

```
package com.fastcampus.ch2;

import java.io.File;
import java.io.IOException;
import java.lang.reflect.Method;
import java.lang.reflect.Parameter;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Scanner;

import org.springframework.ui.Model;
import org.springframework.validation.support.BindingAwareModelMap;

public class MethodCall3 {
    public static void main(String[] args) throws Exception {
        // 1. 요청할 때 제공된 값 - request.getParameterMap();
        Map map = new HashMap();
        map.put("year", "2021");
        map.put("month", "10");
        map.put("day", "1");

        Model model = null;
        Class clazz = Class.forName("com.fastcampus.ch2.YoilTellerMVC");
        Object obj = clazz.newInstance();

        // main 함수를 가져온다.
        // YoilTellerMVC.main(int year, int month, int day, Model model)
        Method main = clazz.getDeclaredMethod("main", int.class, int.class,
        int.class, Model.class);
        // String viewName = (String) main.invoke(obj, new Object[] { 2021, 10, 1,
        model });

        // -----
        Parameter[] paramArr = main.getParameters(); // main메서드의 매개변수 목록을
        가져온다.
        Object[] argArr = new Object[paramArr.length]; // 매개변수 갯수와
        같은 길이의 Object배열을 생성

        for (int i = 0; i < paramArr.length; i++) {
            String paramName = paramArr[i].getName(); // 매개변수의 이름을 알기
            Class paramType = paramArr[i].getType(); // 매개변수의 타입을 알기
            Object value = map.get(paramName); // map에서 못찾으면 value는 null

            // paramType중에 Model이 있으면, 생성 & 저장
            if (paramType == Model.class) { //
                argArr[i] = model = new BindingAwareModelMap();
            } else if (value != null) { // map에 paramName이 있으면,
                // value와 parameter의 타입을 비교해서, 다르면 변환해서 저장
                argArr[i] = convertTo(value, paramType);
            }
        }
    }
}
```

```

    }
    System.out.println("paramArr=" + Arrays.toString(paramArr));
    System.out.println("argArr=" + Arrays.toString(argArr));
    // -----

    // Controller의 main()을 호출 - YoilTellerMVC.main(int year, int month,
int day,
    // Model model)
    String viewName = (String) main.invoke(obj, argArr);
    System.out.println("viewName=" + viewName);

    // Model의 내용을 출력
    System.out.println("[after] model=" + model);

    // 텍스트 파일을 이용한 rendering
    render(model, viewName);
} // main

private static Object convertTo(Object value, Class type) {
    if (type == null || value == null || type.isInstance(value)) // 타입이 같으
면 그대로 반환
        return value;

    // 타입이 다르면, 변환해서 반환
    if (String.class.isInstance(value) && type == int.class) { // String ->
int
        return Integer.valueOf((String) value);
    } else if (String.class.isInstance(value) && type == double.class) { //
String -> double
        return Double.valueOf((String) value);
    }

    return value;
}

private static void render(Model model, String viewName) throws IOException {
    String result = "";

    // 1. 뷰의 내용을 한줄씩 읽어서 하나의 문자열로 만든다.
    Scanner sc = new Scanner(new File("src/main/webapp/WEB-INF/views/" +
viewName + ".jsp"), "utf-8");

    while (sc.hasNextLine())
        result += sc.nextLine() + System.lineSeparator();

    // 2. model을 map으로 변환
    Map map = model.asMap();

    // 3. key를 하나씩 읽어서 template의 ${key}를 value바꾼다.
    Iterator it = map.keySet().iterator();

    while (it.hasNext()) {
        String key = (String) it.next();

```

```
        // 4. replace()로 key를 value 치환한다.
        result = result.replace("${" + key + "}", "" + map.get(key));
    }

    // 5. 렌더링 결과를 출력한다.
    System.out.println(result);
}

/*
 * [실행결과] paramArr=[int year, int month, int day, org.springframework.ui.Model
 * model] argArr=[2021, 10, 1, {}] viewName=yoil [after] model={year=2021,
 * month=10, day=1, yoil=금} <%@ page contentType="text/html; charset=utf-8" %>
 * <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %> <%@ page
 * session="false" %> <html> <head> <title>YoilTellerMVC</title> </head> <body>
 * <h1>2021년 10월 1일은 금요일입니다.</h1> </body> </html>
 */
```