

```
package com.fastcampus.ch2;

import java.lang.reflect.Field;
import java.lang.reflect.Method;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

import org.springframework.util.StringUtils;

public class SetterCall {
    public static void main(String[] args) throws Exception{
        Map<String, String> map = new HashMap<>();
        map.put("year", "2021");
        map.put("month", "10");
        map.put("day", "1");

        Class<?> type = Class.forName("com.fastcampus.ch2.MyDate");

        // MyDate인스턴스를 생성하고, map의 값으로 초기화한다.
        Object obj = dataBind(map, type);
        System.out.println("obj="+obj); // obj=[year=2021, month=10, day=1]
    } // main

    private static Object dataBind(Map<String, String> map, Class<?> clazz) throws
Exception {
        // 1. MyDate인스턴스 생성
        // Object obj = clazz.newInstance(); // deprecated method
        Object obj = clazz.getDeclaredConstructor().newInstance(new Object[0]);

        // 2. MyDate인스턴스의 setter를 호출해서, map의 값으로 MyDate를 초기화
        // 2-1. MyDate의 모든 iv를 돌면서 map에 있는지 찾는다.
        // 2-2. 찾으면, 찾은 값을 setter로 객체에 저장한다.
        Field[] ivArr = clazz.getDeclaredFields();

        for(int i=0;i<ivArr.length;i++) {
            String name = ivArr[i].getName();
            Class<?> type = ivArr[i].getType();

            // map에 같은 이름의 key가 있으면 가져와서 setter호출
            Object value = map.get(name); // 못찾으면 value의 값은 null
            Method method = null;

            try { // map에 iv와 일치하는 키가 있을 때만, setter를 호출
                if(value==null) continue;

                method = clazz.getDeclaredMethod(getSetterName(name), type); //
setter의 정보 얻기
                System.out.println("method="+method);
                method.invoke(obj, convertTo(value, type)); // obj의 setter를 호출
            } catch(Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

    }
}

System.out.println(Arrays.toString(ivArr));

return obj;
}

private static Object convertTo(Object value, Class<?> type) {
    // value의 타입과 type의 타입이 같으면 그대로 반환
    if(value==null || type==null || type.isInstance(value))
        return value;

    // value의 타입과 type이 다르면, 변환해서 반환
    if(String.class.isInstance(value) && type==int.class) // String -> int
        return Integer.valueOf(""+value);

    return value;
}

// iv의 이름으로 setter의 이름을 만들어서 반환하는 메서드("day" -> "setDay")
private static String getSetterName(String name) {
//     return "set"+name.substring(0,1).toUpperCase()+name.substring(1);
    return "set" + StringUtils.capitalize(name); //
org.springframework.util.StringUtils
}
}

/*
[실행결과]
method=public void com.fastcampus.ch2.MyDate.setYear(int)
method=public void com.fastcampus.ch2.MyDate.setMonth(int)
method=public void com.fastcampus.ch2.MyDate.setDay(int)
[private int com.fastcampus.ch2.MyDate.year, private int
com.fastcampus.ch2.MyDate.month, private int com.fastcampus.ch2.MyDate.day]
obj=[year=2021, month=10, day=1]
*/

```