

```
</form>
```

- 서버 재실행
- <http://localhost:8080/articles/new> 접속

A screenshot of a web browser window showing the input page. The page has a header "Navbar" and a main section with fields for "제목" (Title) and "내용" (Content). At the bottom, there are two buttons: "Submit" and "Back". The "Back" button is highlighted with a red box.

A screenshot of a web browser window showing the list page. The page has a header "Navbar" and a table with columns "Id", "Title", and "Content". At the bottom, there is a copyright notice: "© CloudStudying | [Privacy](#) | [Terms](#)". A red arrow points from the "Back" link on the input page to this copyright notice on the list page.

# 〈입력 페이지〉 → 〈상세 페이지〉 이동하기

- ArticleController의 createArticle() 메소드의 return 값에 리다이렉트 정의

```
return "redirect:URL_주소";
```

- 등록한 게시글의 id가 1번이면, articles/1 페이지를 재요청  
" 2번이면, articles/2  
" 3번이면, articles/3

ArticleController.java

```
@PostMapping("/articles/create")
public String createArticle(ArticleForm form){
    log.info(form.toString());
    // 1. DTO를 엔티티로 변환
    Article article = form.toEntity();
    log.info(article.toString());
    // 2. 리파지터리로 엔티티를 DB에 저장
    Article saved = articleRepository.save(article);
    log.info(saved.toString());
    return "redirect:/articles/" + saved.getId();
}
```

# 〈입력 페이지〉 → 〈상세 페이지〉 이동하기

- Article에 getId() 메서드 추가

- 빨간색으로 에러가 표시된 getId()에 마우스를 올리면 나타나는 창에서 [Create method 'getId' in 'Article']을 선택

ArticleController.java

```
@PostMapping("/articles/create")
public String createArticle(ArticleForm form){
    (중략)
    Article saved = articleRepository.save(article);
    log.info(saved.toString());
    return "redirect:/articles/" + saved.getId();  
}
```



```
@GetMapping("/articles/{id}")
public String show(@PathVariable Long id, Mo
    log.info("id = " + id);
```

No candidates found for method call `saved.getId()`.

# 〈입력 페이지〉 → 〈상세 페이지〉 이동하기

- Article에 getId() 메서드를 작성

- 방법 1: 직접 작성

```
public class Article {  
    @Id  
    @GeneratedValue  
    private Long id;  
    @Column  
    private String title;  
    @Column  
    private String content;  
  
    public Long getId() {  
        return id;  
    }  
}
```

- 방법 2: 롬복 사용

```
@NoArgsConstructor  
@AllArgsConstructor  
@ToString  
@Getter  
@Entity  
public class Article {  
    @Id  
    @GeneratedValue  
    private Long id;  
    @Column  
    private String title;  
    @Column  
    private String content;  
}
```

# 〈입력 페이지〉 → 〈상세 페이지〉 이동하기

- 서버 재실행
- localhost:8080/articles/new 접속, 게시글 작성 후, [Submit]
- articles/1 페이지로 리다이렉트 확인

localhost:8080/arti...

Navbar

제목  
가가가가

내용  
1111

Submit Back

© CloudStudying | [Privacy](#) | [Terms](#)

localhost:8080/articles/1

| Id | Title | Content |
|----|-------|---------|
| 1  | 가가가가  | 1111    |

© CloudStudying | [Privacy](#) | [Terms](#)

# 〈상세 페이지〉 → 〈목록 페이지〉 돌아가기

- 〈상세 페이지〉에 〈목록 페이지〉로 돌아갈 수 있게 링크 추가
  - 상세 페이지 URL인 articles/{id}를 받는 컨트롤러 메소드 확인  
→ ArticleController의 show()
  - show 메소드의 return 값 확인: show.mustache
- show.mustache 파일 편집
  - 〈목록 페이지〉로 돌아가는 링크 추가

Templates/articles/show.mustache

(중략)

</table>

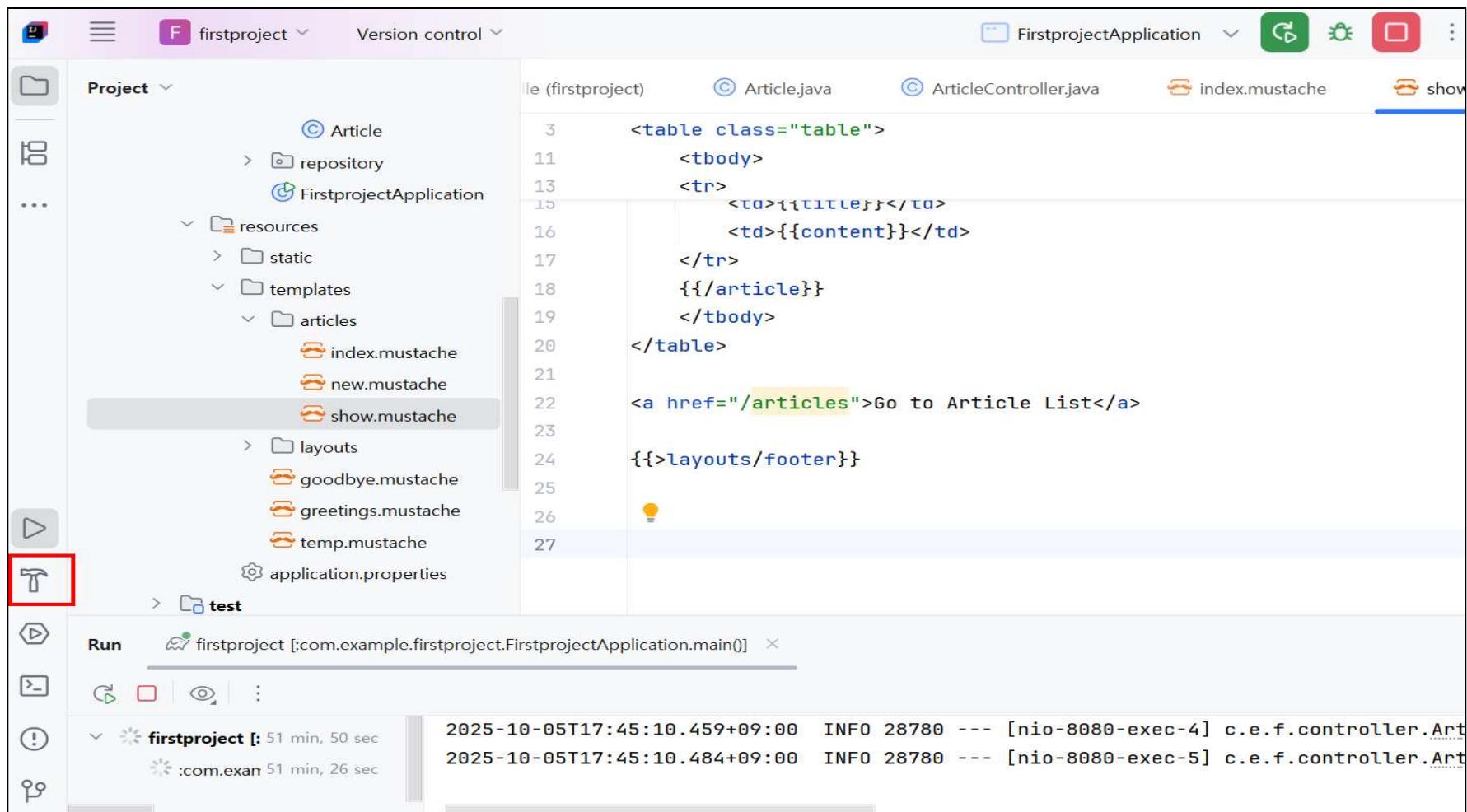
[Go to Article List](/articles)

{{>}layouts/footer}}

# 〈상세 페이지〉 → 〈목록 페이지〉 돌아가기

- **프로젝트 빌드**

- 메뉴 [Build - Build Project] 클릭 또는 아이콘  클릭
  - ✓ 자바 파일이 수정되었을 때 → 서버 재실행
  - ✓ Html 파일이 수정되었을 때 → 프로젝트 빌드



# 〈상세 페이지〉 → 〈목록 페이지〉 돌아가기

- localhost:8080/articles/1 접속

A screenshot of a web browser window titled "Hello, world!". The address bar shows "localhost:8080/arti...". The main content area displays a table with one row:

| Id | Title | Content |
|----|-------|---------|
| 1  | 가가가가  | 1111    |

Below the table is a link "Go to Article List" which is highlighted with a red box and a red arrow points from it to the second browser window.

© CloudStudying | [Privacy](#) | [Terms](#)

A screenshot of a web browser window titled "Hello, world!". The address bar shows "localhost:8080/articles" and has a red arrow pointing to it. The main content area displays a table with one row:

| Id | Title | Content |
|----|-------|---------|
| 1  | 가가가가  | 1111    |

Below the table is a link "Go to Article List" which is highlighted with a red box and a red arrow points from it to the first browser window.

© CloudStudying | [Privacy](#) | [Terms](#)

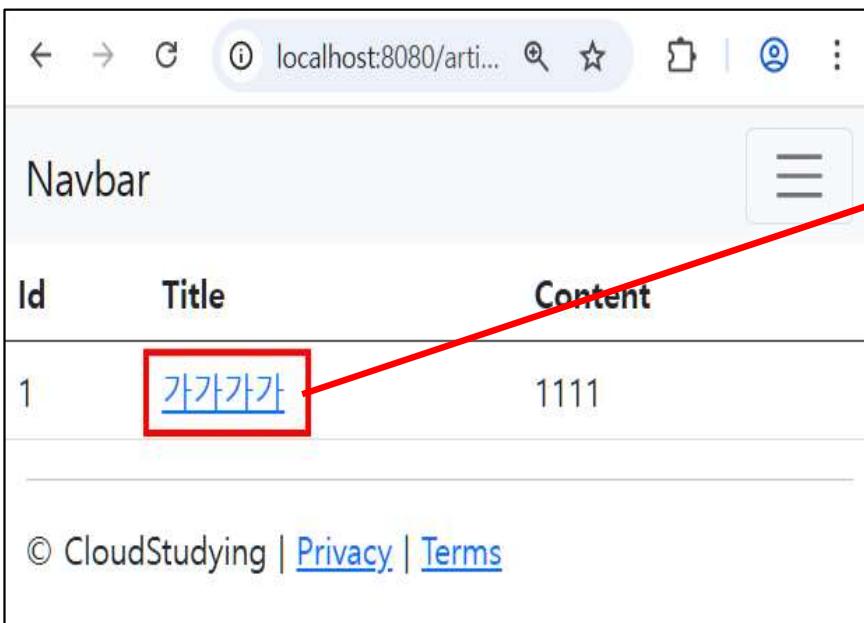
# 〈목록 페이지〉 → 〈상세 페이지〉 이동하기

- 〈목록 페이지〉에서 제목을 클릭하면 해당 글의 〈상세 페이지〉로 이동하기
  - 〈목록 페이지〉의 뷰 파일인 index.mustache 파일 편집
    - ✓ 제목에 링크 추가

```
(중략)
{{#articleList}}
  <tr>
    <td>{{id}}</td>
    <td><a href="/articles/{{id}}">{{title}}</a></td>
    <td>{{content}}</td>
  </tr>
{{/articleList}}
(중략)
```

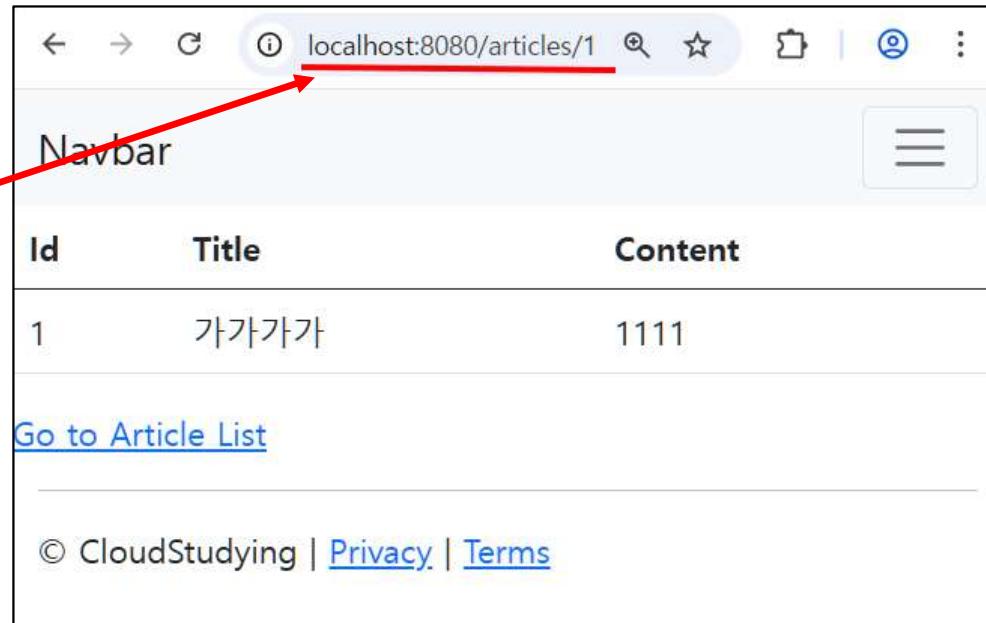
# 〈목록 페이지〉 → 〈상세 페이지〉 이동하기

- 프로젝트 빌드 
- localhost:8080/articles/1 접속



| Navbar |       |         |
|--------|-------|---------|
| Id     | Title | Content |
| 1      | 가가가가  | 1111    |

© CloudStudying | [Privacy](#) | [Terms](#)



| Navbar                             |       |         |
|------------------------------------|-------|---------|
| Id                                 | Title | Content |
| 1                                  | 가가가가  | 1111    |
| <a href="#">Go to Article List</a> |       |         |

© CloudStudying | [Privacy](#) | [Terms](#)