



Spring MVC(Part-2)

- Spring MVC form validation
- Spring MVC Internalization and Localization
- Spring ORM

- Hibernate Validator allows to express and validate application constraints.
- It uses annotations to apply validation rules on properties of a bean.
- you will need the validation-api-1.1.0.Final.jar and hibernate-validator-5.0.1.Final.jar files in order to use the Bean Validation API in your Spring MVC application
- Use @Valid annotation and BindingResult class through which we can get the errors raised by Validator implemented in the controller request handler method.

Hibernate Validators to validate bean properties

- @NotEmpty : Property can not be null or empty.
- @Length : Used to define min and max length of a string.
- @Min : Used to apply constraint on minimum value of number.
- @Max : Used to apply constraint on maximum value of number.
- @Email : used to validate Email Id

- The Bean Validation API allows to declare validation constraints on object models via annotations.

```
public class Employee {  
    @NotEmpty  
    @Email  
    private String email;  
    @NotEmpty  
    @Size(min = 6, max = 15)  
    private String password;  
    // getters and setters  
}
```

Configuring Support For MVC Form Validation

- Spring MVC provides full support for the Bean Validation with minimal configuration.
- Add the following entry to Spring's application context XML file:

```
<mvc:annotation-driven />
```
- Spring MVC will detect and enable the validation support automatically.
- Now in the controller class, annotate the model object that is backing the form by the `@Valid` annotation (`javax.validation.Valid`):

@Controller

```
public class LoginController {  
    @RequestMapping(value = "/login", method = RequestMethod.POST)  
    public String doLogin(@Valid Employee emp, BindingResult result) {  
        // login logic here    }  
}
```

- Spring validate the model object annotated by the @Valid annotation after binding its properties with inputs from JSP page. Any constraint violations will be exposed as errors in the BindingResult object, thus we can check the violation in the controller's method like this:

```
if (result.hasErrors()) {  
    // form validation error  
} else {  
    // form input is ok }
```

- We would return the input form back to the user when any validation errors occurred. And in the JSP form, we can show validation error messages using the Spring's form errors tag as follows:
- `<form:errors path="email" />`
- The error message can be specified in the validation annotation, for example:
- **`@NotEmpty(message = "Please enter your email addresss.")`**

- **Internationalization**
 - It is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.
- **Localization**
 - It is the process of adapting internationalized software for a specific region or language by adding locale-specific components and translating text.
- The terms are frequently abbreviated as i18n (where 18 stands for the number of letters between the first i and last n in internationalization) and L10n respectively, due to the length of the words.

- Declare these files in spring configuration file.
- **ReloadableResourceBundleMessageSource class**
 - `org.springframework.context.support.ReloadableResourceBundleMessageSource`
 - This class defines the message resources.
- **LocaleChangeInterceptor class**
 - `org.springframework.web.servlet.i18n.LocaleChangeInterceptor`
 - This interceptor is configured to intercept the user request and identify the user locale. It intercept any changes in the locale. These changes are then saved in cookies for future request.
- **CookieLocaleResolver class**
 - `org.springframework.web.servlet.i18n.CookieLocaleResolver`
 - This class will be used to set a cookie in the client request. It store the locale changes in cookies.

What is Hibernate?

- Hibernate is an object-relational mapping (ORM) solution for the Java language:
- it provides an easy to use framework for mapping an object-oriented domain model to a traditional relational database.
- Its purpose is to relieve the developer from a significant amount of relational data persistence-related programming tasks.
- It simplifies the data creation, data manipulation and data access.
- Hibernate is free as open source software .

- **DriverManagerDataSource**
 - Used to contain the information about the database such as driver class name, connection URL, username and password.

- **Configuring the SessionFactory object in spring:**
 - Spring helps you create a Hibernate session factory by providing several factory beans in the Spring container.
- **To configure a session factory object in Spring:**
 - Declare a session factory by using the `org.springframework.orm.hibernate3.LocalSessionFactoryBean` class.
 - Or using `org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean`
 - Provide the path of the Hibernate configuration file

- **LocalSessionFactoryBean/AnnotationSessionFactoryBean**

- It is a spring factory bean that creates hibernate sessionfactory.
- The main purpose of this class is to set up the Hibernate SessionFactory in a spring context.
- The hibernate configuration properties can be passed within the XML file.
- The configuration properties include the hibernate mapping resources, hibernate properties and a datasource.



Infogain Corporation, HQ

485 Alberto Way Los Gatos,
CA 95032 USA
Phone: 408-355-6000
Fax: 408-355-7000

Pune

7th Floor, Bhalerao Towers, CTS No.1669 -
1670, Behind Hotel Pride,
Shivaji Nagar, Pune - 411005
Phone : +91-20-66236700

Infogain Irvine

41 Corporate Park,
Suite 390 Irvine, CA 2606 USA
Phone: 949-223-5100
Fax: 949-223-5110

Infogain Austin

Stratum Executive Center Building D
11044 Research Boulevard Suite 200
Austin, Texas 78759

Noida

A-16, Sector 60, Noida Gautam Budh agar,
201301 (U.P.) India
Phone: +91-120-2445144
Fax: +91-120-2580406

Dubai

P O Box 500588 Office No.105,
Building No. 4, Dubai Outsource Zone,
Dubai, United Arab Emirates
Tel: +971-4-458-7336

