

2025년 상반기 K-디지털 트레이닝

스프링 MVC의 Controller

[KB] IT's Your Life

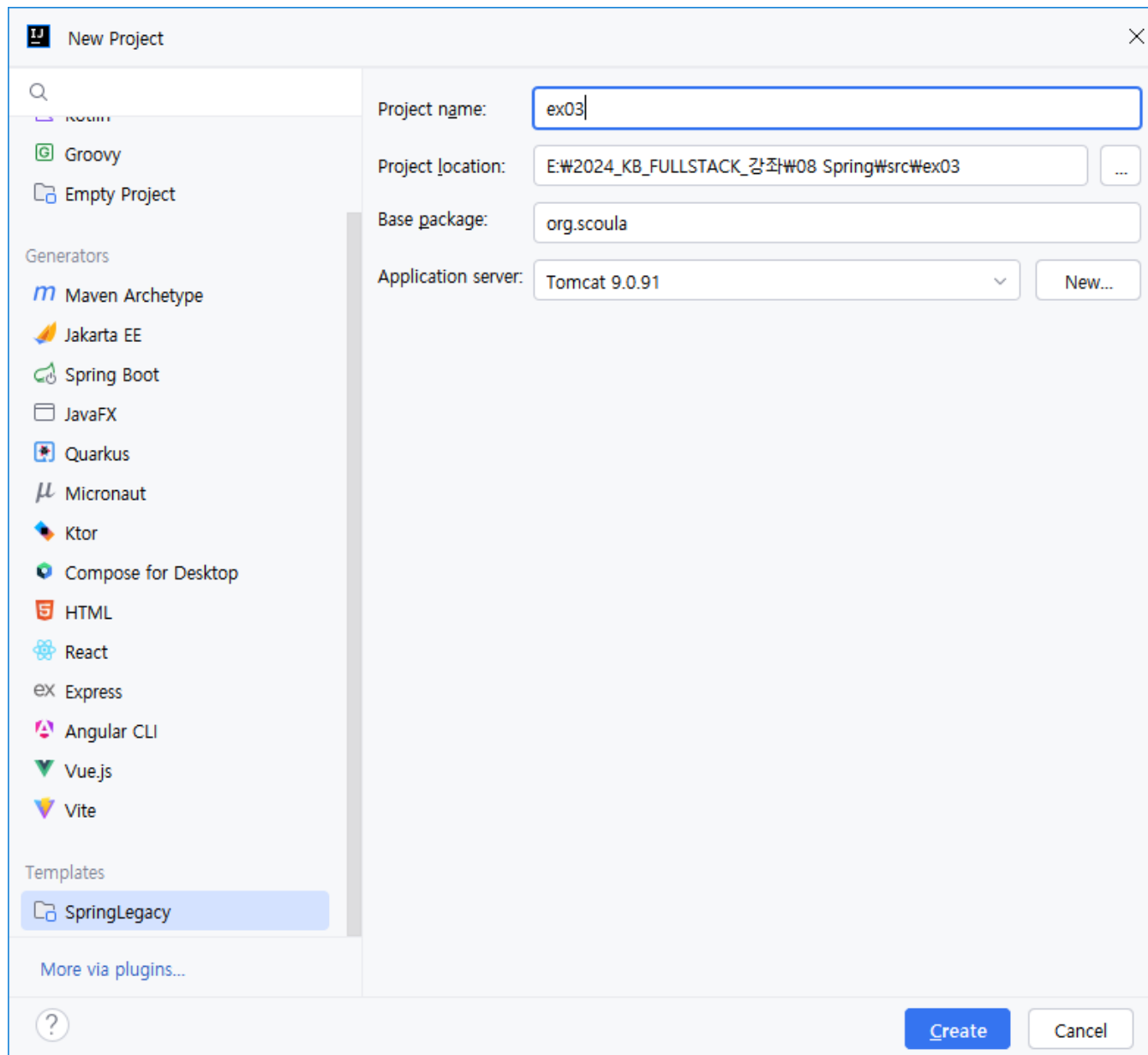
✓ 다음과 같이 프로젝트를 만드세요.

- 템플릿: SpringLegacy
- 프로젝트명: ex03

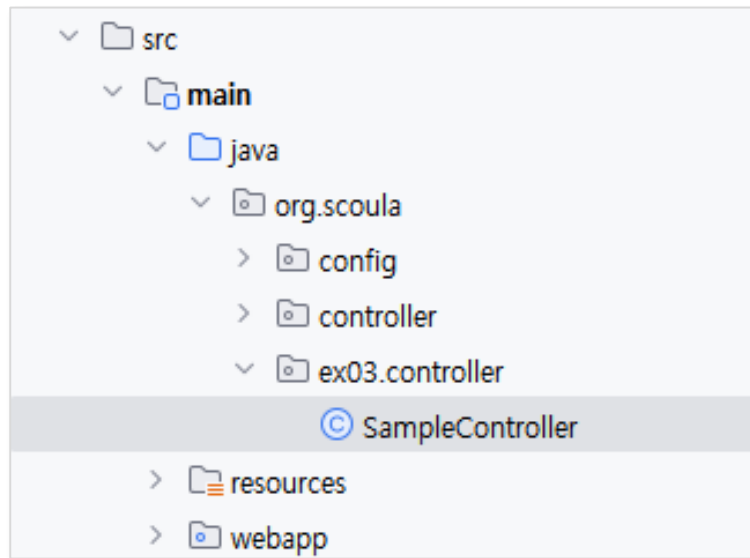
- Tomcat 설정
 - Artifact 조정
 - context path : /

✓ 프로젝트 만들기

- 템플릿: SpringLegacy
- 프로젝트명: ex03



- ✓ 다음과 같이 SampleController를 추가하고, RootConfig의 컴포넌트 스캔 경로에 추가하세요.



ServletConfig.java

```
package org.scoula.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.ViewResolverRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
import org.springframework.web.servlet.view.InternalResourceViewResolver;
import org.springframework.web.servlet.view.JstlView;

@EnableWebMvc
@ComponentScan(basePackages = {
    "org.scoula.controller",
    "org.scoula.ex03.controller"
})
public class ServletConfig implements WebMvcConfigurer {
    ...
}
```

- ✓ SampleController에 다음 작업을 진행하세요.
 - 공통 url을 "/sample"로 설정
 - /sample 경로에 대한 모든 요청 메서드에 대해 호출되는 basic() 메서드 작성
 - 요청이 왔을 경우 로그로 "basic....." 출력

SampleController.java

```
package org.scoula.ex03.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

import lombok.extern.log4j.Log4j2;

@Controller
@RequestMapping("/sample")
@Log4j2
public class SampleController {

    @RequestMapping("")    // url : /sample
    public void basic() {
        log.info("basic.....");
    }
}
```

- ✔ log4j2.xml을 수정하여 HandlerMapping의 등록 정보를 확인하세요.

✓ 확인

- <http://localhost:8080/sample>

- `src/main/resources/log4j2.xml`

```
<Root level="DEBUG">
```

```
<AppenderRef ref="console"/>
```

```
</Root>
```

```
DEBUG: _org.springframework.web.servlet.HandlerMapping.Mappings -
```

```
o.s.c.HomeController:
```

```
{GET [/]}: home()
```

```
DEBUG: _org.springframework.web.servlet.HandlerMapping.Mappings -
```

```
o.s.s.c.SampleController:
```

```
{[/sample]}: basic()
```

```
DEBUG: _org.springframework.web.servlet.HandlerMapping.Mappings - 'beanNameHandlerMapping' {}
```

```
DEBUG: _org.springframework.web.servlet.HandlerMapping.Mappings - 'resourceHandlerMapping'
```

```
{/resources/**=ResourceHttpRequestHandler [ServletContext [/resources/]]}
```

```
DEBUG: org.springframework.jndi.JndiTemplate - Looking up JNDI object with name
```

```
[java:comp/env/spring.liveBeansView.mbeanDomain]
```

```
DEBUG: org.springframework.jndi.JndiLocatorDelegate - Converted JNDI name
```

```
[java:comp/env/spring.liveBeansView.mbeanDomain] not found -
```

- ✓ **SampleController에 다음 작업을 진행하세요.**
 - **/sample/basic** 경로의 GET, POST 요청에 호출되는 **basicGet()** 메서드 작성
 - 요청이 왔을 경우 로그로 "basic get....." 출력
 - **/sample/basicOnlyGet** 경로의 GET 요청에 호출되는 **basicGet2()** 메서드 작성
 - 요청이 왔을 경우 로그로 "basic get only get....." 출력

SampleController.java

```
...
@Controller
@RequestMapping("/sample")
@Log4j2
public class SampleController {

    @RequestMapping("") // url: /sample
    public void basic() {
        log.info("basic.....");
    }

    @RequestMapping(value="/basic", method= {RequestMethod.GET, RequestMethod.POST}) // url: /sample/basic
    public void basicGet(){
        log.info("basic get.....");
    }

    @GetMapping("/basicOnlyGet") // url: /sample/basicOnlyGet
    public void basicGet2(){
        log.info("basic get only get.....");
    }
}
```

- ✓ HomeController에서 다음을 처리하세요.
 - Model을 이용해 name 속성의 값으로 홍길동을 설정
 - index 뷰에서 name 속성을 출력

HomeController.java

```
package org.scoula.controller;

import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
@Slf4j
public class HomeController {

    @GetMapping("/")
    public String home(Model model) {
        model.addAttribute("name", "홍길동");

        return "index";    // View의 이름
    }

}
```

views/index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<h1>${name} 환영합니다.</h1>
</body>
</html>
```

```
@GetMapping("/")
public String home(Model model) {
    model.addAttribute("name", "홍길동");

    return "index";    // View의 이름
}
```

홍길동 환영합니다.

- ✓ 다음과 같이 요청했을 때 SampleController에서 이를 처리하는 ex04() 메서드를 작성하세요.
 - 요청 url: `http://localhost:8080/sample/ex04?name=aaa&age=11&page=9`
 - name과 age는 SampleDTO로 처리
 - page는 단순 int 처리
 - 로그로 SampleDTO와 page 출력
 - 뷰이름: `sample/ex04`
 - `sample/ex04` 뷰에서 SampleDTO와 page 출력

SampleDTI.java

```
...  
import lombok.Data;  
  
@Data  
public class SampleDTO {  
    private String name;  
    private int age;  
}
```


SampleController.java

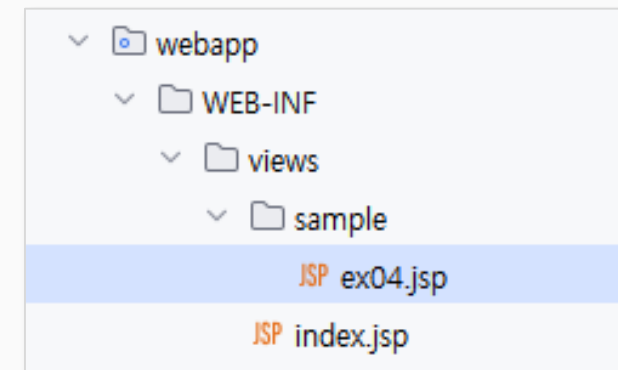
```
...
public class SampleController {
    ...

    @GetMapping("/ex04")
    public String ex04(SampleDTO dto, int page) {
        log.info("dto: " + dto);
        log.info("page: " + page);

        return "sample/ex04";
    }
}
```

views/sample/ex04.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h2>SAMPLE DTO ${sampleDTO}</h2>
    <h2>PAGE ${page}</h2>
</body>
</html>
```



✓ <http://localhost:8080/sample/ex04?name=aaa&age=11&page=9>

INFO : org.scoula.controller.SampleController - dto: SampleDTO(name=aaa, age=11)

INFO : org.scoula.controller.SampleController - page: 9

SAMPLE DTO SampleDTO(name=aaa, age=11)

PAGE

→ page는 뷰에 전달되지 않음

- ✔ 앞의 문제에서 page가 뷰로 넘어가도록 수정하세요.

SampleController.java

```
...
public class SampleController {
    ...

    @GetMapping("/ex04")
    public String ex04(SampleDTO dto, @ModelAttribute("page") int page) {
        log.info("dto: " + dto);
        log.info("page: " + page);

        return "sample/ex04";
    }
}
```

<http://localhost:8080/sample/ex04?name=aaa&age=11&page=9>

SAMPLE DTO SampleDTO(name=aaa, age=11)

PAGE 9

- ✔ SampleController에 다음과 같이 정의되어있을 때 이를 위한 뷰 파일을 작성하세요.

```
@GetMapping("/ex05")
public void ex05() {
    log.info("/ex05.....");
}
```

요청 url을 jsp의 경로로 해석

`/sample/ex05` → `/WEB-INF/views/sample/ex05.jsp`

- ✓ SampleController에 다음을 처리하는 ex06 메서드를 작성하세요.
 - GET으로 /sample/ex06 요청시 /sample/ex06-2로 리다이렉트
 - 리다이렉트될 때 name=AAA, age=10을 쿼리 문자열에 추가

SampleController.java

```
...
public class SampleController {
    ...

    @GetMapping("/ex06")
    public String ex06(RedirectAttributes ra) {
        log.info("/ex06.....");
        ra.addAttribute("name", "AAA");
        ra.addAttribute("age", 10);

        return "redirect:/sample/ex06-2";
    }
}
```