

2025년 상반기 K-디지털 트레이닝

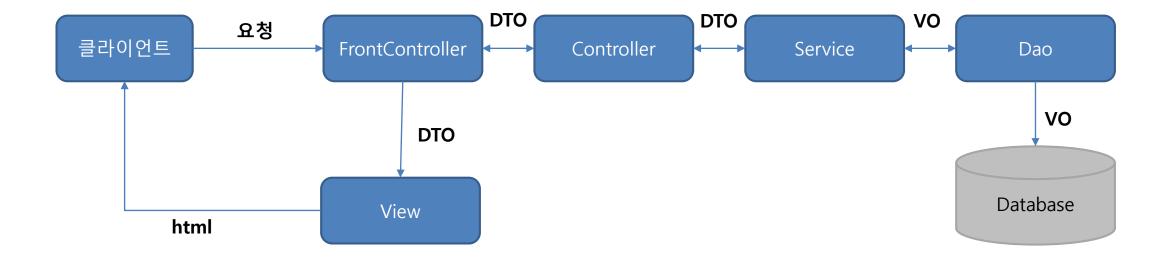
FrontController

[KB] IT's Your Life



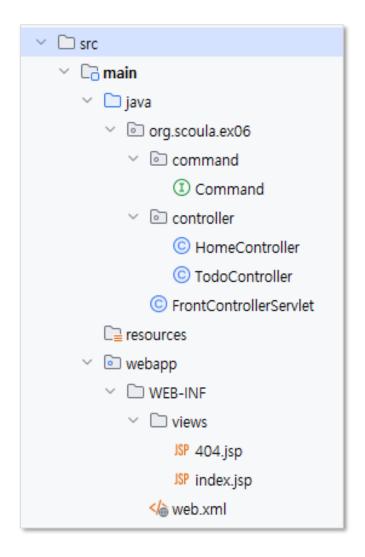
FrontController

- 모든 요청을 받고, 요청의 종류에 따라 작업을 분기하는 Dispatcher Servlet
- o MVC 패턴에서 제일 앞 단에 위치



🗸 프로젝트 만들기

○ 프로젝트 명: ex06



```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    @Override
    public void init() {
    @Override
    protected void doGet(HttpServletReguest reg, HttpServletResponse resp)) throws ServletException, IOException {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)) throws ServletException, IOException {
        doGet(req, resp);
```

FrontController

- 사용자 요청 식별 방법
 - url의 구성
 - http://서비IP번호:포트번호/context명/경로(식별값) 경로(식별값): 수행해야할 명령에 해당

FrontController

○ 식별값 추출하기

```
http://localhost:8090/context명/식별값 getRequestURI()

private String getCommandName(HttpServletRequest request) {
   String requestURI = req.getRequestURI();
   String contextPath = req.getContextPath();
   return requestURI.substring(contextPath.length());
}
```

FrontController

요청별 처리 코드 찾기

- Command 패턴 적용
- Map<String, Command>
 - 키: 요청 식별값
 - 값: Command 인터페이스 구현체(메서드 참조)

```
/create → CreateCommand
/get → GetCommand
...
```

■ Map에 없다면 404 에러

Command.java

```
package org.scoula.ex06.command;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public interface Command {
    String execute(HttpServletRequest req, HttpServletResponse resp) throws IOException;
}
```

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    Map<String, Command> getMap;
    Map<String, Command> postMap;

    public void init() {
        getMap = new HashMap<>();
        postMap = new HashMap<>();
    }
    ...
}
```

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    private String getCommandName(HttpServletRequest req) {
        String requestURI = req.getRequestURI();
        String contextPath = reg.getContextPath();
        return requestURI.substring(contextPath.length());
    private Command getCommand(HttpServletRequest reg) {
        String commandName = getCommandName(req);
        Command command;
        if(req.getMethod().equalsIgnoreCase("GET")) {
            command = getMap.get(commandName);
        } else {
            command = postMap.get(commandName);
        return command;
```

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    public void execute(Command command, HttpServletRequest req, HttpServletResponse resp)
                throws IOException, ServletException {
        String viewName = command.execute(req, resp);
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {
        Command command = getCommand(req);
        if(command != null) {
            execute(command, req, resp);
        } else { // 404 에러 처리
```

🗸 컨트롤러

- 실제 요청을 처리하고 흐름을 제어하는 역할
- 각 메서드는 Command 인터페이스와 일치하게 작성

o GET 요청

- 리턴값은 뷰의 이름
 "<u>/WEB-INF/views/</u>" + 뷰이름 + "<u>.jsp</u>"
 prefix suffix → FrontController에서 처리
- 예: todo/list
 - /WEB-INF/views/todo/list.jsp
- FrontController에서 forward 처리

o POST 요청

- 리턴값은 리다이렉트할 url "redirect:reriect url" 형식
- FrontController에서 redirect 처리

HomeController.java

```
package org.scoula.ex06.controller;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class HomeController {
    public String getIndex(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        return "index";
    }
}
```

view 이름: index;
→ /WEB-INF/views/index.jsp

✓ WEB-INF/views/index.jsp

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    String prefix = "/WEB-INF/views/";
    String suffix = ".jsp";
    HomeController homeController = new HomeController();
    public void init() {
        getMap.put("/", homeController::getIndex);
    public void execute(Command command, HttpServletRequest req, HttpServletResponse resp)
              throws IOException, ServletException {
        String viewName = command.execute(reg, resp;
        if(viewName.startsWith("redirect:")) { // redirect 처리
            resp.sendRedirect(viewName.substring("redirect:".length()));
        } else { // forward 처리
            String view = prefix + viewName + suffix;
            RequestDispatcher dis = req.getRequestDispatcher(view);
            dis.forward(reg, resp);
```

o http://localhost:8080/

FrontController

<u>Todo 목록보기</u>

404에러 처리

- URL 맵핑 맵에 없는 경우
 - WEB-INF/views/404.jsp로 리다렉트

☑ WEB-INF/views/404.jsp

```
public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {
    Command command = getCommand(request);
    if(command != null) {
        execute(command, req, resp;
    } else { // 404 에러 처리
        String view = prefix + "404" + suffix;
        RequestDispatcher dis = req.getRequestDispatcher(view);
        dis.forward(req, resp);
    }
}
```

o http://localhost:8080/abc

404

요청하신 페이지가 존재하지 않습니다.

Todo 만들기

METHOD	URL	컨트롤러 메서드	뷰 이름
GET	/todo/list	getList	todo/list
GET	/todo/view	getView	todo/view
GET	/todo/create	getCreate	todo/create
POST	/todo/create	postCreate	redirect:/todo/list
GET	/todo/update	getUpdate	todo/update
POST	/todo/update	postUpdate	redirect:/todo/list
POST	/todo/delete	postdDelete	redirect:/todo/list

TodoController.java

```
package org.scoula.ex06.controller;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Arrays;
import java.util.List;
import static sun.jvm.hotspot.code.CompressedStream.L;
public class TodoController {
    public String getList(HttpServletRequest reg, HttpServletResponse resp) throws IOException {
        List<String> list = Arrays.asList("Todo 1", "Todo 2", "Todo 3");
        req.setAttribute("todoList", list);
        System.out.println("GET /todo/list");
        return "todo/list";
    public String getView(HttpServletRequest req, HttpServletResponse resp) throws IOException {
        System.out.println("GET /todo/view");
        return "todo/view";
```

TodoController.java

```
public String getCreate(HttpServletRequest req, HttpServletResponse resp) throws IOException {
    System.out.println("GET /todo/create");
   return "todo/create";
public String postCreate(HttpServletRequest reg, HttpServletResponse resp) throws IOException {
   System.out.println("POST /todo/create");
    return "redirect:/todo/list";
public String getUpdate(HttpServletRequest req, HttpServletResponse resp) throws IOException {
    System.out.println("GET /todo/update");
    return "todo/update";
public String postUpdate(HttpServletRequest req, HttpServletResponse resp) throws IOException {
    System.out.println("POST /todo/update");
   return "redirect:/todo/list";
public String postDelete(HttpServletRequest req, HttpServletResponse resp) throws IOException {
    System.out.println("POST /todo/delete");
    return "redirect:/todo/list";
```

WEB-INF/views/todo/list.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    <h1>Todo 목록 보기</h1>
    <div>
        ${todoList}
        <a href="view" >상세보기</a>
    </div>
    <div>
        <a href="create">새 Todo</a>
    </div>
</body>
</html>
```

http://localhost:8080/todo/list

Todo 목록 보기

[Todo 1, Todo 2, Todo 3] <u>상세보기</u> <u>새 Todo</u>

✓ WEB-INF/views/todo/view.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
   <title>Title</title>
</head>
<body>
    <h1>Todo 보기</h1>
    <div>
       <a href="list">목록보기</a> ¦
       <a href="update">수정하기</a>
    </div>
    <form action="delete" method="POST">
       <input type="submit" value="삭제">
    </form>
</body>
</html>
```

Todo 보기

<u>목록보기</u> | <u>수정하기</u> 삭제

WEB-INF/views/todo/create.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
   <title>Title</title>
</head>
<body>
   <h1>새 Todo 생성</h1>
   <form method="POST">
       <input type="submit">
   </form>
                                      새 Todo 생성
</body>
</html>
                                       제출
```

WEB-INF/views/todo/update.jsp

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    TodoController todoController = new TodoController();
    public void init() {
        getMap = new HashMap<>();
        postMap = new HashMap<>();
        getMap.put("/", homeController::getIndex);
        getMap.put("/todo/list", todoController::getList);
        getMap.put("/todo/view", todoController::getView);
        getMap.put("/todo/create", todoController::getCreate);
        qetMap.put("/todo/update", todoController::getUpdate);
        postMap.put("/todo/create", todoController::postCreate);
        postMap.put("/todo/update", todoController::postUpdate);
        postMap.put("/todo/delete", todoController::postDelete);
```

😕 또 다른 애플리케이션 제작 시 FrontController에서 수정되야 하는 부분은 어디인가?

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends HttpServlet {
    HomeController homeController = new HomeController();
    TodoController todoController = new TodoController();
    String prefix = "/views/";
    String suffix = ".jsp";
    public void init() {
        getMap = new HashMap<>();
        postMap = new HashMap<>();
        getMap.put("/", homeController::getIndex);
        getMap.put("/todo/list", todoController::getList);
        getMap.put("/todo/view", todoController::getView);
        getMap.put("/todo/create", todoController::getCreate);
        getMap.put("/todo/update", todoController::getUpdate);
        postMap.put("/todo/create", todoController::postCreate);
        postMap.put("/todo/update", todoController::postUpdate);
        postMap.put("/todo/delete", todoController::postDelete);
```

- ☑ 공통부분은 부모 클래스에서 정의
- ☑ 애플리케이션 특화 부분은 자식 클래스에서 정의

DispatcherServlet.java (부모 클래스)

```
// @WebServlet 애너테이션 붙이지 않음
public class DispatcherServlet extends HttpServlet {
   Map<String, Command> getMap;
   Map<String, Command> postMap;
    String prefix = "/views/";
    String suffix = ".jsp";
    public void init() {
       getMap = new HashMap<>();
       postMap = new HashMap<>();
       createMap(getMap, postMap);
    protected void createMap(Map<String, Command> getMap, Map<String, Command> postMap) {
```

FrontControllerServlet.java

```
@WebServlet(name = "frontControllerServlet", value = "/")
public class FrontControllerServlet extends DispatcherServlet {
    HomeController homeController = new HomeController();
    TodoController todoController = new TodoController();
    @Override
    protected void createMap(Map<String, Command> getMap, Map<String, Command> postMap) {
        getMap.put("/", homeController::getIndex);
        getMap.put("/todo/list", todoController::getList);
        getMap.put("/todo/view", todoController::getView);
        getMap.put("/todo/create", todoController::getCreate);
        getMap.put("/todo/update", todoController::getUpdate);
        postMap.put("/todo/create", todoController::postCreate);
        postMap.put("/todo/update", todoController::postUpdate);
        postMap.put("/todo/delete", todoController::postDelete);
```

상속을 안하는 방법?