

세션 구현하기 실습

—

실습 환경

- <https://github.com/slipp/web-application-server> 저장소를 자신의 계정으로 fork한다.
- Fork한 저장소를 https://youtu.be/xid_GG8kL_w 동영상 참고해 로컬 개발 환경을 구축한다.
- web-application-server 저장소의 was-step3-controller-refactoring 브랜치로 변경한다.
 - 브랜치를 변경하는 방법은 <https://youtu.be/VeTjDYl7UVs> 동영상을 참고한다.

웹 서버 시작 및 테스트

- `webserver.WebServer` 는 사용자의 요청을 받아 `RequestHandler` 에 작업을 위임하는 클래스이다.
- 사용자 요청에 대한 모든 처리는 `RequestHandler` 클래스의 `run()` 메서드가 담당한다.
- `WebServer`를 실행한 후 브라우저에서 <http://localhost:8080>으로 접속해 질문/답변 게시판 서비스 화면이 나타나는지 확인한다.

요구사항

서블릿에서 지원하는 HttpSession API의 일부를 구현해야 한다.

HttpSession API 중 구현할 메소드는 **getId()**, **setAttribute(String name, Object value)**, **getAttribute(String name)**, **removeAttribute(String name)**, **invalidate()** 5개이다.

HttpSession의 가장 중요하고 핵심이 되는 메소드이다.

각 메소드의 역할

- `String getId()`: 현재 세션에 할당되어 있는 고유한 세션 아이디를 반환
- `void setAttribute(String name, Object value)`: 현재 세션에 `value` 인자로 전달되는 객체를 `name` 인자 이름으로 저장
- `Object getAttribute(String name)`: 현재 세션에 `name` 인자로 저장되어 있는 객체 값을 찾아 반환
- `void removeAttribute(String name)`: 현재 세션에 `name` 인자로 저장되어 있는 객체 값을 삭제
- `void invalidate()`: 현재 세션에 저장되어 있는 모든 값을 삭제

요구사항 분리 및 힌트

—

실습

클라이언트와 서버 간에 주고 받을 고유한 아이디를 생성해야 한다. 고유한 아이디는 쉽게 예측할 수 없어야 한다. 예측하기 쉬우면 쿠키 값을 조작해 다른 사용자처럼 속일 수 있다.

힌트

JDK에서 제공하는 UUID 클래스를 사용해 고유한 아이디를 생성할 수 있다.

```
UUID uuid = UUID.randomUUID();
```

실습

앞 단계에서 생성한 고유한 아이디를 쿠키를 통해 전달한다.

힌트

쿠키는 Set-Cookie 헤더를 통해 전달되며 name1=value1; name2=value2 형태로 전달된다. 자바 진영에서 세션 아이디를 전달하는 이름으로 JSESSIONID를 사용한다.

실습

서버 측에서 모든 클라이언트의 세션 값을 관리하는 저장소 클래스를 추가한다.

힌트

HttpSessions와 같은 이름을 가지는 클래스를 추가한다.

이 클래스는 `Map<String, HttpSession>`와 같은 저장소를 통해 모든 클라이언트별 세션을 관리해야 한다. 이 저장소의 키(key)는 앞에서 UUID로 생성한 고유한 아이디이다.

실습

클라이언트별 세션 데이터를 관리할 수 있는 클래스(HttpSession)를 추가한다.

힌트

HttpSession 클래스는 요구사항에 있는 5개의 메소드를 구현해야 하며, 상태 데이터를 저장할 Map<String, Object>가 필요하다.