Using PCore and PConnect Public APIs '24.2

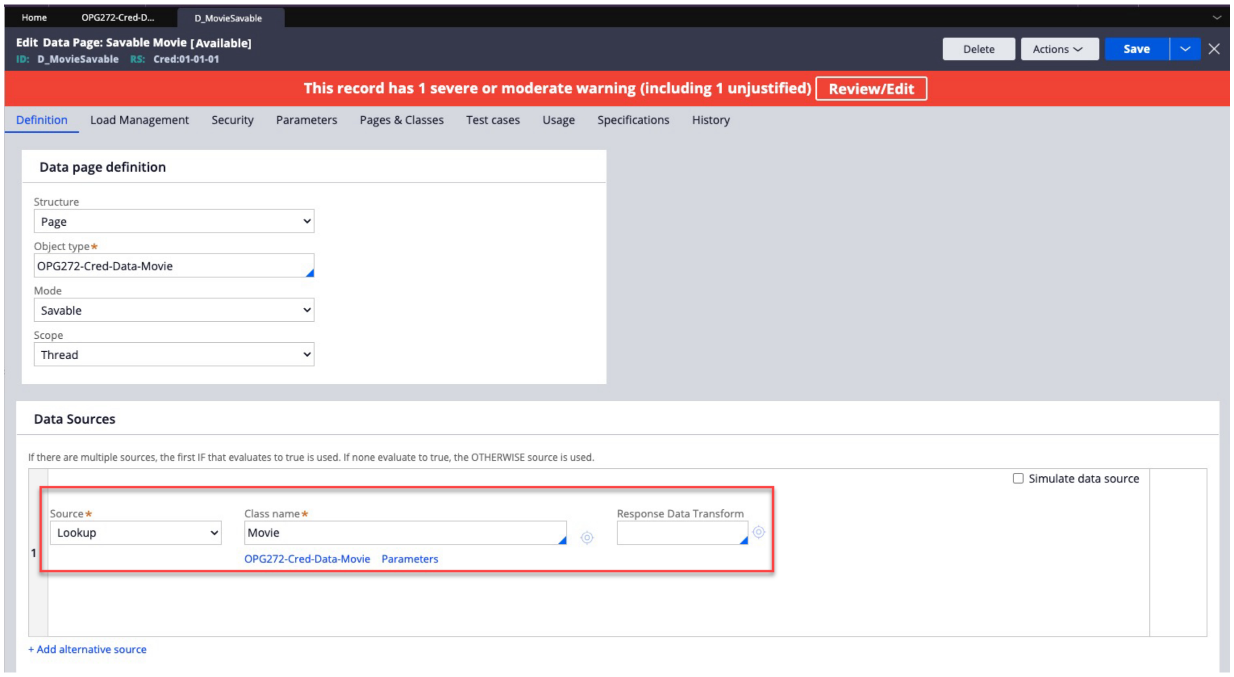
5 September 2025

CONTENTS

Invoking a savable data page

Store data in a data page by invoking a savable data page.

1. In the navigation pane of Dev Studio, click **Records** > **Data Model** > **Data Page**, and open a savable data page.
2. In the **Definition** tab of the data page, under **Data Sources**, enter the required data, and click **Save**.



Edit Data Page: Savable Movie [Available]
ID: D_MovieSavable RS: Cred:01-01-01

This record has 1 severe or moderate warning (including 1 unjustified)

Data page definition

Structure: Page
Object type*: OPG272-Cred-Data-Movie
Mode: Savable
Scope: Thread

Data Sources

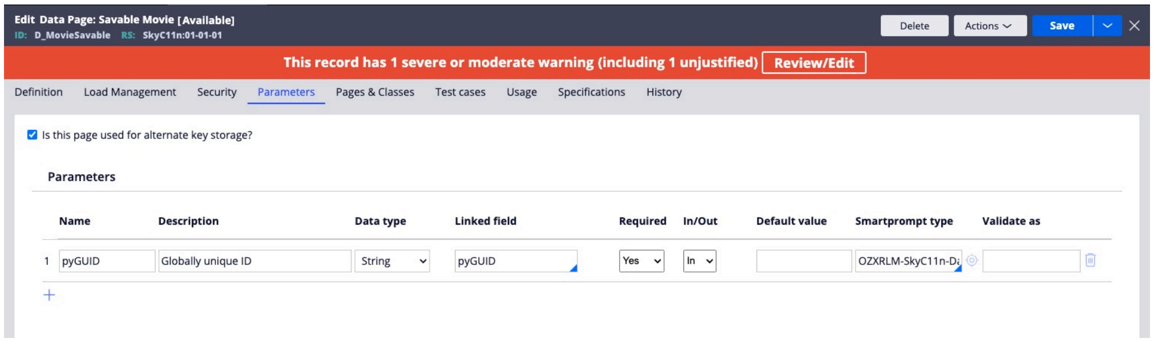
If there are multiple sources, the first IF that evaluates to true is used. If none evaluate to true, the OTHERWISE source is used.

☐ Simulate data source

1 Source*: Lookup | Class name*: Movie | Response Data Transform:
OPG272-Cred-Data-Movie Parameters

Data sources

3. In the **Parameters** tab of the data page, add the required parameters.
4. Select the **Is this page used for alternate key storage?** checkbox.
   - If the data object is configured with an auto-generated key, enter the `pyGUID` key in the **Linked field** text box, and click **Save**.



Edit Data Page: Savable Movie [Available]
ID: D_MovieSavable RS: SkyC11n:01-01-01

This record has 1 severe or moderate warning (including 1 unjustified)

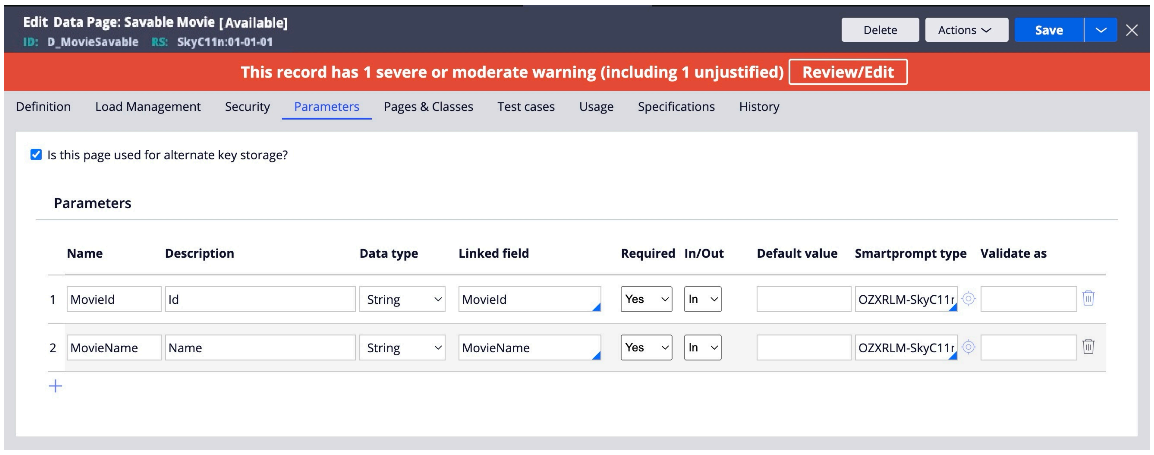
☑ Is this page used for alternate key storage?

Parameters

	Name	Description	Data type	Linked field	Required	In/Out	Default value	Smartprompt type	Validate as
1	pyGUID	Globally unique ID	String	pyGUID	Yes	In		OZXRLM-SkyC11n-Da	

Alternate key details - auto-generated key

- If the data object is configured with custom keys, enter the names of the custom keys in the **Linked field** text box, and click **Save**.



Edit Data Page: Savable Movie [Available]
ID: D_MovieSavable RS: SkyC11n:01-01-01

This record has 1 severe or moderate warning (including 1 unjustified)

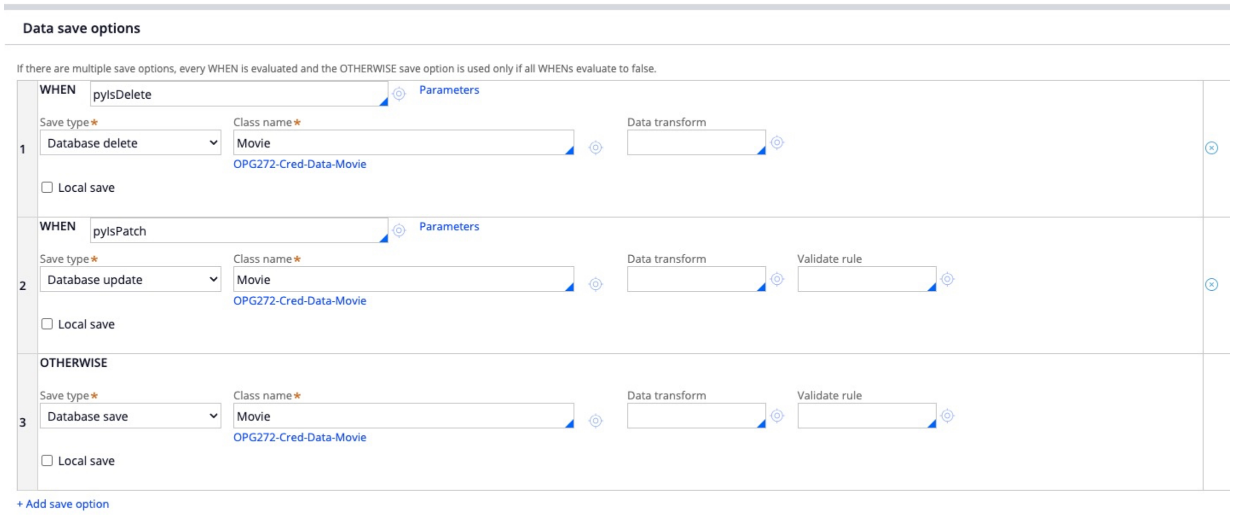
☑ Is this page used for alternate key storage?

Parameters

	Name	Description	Data type	Linked field	Required	In/Out	Default value	Smartprompt type	Validate as
1	MovieId	Id	String	MovieId	Yes	In		OZXRLM-SkyC11r	
2	MovieName	Name	String	MovieName	Yes	In		OZXRLM-SkyC11r	

Alternate key details - custom keys

5. In the **Definition** tab of the data page, under **Data save options**, configure the save plan in the order shown in the below figure.



Data save options

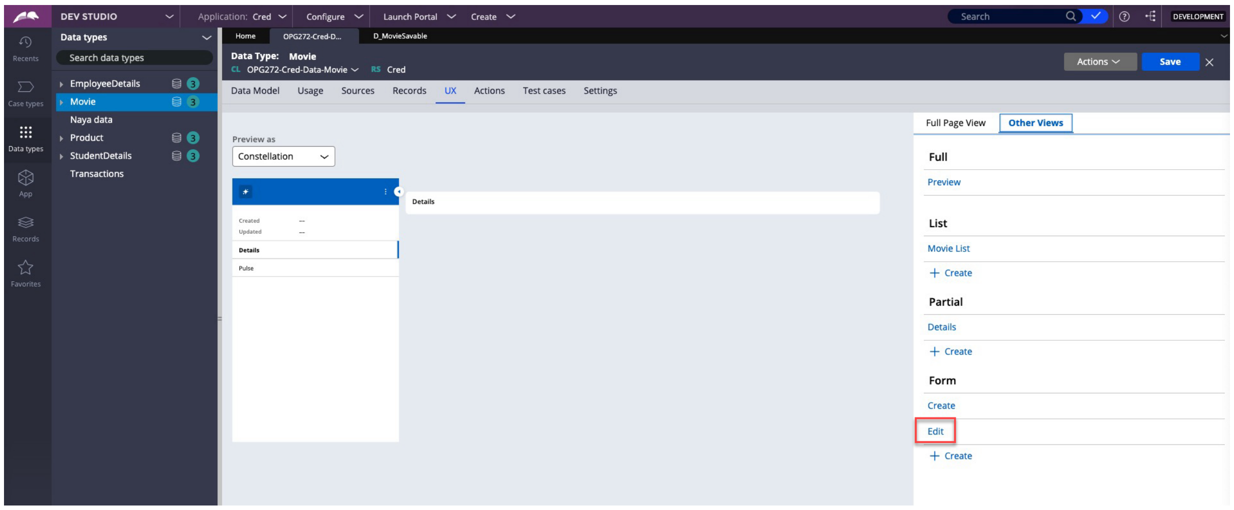
If there are multiple save options, every WHEN is evaluated and the OTHERWISE save option is used only if all WHENs evaluate to false.

1	WHEN pyIsDelete (Parameters)			
	Save type*: Database delete	Class name*: Movie (OPG272-Cred-Data-Movie)	Data transform:	
	☐ Local save			
2	WHEN pyIsPatch (Parameters)			
	Save type*: Database update	Class name*: Movie (OPG272-Cred-Data-Movie)	Data transform:	Validate rule:
	☐ Local save			
3	OTHERWISE			
	Save type*: Database save	Class name*: Movie (OPG272-Cred-Data-Movie)	Data transform:	Validate rule:
	☐ Local save			

+ Add save option

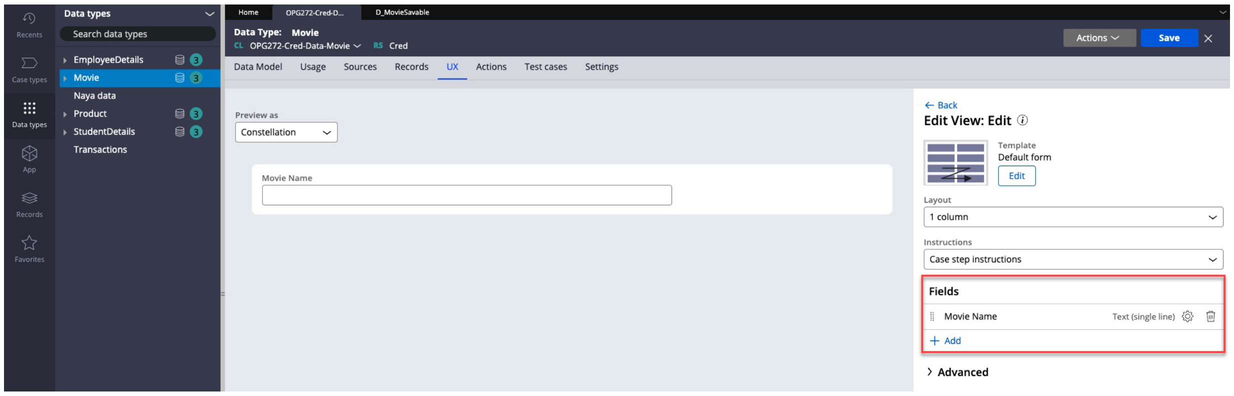
Configuring save plan

6. In the UX tab of the corresponding data type, click the **Other Views** tab, and under the **Form** section, click **Edit.**



Edit form

7. In the **Edit View**, configure the submittable fields of the data type and click **Save**.



Data Type: Movie
CL OPG272-Cred-Data-Movie RS Cred

Preview as: Constellation

Movie Name

Edit View: Edit
Template: Default form
Layout: 1 column
Instructions: Case step instructions
Fields
Movie Name Text (single line)

Configure submittable fields

NOTE: Since primary fields exist in the edit view by default, the fields marked as primary can be updated using this procedure. Since DX APIs allow modifying these fields, mark only those fields as primary that can be modified by the client.

8. To store the data, call the PCore.getRestClient().invokeRestApi API as shown below.

```
PCore.getRestClient().invokeRestApi("updateDataObject", {
                queryPayload: {
                data_view_ID: "< datapageName >"
                },
                body: {
                data: {
                formdata,
                key
                }
                },
                });
```

Example

In this example, the API stores the `MovieName` property in the `D_MovieSavable` data page by passing the `pyGUID` key as a linked field parameter.

```
PCore.getRestClient().invokeRestApi('updateDataObject',{
            queryPayload: {
            data_view_ID: "D_MovieSavable"
            },
            body:{
            data: {
            pyGUID: 'ee793054-a766-49e7-b3f7-fdb44cb266a7',
            MovieName: 'Avengers'
            }
            }
            }).then(response => {
            // Below is the response structure
            /**
            {
            data: {
            responseData: {
            // Record Data
            "MovieName": "Avengers",
            "MovieId": "234",
            "pyGUID": "ee793054-a766-49e7-b3f7-fdb44cb266a7",
            "MovieDescription": "Description Avengers",
            ...
            }
            },
            status:200,
            headers:{
            content-type: "application/json;charset=UTF-8"
            },
                statusText: ""
                }
                */
                });
```

Result:

The data is now stored in the data page.

Related information

- Configuring CRUD data endpoints with Constellation DX API