Cúram Action View 1.0.0

Note

Before using this information and the product it supports, read the information in Notices on page 45

Edition

This edition applies to Cúram Action View 1.0.0.

Contents

Chapter 1 Cúram Action View Business Overview

Organizations that administer social programs require workers to review information from multiple cases and applications in order to understand a client's current situation and determine next steps. Cúram Action View is a workspace that brings this information together in a consolidated, person-centric landing page - giving workers a single view of the most relevant details about the client and their interactions with the organization.

The aim of Cúram Action View is two-fold. Firstly, it provides a single workspace from which workers can access the cross-case information that is most important to day-to-day decision-making, such as applications, cases, payments, relationships, and recent activity. Secondly, it supports more timely and consistent service by surfacing items that require attention and presenting appropriate actions in context, so that workers can progress work from the same view in which they are reviewing the information.

1.1 Core purpose and benefits

Cúram Action View supports organizations in:

- reducing the need to navigate across multiple cases and evidence pages by aggregating person-centric information into a single workspace
- understanding what is currently happening for the client, what has recently changed, and whether any items require follow-up
- identifying issues that may affect eligibility or payment outcomes and directing workers to the information they need in order to resolve those issues
- presenting context-appropriate actions so that workers can undertake common tasks without leaving the workspace
- tailoring the workspace so that it reflects local programs, policies, and priorities

Action View can be configured and extended by the organization so that the cards, layout, available actions, and events reflect local requirements.

1.2 What the caseworker sees

The Action View tab is composed of cards, each focused on a particular aspect of the client’s situation. Together, these cards give caseworkers a consolidated, at-a-glance picture of a client's situation — reducing the need to navigate across multiple screens and enabling faster, more informed decision-making.

- **Quick Actions** – access to the most commonly performed actions from the Action View workspace.
- **Relationships** – a view of the client's personal relationships and shared case participation to help workers understand who else is relevant to the client’s cases.
- **Applications** – an overview of the client’s applications and requested programs, including their current progress.

- **Cases and Entitlement** – an overview of the client's cases, entitlements, and payments, including indicators of outstanding actions that may require review.
- **Recent Activity** – a chronological view of significant events, enabling workers to trace what has happened recently for the client.

Chapter 2 The Cards

The Action View tab is composed of cards, each focused on a particular aspect of the client's situation. Together, these cards give caseworkers a consolidated, at-a-glance picture of a client's situation — reducing the need to navigate across multiple screens and enabling faster, more informed decision-making.

- **Quick Actions** – access to the most commonly performed actions from the Action View workspace.
- **Relationships** – a view of the client's personal relationships and shared case participation to help workers understand who else is relevant to the client's cases.
- **Applications** – an overview of the client's applications and requested programs, including their current progress.
- **Cases and Entitlement** – an overview of the client's cases, entitlements, and payments, including indicators of outstanding actions that may require review.
- **Recent Activity** – a chronological view of significant events, enabling workers to trace what has happened recently for the client.

2.1 Quick Actions

Quick Actions in Cúram Action View provide fast, in-context access to commonly used actions without requiring the caseworker to navigate away from the current view. From the Quick Actions area, the caseworker can perform frequent tasks — such as creating a note — directly from Action View, maintaining focus on the participant, application, or case in context.

Quick Actions are presented in a dedicated area within the Action View layout. All actions in this area follow the same interaction, layout, and accessibility patterns.

What caseworkers can do

Using Quick Actions, the caseworker can:

- Perform frequent tasks directly from Action View without navigating to another page.
- Record observations and decisions at the moment they occur.
- Save information against the correct entity (person, application, or case).
- Access actions that are immediately recognizable, keyboard accessible, and consistent in behavior.

The Quick Actions area is customizable. Organizations can add their own quick actions to suit their business needs. The following quick action is provided out of the box.

Available Quick Actions

New Note

Purpose

The New Note quick action allows the caseworker to create a note directly from Action View, ensuring observations and decisions can be recorded at the moment they occur and saved against the correct entity.

Creating a New Note

Selecting New Note opens a modal window over Action View. From the modal, the caseworker can:

- Enter the note content.
- Select where the note should be saved (the person, application, or case).
- Save the note without leaving the current context.

Selecting where the note should be saved

When the modal opens, a default save location is pre-selected:

- Where a case exists, the most recent case is selected by default.
- Where a case doesn't exist, the person is selected by default.

The caseworker can change the selection using a dropdown. Items in the dropdown are grouped and ordered as follows:

- **Active** – contains active or in-progress items, sorted by most recently created first.
- **Closed / Disposed** – contains closed or disposed items, sorted by most recently closed first.

Status labels are applied where feasible to help the caseworker identify the correct save location.

Quick Action Shortcuts

Keyboard shortcuts are provided for each quick action, allowing caseworkers to perform the action without using the mouse. This improves efficiency for power users and supports faster task execution.

New Note Shortcut

Pressing **Shift + N** opens the New Note modal immediately, without needing to select the New Note button.

When a shortcut is not active

A shortcut is disabled in the following situations:

- **Focus is inside an editable form control** (e.g. an input field or text area). The shortcut does not trigger any action, allowing the caseworker to type normally.
- **Focus is outside the Cúram Action View page** (e.g. on the context panel or another part of the application). The shortcut has no effect.

- **The shortcut is reserved or blocked** by the browser or accessibility tooling.

Toast Notifications

Toast notifications provide immediate, non-intrusive feedback when a caseworker completes an action from Action View.

New Note Toast Notifications

When a note is saved from the New Note quick action, a toast notification is displayed in the header of the Quick Actions card.

- **On success:** "Note saved successfully." The notification is preceded by a checkmark in a green circle.
- **On failure:** "Note not saved." The notification is preceded by a red circle with a diagonal slash.

For more information on toast notifications and their use in Cúram Action View, see Toast Notifications.

Sensitivity and Access Control

Quick Actions respect the sensitivity levels configured for participants and users. A caseworker can only see details of cases or applications they have permission to view. This ensures that sensitive client information is protected and that caseworkers only interact with entities they are permitted to access.

Creating notes for cases and applications

For each case or application, the caseworker's sensitivity must be equal to or greater than the sensitivity of the primary client on the case or application.

- If the caseworker does not have the required sensitivity level for a case or application, that case or application is not available in the save location dropdown. The caseworker cannot select it as a destination for the note.
- If the primary client is not sensitive, but another member on the case or application is, the case or application can still be selected as a save location.

2.2 Relationships

The Relationships card in Cúram Action View provides a consolidated view of a client's personal relationships and shared case participation. From a single location, the caseworker can see who is connected to the client, how they are related, and whether they share any cases — helping to quickly understand the client's support network without leaving Action View.

When the person in context changes, the card title and content update automatically. For example, if the person in context is Maria, the card title is displayed as *Maria's Relationships*.

What caseworkers can do

Using the Relationships card, the caseworker can:

- View the client's active relationships, sorted alphabetically by the related participant's first name.
- Identify how each person is related to the client and when the relationship started.
- Assess shared case participation at a glance.
- Spot risk or vulnerability indicators, such as special cautions.
- Navigate directly to a related participant's Action View to review their details.
- Add, edit, and delete relationships directly from the card.
- View or mask identification details for related participants as needed.

Relationship Card Views

Relationship Card Views

Quick scan (collapsed view)

The initial, collapsed view is intended for rapid assessment of relationships. Each relationship entry provides a concise summary that allows the caseworker to determine:

- who is connected to the client and how they are related
- which relationships are most recent
- whether any related participants have active special cautions
- how many cases the client shares with each related participant

Review in Context (expanded view)

When the caseworker expands a relationship, additional information is displayed to support day-to-day decisions, including:

- **Address** – the participant's primary or preferred address. The preferred address takes precedence where available.
- **Identification** – the preferred identification type and reference number.
  - The full identification value is displayed by default as plain text with a hide icon beside it.
  - The caseworker can mask the value using the hide control; the value is then masked and a show icon appears beside it.
- **Shared cases** – all cases where both the person in context and the related participant are active members.
  - Format: *Case Type – Case Reference*, for example *Income Support – 272*.
  - A count appears in the collapsed view (for example *2 Shared Cases*); the full list appears in the expanded view.
  - Shared case counts and lists update dynamically when participants are added to or removed from cases.

- **Special cautions** – any active special cautions associated with the participant, for example *Fraud Sanction*.
  - In the collapsed view, a special caution icon appears next to the participant's name when an active special caution exists.
  - In the expanded view, special cautions are displayed in full detail at the end of the record. The icon is not shown when the record is expanded.

To support efficient review of households with multiple relationships, the card provides **Expand All** and **Collapse All** controls at the top right of the card. Only one control is shown at a time: Expand All when all records are collapsed, and Collapse All when at least one record is expanded. Individual records can still be expanded or collapsed independently, and records retain their state when the caseworker navigates away and returns during the same session. The controls are hidden when no relationship records exist.

Which Relationships are displayed

- The card only displays current and future relationships; any relationship with an end date in the past is not displayed, even if its status is still active.
- Cancelled relationships are excluded.

Navigating from Relationships

From the Relationships card, the caseworker can navigate directly to related records. Depending on the participant and their sensitivity level:

- For **related participants**, the caseworker can open the participant's name link to navigate to their Action View tab and review their demographic, case, and application details.
- For **shared cases**, the caseworker can view and access each case shared between the client and the related participant.

These links help the caseworker move quickly from a high-level overview of the client's relationships to detailed information when required.

Taking Action on Relationships

Depending on system configuration, the caseworker can add, edit, and delete relationships directly from the card. The available actions and their behavior vary depending on whether Person Evidence Case (PEC) is enabled.

When PEC is disabled (`curam.pec.enabled = NO`)

The following actions are available:

- **New Relationship** – available from the card-level action menu. Opens a modal where the caseworker can search for a participant and specify the relationship type and start date. On save, the card refreshes automatically.

- **Edit** – available at the row level when a record is selected. Opens a modal displaying the current relationship details. On save, the card refreshes automatically.
- **Delete** – available at the row level when a record is selected. A confirmation prompt is displayed. On confirmation, the card refreshes and the deleted record is removed.

When PEC is enabled (`curam.pec.enabled = YES`)

The following additional behaviors apply:

- **New relationships** are created with an In-Edit status and must be activated using Apply Changes before they become active.
- **Edit** is replaced by **Continue Editing** for records in In-Edit status. The record remains in In-Edit status until changes are applied. The latest version of the evidence is displayed when editing an active record. Records which are in In-Edit status display a visual indicator.
- **Delete** on an active relationship creates a pending deletion that must be applied using Apply Changes.
- **Discard** – available at the row level for In-Edit records. On confirmation,In-Edit changes are discarded. If the record had been activated before the edit, the most recent active version is restored. If the record was newly added and never activated, it is removed entirely.
- **Apply Changes** – available from the card-level action menu. Opens a modal listing all records currently in In-Edit status. On save, selected records are activated and the card refreshes. Available only when both Person Data Case (PDC) and PEC configurations are enabled.

Viewing Relationships from Multiple Sources

Administrative Settings (Display Controls)

System administrators can use the `Curam.ActionView.DisplayPersonRelationships` property to control whether person relationships are shown in the Cúram Action View – Relationships card.

- **Default Setting (YES):** Person-level relationships are always visible.
- **Restricted Setting (NO):** Person-level relationships are hidden to reduce clutter.

> **Note:** If the organization has only person relationships (and no other relationship types), those person relationships are always displayed in the Relationships card, regardless of this setting.

This configuration can be managed by an admin under **Cúram Action View – Action View settings**.

Viewing multiple relationships between participants

Caseworkers can capture relationships between people at the person level or within an Integrated Case. When a person is associated with multiple Integrated Cases, relationship information may be maintained independently as evidence on each case. Because relationships may not be synchronized across Integrated Cases, it is possible for different relationship values to exist between the same two participants — for example, one value recorded at the person level and another recorded on an Integrated Case, or differing values across two separate Integrated Cases.

The Relationships card consolidates these into a single view so the caseworker can identify and review any discrepancies.

When relationships are recorded from more than one source, the total number of sources is displayed for each related participant— for example, *3 relationships recorded*. Selecting the expand option opens a popover showing the details for each source.

The Relationship Card dynamically adjusts its header based on the relationship information:

- **Single value:** If the relationship is defined only once, the standard quick-scan view is displayed.
- **Multiple values:** If the relationship is defined more than once at the person level and/or at the case level, the total number of sources (e.g., *3 relationships recorded*) is displayed below the participant's name, alongside an expand arrow.

Clicking the expand button opens a popover window showing the specific details for each source:

- **Source:** Shows exactly where the data originates. People are listed first by their full name (e.g., *Maria Hernandez (Participant)*). Relevant cases follow in alphabetical order, showing the case name and reference number as a clickable link that takes you straight to that case's homepage (e.g., *Income Support – 272*). In this release, the below cases and evidence types are available to be displayed:

Evidence	Sources
Member Relationship	Insurance Affordability Application Case, Insurance Affordability Integrated Case
Household Relationship	Income Support Integrated Case
Relationships	Person (PDC On)

- **Relationship Type:** Displays the relationship type, such as *Spouse*.
- **Start Date:** Displays the date the relationship officially began.

To protect data integrity, the system automatically hides certain action buttons (like **Apply Changes**, **New Relationship**, **Edit**, or **Delete**) under the following conditions:

- If the relationship data is being pulled from Income Support or Insurance Affordability sources, you cannot manually add, edit, or delete them from this card — these actions are hidden, so the user can navigate to each source, review and then update the information. The card only displays information from **active** cases and **active** evidence.
- The **Edit** and **Delete** options are hidden if the same two people have more than one relationship type defined between them.

Sensitivity and Access Control

Relationships may involve participants with sensitivity levels that differ from the caseworker's. The card dynamically adjusts what is visible to ensure privacy and compliance.

A caseworker can only access a participant's details if the caseworker's sensitivity level is equal to or higher than the participant's sensitivity level.

When access is restricted

When the caseworker's sensitivity level is lower than a related participant's level:

- The participant's name is displayed as plain text rather than a navigable link.
- Age, gender, and shared cases are hidden.
- In the expanded view, a message is displayed: *You do not have access to this client's details.*
- A sensitivity indicator icon is displayed next to the participant's name when the record is collapsed.
- The relationship type and start date remain visible regardless of sensitivity.
- Edit and delete actions are not available for restricted participants.

Configuration variations

Participant Data Case (PDC) is an existing case type that manages person-level data as evidence. PDC can be enabled or disabled through the below listed configuration. Depending on whether PDC is enabled or disabled, the data that is displayed on the relationships card changes.

- **PDC enabled (`curam.evidence.pdc.personenabled = YES`)** – restricted participants are displayed with limited information as described above, and the sensitivity indicator is shown.
- **PDC disabled (`curam.evidence.pdc.personenabled = NO`)** – participants with higher sensitivity than the caseworker are not displayed on the card at all.

2.3 Applications

The Applications card in Cúram Action View provides a consolidated view of a client's benefit applications and requested programs. From a single location, the caseworker can see which applications have been submitted, which are in progress, any factors that are blocking progress, and the next appropriate actions, without leaving Action View.

What caseworkers can do

Using the Applications card, the caseworker can:

- View the client's current applications, with the most recently created applications displayed first.
- Open and review an application to determine who is included, which programs were requested, and the current stage of the application.
- Identify common blockers, such as outstanding issues or items requiring verification.
- Access the application document (PDF), when available, to support review and follow-up.
- View the progress of each application, using a visual representation and timeline indicators.
- Perform context-appropriate actions directly from the card, where permitted.
- Access historical applications, with the most recently closed displayed first.

Application Card Views

Quick scan (collapsed view)

The initial, collapsed view is intended for rapid assessment of applications. Each application entry provides a concise summary that allows the caseworker to determine:

- which applications are most recent
- which applications are actively progressing
- which applications may be stalled or require attention

Review in Context (expanded view)

When the caseworker expands an application, additional information is displayed to support day-to-day decisions, including:

- **Participants included on the application** – all participants linked to the application.
- **Requested programs and status** – each program requested, and its status (for example, pending, withdrawn, approved, or denied).
- **Application form** – a link to the application form (PDF), when available.

Application Progress Indicator (within expanded view)

When the caseworker expands an active application, an Application Progress cluster is displayed. This cluster contains a vertical step graph that tracks the application's progression through its lifecycle, showing completed, in-progress, and upcoming steps at a glance.

The progress indicator is displayed only for active applications – that is, applications that are not currently closed or disposed.

The graph represents the application's current position in its lifecycle at a single point in time. It is not a full historical record of every status change. If an application returns to a previous stage – for example, moving from Ready for Determination back to Submitted – the graph returns to that step, the date is updated to reflect the most recent occurrence, and all subsequent steps are reset to projected.

Visual indicators

Each step on the graph uses a distinct visual indicator to communicate its status:

- **Completed** – a filled circle with a check mark indicates that the step has occurred.
- **In progress** – a half circle indicates that the step is currently underway.
- **Projected** – an empty circle indicates that the step has not yet occurred but is expected as part of the standard lifecycle.
- **Stalled** – a circle with an alert indicator signals that the application has stalled at this step, for example, because of a submission failure or an authorization failure.

Lifecycle steps

- **Started**
  Not associated with any particular status. This step is always displayed regardless of the current application status. It shows the date the application was initiated. If the application form is still being worked on, the last saved date is also shown beneath the start date.

- **Submitted**
  Associated with the 'Open' or 'Submitted', 'Submit Failed', and 'Submitting' statuses. Indicates that the application has been submitted or submission was attempted. If the application is currently being submitted, this step appears as in progress. If submission has failed, this step is shown with the stalled indicator. When an application is resubmitted – for example, after a program is withdrawn and reopened – the existing Submitted step is updated with the new date rather than a new step being added.
- **Ready for Determination**
  Associated with the 'Ready for Determination' status. Displayed only when this step is configured for the application type. Out of the box, this applies to Income Support applications. Shows the date the application was marked ready for determination.
- **Closed / Disposed**
  Associated with the 'Closed', 'Disposed', 'Open', 'Ready for Determination', 'Authorization Failed', and 'Authorization in Progress' statuses. This is the final step in the lifecycle. This step appears as in progress when at least one program has been actioned but not all. For this event, the individual program names and their current statuses are listed beneath the step – for example, Food Assistance: Denied or Cash Assistance: Pending. Once all programs have been actioned, the application moves to the closed applications list and becomes read-only — no further actions or updates are available.

See section 'Taking Action on Applications' for the actions that are available during each step.

Taking Action on Applications

The actions available at each step depend on the application's configuration and its current status. The Applications card displays only the actions relevant to the application and its programs at that point in the lifecycle.

Actions available during the Started step

During this first step, the application has been initiated but not yet submitted. The following actions may be available:

- **Resume** – Continue the application from where it was last saved. Available when the application was started by a caseworker and saved but not yet submitted.
- **Submit** – Submit the application without leaving the Applications card. Available when submission of incomplete or partially completed applications is permitted.
- **Delete** – Remove the application if it is no longer required. Available only when the application has not yet been submitted.

These actions are also available when submission has previously failed (for example, where the application is in a Submit Failed state), allowing the caseworker to retry submission, resume processing, or delete the application as appropriate.

Common actions

The following actions are available across the remaining lifecycle steps (Submitted, Ready for Determination, and Closed/Disposed while the application is still in progress):

- **Check eligibility** – Evaluate eligibility for the requested programs.

- **Resolve prospect applicants** – Resolve unregistered participants before continuing processing. Only available if there are unregistered participants.
- **Add program** – Add additional programs to the application.
- **Authorize** – Approve or deny the programs associated with the application. Only available after eligibility has been checked.
- **Program-level actions** – Withdraw, reopen, deny, or authorize an individual program, depending on the program type.

Actions available during the Submitted step

Once the application has been submitted, the common actions listed above become available. In addition, the following action is available during this step:

- **Ready for Determination** – Moves the application from Submitted to Ready for Determination. Not all applications require this step; it is a configurable step.

Actions available during the Ready for Determination step

When an application reaches the Ready for Determination step, the common actions listed above remain available. There are no additional step-specific actions at this point.

Actions available during the Closed / Disposed step

While at least one program has been actioned but not all programs have reached a final status, the application is considered in progress within the Closed / Disposed step. The common actions listed above remain available during this period.

Once all programs have been actioned, the application moves to the closed applications list and becomes read-only — no further actions or updates are available.

> **Note:** When the application is in a temporary processing state (for example, Submitting), actions will not be available until processing completes.

Navigating from Applications

The application name and reference number displayed on each application entry is a clickable link. Selecting it navigates the caseworker directly to the application home page, where they can access the full application record.

Managing Work

The Applications card highlights common reasons an application may not be progressing, such as:

- outstanding issues
- items that still require verification
- information currently being edited

These indicators help the caseworker identify the next actions required to move the application forward and reduce unnecessary delays. These indicators are also links allowing the caseworker to navigate directly to the relevant page to take further action.

Sensitivity and Access Control

The Applications card respects the sensitivity levels configured for participants and users. A caseworker can only see details of applications they have permission to view. This ensures that sensitive client information is protected and that caseworkers only see what they are permitted to see.

Viewing applications

Sensitivity is specified at both the participant level and the user level. For each application, the caseworker's sensitivity must be equal to or greater than the sensitivity of the primary client on the application.

- If the caseworker does not have the required sensitivity level, the application is not displayed within the Applications card. No details of the application are visible.
- If the primary client of the application is not sensitive, but another member on the application is, the application can still be viewed within the Applications card. The sensitive member is included in the number of clients displayed for the application, and their name and age are included in the list of members.

Actions on applications with sensitive members

When an application includes a sensitive member applicant, some actions are restricted. The following actions remain available:

- Resume
- Submit
- Delete
- Ready for Determination
- Check Eligibility
- Add Program
- Program-level actions (Withdraw Program, Re-open Program)

The following cannot be carried out:

- Link to Issues
- Resolve Prospects (where the prospect is sensitive)
- Link to In Edit evidence
- Link to Items to Verify
- Authorize (including Approve, Decline, Deny, and Withdraw for platform programs)

2.4 Cases and Entitlement

The Cases and Entitlement card in Cúram Action View provides a consolidated view of a client's integrated cases, product deliveries, entitlements, and payment information. From a single

location, the caseworker can view which integrated cases exist, which product deliveries are associated with each case, who the case members are, what the person is entitled to receive and what payments are expected or have been made – without leaving Action View.

The card is designed to support quick awareness of case status, membership, entitlements, and payments while allowing the caseworker to drill into more detail when required.

What caseworkers can do

Using the Cases and Entitlement card, the caseworker can:

- View all integrated cases (open and closed) associated with a client, and expand any case to view its product deliveries, case members, entitlement amounts and periods, case owner, and nominee and component details.
- Navigate to related records — use inline links to go directly to integrated cases, product deliveries, items to verify, evidence in edit, and outstanding issues.
- Navigate to further details with a link to payment and determination history for each benefit.
- Take action on any outstanding work using links to evidence issues, evidence in-edit or outstanding verifications.
- Identify the primary client on both the integrated case and each product delivery.
- View case owner information, entitlement amounts and period for each product delivery.
- View a detailed breakdown of both the expected next payments and the previous payment. This is grouped by nominee and delivery method, and lists the components payable along with any underpayments or deductions applied.

Cases and Entitlement Card Views

Quick scan (collapsed view)

The initial, collapsed view is intended to provide a high level overview. Each integrated case entry is displayed as a separate entry on the card and provides a concise summary that allows the caseworker to determine:

- which cases are currently open or closed
- how many product deliveries are associated with each case
- who the case members are at a glance

Review in context (expanded view)

When the caseworker expands an integrated case, additional detail is displayed, including case members and associated product deliveries. Expanding a product delivery reveals entitlement details, case owner, nominee details, and payment information.

Each of these areas is described in full in the sections that follow.

Navigating from Cases and Entitlement

The integrated cases view provides links as follows:

- **Integrated Cases** – navigates to the integrated case record, where the caseworker can view case-level information, members, status, and activity.
- **Items to Verify** – navigates to outstanding verification requirements for the case.
- **Evidence in edit** – navigates to evidence currently being edited for the case, allowing the caseworker to review or complete changes.
- **Verifications** – navigates to the verifications area for the case, where the caseworker can review verification status and take action.

The expanded product delivery provides links that allow the caseworker to navigate directly to related records for further review or action. The following links are available:

- **Product Deliveries** – navigates to the product delivery record, where the caseworker can view entitlement, nominee, and delivery details.
- **Payments** – navigates to a list of payment records and detailed payment information.
- **Determinations history** – navigates to a list of eligibility and entitlement determinations.

Integrated Cases

Integrated Cases are displayed with the integrated case name and reference number. The number of associated product deliveries and the case start date are also displayed.

When the caseworker views the Closed integrated cases tab, the case end date is displayed.

The integrated case is rendered as a navigable link. Selecting the case name navigates directly to the integrated case record.

Integrated case members

When an integrated case is expanded, the card displays the members associated with the case. Member information includes the full name, age, and primary client indicator.

The primary client is always displayed first in the members list and is clearly identified with a *(Primary)* label shown in brackets and italics. Other members appear after the primary client, ordered alphabetically by first name.

Managing Work

The integrated case may highlight common actions that may exist on a case that require action such as:

- outstanding issues
- items that still require verification
- information currently being edited

These indicators help the caseworker identify the next actions required to move the case forward and reduce unnecessary delays. These indicators are also links allowing the caseworker to navigate directly to the relevant page to take further action.

Product Deliveries

Product deliveries are listed under their associated integrated case. Each product delivery includes the program name, coverage type (where applicable), and reference number.

When a program has a coverage type, the product delivery name follows a consistent format: Program name – Coverage type – Reference number. Both the program name and coverage type are displayed, and the full entry is rendered as a navigable link. If the program name and coverage type are the same, the name is displayed once to avoid duplication – for example, *Insurance Assistance – 1234*.

Selecting the product delivery name navigates directly to the product delivery record.

Each product delivery will include the case status, start date, end date (if applicable), members and payments. The expanded view displays the entitlement details, nominees and case owner.

Across all sections of the expanded product delivery view, a dash is displayed when details are not yet available.

Product delivery members

Each product delivery displays the members associated with that program. The primary client is identified with the same *(Primary)* label and is listed first. Other members appear after the primary client, ordered alphabetically by first name.

Entitlement Details

When a product delivery is expanded, additional entitlement information is shown. The level of detail displayed depends on the current status of the product delivery.

Entitlement amount and period

For product deliveries with an active, closed, pending closure, or suspended status, the entitlement details include:

- **Entitlement amount** – the currency symbol and amount, displayed in bold.
- **Entitlement period** – the start and end dates of the entitlement period, shown in brackets.

When a product delivery has a status of open, submitted, or approved, entitlement details are not yet available and are not displayed.

Reopened product deliveries

When a product delivery has been closed and then reopened, the entitlement details that applied at the time the case was closed continue to be displayed. This ensures the caseworker retains visibility of the most recent entitlement information, even though the case status has returned to open.

Case Owner

The case owner is always displayed, regardless of the product delivery status. When the case owner is updated, the change is reflected on the card.

Nominee Details

The expanded view displays nominee and component details under the heading Nominees.

Nominee details are shown when the product delivery status is active, closed, pending closure, or suspended:

- Where there is one component, the nominee name and component are displayed — for example, *Maria Hernandez, Income Assistance*.
- Where there are multiple nominees or components, each nominee appears. If a nominee is eligible for more than one component, the name is shown once with comma-separated components listed — for example, *Maria Hernandez, Income Assistance* and *Juan Hernandez, Medical Assistance, Rental Allowance*.

When a product delivery has been closed and then reopened, the nominee details from the time of closure continue to be displayed, consistent with the behaviour described in Reopened product deliveries above.

Payment Information

Expected next payment

The card displays the total expected amount and date for the next payment. If multiple payments fall on the same date, amounts are combined into a single total.

Previous payments

The card displays the total amount paid and date for the most recent payment, where available.

Additional payment information

From the expanded product delivery, caseworkers can select:

- View Payments - when selected the caseworker can view the list of payment transactions for the product delivery.
- View Determination history - when selected the caseworker can view the list of decisions for the product delivery.
- Payment information icon - a blue information icon appears to the left of the payment amount when payment information is available. Clicking it opens a popover headed Expected Next Payment or Previous Payment accordingly. The popover closes when clicking anywhere outside it.

Payment information icon
Payment Popover

The popover provides a detailed breakdown of payment information, organized by nominee and delivery method.

Structure

Payment details are grouped by nominee and delivery method, with each group containing:

- **Nominee and delivery method** – displayed as a subheading (e.g., *Maria Hernandez, by check*)
- **Payment components** – name, currency, amount, and coverage period
- **Deductions** – name and deducted amount
- **Benefit underpayments** – amount with the label *(One-off Payment)*
- **Sub-total** – total for that nominee/delivery method group

The **total payment amount** is displayed at the bottom, calculated by summing all sub-totals.

Multiple nominees and delivery methods

Each nominee and delivery method combination is displayed as a separate group. For example, one nominee with two delivery methods appears as two groups (*Maria Hernandez, by check* and *Maria Hernandez, by EFT*).

How totals are calculated

Each sub-total = component amount + underpayments − deductions.

The total payment amount = sum of all sub-totals.

The popover adjusts its size to accommodate the content displayed.

Currency and Localization

The Cases and Entitlement card supports localized currency display across all payment and entitlement amounts, including expected next payments, previous payments, the payment popover, and entitlement amounts in the expanded product delivery view.

Currency formatting is determined by the organization's configuration and the caseworker's locale. The currency symbol, its placement relative to the amount, and formatting rules for decimal and thousands separators are all applied automatically. This ensures that amounts are displayed in a consistent and familiar format regardless of region.

Sensitivity and Access Control

The Cases and Entitlement card respects the sensitivity levels configured for participants and users. A caseworker can only see details of cases they have permission to view. This ensures that sensitive client information is protected and that caseworkers only see what they are permitted to see.

Viewing cases

Sensitivity is specified at both the participant level and the user level.

- For each case, the caseworker's sensitivity must be equal to or greater than the sensitivity of the primary client on the application. If the caseworker does not have the required sensitivity level, the case is not displayed within the Cases and Entitlement card. No details of the case are visible.
- If the primary client of the case is not sensitive, but another member on the case is, the case can still be viewed within the Cases and Entitlement card. The sensitive member is displayed, and their name and age are included in the list of members.

2.5 Recent Activity

The Recent Activity card in Cúram Action View provides a single, chronological timeline of key client-related events. It brings together recent activity across a client's applications, cases, and other related records, helping caseworkers quickly understand what changed, when it happened, and how events connect within the client's overall journey.

What caseworkers can do

Using the Recent Activity card, the caseworker can:

- Review client-related activity in chronological order.
- See the most recent updates captured in notes, across all of the client's cases and applications.
- Understand how activity across people, applications, and cases impacts the client's journey.
- Navigate directly to underlying records when more detail is required.
- Filter the timeline to focus on specific time periods.

Recent Activity Card Views

Timeline overview

The card displays events in a unified timeline, ordered with the most recent activity first.

Event types and layout

The Recent Activity timeline currently includes the following event types:

- **Person registration** – when a person is first registered or created in the system.
- **Notes** – a consolidated view of all note activity (such as notes created, edited, or deleted) across all of the client's applications and cases, shown in one place.

Each event entry follows a consistent layout:

- **Event headline** – a concise summary of the event, focused on what changed and, where applicable, who performed the action.
- **Event details** – optional supporting information, such as the related application or case, program details, or in the case of notes, a preview of the note text.

- **Date and time** – when the event occurred. To support scanning longer histories, events are grouped by month and year. As the caseworker scrolls, the current month-year label remains visible and updates automatically when a new month is reached.

Navigating from Recent Activity

From the Recent Activity card, the caseworker can navigate directly to related records. Depending on the event, this may include:

- For **person registration events**, the caseworker can open the person record and view additional demographic or identifying details.
- For **note events**, the caseworker can open the full note and, where applicable, navigate to the related application or case.

These links help the caseworker move quickly from a high-level overview to detailed information when required.

Taking Action from Recent Activity

Some events support simple follow-up actions directly from the card.

Notes

For note-related events, the caseworker may be able to:

- View the note.
- Edit the note.
- Delete the note.

These actions allow quick updates to notes without navigating to another page. Other event types are read-only on the Recent Activity card, with related actions provided elsewhere in Action View.

Filtering and Result Awareness

The Recent Activity card supports filtering so that the caseworker can focus on the activity most relevant to their current task.

Using filters, the caseworker can:

- Show only specific types of events (for example, notes or person registration).
- Limit the timeline to a defined time period.

When filters are applied:

- The timeline updates immediately.
- An indication of how many events are currently displayed is shown.
- Chronological ordering is preserved.

Filtering affects visibility only; it does not change the underlying events.

Empty and Minimum States

The Recent Activity card is designed to always provide a useful starting point:

- The card is never presented as completely empty.
- At a minimum, it includes the person registration event to anchor the timeline.

This ensures that the caseworker always has a clear entry point for understanding the client's history.

Sensitivity and Access Control

The Recent Activity card respects the sensitivity levels configured for participants, users, and individual notes. A caseworker can only see details of events they have permission to view. This ensures that sensitive client information is protected and that caseworkers only see what they are permitted to see.

Viewing note events on the timeline

Sensitivity is checked at two levels for note events: the **participant level** and the **note level**.

Participant-level sensitivity

For each note, the caseworker's sensitivity must be equal to or greater than the sensitivity of the primary client on the related case or application.

- **Case notes:** If the caseworker does not have the required sensitivity for the primary client on the case, no details of that case are visible — including any case notes. Nothing is plotted on the timeline for these notes.
- **Application notes:** If the caseworker does not have the required sensitivity for the primary client on the application, no details of that application are visible — including any notes. Nothing is plotted on the timeline for these notes.
- If the primary client is not sensitive, but another member on the case or application is, the note can still be viewed on the timeline.

Note-level sensitivity

In addition to participant-level sensitivity, individual notes can have their own sensitivity level. The caseworker's sensitivity must be equal to or greater than the sensitivity of the note itself.

If the caseworker does not have the required sensitivity to view a specific note, the note event is still plotted on the timeline based on the date and time the event occurred. The note subject is visible, but the note text and priority are not displayed and the note actions (view, edit, delete) are not available. Instead, an alert is shown with the message: *"You do not have access to this note."*

Chapter 3 General Behaviors

Cúram Action View includes a number of cross-cutting behaviors that apply across multiple cards and actions rather than belonging to a single feature.

The topics in this section describe shared behaviors and interaction patterns that are common across Cúram Action View. These behaviors — such as toast notifications — are not specific to any individual card or action, but instead define how the workspace responds to caseworker activity, communicates outcomes, and keeps information current.

Understanding these behaviors helps caseworkers know what to expect as they work within Action View, and helps organizations plan any customizations that may affect how the workspace responds to user interactions.

3.1 Toast Notifications

Toast notifications provide immediate, non-intrusive feedback when a caseworker completes an action from Action View. The notification appears temporarily in the header of the card where the action was triggered, confirming the outcome without requiring the caseworker to navigate away from the current page or refresh the view.

How Toast Notifications Work

When a caseworker performs an action, a toast notification is displayed in the header of the card where the action was initiated. The notification confirms the outcome of the action and disappears automatically after the display duration.

- Notifications display for 12 seconds, and screen readers announce their content.
- Each toast notification is displayed independently. If the caseworker performs actions across multiple cards, each card displays its own toast notification without dismissing, overriding, or delaying notifications on other cards.
- Only the card where the action was triggered is refreshed to display the notification.

Toast Notification States

Toast notifications indicate the outcome of an action using the following states:

- **Success** – a checkmark in a green circle confirms that the action completed successfully.
- **Error** – a red circle with a diagonal slash indicates that the action was not completed.
- **Loading** – a loading indicator displays while the action is being processed.

Where Toast Notifications Are Used

Toast notifications are currently used on the Cúram Action View screen for actions where the caseworker needs confirmation that an action has completed. The notification is displayed in the header of the card where the action was triggered. Examples of toast notifications include:

- **Quick Actions Card – New Note**: When a caseworker creates a note using the New Note quick action, a toast notification is displayed in the header of the Quick Actions card.

- **Applications Card – Delete Application**: When a caseworker deletes an application from the Applications card, a toast notification is displayed in the header of the Applications card.

Chapter 4 Cúram Action View Technical Overview

This chapter provides a technical overview of Cúram Action View, covering its architecture, development and production environments, and key design considerations.

Cúram Action View follows Cúram's component-based architecture. The Cúram Action View installer introduces new EJBServer and webclient components that contain the content required to run Cúram Action View.

4.1 Versioning and Compatibility

Cúram Action View releases differ from classic Cúram releases. The versions use **Semantic Versioning** (major, minor, patch), and each version is compatible with all product lines from version 8 forward. For example, you can install the Cúram Action View v1.0.0 installer on top of v8.0.1, v8.0.2, v8.0.3, v8.1.0, and later versions, making the feature set available to all v8+ installations.

4.2 Release Cycle

Cúram Action View is released monthly, so it can take advantage of fast iteration on enhancements and new features. The upgrade process is lightweight to support easy adoption of the latest content into a production environment.

4.3 Development Environment

Cúram Action View uses a different development approach from the classic UIM/Facade style client. Developers build the user interface with ReactJS components based on the Carbon Design System to align with the Cúram style guidelines. A new lightweight development environment supports rapid development of new cards, using Storybook as a sandbox to fully develop the user interface independently from the Cúram development environment.

4.4 Production Environment

Cúram Action View uses REST and GraphQL APIs to serve data. You must deploy the Cúram REST client to serve requests.

When planning your deployment, consider the following:

- **REST client deployment** — If you have not already deployed the REST client, plan for this as part of your Cúram Action View deployment.
- **Sizing** — Assess whether your current REST client sizing will meet the additional demand generated by the Cúram Action View page.
- **Workload isolation** — If your current REST client supports the Citizen Portal, consider deploying a separate client to isolate workloads and prevent impacts to either feature.

4.5 Page Architecture

Cúram Action View uses “cards” to group business functionality. For example, a Cases and Entitlement card provides a comprehensive view of Integrated Cases and Product Deliveries and the benefits information related to these case types.

Cards, and the data they load, are decoupled from each other, so you can add and remove them from the page without impacting other cards. Each card gathers its own data. A global communication system allows cards to react to actions from other cards without tightly coupling them. For example, creating a New Note can notify the page of this action, and the Recent Activity card can refresh its content to display the newly added note.

4.6 Data Transfer (REST & GraphQL)

Cúram Action View supports REST and GraphQL APIs to serve data to cards.

All custom card integrations must use REST APIs. GraphQL is restricted to internal, product-managed capabilities only (e.g Cases and Entitlement card). New cards can use custom-built REST APIs or reuse the product-provided REST APIs.

4.7 Performance Characteristics and Operational Guidance

The Action View consolidates information from across the application onto a single page, giving caseworkers a unified starting point for a person or case. This consolidation is deliberate and valuable - from a performance perspective it introduces a different profile from the rest of the application. This section explains that profile, sets out how to reason about it in the context of overall system performance, and describes the practices that keep it healthy.

Why the Action View behaves differently

Most pages in the application are transactional. Each performs a single focused task, reads a narrow slice of data, and costs broadly the same amount each time it is rendered. The Action View is aggregative. Its purpose is to bring together, in one place, information that is otherwise distributed across many separate pages. A single render therefore performs the work of many pages at once, drawing on a wide range of underlying data.

The page is built as a React JavaScript front end, and this shapes how it retrieves its data. Rather than being assembled in full on the server and delivered as a single finished page, the Action View is rendered in the browser and populated predominantly through REST services, with each card retrieving its own data via REST calls. This is what enables the enhanced user experience the page is designed to provide: because cards fetch independently, the page can render its structure immediately and fill each card in as its data arrives, rather than making the user wait for the whole page to be ready at once. Cards can load in parallel, update without a full-page refresh, and respond to interaction on the client.

The trade-off is operational. A single page load fans out into multiple requests served by the REST application (the REST EAR) rather than running as one self-contained operation, so the page's responsiveness depends on the capacity of the REST tier as well as on the cost of the data it gathers.

To keep the page responsive despite this aggregative profile, it is designed to spread its cost rather than incur it all at once. Cards load their data independently rather than as a single blocking operation, so the page becomes usable as each card resolves and a single slow card does not hold up the rest. Additionally, data is fetched as it is needed to populate what is on screen rather than up front, and each card retrieves only the slice of data relevant to it following a progressive disclosure pattern.

Operational Guidance

Action View has been performance-tested using realistic caseworker workflows on Kubernetes infrastructure representative of a production environment. Under these conditions the page proved stable and reliable throughout a sustained load test. As each environment and workload differs, the following practices protect against performance issues on Action View.

- **Test the page as its own workload.** Benchmark it separately, using production-representative data that includes the heaviest cases and persons in the system, and at realistic levels of concurrency.
- **Govern content additions to the page.** Every section or panel configured into Action View adds to the cost of every render. Treat additions to Action View as spending against a performance budget and review proposed changes on that basis rather than assuming additional content is free.
- **Tune which cards are displayed.** Display only the cards your caseworkers genuinely need. Action View 1.0.0 supports turning cards on and off at the page level. Future releases will extend this to individual-card tuning, limiting the volume of data a given card retrieves, and so reducing its contribution to each render.
- **Consider your REST Application Deployment.** The REST Application can be deployed alongside the Cúram Application meaning resources are shared across both workloads. The REST Application could be further isolated to separate the workload of Cúram Action View from the existing caseworker workload if necessary.
- **Isolate workflows for different users.** It is recommended that if you deploy the REST Application to serve the Citizen Portal (Citizen Engagement) that a separate REST Application is used to serve the caseworker user requests, ensuring that spikes in load do not impact each user type.
- **Maintain the data layer.** Indexing and up-to-date statistics deliver a disproportionate benefit here precisely because the page is a read amplifier. Routine data-layer health work pays back first and most visibly on this page.

Summary

The Action View concentrates, into a single page, work that is otherwise spread across the application. Based on its function it has a different performance profile from other pages in the application. Sizing, testing and monitoring it on that basis — together with governing what is placed on it, tuning the data each card returns, and sizing the REST services that deliver it — keeps it performing well and protects the wider system from the load it can generate.

Chapter 5 Installation and Upgrade

This chapter provides instructions for installing or upgrading Cúram Action View in an existing Cúram environment.

5.1 Prerequisites

Cúram Action View requires the following prerequisites:

1. Cúram version 8.0.0.0 or higher.
2. Cúram Action View requires the Cúram REST Client. Follow the Cúram REST Guide for deployment instructions.

5.2 Installation Scenarios

Your installation will fall into 1 of 3 categories.

1. **New installation** — Adding Cúram Action View to a new Cúram installation that is not yet being used in production.
2. **Upgrade** — Adding Cúram Action View to an existing production deployment of Cúram.
3. **Cúram Action View upgrade** — Upgrading an existing production deployment of Cúram Action View to a newer version of Cúram Action View. For example, upgrading v1.0.0 to v2.0.0.

> **Note:** Do **not** use the standard Cúram upgrade guide or upgrade pack for Cúram Action View. This guide streamlines the process and removes unnecessary steps.
>
> Scenarios 2 and 3 (upgrading) are technical upgrades that may require database schema changes or data updates. Although the process is streamlined, plan accordingly.
>
> Your environment may be customised. Review all steps in the context of your target environment and consult support if compatibility is unclear.

5.3 Process Summary

Recommended: Test all steps in a non-production environment before applying them to production.

1. Complete the pre-installation steps.
2. Run the latest installer for both new installs and upgrades.
3. Complete the post-installation steps for each version you have not previously applied. For example, if upgrading from v4.0.0 to v7.0.0, run v5.0.0, v6.0.0, and v7.0.0 steps only.
4. Build and deploy the updated EAR files, including the REST Client.

Note: If deploying the REST Client for the first time use the CúramREST API Guide for guidance.

5.4 Installation Artifacts

- **Setup Guide** - this chapter.
- **Installer** - the executable installer file.
- ***curam-action-view-upgrade-pack*** - zip with versioned folders containing upgrade scripts.
  - **Post-install scripts** - remove obsolete files after install/upgrade.
  - **Upgrade SQL scripts** - update database schema/data for compatibility.

5.5 Installation Steps

Pre-installation

Back up any customisations to the following files:

- `EJBServer/project/config/deployment_packaging.xml`
- `EJBServer/project/config/datamanager_config.xml`

Identify and back up custom values for these environment variables (usually set in `SetEnvironment` or `CustomEnvironment` script files that are run during installation):

- `SERVER_COMPONENT_ORDER`
- `CLIENT_COMPONENT_ORDER`

After installation, compare your backed-up files with the post-install versions and reapply any customisations that were removed.

Installation

1. Extract the release zip to a temporary directory and change to that directory.
2. Run the installer:

```
java -jar "Curam Action View Development.jar"
```

3. Follow the wizard to complete installation against your Cúram Development environment.

Post-installation

1. Run **post-install** scripts (not SQL upgrade scripts) from the `curam-action-view-upgrade-pack` in version order. Run scripts for versions you may be skipping between your previous installed and current installed versions. For example, if upgrading from v1.0.0 to v4.0.0, post-install scripts should be run for v2.0.0, v3.0.0 and v4.0.0.
2. Enable the GraphQL end point.

- Set the property `curam.graphql.endpoint.enabled` to true to enable the GraphQL endpoint used by the Cases and Entitlement card.
- To apply this in your custom environment, add the following entry to your `Application.prx` file. For more information about `Application.prx` files, see the Cúram Server Developers Guide.

```
<property name="curam.graphql.endpoint.enabled" dynamic="no"
 constant="ENV_GRAPHQL_ENABLED">
  <type>BOOLEAN</type>
  <value>true</value>
  <default-value>false</default-value>
  <category>SECURITY</category>
  <locales>
    <locale language="en">
      <display-name>Enable GraphQL</display-name>
      <description>Setting that defines if the GraphQL endpoint URL is enabled
 or disabled.
      </description>
    </locale>
  </locales>
</property>
```

3. Update the link configuration **if required**.
   - The Action View contains hyperlinks and buttons that open other parts of the application — for example, links that open an integrated case tab or a modal to add a new note. If your project configuration or customisations introduce custom versions of these features, you will need to update the link targets.
   - Linking, and the different variants of links, are discussed in the customisation section of this guide, including information on how to customise the link target.
   - **For the initial installation** defer customising links. Continue with the steps below to deploy your development environment. Test the feature set for compatibility with the target environment customisations, and then use the customisation guide to fix any broken links.
4. Build the Cúram server:

```
cd EJBServer
./build server
```

5. Build the Cúram client:

```
cd webclient
./build client
```

6. Build the REST Client:

```
cd EJBServer
./build rest
```

7. Update navigation configuration in your database to add the Cúram Action View tab.
   - By default, the installer adds the Action View as the last tab on the Person page. For best usability, move it to the first position so that it opens first when you click a Person link. Refer to the Cúram Web Client Reference manual for more information on configuring your Application Views and working with tabs.

8. Insert new application configuration data:

```
cd EJBServer
./build inserttabconfiguration
```

9. **[Upgrade only]** Reinsert Application properties.

   - Each version of Cúram Action View may introduce new application properties. These properties are stored on the `PROPERTIES` table of the database.
   - If upgrading an existing deployment the `build database` target can not be used.
   - Run `build insertproperties` to re-insert all properties from the `Application.prx` files in the project's source control. For more information about `Application.prx` files, see the Cúram Server Developers Guide.
   - **Note:** The `insertproperties` command invokes the Properties Manager and reinserts **all** properties from the `PROPERTIES` and `PROPDESCRIPTION` tables that the `build database` target would normally insert. Before running this command, verify that your `Application.prx` files are up to date — any properties you added at runtime without a source-controlled equivalent will be lost. To identify differences, compare the output of `build extractproperties` against your production and previous development environments.

10. **[Upgrade only]** Run SQL upgrade scripts from the `curam-action-view-upgrade-pack`.

    - Use the SQL scripts provided in the `curam-action-view-upgrade-pack`.
    - Upgrade scripts may contain schema changes, but typically insert or update configuration data in existing tables, such as codetable entries.
    - Run the scripts in order of version.
    - Only run scripts that have not already been run by previous upgrades.
    - Some upgrades may include SQL scripts that remove data that is no longer required.

11. Deploy and test the Cúram Action View.

    - If deploying in a development environment, such as running the application from the Eclipse IDE, follow the documented deployment process in the *Cúram Development Environment Installation Guide*.
    - If deploying to an Application Server, follow the standard process documented in the *Cúram Deployment Guide for WebSphere Application Server* or *Cúram Deployment Guide for WebLogic Server* and the *Cúram REST API Guide* to build the EAR files and deploy them to the Application Server.
    - Once deployed, open the Cúram application, log in, and either register a new person or search for an existing one. Open the person tab.
    - If you configured everything correctly, the Action View loads in the main content area of the tab, below the context panel.
    - **Note:** Testing the features of Action View requires a case history to populate the cards with content. For example, the Applications card will require historic or in-progress Applications to exist before the card is populated with data.

Chapter 6 Customizing Cúram Action View

This chapter describes how to customize Cúram Action View, covering the data access layer, the user interface framework, and the tools available to build and validate your customizations.

You can add new cards to the page, remove existing cards, and reorder card positions. To customize Cúram Action View, you need expertise in Cúram API development for the data layer and in JavaScript and ReactJS for the user interface.

6.1 Customizing the Data Access Layer

Cúram Action View serves data to cards through REST APIs. Use the Cúram REST API Guide for advice on creating new APIs. You can reuse existing APIs listed under the Cúram Action View tag in Swagger.

> **Note:** You cannot customize or extend REST APIs. You must create a custom version of an API.

The following sections on customizing the user interface describe how you can use REST APIs to populate card data.

6.2 Customizing the User Interface

You customize Cúram Action View by using the `curam-customizable-ui-starter-pack` The `curam-customizable-ui` framework makes Cúram Action View customizable and extensible. It provides the architecture, cards, data-service hooks, stores, and integration patterns that support Action View. You can use it to:

- Reorder, hide, or disable grid cards.
- Add new cards, messages, and data-service hooks.
- Extend mock data.

The framework preserves upgrade compatibility while enabling customer-specific behavior.

This section introduces the Storybook documentation that ships with the `curam-customizable-ui-starter-pack`. It does not repeat the detailed instructions; instead, each entry gives a short introduction to a Storybook topic, followed by a link where the full guide, examples, and interactive references can be found.

Prerequisites

The detailed documentation lives in the starter-pack Storybook, and that is the source of truth your team should follow. Before using any of the guides below, install the `curam-customizable-ui-starter-pack` and run Storybook locally by following its installation

steps. Once Storybook is running, every page linked here is available from its left-hand navigation.

See more in Storybook:

- **1. Welcome** - Introduces the framework, the problem it solves, and the value it brings to Cúram Action View.
- **2. Using Storybook** - Explains how the Storybook content is organised and how teams use it to learn, build, and validate changes.
- **3. Build and Deploy** - Walks through installing, building, and deploying the starter-pack.

Foundations

The framework ships a fully functional default Cúram Action View and provides the tools you need to customize and extend it. Use the `curam-customizable-ui-starter-pack` as your development environment, where all of your customizations live, while the framework itself is consumed as a versioned dependency. You get the page layout, React components for each card, shared styling and design tokens, state management, data services, and internationalization out of the box. You can customize:

- **Grid order and visibility** — reorder or hide tiles.
- **Translations** — add or override internationalization messages.
- **Data-service hooks** — connect components to new APIs.
- **New components** — add new cards to the grid.
- **Mock data** — develop and test without a live back end.

Your changes reach the default UI through the framework registers, from installing the framework through to deploying your custom bundle. The separation between the framework and the starter-pack keeps your changes safe across upgrades.

A detailed walkthrough is available in Storybook under *4. Foundations — 4.1. General Overview*.

Grid Configuration System

You control tile composition, ordering, visibility, and extensibility across the grids through the Grid Configuration System. Use this file-based mechanism to define the configuration shape behind every grid. The configuration covers the tiles each grid declares, the column-based layout, disabled tiles, additional injected tiles, and heading levels. You can reorder tiles, hide built-in tiles, and append new ones. The independent Quick Actions grid uses its own registration function, so you configure each grid in isolation.

The Storybook starter-pack includes interactive examples under *4. Foundations — 4.2. Grid Configuration System*.

Data Services (GraphQL & REST)

The data-services layer decouples the UI components from the details of how data is fetched, supporting both GraphQL and REST through a consistent, hook-based interface so components stay transport-agnostic.

> **Note:** While `curam-customizable-ui` infrastructure supports GraphQL, it is not yet supported for use in custom built cards. REST APIs must be used for custom development.

The Data Services section describes how a request flows from a component, through a domain hook and an API module, down to the REST or GraphQL service and back as normalized, camelCase data. It outlines the available modules and the standard file layout for each one. It also covers the shared infrastructure they rely on and the conventions that keep responses consistent regardless of the back end.

For implementation details, see *4. Foundations — 4.3. Data Services (GraphQL & REST)* in Storybook.

Mock Data Service (MSW)

The Mock Data Service enables rapid development by removing the dependency on a running Cúram application. Using Mock Service Worker (MSW), the framework intercepts REST and GraphQL requests and returns fixture data, so UI features can be built and iterated on in a lightweight environment, frontend work can begin before back-end APIs exist, and Storybook stories and unit tests stay fully self-contained.

This section explains how the mock layer is structured, how it behaves in both the browser (Storybook and the dev server) and Node.js (Jest), the conventions each handler follows, and how to use and override handlers within unit tests.

A step-by-step guide is available in Storybook under *4. Foundations — 4.4. Mock Data Service (MSW)*.

Supported Customizations

Use the starter-pack as your development environment for customizing Cúram Action View. Apply your changes by calling framework registers with your customizations rather than modifying the framework files directly. The starter-pack organizes where you register customizations and build new components, services, stores, and mock data. It also identifies which folders are reserved. You can configure existing behavior by reordering or hiding tiles, or extend the UI with new functionality such as new tiles, translations, API hooks, mock data, and stores. Customization entry points activate when you import them.

Full details and worked examples are in Storybook under *5. Customizing the UI — 5.1. Overview*.

Modify Page Grid Order

To reorder the Action View grid from your starter-pack, register a `columns` configuration, which is merged on top of the default layout when the page renders. The column-based layout controls where tiles appear. Supplying `columns` replaces the entire default layout, so any omitted tile does not appear. Use the built-in tile identifiers safely, review the before-and-after examples, and use the interactive builder to generate the snippet.

The interactive builder and before-and-after examples are in Storybook under *5.2. Modify Page Grid Order*.

Disable Page Grid Card

To hide or disable built-in cards from the Action View grid, add their tile identifiers to a `disabledTiles` list in your grid customization config. Your disabled list is merged with any tiles the framework already disables. A hidden tile's slot is preserved so the rest of the layout does not reflow. Use this approach to hide a card that is not relevant to a particular deployment or to turn one off temporarily while a feature is still being built.

For the full configuration reference, see *5.3. Hide and Disable Page Grid Card* in Storybook.

Add a New Card to the Page Grid

To add a new React component to the Action View grid, register it under `additionalTiles` and place it in the layout. Create the tile component, register it with a unique identifier, control exactly where it appears, and forward any configuration it needs. Use the full additional-tile entry shape and complete example to combine new tiles, hidden tiles, and a custom column layout.

A complete worked example is available in Storybook under *5.4. Add a New Card to the Page Grid*.

Add New Internationalization Messages

This section shows how to add localized text to your starter-pack components by registering custom translations that the framework merges with its own at startup. It covers co-locating message definitions with the component that uses them, consuming those messages in the component, adding the strings to a locale file (including the pseudo-locale used for translation testing), and registering the locale map. It also explains how merging lets you override an existing framework message or introduce entirely new text for your own components.

The Storybook guide walks through each step under *5.5. Add New Internationalization Messages*.

Add New Data Service Hooks (Support New APIs)

This section explains how to connect a starter-pack component to a new API by adding a data-service hook, keeping fetch logic out of the components themselves. It walks through three steps: registering the API endpoint, writing a thin API function that calls the framework's REST or GraphQL service, and building a data-service hook. The hook manages loading, error, and data

states. It also cancels in-flight requests when a component unmounts. It then shows how to export the hook and consume it in a component.

For working code and hook patterns, see *5.6. Add New Data Service Hooks* in Storybook.

Add New Mock Data

This section shows how to add mock data for your new API endpoints in the starter-pack so features can be developed and tested without a live back end. It explains where to place a data file, how the code-generation step wires your data into the browser mock layer automatically, and how the framework selects between the generated handlers and its own defaults. It also describes how to override existing framework fixture data when you need different sample responses.

The code-generation workflow is documented in Storybook under *5.7. Add New Mock Data*.

Add New Stores

This section explains how to add shared global state for your starter-pack components using the framework's store utility, which provides provider-free, selector-based state management without wrapping the application in context. It walks through creating a store, exposing a stable actions hook, and subscribing to state with selectors. Components only re-render when their selected slice changes. It also covers how updates are merged and how named stores appear in developer tooling. The key rules are: always select, separate the actions hook, keep state flat, and use one store per domain.

Detailed patterns and developer-tooling tips are in Storybook under *5.8. Add New Stores*.

Feature-Specific Extensions — Quick Actions

These guides cover customizing one specific feature. In the current Storybook content, that feature is the Quick Actions tile, which has its own independent grid and its own registration function.

Reorder Quick Actions Grid

This section explains how to control the display order of the buttons in the Quick Actions tile using an `order` list. It describes how listed buttons appear first in the sequence you specify while any unlisted buttons follow in their original order, and how ordering works across both built-in and custom buttons so they can be arranged together freely.

The full ordering reference is in Storybook under *5.9.1.1. Reorder Quick Actions Grid*.

Disable a Quick Action

This section shows how to control button visibility in the Quick Actions tile. *I*t distinguishes between removing a button entirely and keeping it visible but non-interactive. Both lists are merged with the framework's defaults rather than replacing them. Keeping a button visible but inactive is useful when an action is only temporarily unavailable.

For the complete configuration options, see *5.9.1.2. Hide and Disable a Quick Action* in Storybook.

Add a New Quick Action

This section explains how to add a new button to the Quick Actions tile by registering an additional entry. It covers defining the button's label, optional icon, optional keyboard shortcut, and the action it triggers. Keep translatable labels in a dedicated messages module. It documents the full shape of a Quick Action entry and how ordering, hiding, and inactivating interact with the new button.

See more in Storybook under *5. Customizing the UI — 5.9. Feature-Specific Extensions — 5.9.1. Quick Actions List — 5.9.1.3. Add a New Quick Action*.

Utilities

This section introduces the CSS utility classes — single-purpose classes that apply a specific style directly to an element without custom CSS. It explains why they are useful for keeping design values such as spacing and color consistent across the project, advises using them sparingly so markup stays readable, and shows how to apply them through the `className` attribute.

See more in Storybook under *Utilities — 1. Introduction*.

Accessibility

This section covers the accessibility-focused utility classes, including the helper that hides an element visually while keeping it available to screen readers - useful, for example, for providing a text equivalent alongside a decorative icon.

See more in Storybook under *Utilities — 2. Accessibility*.

Typography

This section groups the text-related utilities: balanced line wrapping, start and end text alignment, capitalization, and the primary and secondary text colors. Each utility is shown with the class name to apply.

See more in Storybook under *Utilities — 3. Typography*.

Color

This section documents the support colors used to reflect system status - success, warning, error, information, and the minor and major caution states - along with the guidance that color should always be used with purpose and never as the only means of conveying information.

See more in Storybook under *Utilities — 4. Color*.

Spacing

This section describes the margin and padding utilities for laying out components, including the directional options and the vertical and horizontal shortcuts. It also points to the layout components to consider when you need uniform spacing for dynamic content.

See more in Storybook under *Utilities — 5. Spacing*.

Alignment

This section explains the vertical alignment utilities for inline and table-cell content - top, baseline, middle, bottom, text-top, text-bottom, sub, and super - and notes where these utilities apply and when to prefer a layout component instead.

See more in Storybook under *Utilities — 6. Alignment*.

Action View Tools

This section covers the interactive tools that ship with the Storybook starter-pack to help you configure and test your customizations.

Grid Configuration Builder

This section provides an interactive, drag-and-drop builder for generating the Action View grid customization configuration. You drag tiles between columns to design your layout, reorder them within a column, add or remove whole columns, and the corresponding `columns` snippet updates live so you can copy it straight into your customization file. It also summarises the configuration shape the builder produces.

See more in Storybook under *Action View — Grid Configuration Builder*.

Where to Go From Here

Together, these sections trace a complete path - from understanding the foundations the framework provides, to configuring and extending Cúram Action View, to the utilities and tools that refine the result. Treat this section as your map: a concise orientation to what is possible, with each link opening the full Storybook guide where the detailed instructions, working code, and interactive examples live. Start with the foundations, then follow the Storybook guides as your requirements grow. The starter-pack preserves your customizations across framework upgrades.

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.