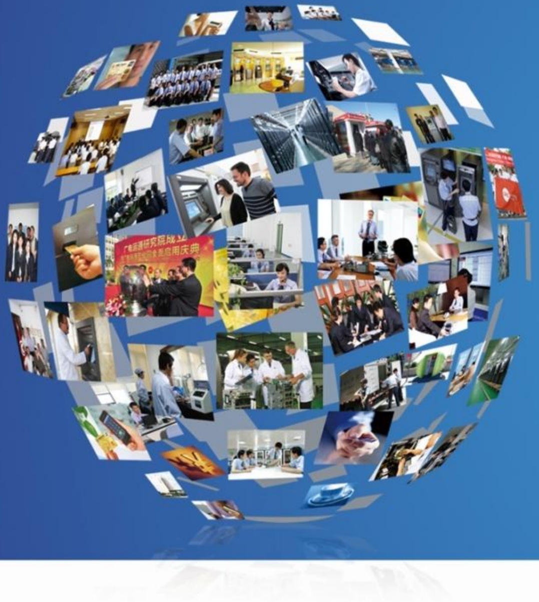


Online Note System

-- William Ouyang

2/4/2016

Project Introduction

- Note it’s very important in our daylily life . Sometimes we may want to record out think anytime and anywhere
- Online Note System make easy and economy then page Note Book
- Project Function
    1. User Management , Note Class Management ,Note Management
- Project Extension(Future Function)
    1. Connection By Android And Iphone device by making a web webservices interface
    2. Real time reminder by Email or Short Message by Create Spring Scheduling

Project Management

- Project Plan and Monitor

  Plan

  Requirement Analysis

  WBS

Schedule	Things To Do	Status
Jan 31	Modeling and Mapping	Meet
Feb 1	Coding	Meet
Feb 2	Integrating & Testing	Meet
Feb 3	Document and Preparation	Meet

Project Demonstration

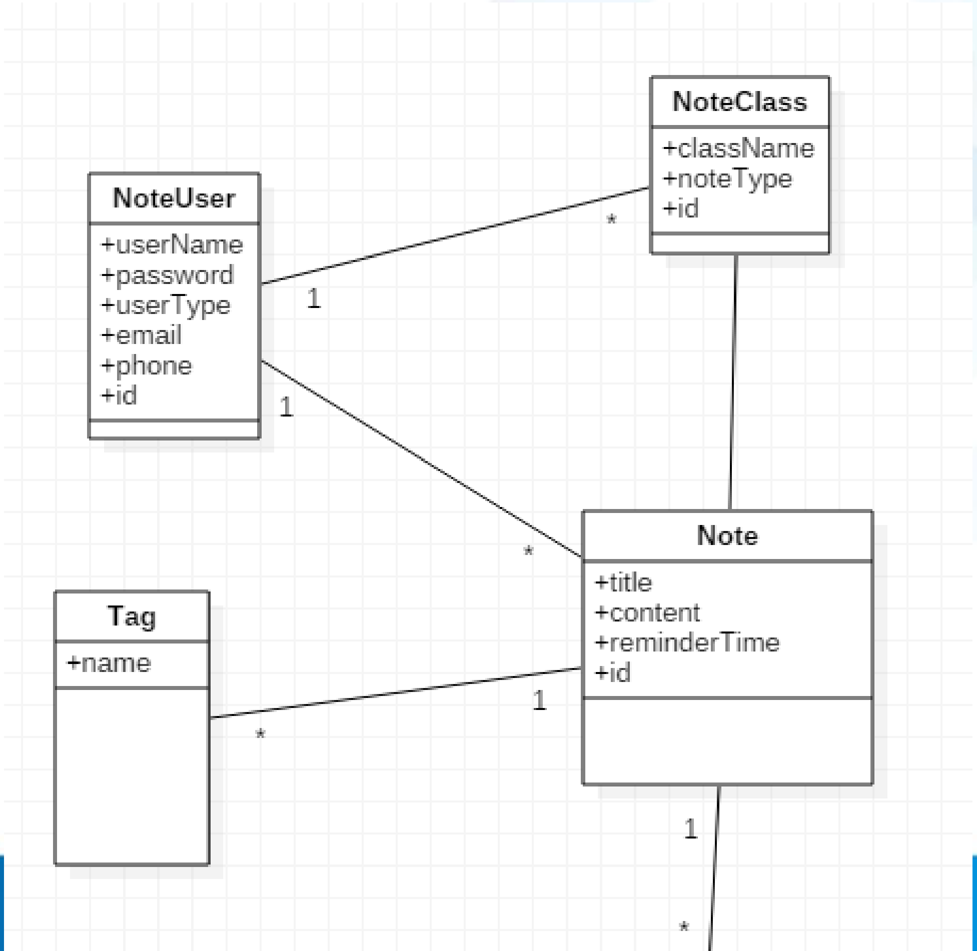
Show how to implement project

- Modeling

- All Project using

  Domain-Driven Design

  Object Oriented



Show how to implement proj

- Project Structure

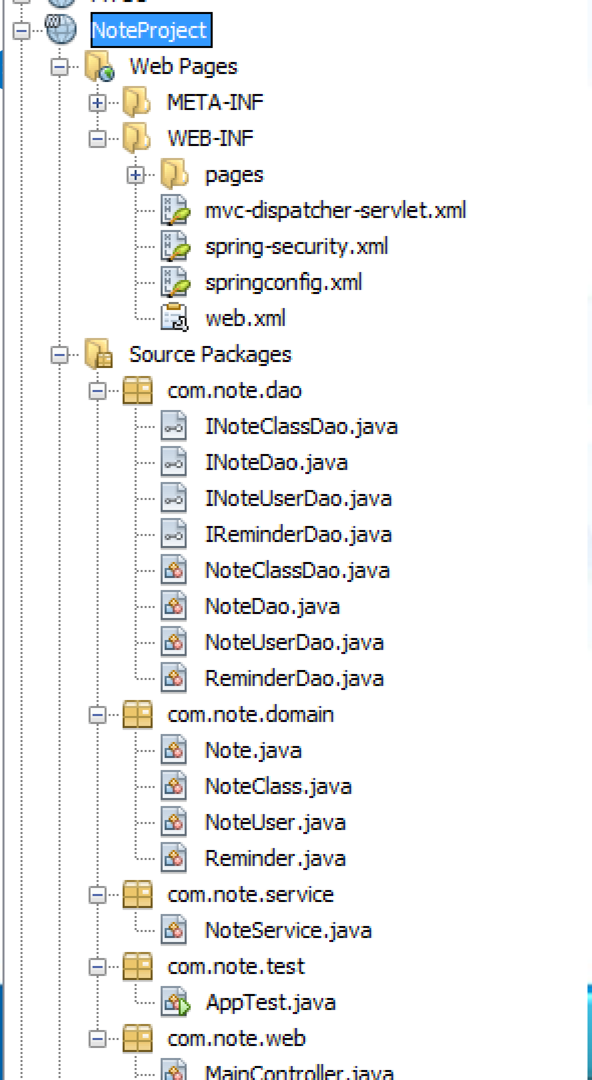
S View

Controller

Service

Domain

DAO

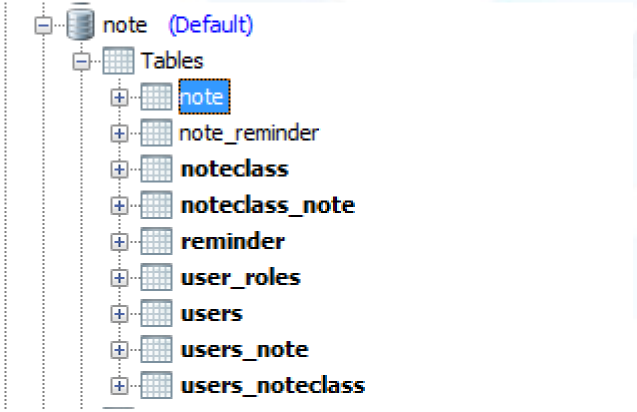


NoteProject
Web Pages
META-INF
WEB-INF
pages
mvc-dispatcher-servlet.xml
spring-security.xml
springconfig.xml
web.xml
Source Packages
com.note.dao
INoteClassDao.java
INoteDao.java
INoteUserDao.java
IReminderDao.java
NoteClassDao.java
NoteDao.java
NoteUserDao.java
ReminderDao.java
com.note.domain
Note.java
NoteClass.java
NoteUser.java
Reminder.java
com.note.service
NoteService.java
com.note.test
AppTest.java
com.note.web
MainController.java

Show how to implement project

- **Mapping**

```
@Entity(name="users")
public class NoteUser {
        @Id
        @GeneratedValue
          private int id;   private String username;   private String password;
          private String userType;   private String email;
          private String phone;
          private int enabled;
         @OneToMany(cascade=CascadeType.ALL)
          List<NoteClass> listNoteClass= new ArrayList<NoteClass>();
         @OneToMany(cascade=CascadeType.ALL)
          List<Note> listNote= new ArrayList<Note>();
```



```
note (Default)
  Tables
    note
    note_reminder
    noteclass
    noteclass_note
    reminder
    user_roles
    users
    users_note
    users_noteclass
```

Show how to implement project

- **Hibernate HQL – select all Note by userID and Note ClassID**

```
@Override
public List<Note> getNoteByNoteClassIDAndUserID(int userId, int
noteclassId) {
        return sessionFactory.getCurrentSession().createQuery(""
select distinct note from users users1 join users1.listNoteClass
noteClass join noteClass.listNote note where users1.id=:userId and
noteClass.id=:noteclassId ).setParameter("userId",
userId).setParameter("noteclassId", noteclassId).list();
    }
```

Show how to implement project

- Spring injection

```
<tx:annotation-driven transaction-manager="txManager" />
<bean id="txManager“  class="org.springframework.orm.hibernate4.HibernateTransactionManager">
                                        <property name="sessionFactory" ref="sessionFactory" />
</bean>
<bean id="noteDao" class="com.note.dao.NoteDao">
                                        <property name="sessionFactory" ref="sessionFactory" />
</bean>
        <bean id="noteClassDao" class="com.note.dao.NoteClassDao">
                                        <property name="sessionFactory" ref="sessionFactory" />
                    </bean>
        <bean id="noteUserDao" class="com.note.dao.NoteUserDao">
                                        <property name="sessionFactory" ref="sessionFactory" />
                    </bean>
<bean id="noteService" class="com.note.service.NoteService">
                    <property name="iNoteClassDao" ref="noteClassDao" />
                     <property name="iNoteDao" ref="noteDao" />
                    <property name="iNoteUserDao" ref="noteUserDao" />
                     <property name="iReminderDao" ref="reminderDao" />
</bean>
```

Show how to implement project

- **Spring Security**

```
<authentication-manager>
	<authentication-provider>
	<jdbc-user-service data-source-ref="dataSource"
	users-by-username-query="select username,password, enabled from
	users where username=? “
	authorities-by-username-query="select username, role from user_roles
	where username =?  " />
	</authentication-provider>
</authentication-manager>
```

Show how to implement project

- **Spring MVC**

```
@Controller
public class MainController {
    @Resource
     NoteService noteService;
  @RequestMapping(value =  "/", method = RequestMethod.GET)
    public ModelAndView defaultPage() {
            ModelAndView model = new ModelAndView();
            model.setViewName("login");
            return model;
}
```

Show how to implement project

- **Transaction**

```
@Transactional
public class NoteService {
    INoteClassDao iNoteClassDao;
    INoteDao iNoteDao;
    INoteUserDao iNoteUserDao;
    IReminderDao iReminderDao;
```

Summary & Lesson Lean

Summary

1. know how to Entity mapping and HQL
2. Know how to DI and AOP
3. Use other Spring function: Spring Security,Spring RMI,Spring JSM,Spring Data

Lesson Lean

① Create a test function class it's more fast then start and redeploy tomcat, need to copy spring config file to "Other" folder

② Spring security 405 Error

Thank You