

Spring MVC

Validator 커스터마이징

Validator

- JSR-303, JSR-380 스펙을 이용한 유효성 검사 방법을 학습하였습니다.
- 지금까지는 유효성 검사를 하기 위해 Hibernate 를 사용하였습니다.
- 이번에는 Validator를 직접 만들어 보도록 하겠습니다.
- JSR-303, JSR-380 스펙으로 유효성 검사를 한 후 추가적으로 다른 유효성 검사를 하고자 할 때 사용합니다.

Validator 인터페이스 구현

- 먼저 Validator 인터페이스를 구현해야 합니다.
 - supports : 유효성 검사할 데이터를 가지고 있는 객체가 유효성 검사가 가능한지 확인합니다.
 - validate : 유효성 검사를 하는 메서드
-

Validator를 컨트롤러에서 등록

- Validator를 컨트롤러에서 등록해줍니다.
- 사용할 Validator가 하나면 setValidator, 한 개 이상이면 addValidators 메서드를 사용합니다.

```
@InitBinder
protected void initBinder(WebDataBinder binder) {
    DataBeanValidator validator1 = new DataBeanValidator();
    binder.setValidator(validator1);
}
```

ValidateUtils

- 지정된 값에 대해 유효성 검사를 하는 메서드
- `rejectIfEmpty(error객체, "프로퍼티이름", "코드이름")` : 값이 비어 있는지 확인합니다. 공백은 글자로 취급합니다.
- `rejectIfEmptyOrWhitespace(error객체, "프로퍼티이름", "코드이름")` : 값이 비어 있거나 공백으로만 구성되어 있는지 확인합니다.
- 입력값에 문제가 있다면 `error` 객체에 오류정보를 저장합니다. 사용할 오류 메시지는 "`코드이름.bean객체이름.프로퍼티이름`"으로 구성됩니다.

rejectValue

- 유효성 조건을 직접 만들어 검사할 때 사용합니다.
- If문으로 유효성 검사를 해주고 위배시 rejectValue를 통해 오류 정보를 지정합니다..
- rejectValue("프로퍼티이름", "코드이름")
- 입력값에 문제가 있다면 error 객체에 오류정보를 저장합니다.
사용할 오류 메시지는 "코드이름.bean객체이름.프로퍼티이름"으로 구성됩니다.

학습정리

○ Validator 커스터마이징