Application Localization and Styling

Learning Objectives

- Understand the processes of localizing and internationalizing Liferay DXP applications
- Learn the methods by which CSS and JavaScript resources are added to applications in Liferay DXP

Tasks to Accomplish

- Add Localization Resources
- Add CSS Resources
- Add JavaScript Resources (optional)

Exercise Prerequisites

- Java JDK installed to run Liferay
  - Download here: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
  - Instructions on installation here: https://www.java.com/en/download/help/download_options.xml
- Preferred development tools (e.g. Blade CLI, Gradle, IntelliJ IDEA with Liferay plugin, etc.) installed with the "Gradebook Workspace" already created
  - This was done in the first training module
- Exercise Prereqs added to workspace or previous training modules completed

Table of Contents

Update Application with Localizable Elements

Exercise Goals

- Update the service.xml for the Gradebook Application by making the Title field localizable
- Modify the Service module for a localizable Title field
- Modify the API module for a localizable Title field
- Modify the Web module for a localizable Title field

Update the service.xml

1. **Start** your IDE with the Gradebook-workspace if it's not already started.
    - Make sure to start the server as well.
2. **Open** the `service.xml` file.
3. **Remove** the `title` column.
4. **Create** a new section below the `<!-- Audit fields -->` with the following code:

```
<!-- Localization Fields -->


<column name="title" type="String" localized="true"></column>
```

Build the Service

1. **Execute** the `buildService` task in the `gradebook-workspace/modules/gradebook/build` directory.

Update Assignment LocalServiceImpl

1. **Open** *AssignmentLocalServiceImpl* in the gradebook-service module.
2. **Find** the `addAssignment` class.
3. **Replace** `String title` with `Map<Locale, String> titleMap`.
4. **Replace** all instances of the `title` variable with `titleMap`.
5. **Find** the `updateAssignment` overload.
6. **Replace** `String title` with `Map<Locale, String> titleMap`.
7. **Replace** all instances of the `title` variable with `titleMap`.
    - The completed class should look like this:

```
/**
 * Copyright (c) 2000-present Liferay, Inc. All rights reserved.
 *
 * This library is free software; you can redistribute it and/or modify it under
 * the terms of the GNU Lesser General Public License as published by the Free
 * Software Foundation; either version 2.1 of the License, or (at your option)
 * any later version.
 *
 * This library is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
 * FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
 * details.
 */
package com.liferay.training.gradebook.service.impl;
import com.liferay.portal.aop.AopService;
import com.liferay.portal.kernel.dao.orm.Disjunction;
```

```
import com.liferay.portal.kernel.dao.orm.DynamicQuery;
import com.liferay.portal.kernel.dao.orm.RestrictionsFactoryUtil;
import com.liferay.portal.kernel.exception.PortalException;
import com.liferay.portal.kernel.model.Group;
import com.liferay.portal.kernel.model.User;
import com.liferay.portal.kernel.service.ServiceContext;
import com.liferay.portal.kernel.util.OrderByComparator;
import com.liferay.portal.kernel.util.Validator;
import com.liferay.training.gradebook.model.Assignment;
import com.liferay.training.gradebook.service.base.AssignmentLocalServiceBaseImpl;
import com.liferay.training.gradebook.validator.AssignmentValidator;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Reference;

/**
 * The implementation of the assignment local service.
 *
 * <p>
 * All custom service methods should be put in this class. Whenever methods are
 * added, rerun ServiceBuilder to copy their definitions into the
 * <code>com.liferay.training.gradebook.service.AssignmentLocalService</code>
 * interface.
 *
 * <p>
 * This is a local service. Methods of this service will not have security
 * checks based on the propagated JAAS credentials because this service can only
```

```
 * be accessed from within the same VM.
 * </p>
 *
 * @author Brian Wing Shun Chan
 * @see AssignmentLocalServiceBaseImpl
 */
@Component(
   property = "model.class.name=com.liferay.training.gradebook.model.Assignment",
   service = AopService.class
)

public class AssignmentLocalServiceImpl extends AssignmentLocalServiceBaseImpl {

   public Assignment addAssignment(long groupId, Map<Locale, String> titleMap, String des
         Date dueDate, ServiceContext serviceContext) throws PortalException {
        // Validate assignment parameters.
                        _assignmentValidator.validate(titleMap, description, dueDate);
      // Get group and user.
      Group group = groupLocalService.getGroup(groupId);
      long userId = serviceContext.getUserId();
      User user = userLocalService.getUser(userId);
      // Generate primary key for the assignment.
      long assignmentId = counterLocalService.increment(Assignment.class.getName());
      // Create assigment. This doesn't yet persist the entity.
      Assignment assignment = createAssignment(assignmentId);
      // Populate fields.
      assignment.setCompanyId(group.getCompanyId());
      assignment.setCreateDate(serviceContext.getCreateDate(new Date()));
      assignment.setDueDate(dueDate);
      assignment.setDescription(description);
      assignment.setGroupId(groupId);
      assignment.setModifiedDate(serviceContext.getModifiedDate(new Date()));
```

```
   assignment.setTitleMap(titleMap);
   assignment.setUserId(userId);
   assignment.setUserName(user.getScreenName());
   // Persist assignment to database.
   return super.addAssignment(assignment);
}

public Assignment updateAssignment(long assignmentId, Map<Locale, String> titleMap,
      String description, Date dueDate, ServiceContext serviceContext) throws PortalE
        // Validate assignment parameters.
        _assignmentValidator.validate(titleMap, description, dueDate);
   // Get the Assignment by id.
   Assignment assignment = getAssignment(assignmentId);
   // Set updated fields and modification date.
   assignment.setModifiedDate(new Date());
   assignment.setTitleMap(titleMap);
   assignment.setDueDate(dueDate);
   assignment.setDescription(description);
   assignment = super.updateAssignment(assignment);
   return assignment;
}

public List<Assignment> getAssignmentsByGroupId(long groupId) {
   return assignmentPersistence.findByGroupId(groupId);
}
public List<Assignment> getAssignmentsByGroupId(long groupId, int start, int end) {
   return assignmentPersistence.findByGroupId(groupId, start, end);
}
public List<Assignment> getAssignmentsByGroupId(long groupId, int start, int end,
      OrderByComparator<Assignment> orderByComparator) {
   return assignmentPersistence.findByGroupId(groupId, start, end, orderByComparator);
}
```

```
public List<Assignment> getAssignmentsByKeywords(
   long groupId, String keywords, int start, int end,
   OrderByComparator<Assignment> orderByComparator) {
   return assignmentLocalService.dynamicQuery(
      getKeywordSearchDynamicQuery(groupId, keywords), start, end,
      orderByComparator);
}
public long getAssignmentsCountByKeywords(long groupId, String keywords) {
   return assignmentLocalService.dynamicQueryCount(
      getKeywordSearchDynamicQuery(groupId, keywords));
}
private DynamicQuery getKeywordSearchDynamicQuery(
   long groupId, String keywords) {
   DynamicQuery dynamicQuery = dynamicQuery().add(
      RestrictionsFactoryUtil.eq("groupId", groupId));
   if (Validator.isNotNull(keywords)) {
      Disjunction disjunctionQuery =
         RestrictionsFactoryUtil.disjunction();
      disjunctionQuery.add(
         RestrictionsFactoryUtil.like("title", "%" + keywords + "%"));
      disjunctionQuery.add(
         RestrictionsFactoryUtil.like(
            "description", "%" + keywords + "%"));
      dynamicQuery.add(disjunctionQuery);
   }
   return dynamicQuery;
}

@Override
public Assignment addAssignment(Assignment assignment) {
   throw new UnsupportedOperationException("Not supported.");
}
```

```
    @Override
    public Assignment updateAssignment(Assignment assignment) {
        throw new UnsupportedOperationException("Not supported.");
    }
    @Reference
    AssignmentValidator _assignmentValidator;
}
```

Update AssignmentServiceImpl

1. **Open** `AssignmentServiceImpl.java`.
2. **Find** the `addAssignment` class.
3. **Replace** `String title` with `Map<Locale, String> titleMap`.
4. **Replace** all instances of the `title` variable with `titleMap`.
5. **Find** the `updateAssignment` overload.
6. **Replace** `String title` with `Map<Locale, String> titleMap`.
7. **Replace** all instances of the `title` variable with `titleMap`.
    - The completed class should look like this:

```
/**
 * Copyright (c) 2000-present Liferay, Inc. All rights reserved.
 *
 * This library is free software; you can redistribute it and/or modify it under
 * the terms of the GNU Lesser General Public License as published by the Free
 * Software Foundation; either version 2.1 of the License, or (at your option)
 * any later version.
 *
 * This library is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
 * FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more
 * details.
 */
```

```
package com.liferay.training.gradebook.service.impl;

import com.liferay.portal.aop.AopService;
import com.liferay.portal.kernel.exception.PortalException;
import com.liferay.portal.kernel.service.ServiceContext;
import com.liferay.portal.kernel.util.OrderByComparator;
import com.liferay.training.gradebook.model.Assignment;
import com.liferay.training.gradebook.service.base.AssignmentServiceBaseImpl;

import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.Map;

import org.osgi.service.component.annotations.Component;

/**
 * The implementation of the assignment remote service.
 *
 * <p>
 * All custom service methods should be put in this class. Whenever methods are added, re
 *
 * <p>
 * This is a remote service. Methods of this service are expected to have security checks
 * </p>
 *
 * @author Brian Wing Shun Chan
 * @see AssignmentServiceBaseImpl
 */
@Component(
        property = {
                "json.web.service.context.name=gradebook",
```

```
                "json.web.service.context.path=Assignment"
        },
        service = AopService.class
)
public class AssignmentServiceImpl extends AssignmentServiceBaseImpl {


        /*
         * NOTE FOR DEVELOPERS:
         *
         * Never reference this class directly. Always use <code>com.liferay.training.gra
         */
        public Assignment addAssignment(
                long groupId, Map<Locale, String> titleMap, String description,
                Date dueDate, ServiceContext serviceContext)
                throws PortalException {

                return assignmentLocalService.addAssignment(
                        groupId, titleMap, description, dueDate, serviceContext);
        }

        public Assignment deleteAssignment(long assignmentId)
                throws PortalException {

                Assignment assignment =
                        assignmentLocalService.getAssignment(assignmentId);

                return assignmentLocalService.deleteAssignment(assignment);
        }

        public Assignment getAssignment(long assignmentId)
                throws PortalException {
```

```
        Assignment assignment =
                assignmentLocalService.getAssignment(assignmentId);

        return assignment;
}

public List<Assignment> getAssignmentsByGroupId(long groupId) {

        return assignmentPersistence.findByGroupId(groupId);
}

public List<Assignment> getAssignmentsByKeywords(
        long groupId, String keywords, int start, int end,
        OrderByComparator<Assignment> orderByComparator) {

        return assignmentLocalService.getAssignmentsByKeywords(
                groupId, keywords, start, end, orderByComparator);
}

public long getAssignmentsCountByKeywords(long groupId, String keywords) {

        return assignmentLocalService.getAssignmentsCountByKeywords(
                groupId, keywords);
}

public Assignment updateAssignment(
        long assignmentId, Map<Locale, String> titleMap, String description,
        Date dueDate, ServiceContext serviceContext)
        throws PortalException {

        return assignmentLocalService.updateAssignment(
                assignmentId, titleMap, description, dueDate, serviceContext);
```

```
        }
}
```

Rebuild and Deploy the Service

1. **Run** the `buildService` Gradle task to regenerate the service.
    - Don't worry about errors you get for the method validate as we will resolve these in a later step.
2. **Start** the server if it's not running.
3. **Deploy** the *gradebook-api* and *gradebook-service* modules if they're not already deployed.

You should see a success log message if modules were deployed successfully:

```
2019-03-20 11:31:59.667 INFO  [pipe-start 984][BundleStartStopLogger:39] STARTED com.life
2019-03-20 11:32:02.573 INFO  [pipe-start 985][BundleStartStopLogger:39] STARTED com.life
```

Update entry_search_columns.jspf

1. **Open** `entry_search_columns.jspf` from the gradebook-web module.
2. **Replace** every `entry.getTitle` with `entry.getTitle(locale)`.

```
<%-- Generate assignment view  URL. --%>

<portlet:renderURL var="viewAssignmentURL">
        <portlet:param name="mvcRenderCommandName" value="<%=MVCCommandNames.VIEW_ASSIGNM
        <portlet:param name="redirect" value="${currentURL}" />
        <portlet:param name="assignmentId" value="${entry.assignmentId}" />
</portlet:renderURL>

<c:choose>

        <%-- Descriptive (list) view --%>

        <c:when
```

```
test='${assignmentsManagementToolbarDisplayContext.getDisplayStyle().equa

<%-- User --%>

<liferay-ui:search-container-column-user
        showDetails="<%=false%>"
        userId="<%=entry.getUserId()%>"
/>

<liferay-ui:search-container-column-text colspan="<%=2%>">

        <%
                String modifiedDateDescription =
                        LanguageUtil.getTimeDescription(
                                        request, System.currentTimeMillis
                                        - entry.getModifiedDate().getTime
        %>

        <h5 class="text-default">
                <liferay-ui:message
                        arguments="<%=new String[] {entry.getUserName(),
                        key="x-modified-x-ago" />
        </h5>

        <h4>
                <aui:a href="${viewAssignmentURL}">
                        ${entry.getTitle(locale)}
                </aui:a>
        </h4>

</liferay-ui:search-container-column-text>
```

```
		<liferay-ui:search-container-column-jsp
			path="/assignment/entry_actions.jsp" />
	</c:when>

	<%-- Card view  --%>

	<c:when
		test='${assignmentsManagementToolbarDisplayContext.getDisplayStyle().equa

		<%
			row.setCssClass("lfr-asset-item");
		%>

		<liferay-ui:search-container-column-text>

			<%-- Vertical card. --%>

			<liferay-frontend:icon-vertical-card
				actionJsp="/assignment/entry_actions.jsp"
				actionJspServletContext="<%= application %>"
				icon="cards2" resultRow="${row}"
				title="${entry.getTitle(locale)}"
				url="${viewAssignmentURL}">

				<liferay-frontend:vertical-card-sticker-bottom>

					<liferay-ui:user-portrait
						cssClass="sticker sticker-bottom"
						userId="${entry.userId}"
					/>
				</liferay-frontend:vertical-card-sticker-bottom>
```

```
                        <liferay-frontend:vertical-card-footer>

                                <div class="truncate-text">

                                        <%-- Strip HTML --%>

                                        <%=HtmlUtil.stripHtml(entry.getDescript
                                </div>
                        </liferay-frontend:vertical-card-footer>
                </liferay-frontend:icon-vertical-card>
        </liferay-ui:search-container-column-text>
</c:when>

<%-- Table view --%>

<c:otherwise>

        <liferay-ui:search-container-column-text
                href="${viewAssignmentURL}"
                name="title"
                value="<%= entry.getTitle(locale) %>"
        />

        <liferay-ui:search-container-column-user
                name="author"
                userId="${entry.userId}"
        />

        <liferay-ui:search-container-column-date
                name="create-date"
                property="createDate"
        />
```

```
                <liferay-ui:search-container-column-jsp
                        name="actions"
                        path="/assignment/entry_actions.jsp"
                />
        </c:otherwise>
</c:choose>
```

Update AddAssignmentMVCActionCommand

1. **Open** `AddAssignmentMVCActionCommand` class.
2. **Find** the comment `// Get parameters from the request..`
3. **Add** code for localization and the titleMap below that.
    - See the example below
4. **Add** the missing imports:

```
package com.liferay.training.gradebook.web.portlet.action;


import com.liferay.portal.kernel.exception.PortalException;
import com.liferay.portal.kernel.portlet.bridges.mvc.BaseMVCActionCommand;
import com.liferay.portal.kernel.portlet.bridges.mvc.MVCActionCommand;
import com.liferay.portal.kernel.service.ServiceContext;
import com.liferay.portal.kernel.service.ServiceContextFactory;
import com.liferay.portal.kernel.servlet.SessionErrors;
import com.liferay.portal.kernel.servlet.SessionMessages;
import com.liferay.portal.kernel.theme.ThemeDisplay;
import com.liferay.portal.kernel.util.DateFormatFactoryUtil;
import com.liferay.portal.kernel.util.LocalizationUtil;
import com.liferay.portal.kernel.util.ParamUtil;
import com.liferay.portal.kernel.util.WebKeys;
import com.liferay.training.gradebook.exception.AssignmentValidationException;
import com.liferay.training.gradebook.model.Assignment;
import com.liferay.training.gradebook.service.AssignmentService;
```

```
import com.liferay.training.gradebook.web.constants.GradebookPortletKeys;
import com.liferay.training.gradebook.web.constants.MVCCommandNames;

import java.util.Date;
import java.util.Locale;
import java.util.Map;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;

import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Reference;

/**
 * MVC Action Command for adding assignments.
 *
 * @author liferay
 */
@Component(
        immediate = true,
        property = {
                "javax.portlet.name=" + GradebookPortletKeys.GRADEBOOK,
                "mvc.command.name=" + MVCCommandNames.ADD_ASSIGNMENT
        },
        service = MVCActionCommand.class
)
public class AddAssignmentMVCActionCommand extends BaseMVCActionCommand {

        @Override
        protected void doProcessAction(
                ActionRequest actionRequest, ActionResponse actionResponse)
                throws Exception {
```

```
        ThemeDisplay themeDisplay =
                (ThemeDisplay) actionRequest.getAttribute(WebKeys.THEME_DISPLAY);

        ServiceContext serviceContext = ServiceContextFactory.getInstance(
                Assignment.class.getName(), actionRequest);

        // Get parameters from the request.

        // Use LocalizationUtil to get a localized parameter.

        Map<Locale, String> titleMap =
                LocalizationUtil.getLocalizationMap(actionRequest, "title");

String description = ParamUtil.getString(actionRequest, "description", null);

Date dueDate = ParamUtil.getDate(actionRequest, "dueDate", null);

try {

                // Call the service to add a new assignment.

                _assignmentService.addAssignment(
                        themeDisplay.getScopeGroupId(), titleMap, description, du

                // Set the success message.

                SessionMessages.add(actionRequest, "assignmentAdded");

                sendRedirect(actionRequest, actionResponse);
        }
        catch (AssignmentValidationException ave) {
```

```
                        // Get error messages from the service layer.

                        ave.getErrors().forEach(key -> SessionErrors.add(actionRequest, k

                        ave.printStackTrace();

                        actionResponse.setRenderParameter(
                                "mvcRenderCommandName", MVCCommandNames.EDIT_ASSIGNMENT);

                }
                catch (PortalException pe) {

                        // Set error messages from the service layer.

                   SessionErrors.add(actionRequest, "serviceErrorDetails", pe);

                        pe.printStackTrace();

                        actionResponse.setRenderParameter(
                                "mvcRenderCommandName", MVCCommandNames.EDIT_ASSIGNMENT);
                }
        }

        @Reference
        protected AssignmentService _assignmentService;

}
```

Update EditAssignmentMVCActionCommand

1. **Open** `EditAssignmentMVCActionCommand` class.
2. **Find** the comment `// Get parameters from the request..`
3. **Add** code for localization and titleMap below that:
4. **Add** the missing imports.

```
package com.liferay.training.gradebook.web.portlet.action;


import com.liferay.portal.kernel.exception.PortalException;

import com.liferay.portal.kernel.portlet.bridges.mvc.BaseMVCActionCommand;

import com.liferay.portal.kernel.portlet.bridges.mvc.MVCActionCommand;

import com.liferay.portal.kernel.service.ServiceContext;

import com.liferay.portal.kernel.service.ServiceContextFactory;

import com.liferay.portal.kernel.servlet.SessionErrors;

import com.liferay.portal.kernel.servlet.SessionMessages;

import com.liferay.portal.kernel.util.DateFormatFactoryUtil;

import com.liferay.portal.kernel.util.LocalizationUtil;

import com.liferay.portal.kernel.util.ParamUtil;

import com.liferay.training.gradebook.exception.AssignmentValidationException;

import com.liferay.training.gradebook.model.Assignment;

import com.liferay.training.gradebook.service.AssignmentService;

import com.liferay.training.gradebook.web.constants.GradebookPortletKeys;

import com.liferay.training.gradebook.web.constants.MVCCommandNames;


import java.util.Date;

import java.util.Locale;

import java.util.Map;


import javax.portlet.ActionRequest;

import javax.portlet.ActionResponse;


import org.osgi.service.component.annotations.Component;
```

```
import org.osgi.service.component.annotations.Reference;


/**
 * MVC Action Command for editing assignments.
 *
 * @author liferay
 *
 */
@Component(
        immediate = true,
        property = {
                "javax.portlet.name=" + GradebookPortletKeys.GRADEBOOK,
                "mvc.command.name=" + MVCCommandNames.EDIT_ASSIGNMENT
        },
        service = MVCActionCommand.class
)
public class EditAssignmentMVCActionCommand extends BaseMVCActionCommand {


        @Override
        protected void doProcessAction(
                ActionRequest actionRequest, ActionResponse actionResponse)
                throws Exception {


                ServiceContext serviceContext =
                        ServiceContextFactory.getInstance(Assignment.class.getName(), act


                // Get parameters from the request.


                long assignmentId = ParamUtil.getLong(actionRequest, "assignmentId");


                // Use LocalizationUtil to get a localized parameter.
```

```
Map<Locale, String> titleMap =
        LocalizationUtil.getLocalizationMap(actionRequest, "title");

String description = ParamUtil.getString(actionRequest, "description", nu

Date dueDate = ParamUtil.getDate(actionRequest, "dueDate", null);

try {

        // Call the service to update the assignment

        _assignmentService.updateAssignment(
                assignmentId, titleMap, description, dueDate, serviceCont

        // Set the success message.

        SessionMessages.add(actionRequest, "assignmentUpdated");

        sendRedirect(actionRequest, actionResponse);
}
catch (AssignmentValidationException ave) {

        // Get error messages from the service layer.

        ave.getErrors().forEach(key -> SessionErrors.add(actionRequest, k

        ave.printStackTrace();

        actionResponse.setRenderParameter(
                "mvcRenderCommandName", MVCCommandNames.EDIT_ASSIGNMENT);

}
```

```
                catch (PortalException pe) {

                        // Get error messages from the service layer.

                        SessionErrors.add(actionRequest, "serviceErrorDetails", pe);

                        pe.printStackTrace();

                        actionResponse.setRenderParameter(
                                "mvcRenderCommandName", MVCCommandNames.EDIT_ASSIGNMENT);
                }
        }

        @Reference
        protected AssignmentService _assignmentService;
}
```

Update the Assignment Validator Interface

1. **Open** the `AssignmentValidator` interface in the gradebook-api module.
2. **Replace** `String title` with `Map<Locale, String> titleMap`.
3. **Add** missing imports.

```
package com.liferay.training.gradebook.validator;

import com.liferay.training.gradebook.exception.AssignmentValidationException;

import java.util.Date;
import java.util.Locale;
import java.util.Map;

public interface AssignmentValidator {
```

```
        /**
         * Validates an Assignment
         *
         * @param titleMap
         * @param description
         * @param dueDate
         * @throws AssignmentValidationException
         */
        public void validate(
            Map<Locale, String> titleMap, String description, Date dueDate)
                        throws AssignmentValidationException;
}
```

Update the Assignment Validator Service Component

1. **Open** the `AssignmentValidatorImpl` class.
2. **Update** the `title` variable from a `String` to `titleMap` defined as `Map<Locale, String>`.
3. **Add*** missing imports.

```
package com.liferay.training.gradebook.util.validator;


import com.liferay.portal.kernel.util.LocaleUtil;
import com.liferay.portal.kernel.util.MapUtil;
import com.liferay.portal.kernel.util.Validator;
import com.liferay.training.gradebook.exception.AssignmentValidationException;
import com.liferay.training.gradebook.validator.AssignmentValidator;


import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.Map;
```

```
import org.osgi.service.component.annotations.Component;

@Component(
        immediate = true,
        service = AssignmentValidator.class
)
public class AssignmentValidatorImpl implements AssignmentValidator {

        /**
         * Validates assignment values and throws
         * {AssignmentValidationExceptionException} if the assignment values are not
         * valid.
         *
         * @param titleMap
         * @param description
         * @param dueDate
         * @throws AssignmentValidationException
         */
        public void validate(
                Map<Locale, String> titleMap, String description, Date dueDate)
                throws AssignmentValidationException {

                List<String> errors = new ArrayList<>();

                if (!isAssignmentValid(titleMap, description, dueDate, errors)) {
                        throw new AssignmentValidationException(errors);
                }
        }

        /**
         * Returns <code>true</code> if the given fields are valid for an
```

```
     * assignment.
     *
     * @param titleMap
     * @param description
     * @param dueDate
     * @param errors
     * @return boolean <code>true</code> if assignment is valid, otherwise
     *         <code>false</code>
     */
    private boolean isAssignmentValid(
            final Map<Locale, String> titleMap, final String description,
            final Date dueDate, final List<String> errors) {

        boolean result = true;

        result &= isTitleValid(titleMap, errors);
        result &= isDueDateValid(dueDate, errors);
        result &= isDescriptionValid(description, errors);

        return result;
    }

    /**
     * Returns <code>true</code> if description is valid for an assignment. If
     * not valid, an error message is added to the errors list.
     *
     * @param description
     * @param errors
     * @return boolean <code>true</code> if description is valid, otherwise
     *         <code>false</code>
     */
    private boolean isDescriptionValid(
```

```
        final String description, final List<String> errors) {

                boolean result = true;

                // Verify the map has something

                if (description == "") {
                        errors.add("assignmentDescriptionEmpty");
                        result = false;
                }
        return result;
}



/**
 * Returns <code>true</code> if due date is valid for an assignment. If not
 * valid, an error message is added to the errors list.
 * Note that this error can't ever happen in the user interface because
 * we are always getting a default value on the controller layer (Action Commands
 * However, this service could be access through the WS Api, which is why we neec
 *
 * @param dueDate
 * @param errors
 * @return boolean <code>true</code> if due date is valid, otherwise
 *         <code>false</code>
 */
private boolean isDueDateValid(
        final Date dueDate, final List<String> errors) {

        boolean result = true;

        if (Validator.isNull(dueDate)) {
```

```
                errors.add("assignmentDateEmpty");
                result = false;
        }

        return result;
}

/**
 * Returns <code>true</code> if title is valid for an assignment. If not
 * valid, an error message is added to the errors list.
 *
 * @param titleMap
 * @param errors
 * @return boolean <code>true</code> if the title is valid, otherwise
 *         <code>false</code>
 */
private boolean isTitleValid(
        final Map<Locale, String> titleMap, final List<String> errors) {

        boolean result = true;

        // Verify the map has something

        if (MapUtil.isEmpty(titleMap)) {
                errors.add("assignmentTitleEmpty");
                result = false;
        }
        else {

                // Get the default locale.

                Locale defaultLocale = LocaleUtil.getSiteDefault();
```

```
                        if ((Validator.isBlank(titleMap.get(defaultLocale)))) {
                                errors.add("assignmentTitleEmpty");
                                result = false;
                        }
                }

                return result;
        }
}
```

Test Localization

1. **Go to** localhost:8080 in your browser.
2. **Refresh** the page.
   - Make sure the module has restarted successfully before refreshing the page.
3. **Click** the *Options* icon next to the *Test Assignment*.
4. **Choose** *Edit*.
5. **Click** the *Localization* icon (default should be an American flag with *en-US*).
6. **Choose** the French option (*fr-Fr* with the French flag).
7. **Type** `Test d'exercices` for *Title* (The original English title will display beneath the field).
8. **Click** *Save*.

Add Localization Resources

Exercise Goals

- Review portlet component properties
- Change the localization file encoding
- Provide localization resources
- Test the user interface

Take a look at the component properties of `GradebookPortlet.java`. See the resource bundle property:

```
@Component(
        immediate = true,
        property = {
                "com.liferay.portlet.display-category=category.training",
                "com.liferay.portlet.instanceable=false",
                "javax.portlet.display-name=Gradebook",
                "javax.portlet.init-param.template-path=/",
                "javax.portlet.init-param.view-template=/view.jsp",
                "javax.portlet.name=" + GradebookPortletKeys.GRADEBOOK,
                "javax.portlet.resource-bundle=content.Language",
                "javax.portlet.security-role-ref=power-user,user",
                "javax.portlet.init-param.add-process-action-success-action=false"
        },
        service = Portlet.class
)
public class GradebookPortlet extends MVCPortlet {
}
```

The value of the property translates to a file system path `src/main/resources/content/Language.properties`, which was done automatically by the module template. By setting this property, localization resources of `Language.properties` are available for Liferay JSP tag libraries automatically. To add support for another language, let's say for German, just add a new language properties file `src/main/resources/content/Language_de_DE.properties`.

Default encoding for the language text files is ISO-8859-1. While it usually works for English, localizations for other languages with non-standard ASCII special characters will break.

Let's change the file encoding to UTF-8.

Change the Localization File Encoding

1. **Right-click** on the `src/main/resources/content/Language.properties` file in the *gradebook-web*.
2. **Click** on *Properties*.
3. **Change** the *Text file encoding* to UTF-8 and close the dialog.

Provide Localization Resources

1. **Open** the `gradebook-web/src/main/resources/content/Language.properties` file.
2. **Implement** the file as follows:

```
#
# This is the default localization file containing the key to display messages mappings.
#


#
# Standard portlet display messages
#
javax.portlet.description.com_liferay_training_gradebook_web_portlet_GradebookPortlet=Gra
javax.portlet.display-name.com_liferay_training_gradebook_web_portlet_GradebookPortlet=Gr
javax.portlet.keywords.com_liferay_training_gradebook_web_portlet_GradebookPortlet=Gradeb
javax.portlet.short-title.com_liferay_training_gradebook_web_portlet_GradebookPortlet=Gra
```

```
javax.portlet.title.com_liferay_training_gradebook_web_portlet_GradebookPortlet=Gradebook

#
# Application category
#
category.training=Liferay Training


#
# Asset name localization (Asset Publisher, search and permissions UI)
#
model.resource.com.liferay.training.gradebook.model.Assignment=Assignment
model.resource.com.liferay.training.gradebook.model=Gradebook


#
# Application Display Template localization
#
application-display-template-type=Gradebook template


#
# Permission localizations
#
action.ADD_ASSIGNMENT=Add Assignment
action.ADD_SUBMISSION=Add Submission
action.DELETE_SUBMISSION=Delete Submission
action.EDIT_SUBMISSION=Edit Submission
action.GRADE_SUBMISSION=Grade Submissions
action.VIEW_SUBMISSIONS=View Submissions


#
# Other messages
#
add-assignment=Add New Assignment
add-submission=Add New Submission
```

```
add-submission-for-x=Add New Submission for {0}

are-you-sure-to-delete=Are you sure to delete this?

assignment-added-successfully=Assignment added successfully

assignment-deleted-successfully=Assigment was deleted succesfully

assignment-duedate=Assignment Due Date

assignment-information=Assignment Information

assignment-updated-successfully=Assignment updated successfully

assignments=Assignments

assignments-help-text=Please click the plus sign to add a new assignments.

edit-assignment=Edit

edit-submission-for-x=Edit Submission for {0}

error.assignment-date-empty=Due date cannot be empty.

error.assignment-description-empty=Description cannot be empty.

error.assignment-service-error=Service request failed with message: {0}

error.assignment-title-empty=Please enter a valid title.

error.only-one-submission-allowed=Only one submission per student is allowed.

error.assignment-title-format=Please enter letters, words, numbers, or standard punctuat

error.submission-is-too-late=Your submission is too late.

error.submission-service-error=There was a problem with your submission.

error.submission-text-null=Submission text cannot be empty.

error.submission-text-too-short=Submission text too short.

error.submission-text-too-long=Submission text too long.

grade=Grade

gradebook-example-adt=Gradebook Application Display Template example

gradebook-portlet-instance-configuration-name=Gradebook Portlet Configuration

gradebook-service-configuration-name=Gradebook Service Configuration

grade-submission=Grade

grading=Grading

no-assignments=No assignments yet

no-comment=No comments yet.

no-submissions=No submissions for this assignment yet

not-graded=Not graded yet.
```

```
student=Student
submission-comment=Submission Comment
submission-added-succesfully=Submission was created succesfully
submission-deleted-succesfully=Submission was deleted succesfully
submission-graded-succesfully=Submission was graded succesfully
submission-information=Submission Information
submission-not-graded=Not graded
submission-settings=Submission Settings
submission-text=Submission Text
submissions=Submissions
submit-date=Submitted
this-is-an-example-adt=This is an Example ADT for the Gradebook
viewing-submissions-not-allowed=You don't have permissions to view submissions.
your-submission=Your Submission
```

> Note: Some of the keys are used at later and optional exercise steps.

Test the User Interface

1. **Go to** localhost:8080 in your browser after redeploying the module.
2. **Open** the Gradebook application.
3. **Refresh** the page.
    - You should now be able to see the labels and messages correctly:

Add CSS Resources

Exercise Goals

- Add CSS resources
- Configure the portlet component to use the provided CSS resources
- Test the changes

The styling of the assignment list needs polishing. In the *Table* view, the author column is not aligned and the links should be underlined.

We can provide CSS resources for the Gradebook portlet to fix the issues.

Let's first create a CSS file to provide our custom styles for the *gradebook-web* module.

Add CSS Resources

1. **Create** a folder file `resources/META-INF/resources/css` if it doesn't already exist.
2. **Create** (or Open) a file `resources/META-INF/resources/css/main.scss` and implement as follows:

```
.gradebook-portlet {


        h1 {
                font-size: 1.7rem;
                margin: 20px 0 10px 0;
        }


        h2 {
                margin: 30px 0 10px 0;
        }
```

```
.lfr-search-container-wrapper {
	a {
		text-decoration: underline;
	}

	.user-icon {
		float: left;
	}

	.user-details {
		vertical-align: sub;

		.user-name {
			color: inherit;
		}
	}
}

.submission-text {
	border: 1px solid #eee;
	border-radius: 5px;
	padding: 20px;
}

.assignment-metadata,
.submission-metadata {
	font-size: .9em;

	dt {
		margin-top: 15px;
	}
```

```
                dd {


                }
        }

        .edit-assignment {


                .assignment-description {
                        font-size: .875rem;
                        font-weight: 600;


                        .reference-mark {
                                font-size: 6px;
                        }
                }
        }
}
```

The portlet component needs to know where to load the CSS resources from. Also, it's a good practice to encapsulate portlet styles by wrapping the portlet in a CSS class.

Configure the Portlet Component

1. **Open** the `GradebookPortlet.java` portlet class.
2. **Add** the following component property:

```
"com.liferay.portlet.css-class-wrapper=gradebook-portlet",
```

Test the Changes

1. **Refresh** the browser to see the changes after redeploying the module.
2. **Switch** the list to the *Table* view using the button on the left side of the search bar, if necessary. The Author column is now better aligned and the links have underlining:

Application Localization and Styling Module Quiz

1. The key to display term mapping is done in which of the following files?

   A. Language.properties

   B. Language.format

   C. Terms.properties

   D. Terms.format

2. Which of the following language file names follow proper naming syntax?

   A. Language.en.US.properties

   B. Language_en.US.properties

   C. Language_en_US.properties

   D. Language_en_US_properties

3. UI taglibs fall back to the portal bundle if a portlet bundle is not found.

   A. True

   B. False

4. Which of the following are CSS-related portlet component properties? (Choose all correct answers).

   A. com.liferay.portlet.header-page-css

   B. com.liferay.portlet.header-portal-css

   C. com.liferay.portlet.footer-page-css

   D. com.liferay.portlet.footer-portal-css

   E. com.liferay.portlet.header-portlet-css

5. Files with the .css suffix are SASS-compiled to .scss and referenced in the properties with .css.

   A. True

   B. False

Answer Key

1. A
2. C
3. True
4. B, D, and E
5. False