Cúram 8.1.2

Appeals Developer Guide

Note

Before using this information and the product it supports, read the information in Notices on page 13

Edition

This edition applies to Cúram 8.1, 8.1.1, and 8.1.2.

Contents

1 Developing Appeals

Use this information to extend the default features of Merative ™ SPM Appeals. By default, Appeals is configured for case types of product delivery, issue, and integrated case. Appeals can be extended to handle extra case types. Case objects can be configured to be appealed, rather than the parent case itself.

1.1 Enabling the appeal of case objects

Complete the following tasks to allow objects on a case to be appealed, rather than the case itself. An appealable object can be anything on a case type that has a unique identifier.

Implementing the Appealable interface

You must implement the Appealable interface. The data that is returned populates the Description column on the "Items Under Appeal" page.

There are two methods to be implemented in the Appealable interface.

```
LocalisableString getAppealObjectDescription(APPEALOBJECTTYPEEntry objectType, long
 objectID)
```

This method returns the globalized description for the object.

```
String getHomePageURI(APPEALOBJECTTYPEEntry objectType, long objectID)
```

This method returns the home page for the object.

Adding a code table entry

You must add a code table entry to the ct_AppealObjectType.ctx code table. You must set the Java identifier for the entry as it is used in binding the Java implementation.

```
<code
      default="false"
      java_identifier="EXAMPLEOBJECT"
      status="ENABLED"
      value="AOT1001"
    >
      <locale
        language="en"
        sort_order="0"
      >
        <description>Exampleobject</description>
        <annotation/>
      </locale>
    </code>
```

Binding the code table to the implementation

Bind the implementation of the AppealableObjectType interface to the AppealObjectType code in a Guice module as shown.

```
final MapBinder<APPEALOBJECTTYPEEntry, Appealable> mapbinder = MapBinder
        .newMapBinder(binder(), APPEALOBJECTTYPEEntry.class, Appealable.class);

    mapbinder.addBinding(APPEALOBJECTTYPEEntry.EXAMPLEOBJECT).to(
        AppealableExampleobjectImpl.class);
```

Implementing the client wizard

Implement the wizard framework that handles the creation of an Appeal case with a list of appealable objects. You use this framework to avoid any compile dependencies on the Appeals component.

Procedure

1. Implement the AppealableCaseType interface for the parent case type.
2. Create the first wizard page, which presents a list of objects on the case to be appealed. This page must pass a delimited list of objects to the predefined second wizard screen (Appeal_createWizard). The format of the delimited list is:

   ```
   ObjectID,ObjectTypeCode|
   ```

   For example, "1001,AOT1|2001,AOT2|2002,AOT2|" Typically, a MULTISELECT list is used on the client page, so a façade helper class is required to convert from the multiselect to this delimited format. A façade method is also required to return the wizard properties file.
3. Create the wizard properties file, defining the following details:

   ```
   Number.Wizard.Pages=3

   {FirstWizardPage}.Wizard.Item.Text=Select {ObjectType}
   {FirstWizardPage}.Wizard.Page.Title=Step 1:
   {FirstWizardPage}.Wizard.Page.Desc=Select {ObjectType}
   Wizard.PageID.1={FirstWizardPage}

   AppealDetermination_selectParticipants.Wizard.Item.Text=Select Appealant/Respondent
   AppealDetermination_selectParticipants.Wizard.Page.Title=Step 2:
   AppealDetermination_selectParticipants.Wizard.Page.Desc=Select Appealant/Respondent
   Wizard.PageID.2=AppealWizard_SelectParticipants

   AppealDetermination_createAppeal.Wizard.Item.Text=Record Appeal Details
   AppealDetermination_createAppeal.Wizard.Page.Title=Step 3:
   AppealDetermination_createAppeal.Wizard.Page.Desc=Record Appeal Details
   Wizard.PageID.3=AppealWizard_createAppeal
   ```

   Where {FirstWizardPage} is the name of a client page created in the previous step and {ObjectType} is the name of the object.

1.2 Enabling the appeal of another case type

By default, the Appeals component is configured to work only with case types of Product Delivery, Issue, and Integrated Case. Complete the following tasks to enable the appeal of another case type by using the Appealable Case Type interface.

Implementing the AppealableCaseType interface

Implement the AppealableCaseType interface for the new case type.

There are five methods to be implemented on the AppealableCaseType interface.

Use the following three methods to define the business logic for the case type:

```
boolean isContinueBenefitsEnabled(CaseID caseID);
```

This method returns true if Continue Benefits functionality should be enabled for this instance of the Case Type.

```
AppealableCaseTypeDetailsList listAppealableCaseDetails();
```

This method lists all of the case configurations for the abstract Case Type that can be configured for appeals.

```
boolean isCaseAppealable(CaseID caseID);
```

This method returns true if the case can be appealed in its current state. For example, if the case must be in a state of "Active", then implement the logic to check for this state in this method.

Use the following two methods to implement the wizard framework for appealing case objects:

```
String getCreateWizardProperties();
```

This method returns the name of the wizard properties file.

```
ClientURI getCreateWizardURI(CaseID caseID);
```

This method returns the initial screen in the wizard.

Binding the code table to the implementation

Bind the implementation of the AppealableCaseType interface to the Case Type code in a Guice module as shown.

```
final MapBinder<CASETYPECODEEntry, AppealableCaseType> appealableCaseTypeBinder =
 MapBinder
.newMapBinder(binder(), CASETYPECODEEntry.class,
            AppealableCaseType.class);

    appealableCaseTypeBinder.addBinding(CASETYPECODEEntry.APPLICATION_CASE).to(
        ApplicationAppealableCaseType.class);
```

Updating the Appeals Client navigation

Update the client configuration to show the Appeals pages for the new case type.

The following changes need to be made:

1. Add the following entries to the Workspace Section File:

```
<sc:tab id="AppealHearing"/>
<sc:tab id="AppealHearingCaseHome"/>
<sc:tab id="AppealHearingCaseHomeIC"/>
<sc:tab id="AppealHearingIC"/>
<sc:tab id="AppealHearingReviewHearing"/>
<sc:tab id="AppealHearingReviewHearingIC"/>
<sc:tab id="AppealHearingReviewHome"/>
<sc:tab id="AppealHearingReviewHomeIC"/>
<sc:tab id="AppealJudicialReviewHome"/>
<sc:tab id="AppealJudicialReviewHomeIC"/>
<sc:tab id="LegalActionsForHearing"/>
<sc:tab id="LegalActionsForImmediateDetentionDecision"/>
<sc:tab id="LegalActionsForPetition"/>
<sc:tab id="AppealDeskHearing"/>
<sc:tab id="AppealDeskHearingIC"/>
<sc:tab id="LegalActionOrganizationHome"/>
<sc:tab id="AppealSearch"/>
<sc:tab id="AppealHearingIssue"/>
```

2. Add a link to create an Appeal case to the Case menu file:

```
<mc:menu-item dynamic="true"
  id="CaseAppeal"
  page-id="{pageID}"
  title="MenuItem.Title.CaseAppeal"
  tooltip="MenuItem.Tooltip.CaseAppeal"
  open-as="modal"
/>
```

Where {pageID} is Appeal_resolveCaseAppealWizard or, if appealing an object, the name of the first wizard screen.

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

Information concerning non-Merative products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Merative has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-Merative products. Questions on the capabilities of non-Merative products should be addressed to the suppliers of those products.

Any references in this information to non-Merative websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Merative product and use of those websites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to Merative, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy

The Merative privacy policy is available at https://www.merative.com/privacy.

Trademarks

Merative ™ and the Merative ™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Adobe™, the Adobe™ logo, PostScript™, and the PostScript™ logo are either registered trademarks or trademarks of Adobe™ Systems Incorporated in the United States, and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft™, Windows™, and the Windows™ logo are trademarks of Microsoft™ Corporation in the United States, other countries, or both.

UNIX™ is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.