

When an Action method is called we want to grab the data sent to the server.

The old fashion way is getting the value, as a String, from the request object, check if it is null, or maybe with length 0, then try to convert it in a number, or a Date, or any other type, checking any format/parse exception to validate it.

And maybe, if conversion succeeds, we want to perform additional validation.

The framework does all this work for you. Use a `@Param` annotation for every method parameter to bind it to the request parameter received:

```
<input type="text" name="myField">
```

```
protected Response myMethod(@Param("myField") String userName)...
```

Now you have the value received in the request parameter "myField" in the method parameter String userName.

If you want the parameter userName to be compulsory, just inform the framework about it:

```
protected Response myMethod(  
    @Param(value="myField", mandatory=true) String userName)...
```

This way, in the case the value is null or zero length, a mandatory validation message is generated and the method is not called.

Compulsoriness is not the only validation you can perform: **minLength** and **maxLength** can be specified for a String parameter,

minValue

and

maxValue

for a Number parameter, and you can specify a regular expression in the

match

property.

```
@Param(value="myStringField", minLength=5, maxLength=20, match="[a-zA-Z]{1,}.*") String  
name  
@Param(value="myNumberField", minValue=0, maxValue=100) int age
```

When a validation message is generated, the appropriate message key is used to display an error message, as defined in the resource bundle file. The message parameter {0} is used to display the field name that generated the message:

```
myClass.myMethod.myStringField.mandatory=Field {0} is mandatory!
```

By default the request parameter name is used when generating the error message, but this is not always advisable, in this case we would have the message "Field myStringField is mandatory!". If a display name is specified for the parameter, this is used instead:

```
@Param(value="myStringField", displayName="Name")
```

Now the error message would look like "Field Name is mandatory!".