[ 10 ] RequestMapping

1. Get 방식과 Post 방식
   * @RequestMapping에서 요청을 방을 때 Get방식(url에 데이터를 실어서 보냄)과 Post방식(header에 데이터를 실어서 보냄)을 구분 할 수 있다
   * @RequestMapping 어노테이션을 이용해서 value값은 요청경로. method값에는 요청방식이 들어간다. 방식이 맞지 않으면 405에러

(예제)

**web.xml 수정**

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<!-- 한글 처리 부분 -->

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

**webapp폴더에 index.jsp 추가**

<body>

<jsp:forward page=*"main.do"*/>

</body>

<body>

<h1>GET방식</h1>

<form action=*"*${conPath }*/student.do"* method=*"get"*>

ID <input type=*"text"* name=*"id"*>

<input type=*"submit"* value=*"GET방식으로 전송"*>

</form>

<hr>

<h1>POST방식</h1>

<form action=*"*${conPath }*/student.do"* method=*"post"*>

ID <input type=*"text"* name=*"id"*>

<input type=*"submit"* value=*"POST방식으로 전송"*>

</form>

<hr>

<h1>redirect 키워드 예제</h1>

<form action=*"studentConfirm.do"*>

ID <input type=*"text"* name=*"id"*>

<input type=*"submit"* value=*"confirmId"*>

</form>

</body>

**Ex1StudentController.java**

@Controller

**public** **class** Ex1StudentController {

@RequestMapping(value="main", method=RequestMethod.***GET***)

**public** String main() {

**return** "ex1/main";

}

// @RequestMapping(value="student")

// public String student(HttpServletRequest request, String id, Model model) {

// String method = request.getMethod();

// System.out.println(method+" 방식 student메소드");

// //String id = request.getParameter("id");

// model.addAttribute("method", method);

// model.addAttribute("id",id);

// return "ex1/studentId";

// }

@RequestMapping(value="student", method = RequestMethod.***GET***) // 메소드방식이 다르면 405에러

**public** String student(String id, Model model) {

System.***out***.println("get 방식 student메소드");

model.addAttribute("id", id);

model.addAttribute("method", "GET");

**return** "ex1/studentId";

}

@RequestMapping(value="student", method = RequestMethod.***POST***) // 메소드방식이 다르면 405에러

**public** ModelAndView student(String id, ModelAndView modelAndView) {

System.***out***.println("POST 방식 student메소드");

modelAndView.addObject("id", id);

modelAndView.addObject("method", "POST");

modelAndView.setViewName("ex1/studentId");

**return** modelAndView;

}

**studentID.jsp**

<body>

<h1>studentId.jsp입니다</h1>

<h2>전송방식 : ${method }</h2>

<h2>아이디 : ${id }</h2>

<button onclick="history.back()">이전</button>

<button onclick=*"location.href='*${conPath}*/main.do'"*>처음으로</button>

</body>

1. redirect 키워드 : 다른 페이지로 이동할 때 사용
   * 입력하거나 넘겨주는 요청값(model attribute)이 필요 없이 controller단을 호출할 때
2. forward 키워드 : 다른 페이지로 이동할 때 사용
   * 요청값(model attribute)도 함께 넘겨주면서 controller단을 호출할 때

(예제)

**<** **redirect 키워드 와 forward 키워드** **비교>**

**RedirectController.java**

@Controller

**public** **class** Ex1RedirectController {

@RequestMapping(value="studentConfirm.do", method = RequestMethod.***GET***)

**public** String confirmID(String id, Model model) {

model.addAttribute("id", id);

**if**(id.equals("aaa")) {

//return "redirect:studentOk.do?id="+id;

**return** "forward:studentOk.do";

}

//return "redirect:studentNg.do?id="+id;

**return** "forward:studentNg.do";

}

@RequestMapping(value="studentOk.do", method=RequestMethod.***GET***)

**public** String studentOk() {

**return** "ex1/studentOk";

}

@RequestMapping(value="studentNg.do", method=RequestMethod.***GET***)

**public** String studentNg(String id, Model model) {

model.addAttribute("id", id);

**return** "ex1/studentNg";

}

@RequestMapping(value="fullpath.do", method = RequestMethod.***GET***)

**public** String fullpath(){

//return "redirect:http://localhost:8090/ch09/student/fullpath.jsp";

**return** "redirect:http://www.naver.com";

}

}

**student/studentNg.jsp**

<body>

<h1>studentNg.jsp입니다. ID confirm 실패</h1>

<h2>model의 ID : ${id }</h2>

<h2>param의 ID : ${param.id }</h2>

<a href=*"*${conPath }*/fullpath.do"*>FULLPATH로</a>

</body>

**student/studentOk.jsp**

<body>

<h1>studentOk.jsp입니다. ID confirm 완료</h1>

<h2>model의 ID : ${id }</h2>

<h2>param의 ID : ${param.id }</h2>

<hr>

<a href=*"fullpath.do"*>fullpath로</a>

</body>

**Student/fullPath.jsp**

<body>

<h1>student폴더안의 fullpath.jsp입니다</h1>

<h2>redirect시 전체 풀경로를 다 써 줄 수도 있다</h2>

</body>

>