

SlangDictionary

Documentation

Student:

Bleard Rexhaj, ID: 611019

Erdenesaikhan Tserendavga, ID: 611104

Enkhjargal Gansukh, ID: 611031

Team name: Captains

Team slogan: None of us is as smart as all of us

Professor:

Joe Bruen

Course:

Enterprise Architecture CS544, August block of 2020

Contents

User Module	3
Term and REST module	5
Category module	7

User Module by Bleard Rexhaj

- The implementation of a back to front module was made by me, A user will be able to register / login, the technologies used on this module are Validation, Aspect Oriented Programing, Hibernate JPA, Security, Database ORM, Spring Email, Testing.
- Validation – Validation was made by spring validation, main validation annotation used are @NotEmpty, @Email and custom validations like @EmptyOrSize, @FieldMatch that will check whether the password and confirm password will match.
- Spring Email - Spring Email was used in order to complete the implementation of the user, so whenever a user is registered it will be in a disabled state, an email will be generated and sent out and when the user clicks the generated link that includes the token his data will be enabled and ready for use.
- AOP – Aspect oriented programing was used in the User Module with the @Before and @After annotations, there was the need to encrypt the password and add a default role before the user was being persisted to the database and also to send an confirmation email to the user in order for him to confirm his account, and aop was the perfect tool to solve this cross-cutting concern.
- Security - Spring security was user to authorize the user based on his authorities.
- ORM/Hibernate - Hibernate was used to persist ORM java objects.
- Testing – UserIntegrationTest was created to test whether a user will be registered successfully, and whether a user will be able to access a protected path (MockMvcUser).

Package declaration: com.captains.slangdictionary;

1. The implementation of validation that is related to User module can be found on model:

- User.java
- UserCredentials.java
- Address.java

2. The implementation of spring email that is related to User module can be found on:

- service – EmailService.java
- dao ConfirmationTokenDao.java
- aspect UserAspect.java

3. The implementation of aspect-oriented programming can be found on the aspect package

4. UserAspect.java.

5. The implementation of spring security can be found on the security package
SecurityConfig.java

6. The testing of user module can be found on the test package UserIntegrationTest.java

Term and REST module by Enkhjargal Gansukh

Term module is used in Home, My Term list and Rest API pages. This module's main purposed term add, list and search logics to show users and guests by JSP pages.

Rest controller is very good module for external third-party users can access to our data by RESTful API. This module includes add, list and find terms methods.

1. The implementation of Term module entity, term service and data access object.
 - Entity – Term.java
 - Service – TermService.java
 - DAO – TermDao.java
2. The implementation of Term controller and JSP pages
 - Controller – TermController.java
 - JSP – termlist.jsp
3. The implementation of add and search term JSP and controller.
 - Controller – add and search methods in TermController.java
 - JSP – add.jsp, added HTML tags and spring form in termlist.jsp
4. The implementation of REST controller and testing.
 - Controller– RestController.java
 - Can access below URI's

Get all term list: (<http://localhost:8080/api>, method: GET)

Find term by word: (<http://localhost:8080/api/find?word=hello>, method: GET)

Add term: (<http://localhost:8080/api>, method: POST, body: application/json)

Category module by Erdenesaikhan Tserendavga

Category module provide list of categories in category page. The basic requirement for a category page is to categorize how many categories users have. A list of words containing the user-selected category is loaded on the category list page. The developer can add the required categories to the system, and the user cannot add new ones directly.

1. The implementation of Category module entity, category service and data access object.
 - Entity – Category.java
 - Service – CategoryService.java
 - DAO – CategoryDao.java
2. The implementation of Category controller and JSP pages
 - Controller – CategoryController.java
 - JSP – categoryList.jsp
 - JSP – categoryTerms.jsp
3. The implementation of add and search term JSP and controller.
 - Controller – add and search methods in TermController.java
 - JSP – add.jsp, added HTML tags and spring form in termlist.jsp

Get all term list: (<http://localhost:8080/category>, method: GET)