

Cúram 8.1.3

Implementing the Eligibility Viewer Guide

Note

Before using this information and the product it supports, read the information in [Notices on page 15](#)

Edition

This edition applies to Cúram 8.1, 8.1.1, 8.1.2, and 8.1.3.

© Merative US L.P. 2012, 2024

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

Contents

Note..... iii

Edition..... v

1 Implementing the Eligibility Viewer..... 9

 1.1 Implementing a service provider interface for the Eligibility Viewer..... 9

 The Eligibility Viewer data format.....9

 Binding the SPI to your integrated case type..... 11

 Add, Enable, and Configure the Eligibility Viewer.....11

 Implementing key events for the Eligibility Viewer..... 13

Notices..... 15

 Privacy policy..... 16

 Trademarks..... 16

1 Implementing the Eligibility Viewer

The Eligibility viewer (EV) is an enhancement to caseworker functionality that gives caseworkers a holistic view of eligibility for a person or for an integrated case. The information can be presented in two tabs, one on the person view, and one on the integrated case view.

1.1 Implementing a service provider interface for the Eligibility Viewer

You must define the data that you want to display, implement the Service Provider Interface that returns the required data that you specify, and bind the data to the appropriate integrated case.

Before you begin

This information assumes that you are familiar with basic Cúram development tasks.

About this task

This Service Provider Interface implementation is called every time that eligibility information about your integrated case type is needed. For example, whenever someone opens the eligibility tab on your IC, or whenever someone opens the eligibility tab for a person who is a member of your type of IC. This implementation returns the eligibility data for your integrated case type, for a specified case and year. If an eligibility interval is valid at any point in the specified year, it is included when this data is returned.

Procedure

Create a java class to implement

```
curam.eligibilitytimelinecalendar.impl.TimelineCalendarDataRetrieval.
```

The Eligibility Viewer data format

Data that is returned by your Service Provider Interface implementation must be in the following format.

TimelineCalendarRow

TimelineCalendarRow represents a single row in the Eligibility View. For eligibility, it represents a person.

Table 1: TimelineCalendarRow Attribute

Attribute Name	Type	Purpose
ID	String	This attribute contains the ID of this row. For EV, the ID must be the string value of the ConcernRoleID of the person. This attribute must be populated.
title	String	This attribute contains the string of text that represents the person. For example, James Smith (51). This attribute must be populated.
timelines	List<TimelineCalendarTimeline>	This attribute contains a list of eligibility timelines for this person. For overlapping eligibility information, use multiple TimelineCalendarTimeline types.

TimelineCalendarTimeline

TimelineCalendarTimeline represents an eligibility timeline. Eligibility information cannot overlap within this timeline.

Table 2: TimelineCalendarTimeline Attribute

Attribute name	Type	Purpose
ID	String	This attribute contains the ID by which you can reference this timeline. This attribute is not used in EV.
timeline	Timeline<TimelineCalendarTimeline>	This attribute contains a Cúram Timeline of TimelineCalendarIntervalValue in which all the dates and values of eligibility are stored.

TimelineCalendarIntervalValue

TimelineCalendarIntervalValue represents the eligibility value for a given timeline interval.

Table 3:

Attribute Name	Type	Purpose
ID	String	This attribute contains the ID by which the interval is referenced. It is used to configure the color of the eligibility bar in EV. For eligibility, this attribute should be populated by the product type. This attribute must be populated.

Attribute Name	Type	Purpose
title	String	This attribute represents the text that is shown in the eligibility bar. For EV it should reference the name of the product for which the person is eligible in this period.
categoryCode	String	This attribute contains the category code-table code that describes the interval. This attribute is optional.
productNameCode	String	This attribute contains the product name code table code which describes this interval. This attribute is optional.
subtitle	String	This attribute contains a subheading to be displayed in the eligibility overlay. This attribute is optional.
additionalInformation	Map<String, String>	This attribute contains key value pairs that are displayed in tabular format in the eligibility overlay.

Binding the SPI to your integrated case type

Binding the Service Provider to your integrated case type allows the SPI to be discovered and injected when needed.

Procedure

1. Create a java class that extends `com.google.inject.AbstractModule` and override its `configure` method. In this method, create a `MapBinder` with a key of `String` and a value of `TimelineCalendarDataRetrieval.class`, for example, `MapBinder.newMapBinder(binder(), String.class, TimelineCalendarDataRetrieval.class)`. Add your implemented SPI to this map with a key of your integrated case type.
2. Add an entry to your `MODULECLASSNAME.dmx` data file so that your module can be found by Cúram.

Add, Enable, and Configure the Eligibility Viewer

You must add client screens to display the eligibility data for the client. The following tasks describe how to add the EV to your application, enable EV, and configure EV.

Adding and enabling an Eligibility tab to your navigation file

You must add the Eligibility viewer to your chosen integration case type and then enable the EV.

About this task

The Eligibility Viewer needs to be added to your Integrated Case application navigation file (`.nav`). EV contains a timeline calendar widget that visually displays the eligibility information

for your Integrated Case. For accessibility, EV also contains a list view of the information that is visually represented in the widget.

Procedure

- 1. Add the following code fragment to your Integrated Case navigation file (*.nav*).

```
<nc:navigation-page id="EligibilityGraph" page-  
id="IntegratedCase_timelineCalendarGraph"  
title="Page.Title.EligibilityGraphView"/>
```

- 2. Add the following properties to your Integrated Case navigation *.property* property file.

```
Page.Title.EligibilityGraphView=Eligibility
```

- 3. Enable the Eligibility tab on the Person tab. Change the value of the environment variable *curam.core.timelinecalendar.enabled* to true.

Configuring the Eligibility Viewer

The following options to configure the Eligibility Viewer are available by modifying properties in the Application Resource *TimelineCalendar.properties* file. This configuration can be done during either runtime or build time.

About this task

You must be familiar with updating Cúram administration properties.

Table 4: Cúram administration properties

Property name	Type	Purpose
{Integrated Case Type}.showRowTitles	Boolean	This property determines whether the Person's name is displayed in the Eligibility Viewer. This determination is set on a per integrated case type basis using the <i>ProductCategory codetable</i> . For example, <i>CT2001.showRowTitles=false</i> , which hides the names of Persons eligible for the Person tab.
{Product Type}.innerColor	Hex color	The color that is to be used in the border of all eligible intervals of this product type. It refers to the <i>TimelineCalendarIntervalValue.id</i> set while implementing the SPI, for example, #Ab2E4d.
{TimelineCalendarIntervalValue.id}.outerColor	Hex color	The color that is to be used in the fill of all eligible intervals of this product type. It refers to the <i>TimelineCalendarIntervalValue.id</i> that is set while implementing the SPI, for example, #A2e.

Note: This procedure can also be completed at build time by overriding the `TimelineCalendar.properties` entry in a custom `AppResource.dmx` file.

Procedure

1. Log in to the Cúram Administration Application.
2. In the Shortcuts Panel for the Administration Workspace, select **Intelligent Evidence Gathering**.
3. Select **Application Resources**.
4. Use the **Property** category as a filter to search for `TimelineCalendar.properties`.
5. Select **download** from the list row menu.
6. Use the previous table to choose your configurations. Add those configurations to the file and save the file.
7. Open a modal by selecting **Edit** from the list row menu.
8. Browse to the saved file and add the New Content.
9. Click **Save**.

Implementing key events for the Eligibility Viewer

Key events are changes to an integrated case that might affect eligibility. Key changes can be based on either evidence-based changes or non-evidence-based changes. An icon is displayed on the Eligibility Viewer below the month in which the key event occurs. Caseworkers can review key events to quickly determine what has changed on the case and the reason for a change in eligibility.

Customizing the display of evidence-based key events

Customize the key events messages that the eligibility viewer displays based on changes to particular evidence types.

About this task

A key event is automatically generated when a change in evidence occurs. The default key events message is in the following format:

```
Date of change - Evidence Type - Case participant name (Case participant age)
```

The key events message is controlled by the following message:

```
BpoTimelineCalendarKeyEvents.INF_TIMELINE_KEY_EVENTS
```

Note: The eligibility viewer uses the `Bootstrap.properties` file for key dates. Otherwise, it uses the `ApplicationConfiguration.properties` file. For consistent date formatting, ensure that you set the client and server date formats to the same value. For more information, see the related links.

Procedure

1. For your evidence type, create a class that implements the following:

```
curam.core.sl.util.timelinecalendar.impl.TimelineCalendarKeyEventsDataRetrival
```

2. Bind the implementation by using a Guice MapBinder, where your evidence type is the key, for example:

```
MapBinder.newMapBinder(binder(), String.class, TimelineCalendarDataRetrieval.class);
```

Related concepts**Related reference*****Customizing the display of non-evidence-based key events***

Not all events that affect eligibility are based on changes to evidence, for example, when a person becomes 65 years of age. A key event can also be generated for non-evidence base events. Configure non-evidence-based key events based on the type of integrated case.

Procedure

1. For your integrated case type, create a class that implements the following:

```
curam.core.sl.util.timelinecalendar.impl.TimelineCalendarKeyEventsRulesDataRetrival
```

2. Bind the implementation by using a Guice MapBinder, where your integrated case type is the key, for example:

```
MapBinder.newMapBinder(binder(), String.class,
    TimelineCalendarKeyEventsRulesDataRetrival.class);
```

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.