

Servlet 정리

PRG 패턴(Post-Redirect-GET)

POST 방식과 Redirect를 결합해서 사용하는 PRG 패턴

- 사용자는 컨트롤러에 원하는 작업을 POST 방식으로 처리하기를 요청
- POST 방식을 컨트롤러에서 처리하고 브라우저는 다른 경로로 이동(GET)하라는 응답(Redirect)
- 브라우저는 GET 방식으로 이동

PRG 패턴은 일상 생활에서 접하는 게시판이 가장 대표적이다.

예)

- 사용자가 새로운 게시글의 내용을 작성하고 POST 방식으로 전송
- 서버에서 새로운 게시글을 처리한 후에 브라우저의 주소를 목록 화면 경로로 이동하도록 응답(Redirect)
- 브라우저는 목록 화면을 보여주고 사용자는 자신이 추가한 게시글이 추가된 결과를 확인

PRG 패턴은 POST 방식의 처리 후에 바로 다른 주소로 브라우저가 이동하기 때문에 반복적으로 POST 호출이 되는 상황을 막을 수도 있고, 사용자의 입장에서 처리가 끝나고 다시 처음 단계로 돌아간다는 느낌을 주게 된다.

Todo 웹 애플리케이션 와이어 프레임 그리기

- 화면에 해당 페이지를 볼 수 있는 경로(URL)를 명시한다.
- GET 방식으로 동작하고 눈에 보이는 것을 가장 먼저 구성한다.
- POST 방식으로 처리되는 컨트롤러는 다른 그림으로 표현한다.
- Redirect되어 보여지는 경우는 다른 선으로 표현한다.

1. 목록 화면(GET)

- GET 방식으로 조회할 수 있는 화면
- 등록/수정/삭제 후에도 결과를 확인할 수 있는 화면

2. 등록 화면(GET)

- 등록을 위해서는 GET 방식으로 이동하기 때문에 등록 화면 우선적으로 작성되는 화면이다.

3. 등록 화면(POST)

- 등록 화면에서 입력한 내용들을 POST 방식으로 전송되기 때문에 별도의 컨트롤러를 추가하고 처리 후에는 Redirect 방식으로(PRG 패턴) 다시 목록 페이지를 보도록 설계

4. 조회 화면(GET)

- GET 방식으로 동작하게 설계 목록, 화면에서 특정한 Todo 글 번호를 선택하면 동작하도록 설계

5. 수정/삭제 화면(GET)

- 수정/삭제 작업은 조회 화면에서 '수정/삭제'를 선택했을 때 GET 방식으로 이동이 가능하도록 설계, 수정/삭제 화면에서 POST 방식으로 수정이나 삭제를 진행할 수 있는 버튼 추가

6. 수정(POST)

- 실제 수정 작업은 POST 방식으로 처리되어야 하고, 컨트롤러에서 추정 처리된 후에 다시 목록 화면으로 이동

7. 삭제(POST)

- 동일하게 POST 방식으로 처리하도록 설계, 삭제된 후에 목록 화면으로 Redirect되도록 구성

기능	동작 방식	컨트롤러(org.zerock.wi.todo)	JSP
목록	GET	TodoListController	WEB-INF/todo/ <u>list.jsp</u>
등록(입력)	GET	TodoRegisterController	WEB-INF/todo/ <u>register.jsp</u>
등록(처리)	POST	TodoRegisterController	Redirect
조회	GET	TodoReadController	WEB-INF/todo/ <u>read.jsp</u>
수정(입력)	GET	TodoModifyController	WEB-INF/todo/ <u>modify.jsp</u>
수정(처리)	POST	TodoModifyController	Redirect
삭제(처리)	POST	TodoRemoveController	Redirect