**Online Collaboration**

**Online Collaboration**

* Create a Spring Boot project named online Collaborate with web, Spring data JPA, Spring Dev Tools and MySQL server driver package. Extract that project.
* Import the project in Eclipse.
* Open package com. coll. Online Collaborate. config.
* Configuration file as HibernateConfig:

package com.coll.OnlineCollaborate.config;

import java.util.Properties;

import javax.sql.DataSource;

import org.hibernate.SessionFactory;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.boot.autoconfigure.orm.jpa.HibernateJpaAutoConfiguration;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.ComponentScans;

import org.springframework.context.annotation.Configuration;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.orm.hibernate5.HibernateTransactionManager;

import org.springframework.orm.hibernate5.LocalSessionFactoryBean;

import org.springframework.orm.hibernate5.LocalSessionFactoryBuilder;

import org.springframework.transaction.annotation.EnableTransactionManagement;

import org.springframework.web.servlet.ViewResolver;

import org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration

@ComponentScans(value= {@ComponentScan("com.coll.OnlineCollaborate"),

@ComponentScan("model"),

@ComponentScan("controller"),

@ComponentScan("dao"),

@ComponentScan("daoImpl"),

@ComponentScan("config"),

@ComponentScan("serviceImpl"),

@ComponentScan("service")})

@EnableAutoConfiguration(exclude= {HibernateJpaAutoConfiguration.class})

@EnableTransactionManagement

public class HibernateConfig {

public static final String DATABASE\_URL="jdbc:mysql://localhost:3306/Collaboration";

public static final String DATABASE\_DRIVER="com.mysql.cj.jdbc.Driver";

public static final String DATABASE\_DIALECT="org.hibernate.dialect.MySQLDialect";

public static final String DATABASE\_USERNAME="root";

public static final String DATABASE\_PASSWORD="niit@123";

@Bean(name="datasource")

public DataSource getDataSource() {

DriverManagerDataSource datasource=new DriverManagerDataSource();

datasource.setDriverClassName(DATABASE\_DRIVER);

datasource.setUrl(DATABASE\_URL);

datasource.setUsername(DATABASE\_USERNAME);

datasource.setPassword(DATABASE\_PASSWORD);

return datasource;

}

@Bean

public LocalSessionFactoryBean getSessionFactory() {

LocalSessionFactoryBean sessionFactory=new LocalSessionFactoryBean();

sessionFactory.setDataSource(getDataSource());

sessionFactory.setPackagesToScan("com.coll.OnlineCollaborate");

Properties hibernateProperties=new Properties();

hibernateProperties.put("hibernate.dialect", DATABASE\_DIALECT);

hibernateProperties.put("hibernate.show\_sql", "true");

hibernateProperties.put("hibernate.hbm2ddl.auto", "update");

sessionFactory.setHibernateProperties(hibernateProperties);

return sessionFactory;

}

@Bean

public HibernateTransactionManager getTranscationManager() {

HibernateTransactionManager txm=new HibernateTransactionManager();

txm.setSessionFactory(getSessionFactory().getObject());

return txm;

}

@Bean

public ViewResolver jspViewResolver()

{

InternalResourceViewResolver viewResolver=new InternalResourceViewResolver();

viewResolver.setPrefix("/views");

viewResolver.setSuffix(".jsp");

return viewResolver;

}

}

* Create schema as Collaboration
* Active Collaboration Schema by following Commend

Use Collaborate;

Create Tables

1.User Tables:

create table User(

UserId int not null auto\_increment,

firstName varchar(30),

lastName varchar(30) ,

username varchar(30),

Password varchar(20),

email varchar(30),

role varchar(10),

status varchar(10),

isOnline varchar(5),

enabled varchar(5),

primary key(UserId)

);

2.Blog Table:

create table Blog(

blogId int not null auto\_increment,

blogTiTle varchar(30),

blogContent varchar(200),

blogPosted datetime,

status varchar(10),

noOfLikes int,

noComments int,

noOfViews int,

userId int,

username varchar(20),

primary key(BlogId),

);

3.BlogComments Tables:

create table BlogComments(

blogCommentId int not null auto\_increment,

userId int ,

username varchar(20),

userProfileId varchar(20),

title varchar(30),

noOfLikes int,

blogcomment varchar(20),

currentDate date,

blogId int,

primary key(blogCommentId),

);

* Create model class:

1.User

package com.coll.OnlineCollaborate.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Transient;

import org.springframework.stereotype.Component;

@Component

@Entity

public class User extends DomainResponse implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

private int userId;

private String firstName;

private String lastName;

private String username;

private String password;

private String email;

private String role;

private String status;

private String isOnline;

private String enabled;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public String getIsOnline() {

return isOnline;

}

public void setIsOnline(String isOnline) {

this.isOnline = isOnline;

}

public String getEnabled() {

return enabled;

}

public void setEnabled(String enabled) {

this.enabled = enabled;

}

}

* Create Interface:

2.IUserDao

package com.coll.OnlineCollaborate.dao

import java.util.List;

import com.coll.OnlineCollaborate.model.User;

public interface IUserDao {

List<User> userListbyStatus(String status);

List<User> getAllUsers();

User getUserById(int UserId);

User getUserByUserName(String Username);

User validateUser(User user);

boolean addUser(User user);

boolean updateUser(User user);

boolean deleteUser(int userId);

boolean deactiveUser(int userId);

List<User> getAllDeactiveUser();

boolean activeUser(int userId);

boolean updateUserProfile(String file,Integer userId);

}

* create DaoImpl class

3.UserDaoImpl

package com.coll.OnlineCollaborate.DaoImpl

import java.util.List;

import javax.persistence.Query;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.Dao.IUserDao;

//import com.coll.OnlineCollaborate.dao.IUserDao;

import com.coll.OnlineCollaborate.model.User;

@Repository("userDao")

@Transactional

public class UserDaoImpl implements IUserDao {

@Autowired

SessionFactory sessionFactory;

@Override

public List<User> userListbyStatus(String status) {

String q="from User where status='"+status+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

@Override

public List<User> getAllUsers() {

return sessionFactory.getCurrentSession().createQuery("from User",User.class).getResultList();

}

@Override

public User getUserById(int UserId) {

return sessionFactory.getCurrentSession().get(User.class,Integer.valueOf(UserId));

}

@Override

public User getUserByUserName(String userName) {

String query="from User where username=:username";

return sessionFactory.getCurrentSession().createQuery(query,User.class).setParameter("username",userName).getSingleResult();

}

@Override

public User validateUser(User user) {

String username=user.getUsername();

String password=user.getPassword();

String q="from User where username='"+username+"' and password='"+password+"' and enabled='true'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

try {

user=(User)query.getSingleResult();

user.setIsOnline("true");

return user;

}

catch(Exception ex){

ex.printStackTrace();

return null;

}

}

@Override

public boolean addUser(User user) {

try {

sessionFactory.getCurrentSession().save(user);

return true;

}

catch(Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateUser(User user) {

try {

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteUser(int userId) {

try {

sessionFactory.getCurrentSession().delete(getUserById(userId));

return true;

}

catch(Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean deactiveUser(int userId) {

try {

User user=getUserById(userId);

user.setEnabled("false");

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateUserProfile(String file, Integer userId) {

String q="update User set profile=:filename where userId=:id";

Query query=sessionFactory.getCurrentSession().createQuery(q);

query.setParameter("id",(Integer)userId);

query.setParameter("filename",file);

try{

query.executeUpdate();

return true;

}

catch(Exception e) {

e.printStackTrace();

return false;

}

}

@Override

public List<User> getAllDeactiveUser() {

return sessionFactory.getCurrentSession().createQuery("from User where enabled='false'",User.class).getResultList();

}

@Override

public boolean activeUser(int userId) {

try {

User user=getUserById(userId);

user.setEnabled("true");

user.setStatus("Active");

sessionFactory.getCurrentSession().update(user);

return true;

}

catch(Exception ex) {

ex.printStackTrace();

return false;

}

}

}

* Create service interface

4.IUserService Interface

package com.coll.OnlineCollaborate.service;

import java.util.List;

import com.coll.OnlineCollaborate.model.User;

public interface IUserService {

List<User> userListbyStatus(String status);

List<User> getAllUsers();

User getUserById(int UserId);

User getUserByUserName(String Username);

User validateUser(User user);

boolean addUser(User user);

boolean updateUser(User user);

boolean deleteUser(int userId);

boolean deactiveUser(int userId);

List<User> getAllDeactiveUser();

boolean activeUser(int userId);

boolean updateUserProfile(String file,Integer userId);

}

* Create serviceImpl

5.UserServiceImpl

package com.coll.OnlineCollaborate.serviceImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.Dao.IUserDao;

//import com.coll.OnlineCollaborate.dao.IUserDao;

import com.coll.OnlineCollaborate.model.User;

import com.coll.OnlineCollaborate.service.IUserService;

@Service

@Transactional

public class UserServiceImpl implements IUserService {

@Autowired

IUserDao userDao;

@Override

public List<User> userListbyStatus(String status) {

return userDao.userListbyStatus(status);

}

@Override

public List<User> getAllUsers() {

return userDao.getAllUsers();

}

@Override

public User getUserById(int UserId) {

return userDao.getUserById(UserId);

}

@Override

public User getUserByUserName(String Username) {

return userDao.getUserByUserName(Username);

}

@Override

public User validateUser(User user) {

return userDao.validateUser(user);

}

@Override

public boolean addUser(User user) {

return userDao.addUser(user);

}

@Override

public boolean updateUser(User user) {

return userDao.updateUser(user);

}

@Override

public boolean deleteUser(int userId) {

return userDao.deleteUser(userId);

}

@Override

public boolean deactiveUser(int userId) {

return userDao.deactiveUser(userId);

}

@Override

public boolean updateUserProfile(String file, Integer userId) {

return userDao.updateUserProfile(file, userId);

}

@Override

public List<User> getAllDeactiveUser() {

return userDao.getAllDeactiveUser();

}

@Override

public boolean activeUser(int userId) {

return userDao.activeUser(userId);

}

}

* Create controller

6.UserController

package com.coll.OnlineCollaborate.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.coll.OnlineCollaborate.model.User;

import com.coll.OnlineCollaborate.service.IUserService;

@RestController

@CrossOrigin(origins="http://localhost:4200")

@RequestMapping(value="/api")

public class UserCotroller {

@Autowired

IUserService userService;

@PostMapping("save-user")

public boolean saveUser(@RequestBody User user) {

return userService.addUser(user);

}

@GetMapping("user-list")

public List<User> allUsers(){

return userService.getAllUsers();

}

@DeleteMapping("delete-user/{userId}")

public boolean deleteUser(@PathVariable("userId") int userId) {

return userService.deleteUser(userId);

}

@GetMapping("user/{userId}")

public User userById(@PathVariable("userId") int userId) {

return userService.getUserById(userId);

}

@PostMapping("update-user/{userId}")

public boolean updateUser(@RequestBody User user,@PathVariable("userId")int userId) {

user.setUserId(userId);

return userService.updateUser(user);

}

@GetMapping("deactive-list")

public List<User> AllDeactiveUsers(){

return userService.getAllDeactiveUser();

}

@PostMapping("active-user/{userId}")

public boolean activeUser(@PathVariable("userId") int userId) {

return userService.activeUser(userId);

}

@PostMapping("validate-user")

public User validateUser(@RequestBody User user) {

return userService.validateUser(user);

}

}

* Create model class

1.Blog:

package com.coll.OnlineCollaborate.model;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import org.springframework.stereotype.Component;

import com.fasterxml.jackson.annotation.JsonManagedReference;

@Component

@Entity

public class Blog extends DomainResponse implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

int blogId;

String blogTitle, blogContent;

LocalDate blogPosted;

String status;

int noOfLikes, noOfComments,noOfViews;

int userId;

String username;

@OneToMany(mappedBy="blog",fetch=FetchType.EAGER, cascade=CascadeType.ALL)

@JsonManagedReference

List<BlogComments> blogComments;

public int getBlogId() {

return blogId;

}

public void setBlogId(int blogId) {

this.blogId = blogId;

}

public String getBlogTitle() {

return blogTitle;

}

public void setBlogTitle(String blogTitle) {

this.blogTitle = blogTitle;

}

public String getBlogContent() {

return blogContent;

}

public void setBlogContent(String blogContent) {

this.blogContent = blogContent;

}

public LocalDate getBlogPosted() {

return blogPosted;

}

public void setBlogPosted(LocalDate blogPosted) {

this.blogPosted = blogPosted;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public int getNoOfLikes() {

return noOfLikes;

}

public void setNoOfLikes(int noOfLikes) {

this.noOfLikes = noOfLikes;

}

public int getNoOfComments() {

return noOfComments;

}

public void setNoOfComments(int noOfComments) {

this.noOfComments = noOfComments;

}

public int getNoOfViews() {

return noOfViews;

}

public void setNoOfViews(int noOfViews) {

this.noOfViews = noOfViews;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public List<BlogComments> getBlogComments() {

return blogComments;

}

public void setBlogComments(List<BlogComments> blogComments) {

this.blogComments = blogComments;

}

}

* Create interface

2.IBlogDao

package com.coll.OnlineCollaborate.Dao;

import java.util.List;

import com.coll.OnlineCollaborate.model.Blog;

public interface IBlogDao {

List<Blog> getAllBlogs();

List<Blog> getBlogsByStatus(String status);

List<Blog> getUsersBlogs(int id);

Blog getBlogById(int blogId);

boolean addBlog(Blog blog);

boolean updateBlog(Blog blog);

boolean deleteBlog(Blog blog);

}

* Create BolgDaoImpl

3.BlogDaoImpl

package com.coll.OnlineCollaborate.DaoImpl;

import java.util.List;

import org.hibernate.SessionFactory;

import org.hibernate.query.Query;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.Dao.IBlogDao;

import com.coll.OnlineCollaborate.model.Blog;

@Repository("blogDao")

@Transactional

public class BlogDaoImpl implements IBlogDao {

@Autowired

SessionFactory sessionFactory;

@Override

public List<Blog> getAllBlogs() {

return sessionFactory.getCurrentSession().createQuery("from Blog",Blog.class).getResultList();

}

@Override

public List<Blog> getBlogsByStatus(String status) {

String q="from Blog where status='"+status+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

@Override

public List<Blog> getUsersBlogs(int blogId) {

// TODO Auto-generated method stub

return null;

}

@Override

public Blog getBlogById(int blogId) {

return sessionFactory.getCurrentSession().get(Blog.class,Integer.valueOf(blogId));

}

@Override

public boolean addBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().save(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().update(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteBlog(Blog blog) {

try

{

sessionFactory.getCurrentSession().delete(blog);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

}

* create BlogService:

4.IBlogService:

package com.coll.OnlineCollaborate.service;

import java.util.List;

import com.coll.OnlineCollaborate.model.Blog;

public interface IBlogService {

List<Blog> getAllBlogs();

List<Blog> getBlogsByStatus(String status);

List<Blog> getUsersBlogs(int blogId);

Blog getBlogById(int blogId);

boolean addBlog(Blog blog);

boolean updateBlog(Blog blog);

boolean deleteBlog(Blog blog);

}

* Create BlogServiceImpl

5.BlogServiceImpl

package com.coll.OnlineCollaborate.serviceImpl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.Dao.IBlogDao;

//import com.coll.OnlineCollaborate.dao.IBlogDao;

import com.coll.OnlineCollaborate.model.Blog;

import com.coll.OnlineCollaborate.service.IBlogService;

@Service

@Transactional

public class BlogServiceImpl implements IBlogService {

@Autowired

IBlogDao blogDao;

@Override

public List<Blog> getAllBlogs() {

return blogDao.getAllBlogs();

}

@Override

public List<Blog> getBlogsByStatus(String status) {

return blogDao.getBlogsByStatus(status);

}

@Override

public List<Blog> getUsersBlogs(int blogId) {

return blogDao.getUsersBlogs(blogId);

}

@Override

public Blog getBlogById(int blogId) {

return blogDao.getBlogById(blogId);

}

@Override

public boolean addBlog(Blog blog) {

return blogDao.addBlog(blog);

}

@Override

public boolean updateBlog(Blog blog) {

return blogDao.updateBlog(blog);

}

@Override

public boolean deleteBlog(Blog blog) {

return blogDao.deleteBlog(blog);

}

}

* Create BlogController

6.BlogController class

package com.coll.OnlineCollaborate.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.coll.OnlineCollaborate.model.Blog;

import com.coll.OnlineCollaborate.service.IBlogService;

@RestController

@CrossOrigin(origins="http://localhost:4200")

@RequestMapping(value="/api")

public class BlogController {

@Autowired

IBlogService blogService;

@PostMapping("save-blog")

public boolean saveBlog(@RequestBody Blog blog)

{

return blogService.addBlog(blog);

}

@GetMapping("blog-list")

public List<Blog> allBlog()

{

return blogService.getAllBlogs();

}

@DeleteMapping("delete-blog/{blog}")

public boolean deleteBlog(@PathVariable("blog") Blog blog)

{

return blogService.deleteBlog(blog);

}

@GetMapping("blog/{blog\_id}")

public Blog blogById(@PathVariable("blog\_id") int blog\_id)

{

return blogService.getBlogById(blog\_id);

}

@PostMapping("update-blog/{blog}")

public boolean updateBlog(@PathVariable("blog") Blog blog)

{

return blogService.updateBlog(blog);

}

}

* Create BlogComments model class

1.model class:

package com.coll.OnlineCollaborate.model;

import java.io.Serializable;

import java.time.LocalDate;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import org.springframework.stereotype.Component;

import com.fasterxml.jackson.annotation.JsonBackReference;

@Component

@Entity

public class BlogComments extends DomainResponse implements Serializable{

private static final long serialVersionUID =1L;

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

int blogCommentId;

int userId;

String username;

String userProfileId;

String title;

int noOfLikes;

String blogcomment;

LocalDate currentDate;

@ManyToOne

@JoinColumn(name="blogId")

@JsonBackReference

Blog blog;

public int getBlogCommentId() {

return blogCommentId;

}

public void setBlogCommentId(int blogCommentId) {

this.blogCommentId = blogCommentId;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getUserProfileId() {

return userProfileId;

}

public void setUserProfileId(String userProfileId) {

this.userProfileId = userProfileId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public int getNoOfLikes() {

return noOfLikes;

}

public void setNoOfLikes(int noOfLikes) {

this.noOfLikes = noOfLikes;

}

public String getBlogcomment() {

return blogcomment;

}

public void setBlogcomment(String blogcomment) {

this.blogcomment = blogcomment;

}

public LocalDate getCurrentDate() {

return currentDate;

}

public void setCurrentDate(LocalDate currentDate) {

this.currentDate = currentDate;

}

public Blog getBlog() {

return blog;

}

public void setBlog(Blog blog) {

this.blog = blog;

}

public static long getSerialversionuid() {

return serialVersionUID;

}

}

* Create IBlogcommentsDao Interface

2.IBlogComments:

package com.coll.OnlineCollaborate.Dao;

import java.util.List;

import com.coll.OnlineCollaborate.model.Blog;

import com.coll.OnlineCollaborate.model.BlogComments;

public interface IBlogCommentsDao {

List<BlogComments> getAllBlogsComments();

List<BlogComments> getCommentsbyBlog(int blog);

List<BlogComments> getBlogsCommentsByStatus(String status);

BlogComments getBlogCommentsById(int blogcommentsId);

boolean addBlogComments(BlogComments blogComments);

boolean updateBlogComments(BlogComments blogComments);

boolean deleteBlogComments(BlogComments blogComments);

}

* Create BlogComments DaoImpl

3.BlogCommentsDaoImpl

package com.coll.OnlineCollaborate.DaoImpl;

import java.util.List;

import org.hibernate.SessionFactory;

import org.hibernate.query.Query;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.Dao.IBlogCommentsDao;

import com.coll.OnlineCollaborate.model.Blog;

import com.coll.OnlineCollaborate.model.BlogComments;

@Repository("blogCommentsDao")

@Transactional

public class BlogCommentsDaoImpl implements IBlogCommentsDao {

@Autowired

SessionFactory sessionFactory;

@Override

public List<BlogComments> getAllBlogsComments() {

return sessionFactory.getCurrentSession().createQuery("from BlogComments",BlogComments.class).getResultList();

}

@Override

public List<BlogComments> getBlogsCommentsByStatus(String status) {

String q="from BlogComments where status='"+status+"'";

Query query=sessionFactory.getCurrentSession().createQuery(q);

return query.getResultList();

}

public List<BlogComments> getCommentsbyBlog(int blog){

return null;

}

@Override

public BlogComments getBlogCommentsById(int blogcommentsId) {

return sessionFactory.getCurrentSession().get(BlogComments.class,Integer.valueOf(blogcommentsId));

}

@Override

public boolean addBlogComments(BlogComments blogcomments) {

try

{

sessionFactory.getCurrentSession().save(blogcomments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateBlogComments(BlogComments blogcomments) {

try

{

sessionFactory.getCurrentSession().update(blogcomments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

@Override

public boolean deleteBlogComments(BlogComments blogcomments) {

try

{

sessionFactory.getCurrentSession().delete(blogcomments);

return true;

}

catch(Exception ex)

{

ex.printStackTrace();

return false;

}

}

}

* Create IBlogCommentsService Interface

4.IBlogCommentsService:

package com.coll.OnlineCollaborate.service;

import java.util.List;

import com.coll.OnlineCollaborate.model.BlogComments;

public interface IBlogCommentsService {

List<BlogComments> getAllBlogsComments();

List<BlogComments> getBlogsCommentsByStatus(String status);

List<BlogComments> getUsersBlogsComments(int blogcommentsId);

List<BlogComments> mainList();

BlogComments getBlogCommentsById(int blogcommentsId);

boolean addBlogComments(BlogComments blogcomments);

boolean updateBlogComments(BlogComments blogcomments);

boolean deleteBlogComments(BlogComments blogcomments);

}

* Create BlogCommentsServiceDaoImpl class

5.BlogCommentsServiceDaoImpl

package com.coll.OnlineCollaborate.serviceImpl;

import java.util.List;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.coll.OnlineCollaborate.model.BlogComments;

import com.coll.OnlineCollaborate.service.IBlogCommentsService;

@Service

@Transactional

public class BlogCommentsServiceImpl implements IBlogCommentsService{

@Override

public List<BlogComments> getAllBlogsComments() {

// TODO Auto-generated method stub

return null;

}

@Override

public List<BlogComments> getBlogsCommentsByStatus(String status) {

// TODO Auto-generated method stub

return null;

}

@Override

public List<BlogComments> getUsersBlogsComments(int blogcommentsId) {

// TODO Auto-generated method stub

return null;

}

@Override

public List<BlogComments> mainList() {

// TODO Auto-generated method stub

return null;

}

@Override

public BlogComments getBlogCommentsById(int blogcommentsId) {

// TODO Auto-generated method stub

return null;

}

@Override

public boolean addBlogComments(BlogComments blogcomments) {

// TODO Auto-generated method stub

return false;

}

@Override

public boolean updateBlogComments(BlogComments blogcomments) {

// TODO Auto-generated method stub

return false;

}

@Override

public boolean deleteBlogComments(BlogComments blogcomments) {

// TODO Auto-generated method stub

return false;

}

}

* Create BlogCommentsController class

BlogCommentsController

package com.coll.OnlineCollaborate.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.coll.OnlineCollaborate.model.BlogComments;

import com.coll.OnlineCollaborate.service.IBlogCommentsService;

@RestController

@CrossOrigin(origins="http://localhost:4200")

@RequestMapping(value="/api")

public class BlogCommentsController {

@Autowired

IBlogCommentsService blogcommentsService;

@PostMapping("save-blogcomments")

public boolean saveBlogComments(@RequestBody BlogComments blogcomments)

{

return blogcommentsService.addBlogComments(blogcomments);

}

@GetMapping("blogcomments-list")

public List<BlogComments> allBlogComments()

{

return blogcommentsService.getAllBlogsComments();

}

@DeleteMapping("delete-blogcomments/{blogcomments}")

public boolean deleteBlogComments(@PathVariable("blogcomments") BlogComments blogcomments)

{

return blogcommentsService.deleteBlogComments(blogcomments);

}

@GetMapping("blogcomments/{blogcomments\_id}")

public BlogComments blogcommentsById(@PathVariable("blogcomments\_id") int blogcomments\_id)

{

return blogcommentsService.getBlogCommentsById(blogcomments\_id);

}

@PostMapping("update-blogcomments/{blogcomments}")

public boolean updateBlogcomments(@PathVariable("blogcomments") BlogComments blogcomments)

{

return blogcommentsService.updateBlogComments(blogcomments);

}

}

* + Right click OnlineCollaborateApplication
  + Click Run as
  + Click JavaApplication