KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

FrontController (심화1)

[KB] IT's Your Life

KB 국민은행

FrontController

- **다음 조건으로 HomeController를 작성하세요.**
  - 패키지: org.scoula.ex06.controller
  - 메서드:
    - 메서드명: getIndex
    - Command 인터페이스와 동일한 구조를 가짐
    - 뷰이름 index 리턴

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.jsp를 다음과 같이 작성한다.

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>JSP - Hello World</title>
</head>
<body>
  <h1>FrontController</h1>
  <br/>
  <a href="/todo/list">Todo 목록보기</a>
</body>
</html>
```

FrontController

- **다음 조건을 처리하는 부분을 FrontControllerServlet에 추가하세요.**
  - 뷰 이름의 접두어는 /WEB-INF/views/, 접미어는 .jsp로 함
  - 필드
    - String prefix
    - String suffix
    - HomeController

    → 각각 바로 초기화 한다.

  - init() 메서드에 "/"요청에 대한 HomeController의 getIndex 맵핑을 getMap에 추가하세요.

FrontControllerServlet.java

```
@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);
    }
            ...
}
```

FrontControllerServlet의 execute() 메서드에 다음을 추가하세요.

```
@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);
       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(req, resp);
       }
   }
  ...
}
```

FrontController

- URL 맵핑 맵에 없는 경우 404에러 처리를 위한 404.jsp 파일을 작성하세요.

WEB-INF/views/404.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>

    <h1>404</h1>
    <h3>요청하신 페이지가 존재하지 않습니다.</h3>

</body>
</html>
```

FrontController

- FrontController의 doGet() 메서드에서 요청을 처리할 수 없는 경우 404 페이지로 포워딩하는 코드를 추가하세요.

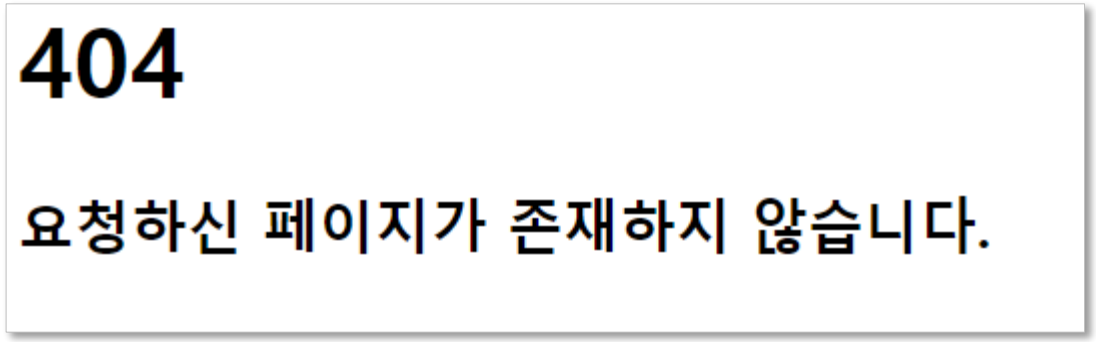
FrontControllerServlet.java

```
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
        String view = prefix + "404" + suffix;
        RequestDispatcher dis = req.getRequestDispatcher(view);
        dis.forward(req, resp);
    }
}
```

FrontController

- 다음 url을 요청하여 그 결과를 확인하세요.
  - http://localhost:8080/

  - http://localhost:8080/test



KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

FrontController (심화2)

[KB] IT's Your Life

KB 국민은행

FrontController

- **다음 테이블 정보를 이용하여 TodoController를 작성하세요.**

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

  - 각 메서드에서는 다음 양식으로 콘솔에 출력한다.
    - [메서드명] [요청 URL]
  - getList()에서는 "Todo 1", "Todo 2", "Todo 3"를 List 로 만들고, request 스코프에 속성명 todoList로 저장한다.

TodoController.java

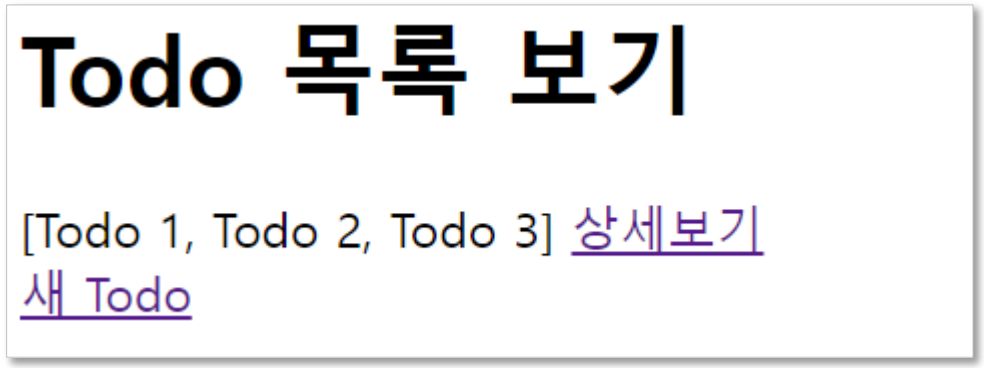
```
package org.scoula.ex06.controller;

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 req, 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 req, 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";
    }
}
```

FrontController

- **다음 조건으로 list.jsp를 작성하세요.**
  - 올바른 뷰 위치에 작성함
  - todoList를 EL로 단순 출력하세요.
  - 상세보기 링크 view
  - 새 Todo 링크 create



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>
```

- **다음 조건으로 view.jsp를 작성하세요.**
  - 올바른 뷰 위치에 작성함
  - 목록보기 링크 list
  - 수정하기 링크 update
  - 삭제를 위한 폼
    - action: delete
    - method: POST

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>
```

- **다음 조건으로 create.jsp를 작성하세요.**
  - 생성 폼 작성
    - action: 현재 url
    - method: POST

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>

</body>
</html>
```

- **다음 조건으로 update.jsp를 작성하세요.**
  - 수정 폼 작성
    - action: 현재 url
    - method: POST

WEB-INF/views/todo/update.jsp

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

FrontController

- 각 요청별로 TodoController의 메서드 맵핑관계를 FrontController의 getMap과 postMap에 추가하세요.
- 각 요청별로 잘 동작하는지 확인하세요.

FrontControllerServlet.java

```
@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);
        getMap.put("/todo/update", todoController::getUpdate);

        postMap.put("/todo/create", todoController::postCreate);
        postMap.put("/todo/update", todoController::postUpdate);
        postMap.put("/todo/delete", todoController::postDelete);
    }
    ...
}
```

FrontController

- **FrontController의 재사용을 위해 DispatcherServlet 부모 클래스를 작성하세요.**
  - 실제 url 맵핑은 createMap() 메서드에서 하게 됨
  - 실제 구현은 자식 클래스가 작성함

DispatcherServlet.java

```
// @WebServlet 애너테이션 붙이지 않음
public class DispatcherServlet extends HttpServlet {
    Map<String, Command> getMap;
    Map<String, Command> postMap;

    String prefix = “/WEB-INF/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) {

    }
    …
}
```

FrontController

- DispatcherServlet을 상속 받아 작성하는 것으로 FrontControllerServlet를 리팩토링하세요.

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);
    }
}
```