[ 11 ] 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>

**main.jsp**

<body>

<h1>GET방식</h1>

<form action=*"student.do"* method=*"get"*>

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

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

</form>

<h1>GET방식</h1>

<form action=*"student.do"* method=*"post"*>

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

<input type=*"submit"* value=*"post방식으로 보내기"*>

</form>

</body>

**MyController.java**

@Controller

**public** **class** MyController {

//@RequestMapping("main.do") 둘 다 가능

@RequestMapping("main")

**public** String goMain() {

**return** "main";

}

@RequestMapping(value="student")

**public** String studentall(HttpServletRequest request, Model model) {

System.***out***.println("RequestMethod.GET");

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

model.addAttribute("method", request.getMethod());

model.addAttribute("id", id);

**return** "studentId";

}

}

**studentID.jsp**

<body>

<h1>studentID.jsp입니다.</h1>

<hr>

<h3>ID : ${id }</h3>

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

</body>

**MyController.java GET방식과 POST방식을 분리해서 로직수행**

@Controller

**public** **class** MyController {

@RequestMapping(value = "main")

**public** String home(Model model) {

**return** "main";

}

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

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

System.***out***.println("RequestMethod.GET");

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

model.addAttribute("id", id);

**return** "studentId";

}

@RequestMapping(value="student", method = RequestMethod.***POST***)

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

System.***out***.println("RequestMethod.POST");

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

modelAndView.addObject("id", id);

modelAndView.setViewName("studentId");

**return** modelAndView;

}

}

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)

**main.jsp**

<body>

<h1>GET방식</h1>

<form action=*"student.do"* method=*"get"*>

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

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

</form>

<h1>GET방식</h1>

<form action=*"student.do"* method=*"post"*>

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

<input type=*"submit"* value=*"post방식으로 보내기"*>

</form>

<h1>redirect 키워드 예제</h1>

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

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

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

</form>

</body>

**RedirectController.java**

@Controller

**public** **class** RedirectController {

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

**public** String confirmID(HttpServletRequest httpServletRequest, Model model) {

String id = httpServletRequest.getParameter("id");

model.addAttribute("id", id);

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

**return** "redirect:studentOk.do"; //새로운 요청 studentOk.do가 요청된다

}

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

}

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

**public** String studentOk() {

**return** "student/studentOk";

}

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

**public** String studentNg() {

**return** "student/studentNg";

}

@RequestMapping(value="fullpath")

**public** String fullpath() {

**return** "redirect:http://localhost:8181/ch11/temp/fullPath.jsp";

// redirect시 전체 풀경로를 다 써 줄 수도 있다.

}

}

**student/studentNg.jsp**

<body>

<h1>studentNg.jsp입니다. ID confirm실패</h1>

<a href=*"fullpath.do"*>fullPath로</a>

</body>

**student/studentOk.jsp**

<body>

<h1>studentOk.jsp입니다. ID confirm완료</h1>

<a href=*"fullpath.do"*>fullPath로</a>

</body>

**Student/fullPath.jsp**

<body>

<h1>temp폴더 안의 fullPath.jsp입니다</h1>

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

</body>

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

@Controller

**public** **class** RedirectController {

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

**public** String confirmID(HttpServletRequest httpServletRequest, Model model) {

String id = httpServletRequest.getParameter("id");

model.addAttribute("id", id);

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

//return "redirect:studentOk.do"; // 새로운 request 요청

**return** "forward:studentOk.do"; // 새로운 요청을 하는게 아니고 forword

}

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

}

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

**public** String studentOk() {

**return** "student/studentOk";

}

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

**public** String studentNg() {

**return** "student/studentNg";

}

@RequestMapping(value="fullpath")

**public** String fullpath() {

**return** "redirect:http://localhost:8181/ch11req/temp/fullPath.jsp";

// redirect시 전체 풀경로를 다 써 줄 수도 있다.

}

}

student/studentOk.jsp

<body>

<h1>studentOk.jsp입니다. ID confirm완료</h1>

<h1>confirmID : ${id }</h1>

<a href=*"fullpath.do"*>fullPath로</a>

</body>

student/studentNg.jsp

<body>

<h1>studentNg.jsp입니다. ID confirm실패</h1>

<h1>confirmID : ${id }</h1>

<a href=*"fullpath.do"*>fullPath로</a>

</body>