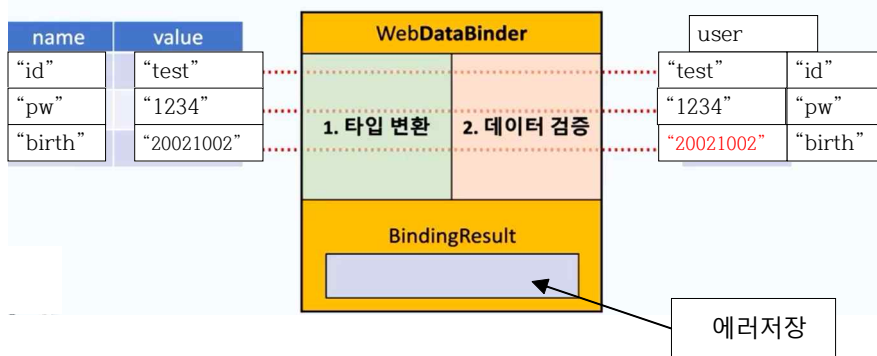


## 데이터변환

WebDataBinder

@PostMapping("/v4")

```
public String test (@ModelAttribute User user , BindingResult result) {
```



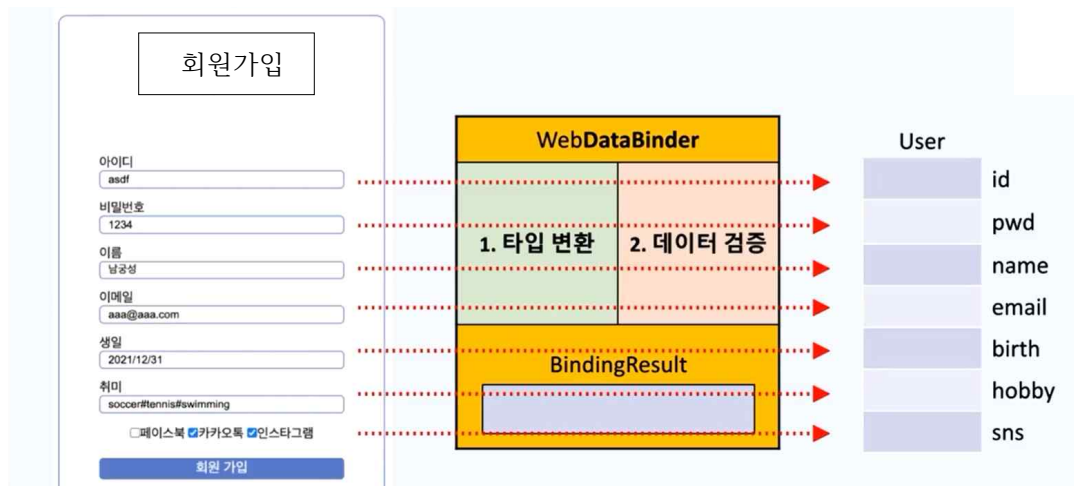
### WebDataBinder

브라우저를 통해서 요청받은 값이 실제 서버의 객체에 바인딩 될 때 중간 역할을 한다

크게 두 가지 작업을 진행한다

1. 타입변환
2. 데이터검증

변환결과나 에러는 BindingResult에 저장



Register Controller에 변환기능 추가하기

```
public class User {
    String id;
    String pw;
    Date birth;
    String[] hobby;
}
```

```
@Setter
@Getter
@ToString
@AllArgsConstructor
@NoArgsConstructor
class User {
    String id;
    String pw;
    @DateFormat(pattern = "yyyy/mm/dd")
    Date birth;
}
```

```
@Controller
public class TestController {
```

- 해당 Controller로 들어오는 요청에 대해
- 추가적인 설정을 하고 싶을 때 사용할 수 있다.
- 모든 요청 전에 InitBinder를 선언한 메소드가 실행된다.

```
@InitBinder
public void toDate( WebDataBinder binder) {
    SimpleDateFormat df = new SimpleDateFormat("yyyy/mm/dd");
    binder.registerCustomEditor( Date.class , new CustomDateEditor( df, false)) ;
    binder.registerCustomEditor(String[].class, new StringArrayPropertyEditor("#"));
    //binder.registerCustomEditor(String[].class, "hobby", new StringArrayPropertyEditor("#"));
}
```

```
@RequestMapping( value="/valid" , method=RequestMethod.GET)
public String reg() {
    return "reg";
}
```

```
@RequestMapping( value="/valid" , method=RequestMethod.POST)
public String reg2(User user , BindingResult result) {
    System.out.println( user);
    System.out.println( result);

    System.out.println( "날짜1 =" + user.getBirth() );
    System.out.println( "날짜 =" + new Date(user.getBirth().getTime()) );
    System.out.println( Arrays.toString(user.getHobby() ) );
    return "reg";
}
```

```
}
```