BlogBean.java | BlogService.... | BlogServicel... | BlogDAO.java | MaxAllowedE... | NoBlogPostsF... | BlogDAOWrapp...

```java
package com.accenture.lkm.bean;

import org.hibernate.validator.constraints.NotEmpty;

public class BlogBean {


    private int blogId;

    @NotEmpty
    private String email;
    @NotEmpty
    private String title;

    private String themes;
    @NotEmpty
    private String content;




    public int getBlogId() {
        return blogId;
    }

    public void setBlogId(int blogId) {
        this.blogId = blogId;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getTitle() {
        return title;
    }

```

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/entity/BlogEntity.java - Eclipse

172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Project Explorer

- Java Resources
  - src
    - com.accenture.lkm.bean
      - BlogBean.java
        - BlogBean
    - com.accenture.lkm.controller
    - com.accenture.lkm.dao
      - BlogDAO.java
      - BlogDAOWrapper.java
    - com.accenture.lkm.entity
      - BlogEntity.java
        - BlogEntity
    - com.accenture.lkm.exception
    - com.accenture.lkm.resources
    - com.accenture.lkm.service
      - BlogService.java
        - BlogService
      - BlogServiceImpl.java
    - com.accenture.lkm.test
      - BlogDAOWrapperTest.java
      - BlogServiceTest.java
  - Libraries
- JavaScript Resources
- build
- WebContent
  - META-INF
  - WEB-INF
    - jspViews
      - ExceptionHandlerPage.jsp
      - Home.jsp
      - ListPosts.jsp
      - NewBlog.jsp
      - Report.jsp
    - lib

BlogBean.java | BlogEntity.java | BlogService.... | BlogServiceI... | BlogDAO.java | NoBlogPostsF... | BlogDAOWrapp...

```
1  package com.accenture.lkm.entity;
2  import javax.persistence.Entity;
7
8  @Entity
9  @Table(name="Blog")
10 public class BlogEntity {
11
12     @Id
13     @GeneratedValue(strategy=GenerationType.AUTO)
14     private int blogId;
15
16     private String email;
17
18     private String title;
19
20     private String themes;
21
22     private String content;
23
24
25
26     public int getBlogId() {
27         return blogId;
28     }
29
30     public void setBlogId(int blogId) {
31         this.blogId = blogId;
32     }
33
34     public String getEmail() {
35         return email;
36     }
37
38     public void setEmail(String email) {
39         this.email = email;
40     }
41
42     public String getTitle() {
43         return title;
44     }
45
```

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/dao/Blo

172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Project Explorer

- Java Resources
  - src
    - com.accenture.lkm.bean
      - BlogBean.java
        - BlogBean
    - com.accenture.lkm.controller
    - com.accenture.lkm.dao
      - BlogDAO.java
      - BlogDAOWrapper.java
    - com.accenture.lkm.entity
      - BlogEntity.java
        - BlogEntity
    - com.accenture.lkm.exception
    - com.accenture.lkm.resources
    - com.accenture.lkm.service
      - BlogService.java
        - BlogService
      - BlogServiceImpl.java
    - com.accenture.lkm.test
      - BlogDAOWrapperTest.java
      - BlogServiceTest.java
  - Libraries
- JavaScript Resources
- build
- WebContent
  - META-INF
  - WEB-INF
    - jspViews
      - ExceptionHandlerPage.jsp
      - Home.jsp
      - ListPosts.jsp
      - NewBlog.jsp
      - Report.jsp
    - lib

BlogBean.java | BlogEntity.java | BlogService.... | BlogServiceI... | BlogDAO.java | NoBlogPostsF... | BlogDAOWrapp...

```
1  package com.accenture.lkm.dao;
2
3  import java.util.List;
10
11 public interface BlogDAO extends CrudRepository<BlogEntity,Integer> {
12
13     /**
14        To-Do Item 1.1: This method should retrieve a list of blog posts associated with the provided email add
15        TODO:
16         - Write a JPQL query and annotate the method with @Query to select blog posts from the BlogEntity tab
17         - Bind the email parameter to the query.
18      */
19      @Query(value="select b from BlogEntity b where b.email=?1 order by b.title")
20      public List<BlogEntity> findPostsByEmail(String email);
21
22     /**
23        To-Do Item 1.2: This method should retrieve the count of blog posts associated with the provided email
24        TODO:
25         - Write a JPQL query and annotate the method with @Query to count the number of blog posts from the Bl
26         - Bind the email parameter to the query.
27      */
28     @Query(value="select count(b.email) from BlogEntity b where b.email = ?1")
29     public long countPostsByEmail(String email);
30
31 }
32
```

Writable | Smart Insert | 15 : 15

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/dao/Blo... - Eclipse 172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- Java Resources
  - src
    - com.accenture.lkm.bean
      - BlogBean.java
        - BlogBean
    - com.accenture.lkm.controller
    - com.accenture.lkm.dao
      - BlogDAO.java
      - BlogDAOWrapper.java
    - com.accenture.lkm.entity
      - BlogEntity.java
        - BlogEntity
    - com.accenture.lkm.exception
    - com.accenture.lkm.resources
    - com.accenture.lkm.service
      - BlogService.java
        - BlogService
      - BlogServiceImpl.java
    - com.accenture.lkm.test
      - BlogDAOWrapperTest.java
      - BlogServiceTest.java
  - Libraries
- JavaScript Resources
- build
- WebContent
  - META-INF
  - WEB-INF
    - jspViews
      - ExceptionHandlerPage.jsp
      - Home.jsp
      - ListPosts.jsp
      - NewBlog.jsp
      - Report.jsp
    - lib

BlogBean.java | BlogEntity.java | BlogService.... | BlogServiceI... | BlogDAO.java | NoBlogPostsF... | BlogDAOWrapp...

```
1  package com.accenture.lkm.dao;
2
3
4⊕ import java.util.ArrayList;
16
17 @Repository
18 @Transactional("txManager")
19 public class BlogDAOWrapper {
20
21⊖ @Autowired
22 private BlogDAO  blogDAO;
23
24⊖     /**
25        To-Do Item 1.3: This method should add a new blog entry to the database.
26
27        TODO: Implement the addblog method:
28         -It should convert the BlogBean object to a BlogEntity.
29         -It should save the BlogEntity to the database.
30         -It should return the blogId(unique identifier) of the newly added blog entry.
31     */
32⊖    public int addblog(BlogBean blog) throws Exception {
33         BlogEntity blogEntity = convertBeanToEntity(blog);
34         BlogEntity blogEntity2 = blogDAO.save(blogEntity);
35         return blogEntity2.getBlogId();
36     }
37
38⊖    /**
39        To-Do Item 1.4: This method should retrieve a list of blog posts for the entered email address.Po
40
41        TODO: Implement the findPostsByEmail method:
42          - It should use the blogDAO reference to call its findPostsByEmail method, retrieving blog post
43          - It should convert each retrieved BlogEntity object to a BlogBean object.
44          - It should add the converted BlogBean objects to a list.
45          - It should return a list of BlogBean objects representing the blog posts.
46      */
47
48⊖   public List<BlogBean> findPostsByEmail(String email) throws Exception {
49        List<BlogEntity> blogEntities = blogDAO.findPostsByEmail(email);
50        List<BlogBean> beans = new ArrayList<>();
51        for(BlogEntity entity : blogEntities) {
52            beans.add(convertEntityToBean(entity));
```

ndson - HON_BlogSystem_Participant/src/com/accenture/lkm/dao/Blo... - Eclipse 172.31.72.9

Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | BlogDAOWrapper....

```
26
27      TODO: Implement the addblog method:
28       -It should convert the BlogBean object to a BlogEntity.
29       -It should save the BlogEntity to the database.
30       -It should return the blogId(unique identifier) of the newly added blog entry.
31    */
32    public int addblog(BlogBean blog) throws Exception {
33        BlogEntity blogEntity = convertBeanToEntity(blog);
34        BlogEntity blogEntity2 = blogDAO.save(blogEntity);
35        return blogEntity2.getBlogId();
36    }
37
38    /**
39       To-Do Item 1.4: This method should retrieve a list of blog posts for the entered email address.Posts should be retrieved in alphabetical or
40
41       TODO: Implement the findPostsByEmail method:
42         - It should use the blogDAO reference to call its findPostsByEmail method, retrieving blog posts based on the provided email address.
43         - It should convert each retrieved BlogEntity object to a BlogBean object.
44         - It should add the converted BlogBean objects to a list.
45         - It should return a list of BlogBean objects representing the blog posts.
46     */
47
48    public List<BlogBean> findPostsByEmail(String email) throws Exception {
49        List<BlogEntity> blogEntities = blogDAO.findPostsByEmail(email);
50        List<BlogBean> beans = new ArrayList<>();
51        for(BlogEntity entity : blogEntities) {
52            beans.add(convertEntityToBean(entity));
53        }
54        return beans;
55    }
56
57    /**
58       To-Do Item 1.5: This method retrieve the count of blog posts associated with the provided email address.
59
60       TODO: Implement the countPostsByEmail method:
61         - It should use the blogDAO reference to call its countPostsByEmail method to retrieve the count of blog posts for the provided email.
62         - It should return the count of blog posts as a long value.
63
64    */
65    public long countPostsByEmail(String email) {
66
```

Writable | Smart Insert | 1 : 1

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/dao/Blo... - Eclipse 172.31.72.9

Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | BlogDAOWrapper....

```
        return beans;
    }

    /**
      To-Do Item 1.5: This method retrieve the count of blog posts associated with the provided email address.

      TODO: Implement the countPostsByEmail method:
        - It should use the blogDAO reference to call its countPostsByEmail method to retrieve the count of blog posts for the provided email.
        - It should return the count of blog posts as a long value.

    */
    public long countPostsByEmail(String email) {

        return blogDAO.countPostsByEmail(email);
    }


    /**
      To-Do Item 1.6: This method should return map of below mentioned themes.

      TODO:
       -Populate  the map with below key-value entries:
        1.Key: Daily Journal  Value: Daily Journal
        2.Key: Cheer Up       Value: Cheer Up
        3.Key: Goal           Value: Goal

     */
     public Map<String, String> populateTheme() {
         Map<String,String> map = new HashMap<>();
         map.put("Daily Journal","Daily Journal");
         map.put("Cheer Up","Cheer Up");
         map.put("Goal", "Goal");
         return map;
     }

    public static BlogEntity convertBeanToEntity(BlogBean bean){
         BlogEntity blogEntityBean = new BlogEntity();
         BeanUtils.copyProperties(bean,blogEntityBean);
         return blogEntityBean;
     }
```

Writable | Smart Insert | 1 : 1

Project Explorer

- Java Resources
  - src
    - com.accenture.lkm.bean
      - BlogBean.java
        - BlogBean
    - com.accenture.lkm.controller
    - com.accenture.lkm.dao
      - BlogDAO.java
      - BlogDAOWrapper.java
    - com.accenture.lkm.entity
      - BlogEntity.java
        - BlogEntity
    - com.accenture.lkm.exception
    - com.accenture.lkm.resources
    - com.accenture.lkm.service
      - BlogService.java
        - BlogService
      - BlogServiceImpl.java
    - com.accenture.lkm.test
      - BlogDAOWrapperTest.java
      - BlogServiceTest.java
  - Libraries
- JavaScript Resources
- build
- WebContent
  - META-INF
  - WEB-INF
    - jspViews
      - ExceptionHandlerPage.jsp
      - Home.jsp
      - ListPosts.jsp
      - NewBlog.jsp
      - Report.jsp
    - lib

BlogBean.java | BlogEntity.java | BlogService.... | BlogServiceI... | BlogDAO.java | NoBlogPostsF... | BlogDAOWrapp...

```
1  package com.accenture.lkm.service;
2
3⊕ import java.util.List;
7
8  public interface BlogService {
9      public int addBlog(BlogBean blog)throws Exception;
10     public List<BlogBean> getAllPosts(String email)throws Exception;
11     public Map<String, String> populateTheme();
12 }
13
```

dson - HON_BlogSystem_Participant/src/com/accenture/lkm/service/BlogServiceImpl.java - Eclipse

172.31.72.9

dit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | BlogDAOWrapper....

```
@Service
public class BlogServiceImpl implements BlogService {

    @Autowired
    private BlogDAOWrapper dao;

    static final int LIMIT=5;

    /**
       To-Do Item 1.7: This method should add a new blog entry to the database.
       TODO: Implement the addblog method:
         -It should use the BlogDAOWrapper reference to call countPostsByEmail method.
         -Check if the number of posts associated with the provided email address is less than LIMIT.
         -If yes, call the addblog method of the blogDAOWrapper to add the blog post.Store the returned blogId.
         -If no, throw a MaxAllowedEntriesReachedException.
         -Create custom MaxAllowedEntriesReachedException class in package com.exception with an error message "User cannot enter more than 5 Bl
         -Finally,method should return newly generated BlogId.
     */

    public int addBlog(BlogBean blog) throws Exception {
        long count = dao.countPostsByEmail(blog.getEmail());
        if(count>LIMIT) {
            throw new MaxAllowedEntriesReachedException("User cannot enter more than 5 Blog entries");
        }
        return dao.addblog(blog);
    }

    public List<BlogBean> getAllPosts(String email) throws Exception {
        List<BlogBean> blogBeans = dao.findPostsByEmail(email);
        return blogBeans;
    }

    @Override
    public Map<String, String> populateTheme() {
        return dao.populateTheme();
    }


}
```

Writable | Smart Insert | 1 : 1

5:24 PM
10/9/2024

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/test/Blo... 172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | BlogDAOW

```
1  package com.accenture.lkm.test;
2
3⊕ import java.util.List;
15
16 @RunWith(SpringJUnit4ClassRunner.class)
17 @ContextConfiguration(locations = { "file:WebContent/WEB-INF/root-config.xml" })
18 @Transactional
19 public class BlogDAOWrapperTest {
20
21⊖     @Autowired
22      private BlogDAOWrapper blogDAOWrapper;
23
24⊖     /**
25       * To-Do Item 1.14 : This method tests the addBlog functionality in
26       * BlogDAOWrapper by adding a dummy blog entry to the database.
27       *
28       * TODO: Implement the testAddBlog method: - It should create a new instance of
29       * BlogBean and set its properties (title, content, email, themes) with dummy
30       * data. - It should invoke the addblog method of blogDAOWrapper, passing the
31       * BlogBean object. - It should verify that a new blog entry has been
32       * successfully inserted into the database table.
33       */
34
35⊖     @Test
36      public void testAddBlog() throws Exception {
37          BlogBean bean = new BlogBean();
38          bean.setTitle("Education");
39          bean.setContent("Happy Learning");
40          bean.setEmail("Alexa@gmail.com");
41          bean.setThemes("DailyJournal");
42          int id = blogDAOWrapper.addblog(bean);
43          Assertions.assertNotNull(id);
44      }
45
46⊖     /**
47       * To-Do Item 1.15 : This method tests the findPostsByEmail function of
48       * BlogDAOWrapper to retrieve a list of blog posts for the entered email
49       * address..
50       *
51       * TODO: Implement the findPostsByEmail method: - It should specify the email
52       * address for which to retrieve blog posts. - It should invoke the
```

Writable | Smart Insert | 1 : 1

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/test/Blo

172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | Blo

```
35    @Test
36    public void testAddBlog() throws Exception {
37        BlogBean bean = new BlogBean();
38        bean.setTitle("Education");
39        bean.setContent("Happy Learning");
40        bean.setEmail("Alexa@gmail.com");
41        bean.setThemes("DailyJournal");
42        int id = blogDAOWrapper.addblog(bean);
43        Assertions.assertNotNull(id);
44    }
45
46    /**
47     * To-Do Item 1.15 : This method tests the findPostsByEmail function of
48     * BlogDAOWrapper to retrieve a list of blog posts for the entered email
49     * address..
50     *
51     * TODO: Implement the findPostsByEmail method: - It should specify the email
52     * address for which to retrieve blog posts. - It should invoke the
53     * findPostsByEmail method of blogDAOWrapper, passing the email address. - It
54     * should verify that the result (list of BlogBean objects) is not null,
55     * indicating successful retrieval of blog posts.
56     */
57    @Test
58    public void testFindPostsByEmail() throws Exception {
59        List<BlogBean> beanList = blogDAOWrapper.findPostsByEmail("Alexa@gmail.com");
60        Assertions.assertNotNull(beanList);
61    }
62
63    /**
64     To-Do Item 1.16 : This method tests the findPostsByEmail functionality in BlogDAOWrapper, verifying the count of blog posts asso
65
66     TODO: Implement the countPostsByEmail method:
67       - It should Specify the email address for which to count the number of blog posts.
68       - It should call the countPostsByEmail method of blogDAOWrapper, passing the email address.
69       - It should verify that the result (count of blog posts) is greater than or equal to 0, ensuring a non-negative count.
70  */
71  @Test
72  public void testCountPostsByEmail() {
73          int actual = 1;
74          long expected = blogDAOWrapper.countPostsByEmail("Alexa@gmail.com");
75          Assertions.assertEquals(expected, actual);
```

Writable | Smart Insert | 1 : 1

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/test/Blo

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.ja... | BlogServiceIm... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapper... | MaxAllowedEntri... | NoBlogPostsFor... | BlogDA

```
        bean.setTitle("Education");
        bean.setContent("Happy Learning");
        bean.setEmail("Alexa@gmail.com");
        bean.setThemes("DailyJournal");
        int id = blogDAOWrapper.addblog(bean);
        Assertions.assertNotNull(id);
    }

    /**
     * To-Do Item 1.15 : This method tests the findPostsByEmail function of
     * BlogDAOWrapper to retrieve a list of blog posts for the entered email
     * address..
     *
     * TODO: Implement the findPostsByEmail method: - It should specify the email
     * address for which to retrieve blog posts. - It should invoke the
     * findPostsByEmail method of blogDAOWrapper, passing the email address. - It
     * should verify that the result (list of BlogBean objects) is not null,
     * indicating successful retrieval of blog posts.
     */
    @Test
    public void testFindPostsByEmail() throws Exception {
        List<BlogBean> beanList = blogDAOWrapper.findPostsByEmail("Alexa@gmail.com");
        Assertions.assertNotNull(beanList);
    }

    /**
     To-Do Item 1.16 : This method tests the findPostsByEmail functionality in BlogDAOWrapper, verifying the count of blog posts associ

     TODO: Implement the countPostsByEmail method:
       - It should Specify the email address for which to count the number of blog posts.
       - It should call the countPostsByEmail method of blogDAOWrapper, passing the email address.
       - It should verify that the result (count of blog posts) is greater than or equal to 0, ensuring a non-negative count.
  */
  @Test
  public void testCountPostsByEmail() {
          int actual = 1;
          long expected = blogDAOWrapper.countPostsByEmail("Alexa@gmail.com");
          Assertions.assertEquals(expected, actual);
  }
}
```

Writable | Smart Insert | 1 : 1

Handson - HON_BlogSystem_Participant/src/com/accenture/lkm/test/BlogServiceTest.java - Eclipse

172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.... | BlogServicel... | BlogDAO.java | BlogDAOWrapp... | MaxAllowedE... | NoBlogPostsF... | BlogDAOWrapp... | BlogService...

```
1  package com.accenture.lkm.test;
2
3  import java.util.List;
15
16 @RunWith(SpringJUnit4ClassRunner.class)
17 @ContextConfiguration(locations = {"file:WebContent/WEB-INF/root-config.xml"})
18 public class BlogServiceTest {
19
20     @Autowired
21     private BlogServiceImpl blogService;
22
23     /**
24        To-Do Item 1.17 : This method tests the addBlog functionality in BlogServiceImpl ensuring that a blog entry is added only
25        when the number of existing posts for the associated email is less than the defined limit (5 entries in our case).
26
27        TODO: Implement the testAddBlogWithLessThanFivePosts_Positive method:
28          - It should create a new instance of BlogBean and set its properties (title, content, email, themes) with dummy data.
29          - It should invoke the addblog method of blogDAOWrapper, passing the BlogBean object.
30          - It should verify that a new blog entry has been successfully inserted into the database table.
31      */
32     @Test
33     public void testAddBlogWithLessThanFivePosts_Positive() throws Exception {
34         BlogBean bean = new BlogBean();
35         bean.setTitle("Digital Citizenship");
36         bean.setContent("Digital Citizens");
37         bean.setEmail("Sam@gmail.com");
38         bean.setThemes("DailyJournal");
39         int id = blogService.addBlog(bean);
40         Assertions.assertNotNull(id);
41     }
42     /**
43        To-Do Item 1.18: This method test the addBlog functionality in BlogServiceImpl to ensure that a blog entry is not added
44        when the number of existing posts for the associated email reaches the defined limit (5 entries in our case).
45
46        TODO: Implement the testAddBlogWithLessThanFivePosts_Negative method:
47         - It should add blog entries for the specified email until the maximum allowed posts (5 in our case) is reached.
48         - It should attempt to add an additional blog entry for the same email.
49         - It ensure that adding the additional blog entry throws a MaxAllowedEntriesReachedException with the message "User cannot enter more t
50      */
51     @Test
52     public void testAddBlogWithLessThanFivePosts_Negative() throws Exception {
```

Writable | Smart Insert | 1 : 1

andson - HON_BlogSystem_Participant/src/com/accenture/lkm/test/Blo

172.31.72.9

Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

BlogBean.java | BlogEntity.java | BlogService.... | BlogServiceI... | BlogDAO.java | BlogDAOWrapp... | MaxAllowedE... | NoBlogPostsF... | BlogDAOWrapp... | BlogService...

```
        - It should verify that a new blog entry has been successfully inserted into the database table.
    */
    @Test
    public void testAddBlogWithLessThanFivePosts_Positive() throws Exception {
        BlogBean bean = new BlogBean();
        bean.setTitle("Digital Citizenship");
        bean.setContent("Digital Citizens");
        bean.setEmail("Sam@gmail.com");
        bean.setThemes("DailyJournal");
        int id = blogService.addBlog(bean);
        Assertions.assertNotNull(id);
    }
    /**
       To-Do Item 1.18: This method test the addBlog functionality in BlogServiceImpl to ensure that a blog entry is not added
       when the number of existing posts for the associated email reaches the defined limit (5 entries in our case).

       TODO: Implement the testAddBlogWithLessThanFivePosts_Negative method:
        - It should add blog entries for the specified email until the maximum allowed posts (5 in our case) is reached.
        - It should attempt to add an additional blog entry for the same email.
        - It ensure that adding the additional blog entry throws a MaxAllowedEntriesReachedException with the message "User cannot enter more than
    */
    @Test
    public void testAddBlogWithLessThanFivePosts_Negative() throws Exception {

    }

    /**
       To-Do Item 1.19: This method test getAllPosts functionality in BlogServiceImpl for retrieving all post associated with a specified email.

       TODO: Implement the testGetAllPosts method:
        - It should call the getAllPosts method of blogService with a specified email.
        - It should assert that the result obtained is not null.
    */
    @Test
    public void testGetAllPosts() throws Exception {
        List<BlogBean> beans = blogService.getAllPosts("Alexa@gmail.com");
        Assertions.assertNotNull(beans);
    }

}
```

Writable | Smart Insert | 1 : 1

Handson - HON_BlogSystem_Participant/WebContent/WEB-INF/jspViews/NewBlog.jsp - Eclipse

172.31.72.9

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.... | BlogServicel.... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapp... | NoBlogPostsF... | BlogDAOWrapp... | BlogSer

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3     <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
4     <%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
5 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
9 <title>Insert title here</title>
10 <style>
11 .error {
12 color: #ff0000;
13 font-style: italic;
14 }
15 </style>
16 </head>
17 <body>
18 <center>
19 <h2>Create New Post</h2>
20 <form:form action="addBlog.html" modelAttribute="blog" >
21
22 <!--
23 To-Do Item 1.8: Complete the NewBlog.jsp page using Spring MVC Form Tag library:
24 TODO: --Create text box for entering email.
25       --Create text box for entering title.
26       --Create a dropdown menu for selecting a theme within the table.
27       --Ensure that the theme dropdown is dynamically populated and mapped to the controller for data retrieval.
28       --Create text box for entering content.
29       --Validations: email,title and content are mandatory fields.
30       --Display appropriate error messages if validation fails.
31       --Create a submit button within the table for submitting the form.
32 -->
33
34     <table  border="1">
35
36         <tr>
37             <td>Email</td><td>
38             <form:input path="email"/>
39             <form:errors path="email" cssClass="error"></form:errors></td>
40
41         </tr>
```

Handson - HON_BlogSystem_Participant/WebContent/WEB-INF/jspViews/NewBlog.jsp - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

BlogBean.java | BlogEntity.java | BlogService.... | BlogServicel.... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapp... | NoBlogPostsF... | BlogDAOWrapp... | BlogSer

```
34    <table  border="1">
35
36        <tr>
37            <td>Email</td><td>
38            <form:input path="email"/>
39            <form:errors path="email" cssClass="error"></form:errors></td>
40
41        </tr>
42
43        <tr>
44            <td>Title</td>
45            <td>
46            <form:input path="title"/>
47            <form:errors path="title" cssClass="error"></form:errors></td>
48        </tr>
49
50        <tr>
51            <td>Theme</td><td>
52            <form:select path="themes">
53
54            <form:option value="" label="---select--"/>
55            <form:options items="${list}" />
56            </form:select></td>
57        </tr>
58
59        <tr>
60            <td>Content</td><td>
61            <form:input path="content"/>
62            <form:errors path="content" cssClass="error">></form:errors></td>
63        </tr>
64
65        <tr>
66            <td colspan="2" align="center">
67            <!-- Add submit button here -->
68            <input type="submit" value="submit">
69
70        </tr>
71        <tr>
72            <td colspan="2" align="center"><B>${MESSAGE}</B></td>
73        </tr>
74
```

Handson - HON_...

File Edit Source Refactor Navigate Search Project Run Window

BlogBean.java | BlogEntity.java | BlogService.... | BlogServicel.... | BlogDAO.java | NewBlog.jsp | BlogDAOWrapp... | NoBlogPostsF... | BlogDAOWrapp...

```
42
43          <tr>
44              <td>Title</td>
45              <td>
46              <form:input path="title"/>
47              <form:errors path="title" cssClass="error"></form:errors></td>
48          </tr>
49
50          <tr>
51              <td>Theme</td><td>
52              <form:select path="themes">
53
54              <form:option value="" label="---select--"/>
55              <form:options items="${list}" />
56              </form:select></td>
57          </tr>
58
59          <tr>
60              <td>Content</td><td>
61              <form:input path="content"/>
62              <form:errors path="content" cssClass="error"></form:errors></td>
63          </tr>
64
65           <tr>
66              <td colspan="2" align="center">
67              <!-- Add submit button here -->
68              <input type="submit" value="submit">
69
70           </tr>
71           <tr>
72              <td colspan="2" align="center"><B>${MESSAGE}</B></td>
73           </tr>
74
75      </table>
76
77           <br> <a href="home.html"> Home</a> <br>
78
79  </form:form>
80  </center>
81  </body>
82  </html>
```