웹 프로그래밍

4장. 쿠키와 세션

동아대학교 컴퓨터공학과
양 선

쿠키/세션의 필요성 (자료 보관하는 제3의 장소의 필요성)

❖ 3장에서 배웠던 데이터 전송은 오직 한 단계
- 여러 단계 후까지 데이터를 넘기려면?
- 중간 화면들은 데이터를 계속 넘겨주기 해야 함

❖ 쿠키/세션은 제3의 장소 역할 (즉, 장바구니 역할)
- 서버와 접속 후에 발생된 고객의 정보를 저장
- 그 정보를 클라이언트가 가지고 있으면 쿠키
  - ✓ 정보가 과자 부스러기처럼 클라이언트 컴퓨터에 남는다고 해서 쿠키라 불림
- 서버가 가지고 있으면 세션

쿠키 VS 세션

- 쿠키를 사용하면 서버에 무리를 주지 않는다는 장점

- 반면 쿠키는 보안에 약하고 데이터 사이즈 작다는 단점
  - 공유 PC 사용 시 쿠키에 저장된 정보를 타인이 알아낼 수 있음!
  - 또한 저장할 수 있는 데이터 사이즈도 한계가 있음
    - ✓ 웹브라우저마다 조금씩 차이가 있긴 하지만, 예를 들어 최대 4KB 쿠키를 300 이하로만 저장 가능하다 이런 식으로 제한을 둠
    - ✓ 이 경우 쿠키를 사용할 수 있는 최대 용량은 4KB * 300개 = 1.2MB

- 세션을 사용하여 이러한 단점 극복
  - 서버에서 관리하므로 보안 강력하게 유지할 수 있음
  - 또한 저장할 수 있는 데이터 사이즈도 매우 큼

세션(session) 코딩 시 자주 헷갈리는 부분

코딩	설명
public String methodName(HttpSession se, ...) { ... }	• 매개변수에 HttpSession 넣어 줌 • 세션 사용하겠다는 의미
se.setAttribute("mid", mid);	• 세션에 한 개 보관하기 • mo.addAttribute("mid", mid); 랑 비슷
String mid = (String)se.getAttribute("mid");	• 세션에 들어있는 항목 1개를 지역변수에 대입 • 리턴 타입이 Object이므로 **형변환 필요** • 단, 지역변수 거치지 않고 바로 모델에 넣을 때는 양쪽 다 Object이므로 형변환 필요 없음 mo.addAttribute("mid", se.getAttribute("mid"))
se.removeAttribute("mid");	• 세션에서 한 개 제거
se.invalidate();	• 세션 무효화 (로그아웃)

여기서 잠깐 팝업창 띄워보기!

- ✓ 세션 사용한 예제 구현하기 전에 잠깐 javascritp 로 팝업창을 띄우는 걸 공부하겠습니다
  .
  - popuptest.html 및 컨트롤러 메소드 작성 후 http://localhost:8080/popuptest

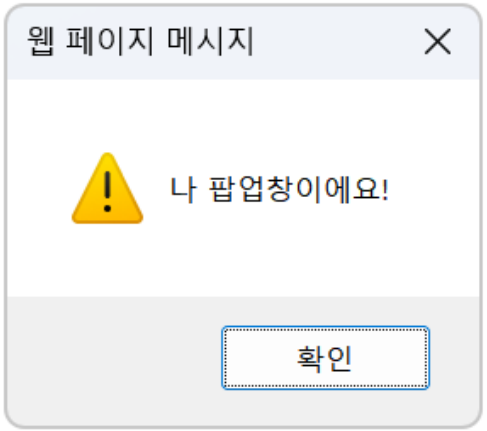
```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>팝업창 테스트</title></head>
<body>

<script>
    alert("나 팝업창이에요!");
    location.href="/";
</script>

</body>
</html>
```

```
@GetMapping("/popuptest")
public String popuptest() {
   return "popuptest";
}
```

<script> ~ </script>는 자바스크립트입니다.

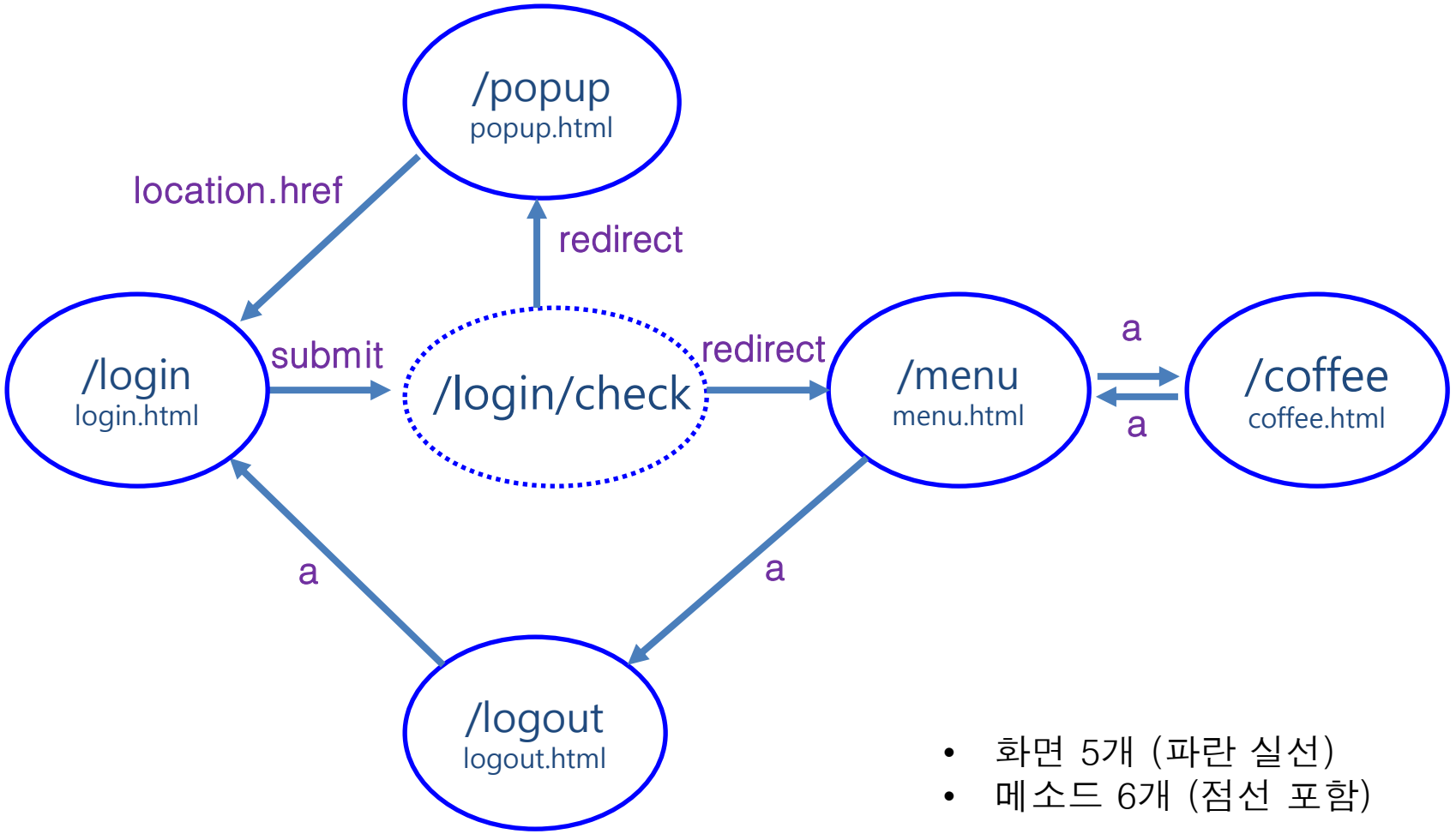
alert은 확인버튼 1개만 있는 팝업창입니다.

location.href 를 이용해서 다른 페이지로 이동합니다.



웹브라우저마다 팝업창
그림은 조금씩 다릅니다.

세션을 사용하는 간단한 로그인/로그아웃 예제 구현



- 화면 5개 (파란 실선)
- 메소드 6개 (점선 포함)

아이디를 틀리게 입력한 경우

front-end	back-end
localhost:8080/login 접속	
	/login 메소드 실행. 로그인화면 보내줌
로그인화면 뜨면 아이디 입력 후 submit	
	/login/check 메소드 실행 아이디 체크 후 만약 없는 아이디이면 /popup 메소드로 redirect (이 때 팝업창에 뜰 메세지를 parameter로 보내줌)
	/popup 메소드 실행. 팝업창화면 보내줌
팝업창화면 뜨고 확인버튼 클릭	
	/login 메소드 실행. 로그인화면 보내줌
로그인화면 뜸	

아이디를 맞게 입력한 경우

front-end	back-end
localhost:8080/login 접속	
	/login 메소드 실행 . 로그인화면 보내줌
로그인화면 뜨면 아이디 입력 후 submit	
	/login/check 메소드 실행. 아이디 체크 후 회원 맞으면 **세션에 아이디 저장** 후 /menu 메소드로 redirect
	/menu 메소드 실행. 메뉴화면 보내줌
메뉴화면 뜨면 커피 클릭	
	/coffee 메소드 실행. 커피화면 보내줌
커피화면 뜨면 메뉴로 돌아가기 클릭	
	/menu 메소드 실행. 메뉴화면 보내줌
메뉴화면 뜨면 로그아웃 클릭	
	/logout 메소드 실행. 세션 무효화 시킨 후 로그아웃화면 보내줌
로그아웃 화면 뜨면 그림 클릭	
	/login 메소드 실행 . 로그인화면 보내줌
로그인화면 뜸	

YourController.java

이제 컨트롤러를 분리하겠습니다.
하나의 프로젝트에 컨트롤러 소스는 여러 개 있어도 됩니다.
이후에 나오는 메소드들은 YourController.java에 넣어주세요.
@Controller 넣는 거 잊지 마세요!

```
package com.web.p1;

import java.util.ArrayList;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

import jakarta.servlet.http.HttpSession;

@Controller
public class YourController {


   /* 여기 메소드 추가 */


}  // class
```

로그인 (login.html)

```
<!DOCTYPE html><html>
<head><meta charset="UTF-8"> <title>Welcome</title>
<style>
  body {background-color:#FBEFF5;
        text-align: center;}
  img {width: 280px; height: 320px; }
  h1 {color:brown; }
  span {color:gray; font-size:0.8em; }
</style></head>
<body>
<h1>Maru Cafe</h1>
(bread, coffee, salad) <p>
<form method="get" action="/login/check">
  ID: <input type="text" name="mid"> <p>
  <input type="submit" value="login">
</form><p>
<img src="/maru.jpg" alt="마루 화동 그림"><br>
<span>(그림 출처: N사 웹툰 maruneun gangjwi)</span>
</body></html>
```

"N사 모죠님 웹툰 그림입니다.
상업적 용도 아니고 교육용입니다!!!"
(저작권 문제 있으면 수업자료에서
삭제하겠습니다 ^^)

```
@GetMapping("/login")
public String login() {
  return "login";
}
```

- img 태그는 그림을 불러옵니다. alt 속성은 해당 그림을 못 불러올 때 대신 문구를 띄워줍니다. 그림파일은 static 폴더에 넣어주시면 됩니다.
- body에 설정된 text-align은 table 안에 있는 글자는 가운데정렬 시킬 수 있지만 table 자체는 가운데정렬 안 됩니다. (나중에 다른 방법 사용)

화면 없는 컨트롤러 메소드 (p. 6 그림 중 점선 메소드)

/login/check

```
@GetMapping("/login/check")
public String loginCheck(HttpSession se,
                @RequestParam("mid") String mid, RedirectAttributes re) {

  var arr = new ArrayList<String>();
  arr.add("고흐");
  arr.add("james");
  arr.add("dooli");
  arr.add("iu");

  if( arr.contains(mid) )
  {
        se.setAttribute("mid", mid);
        return "redirect:/menu";
  }
  else
  {
        re.addAttribute("msg", mid
              + "는 미등록 아이디입니다. 확인 후 로그인 부탁드립니다.");
        return "redirect:/popup";
  }

}
```

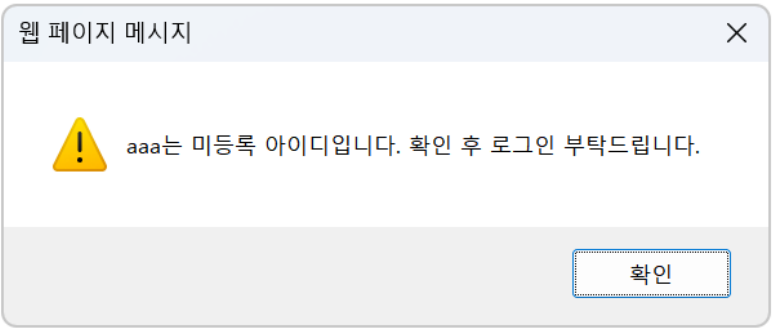
- 지금은 아이디 4명으로 하드코딩 했지만, 나중에는 데이터베이스에서 회원정보 가지고 오게 됩니다.

- 지금까지는 return 다음에 뷰템플릿 파일이름이 있었는데 redirect 콜론 다음에는 주소가 나옵니다.
- 즉, 컨트롤러 안에 있는 @GetMapping("/menu") 메소드로 가서 그 메소드를 실행시키라는 의미입니다.

- msg도 mid처럼 세션에 넣어도 되겠지만 redirect 한 단계 parameter 전송이므로 굳이 세션에 넣지 않고 대신 RedirectAttributes 사용

팝업 (popup.html)

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <head><meta charset="UTF-8">
4  <title>안내글</title></head>
5  <body>
6
7  <script th:inline="javascript">
8    alert([[${msg}]]);
9    location.href="/login";
10 </script>
11
12 </body>
13 </html>
```



- javascript에 타임리프 변수를 넣는 방법입니다.
- 8번 라인에 빨간 오류표시 있어도 웹서버 구동에는 아무런 문제가 없습니다.

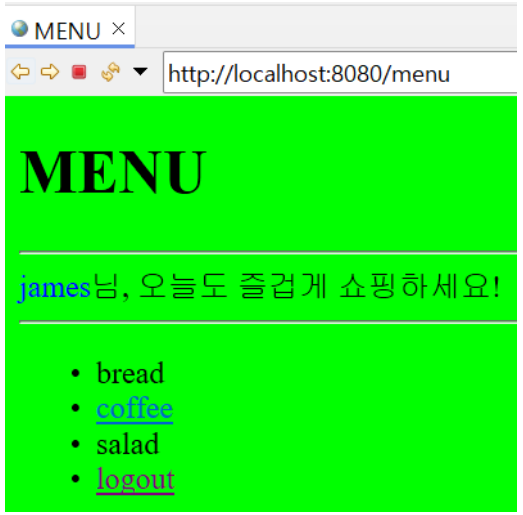
빨간 오류표시 안 나오게 하는 방법도 있습니다.
7~10 라인 대신에 우측처럼 코딩하면 됩니다.
(하지만 우측 코딩은 시험에 나오지는 않습니다.)

```
<script th:inline="javascript">
  /*<![CDATA[*/
    let msg = /*[[${msg}]]*/ 'default';
    alert(msg);
    location.href="/login";
  /*]]>*/
</script>
```

```
@GetMapping("/popup")
public String popup(@RequestParam("msg") String msg, Model mo) {
        /* msg는 사용자 입력 데이터 아니고 redirect parameter */
  mo.addAttribute("msg", msg);
  return "popup";
}
```

메뉴 (menu.html)

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><meta charset="UTF-8">
<title>MENU</title><style>
  body { background-color:lime; }
  #s1 { color:blue; }
</style></head>
<body>
<h1>MENU</h1><hr>
<span id="s1" th:text="${mid}">mid</span>님,
오늘도 즐겁게 쇼핑하세요! <br><hr>
<ul>
  <li>bread
  <li><a href="/coffee">coffee</a>
  <li>salad
  <li><a href="/logout">logout</a>
</ul>
</body></html>
```



- 은 글머리기호 은 번호매기기

- 는 종료태그 생략 가능

- 과 사이에 mid를 넣은 이유는?
  (어차피 타임리프 변수보다 우선순위 밀릴 텐데??)

```
@GetMapping("/menu")
public String menu(HttpSession se, Model mo) {
  mo.addAttribute("mid", se.getAttribute("mid"));
  return "menu";
}
```

커피 선택 화면 (coffee.html)

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><meta charset="UTF-8">
<title>coffee</title><style>
  body {
     background-image: url(/coffee.jpg);
     background-size : cover;
    }
  #s1 { color:rgb(128, 64, 64); }
</style></head>
<body>
<h1><span id="s1" th:text="${mid}">mid</span>님,
원하시는 커피를 선택하세요</h1>
<h3>Coming soon ...</h3>
<a href="/menu">메뉴로 돌아가기</a>
</body></html>
```

- 쿠키/세션 설명을 위해 만든 미완성 화면입니다.
- 나중에 데이터베이스 연결한 후에 커피 뿐만 아니라, 빵, 샐러드도 판매하는 카페 사이트를 완성해 봅시다!!

```
@GetMapping("/coffee")
public String coffee(HttpSession se, Model mo) {
  mo.addAttribute("mid", se.getAttribute("mid"));
  return "coffee";
}
```

로그아웃(logout.html)

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><meta charset="UTF-8">
<title>logout</title><style>
  body { background-color:#FBEFF5; text-align:center; }
  #s1 { color:blue; }
  .s2 { color:gray; font-size:0.8em; }
  img { width: 300px; height: 220px; }
</style></head>
<body>
<span id="s1" th:text="${mid}">mid</span> 님께서 로그아웃하셨습니다.<p>
다음에 또 만나요!!<p>
<span class="s2">(아래 그림을 클릭하시면 첫 화면으로 이동합니다.)</span><br>
<a href="/login">
   <img src="/zakkabye.jpg" alt="바이바이 그림">
</a><br>
<span class="s2">(그림 출처: N사 웹툰 daehak ilgi)</span>
</body></html>
```

"N사 자까님 웹툰 그림입니다.
상업적 용도 아니고 교육용입니다!!!"

- <a>와 사이에 그림이 있으므로 그림을 클릭하면 다른 주소로 넘어갑니다.

```
@GetMapping("/logout")
public String logout(HttpSession se, Model mo) {
  mo.addAttribute("mid", se.getAttribute("mid"));
  se.invalidate();
  return "logout";
}
```