56일차 강의

* 뒤로가기 버튼, 수정 버튼 – 기능도 구현
* 누른 멤버의 정보도 조회되도록
* 삭제버튼 구현 – 기능도
* 로그인, 로그아웃 만들기
* MyMemberSite

에서 오류를 없애고,

회원 가입, 조회, 업데이트 를 다시 만들어보라

* Could not publish server configuration for Tomcat v8.0 Server at localhost.

Multiple Contexts have a path of "/web05\_293".

**[ 이클립스를 사용하여 깃에 올려보자!!! ]**

* 내가 작업한 프로젝트를 깃에 올릴려면 javaEE에서 project Expoloer의 내가 작업한 프로젝트에서 시작한다]
* 프로젝트명에서 마우스 우클릭 -> team
* Team -> share 를 클릭
* 선택사항들을 선택한다

( 어느 레파지토리에 올릴것인지)

* Add to index
* Git Staging 창이 뜬다

(Git Staging 창이 안뜨면 team -> commit 클릭)

* 인덱스를 관리한다
* 커밋 메시지를 작성한다
* 커밋은 (오류가 없고 완벽할 때 하는 것이다)
  + - Brench 는 좀더 다른 것이다 (따로 공부할 것)
* 커밋 한다
* 프로젝트명 – team – show in history

: add to index 한것만 역사로 남아있는 것

* Team – pull

: 다른 사람들이랑 협업할 때 pull 하여,

다른사람과 인덱스를 새로 맞춰줘야 한다는 것

* Team -> synchronize workspace

: 충돌이 있는지 없는지 확인한다

* Team -> push branch master -> preview -> push
* 깃허브 사이트에서 내가 push한 프로젝트가 있는지 확인한다
* 로그인, 로그아웃 기능을 만들어 보자!!
* **데이터 보관소 [ Scope (영역) ]**

**: 데이터를 공유하는 방법**

1. pageContext (JspContext)

: JSP 페이지를 실행하는 동안만 유지된다

* 이 페이지를 떠나는 순간 데이터가 사라진다.
* 다른 페이지에서 여기 있는 정보를 가져갈수 없다

1. request (ServletRequest)

: 클라이언트의 요청이 들어올 때 생성되어

클라이언트에게 응답 할 때까지 유지된다.

* req에 담아서 객체들을 다른 파일에 전달할수 있다.

(영역을 넘나들수 있음)

* 이 페이지에 req의 자료가 소멸되도 다른 페이지에서 데이터가 유지될 수 있음

이 보관소를 포워딩이나 인클루딩하는 서블릿들 사이에서 값을 공유할 때 유용하다

1. session (HttpSession)

: 클라이언트의 최초 요청 시 생성되어 브라우저를 닫을 때 까지 유지된다. 보통 로그인 할 때 이 보관소를 초기화하고, 로그아웃하면 이 보관소에 저장된 값들을 비운다

따라서 이 보관소에 값을 보관하면 서블릿이나 jsp페이지에 상관없이 로그아웃하기 전까지 계속 값을 유지할 수 있다

* 웹은 모든 페이지에서 뭔가를 할때마다 사용자에 대한 권한을 확인 해야한다. (이 사람인지 저 사람인지)
* Session 이 있다면, 모든 페이지에서 뭔가를 할때마다 로그인 하고 뭔가를 할 필요가 없다. 로그아웃하기 전까지 어떤 태그를 타든 로그인상태가 그 화면에서 유지가 된다.
* 페이지를 넘나들 때 소실되서 req에 담아서 자꾸 넘겨야되는데 session을 이용하면 엄청 편리해진다.

1. application (ServletContext)

: 웹 애플리케이션이 시작될 때 생성되어 웹 애플리케이션이 종료될 때까지 유지된다. (자바의 static 과 유사함)

* 브라우저를 닫아도 사라지지않고 어플리케이션을 닫을때까지 유지된다. (홈페이지에 대한 막강한 설정들이 이에 해당)

이 보관소에 데이터를 보관하면 웹 애플리케이션이 실행되는 동안에는 모든 서블릿이 사용할 수 있다.