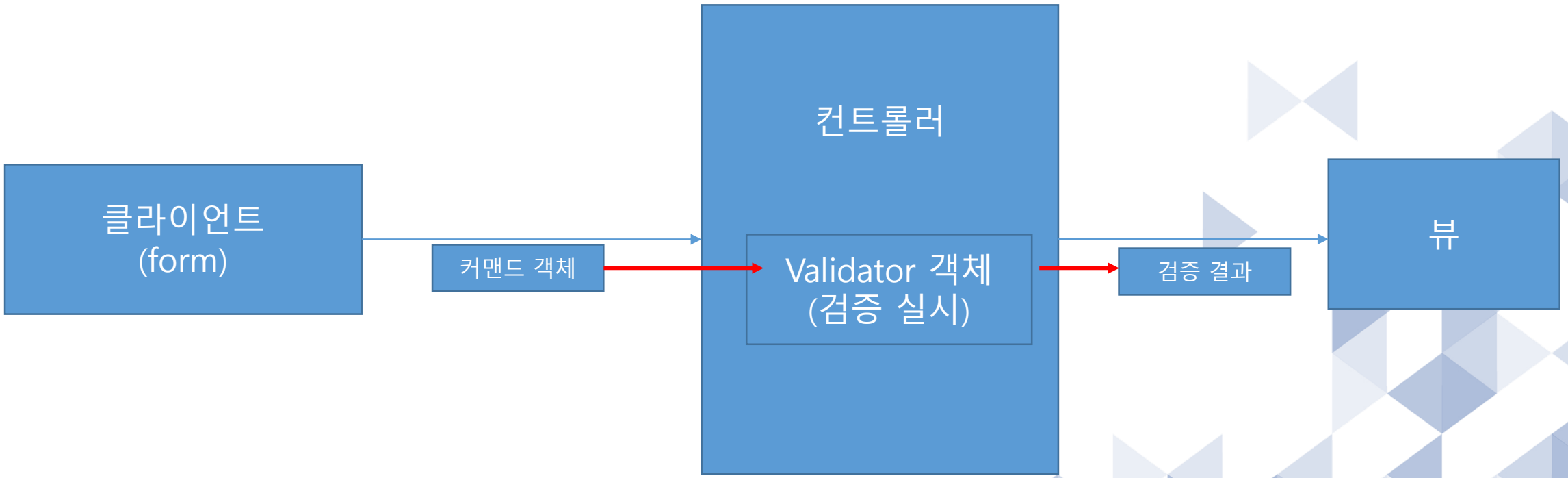
15강. 폼 데이터 값 검증

- Validator를 이용한 검증
- ValidationUtils 클래스
- @Valid와 @InitBinder

Lecturer Kim Myoung-Ho
Nickname 블스
blogstudy@naver.com

15-1. Validator를 이용한 검증

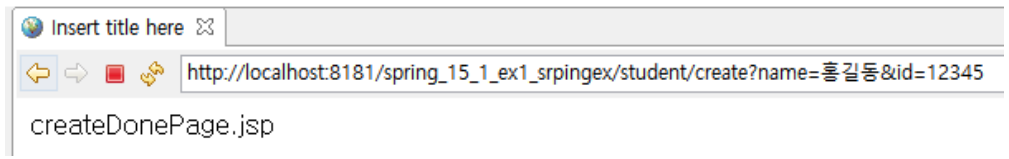
폼에서 전달 되는 데이터를 커맨드 객체에 담아 컨트롤 객체에 전달 한다고 하였습니다.
이때 커맨드 객체의 유효성 검사를 할 수 있습니다.
참고로 javascript을 이용하는 것은 클라이언트에서 검사하는 방법이고, 지금 하는 Validator 인터페이스를 이용하는 방법은 서버에 검사하는 방법 입니다.
(spring_15_1_ex1_srpingex)

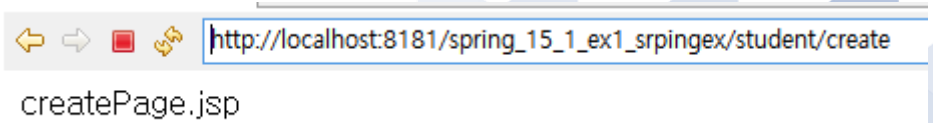


15-1. Validator를 이용한 검증

```java
@RequestMapping("/student/create")
public String studentCreate(@ModelAttribute("student") Student student, BindingResult result) {

    String page = "createDonePage";

    StudentValidator validator = new StudentValidator();
    validator.validate(student, result);
    if(result.hasErrors()) {
        page = "createPage";
    }

    return page;
}
```

```java
public void validate(Object obj, Errors errors) {
    System.out.println("validate()");
    Student student = (Student)obj;

    String studentName = student.getName();
    if(studentName == null || studentName.trim().isEmpty()) {
        System.out.println("studentName is null or empty");
        errors.rejectValue("name", "trouble");
    }

    int studentId = student.getId();
    if(studentId == 0) {
        System.out.println("studentId is 0");
        errors.rejectValue("id", "trouble");
    }
}
```





15-2. ValidationUtils 클래스

데이터 검증을 위해서 Validator 인터페이스의 validate() 메소드를 사용하였습니다.
ValidationUtils 클래스는 validate()메소드를 좀더 편리하게 사용 할 수 있도록 고안된 클래스 입니다.
(spring_15_2_ex1_srpingex)

```
String studentName = student.getName();
if(studentName == null || studentName.trim().isEmpty()) {
    System.out.println("studentName is null or empty");
    errors.rejectValue("name", "trouble");
}
```



```
ValidationUtils.rejectIfEmptyOrWhitespace(errors, "name", "trouble");
```

15-3. @Valid와 @InitBinder

데이터 검증을 하기 위해서 Validator 인터페이스를 구현한 클래스를 만들고, validate()메소드를 직접 호출하여 사용 하였습니다.
이번에는 직접 호출하지 않고, 스프링 프레임워크에서 호출하는 방법에 대해서 살펴 봅니다.
(spring_15_3_ex1_srpingex)

의존 추가

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>4.2.0.Final</version>
</dependency>
```

@InitBinder 추가

```
@InitBinder
protected void initBinder(WebDataBinder binder){
    binder.setValidator(new StudentValidator());
}
```

@Valid 추가

```
@RequestMapping("/student/create")
public String studentCreate(@ModelAttribute("student") Student student, BindingResult result) {

    String page = "createDonePage";

    StudentValidator validator = new StudentValidator();
    validator.validate(student, result);
    if(result.hasErrors()) {
        page = "createPage";
    }

    return page;
}
```

```
@RequestMapping("/student/create")
public String studentCreate(@ModelAttribute("student") @Valid Student student, BindingResult result) {

    String page = "createDonePage";

    StudentValidator validator = new StudentValidator();
    validator.validate(student, result);
    if(result.hasErrors()) {
        page = "createPage";
    }

    return page;
}
```