

# Project Release Notes

Release 4.3 (March 2022)

# What's new

## Custom buttons

Extend action capabilities using custom buttons. Call external APIs and open any URL directly from the action. Custom buttons are available in Create, Edit, Fork, View, and Sign actions. For example, add a custom button to get the line pricing from an external pricing engine in an order management system.

[Read more...](#)

---

## Rollout enhancements

You can now roll out apps to all users of a profile using the Profiles option on the rollout page. For instance, let us say you have a field services app, and the users for this app are part of the 'Technician' profile. You can choose the 'Technician' profile from the rollout screen to make the app available to all technician users instead of selecting each technician individually.

The new Rollout UI offers guided navigation and a dedicated section to invite/revoke users for a smoother flow and improved user experience. The Rollout history page has moved, and it is now accessible per app from the app menu. The rollout summary is more detailed and better organized.

[Read more...](#)

---

## Child section enhancements

Child section enhancements include the following:

- Build-in filter options for selecting / filtering blank records.
- Resizable columns.
- Ability to select all records including records that are yet to be loaded into the child section UI.

[Read more...](#)

# Custom buttons

Business workflows often depend on external services to complete the transaction. You can now boost the capabilities of the standard actions by adding custom buttons to invoke custom actions powered by the custom logic for your business.

For example, you may rely on a custom pricing engine to generate line prices for ordered items in an order management system. Similarly, you may depend on an IoT asset monitoring system to track the movement and monitor the health of your assets.

Custom buttons allow you to extend the capabilities of the create, edit, fork, view, and sign actions to invoke an external API or open any URL directly from the action's UI. Custom buttons can address the scenarios mentioned above:

- Create a custom button to invoke the custom pricing engine for generating line prices.
- Add a custom button to the asset tracking action to open the service provider's asset monitoring webpage using the current record's asset information, such as the serial number and asset tag.

## Types of actions

There are two types of actions that you can associate with the custom button, as described below.

### Call an external API

Use this action type to interact with an external service via REST API. Calling the custom engine for line pricing order management is an example where you can set up a custom

button to call the external engine's API. You should be familiar with REST APIs for setting up this type of action.

### Open the URL

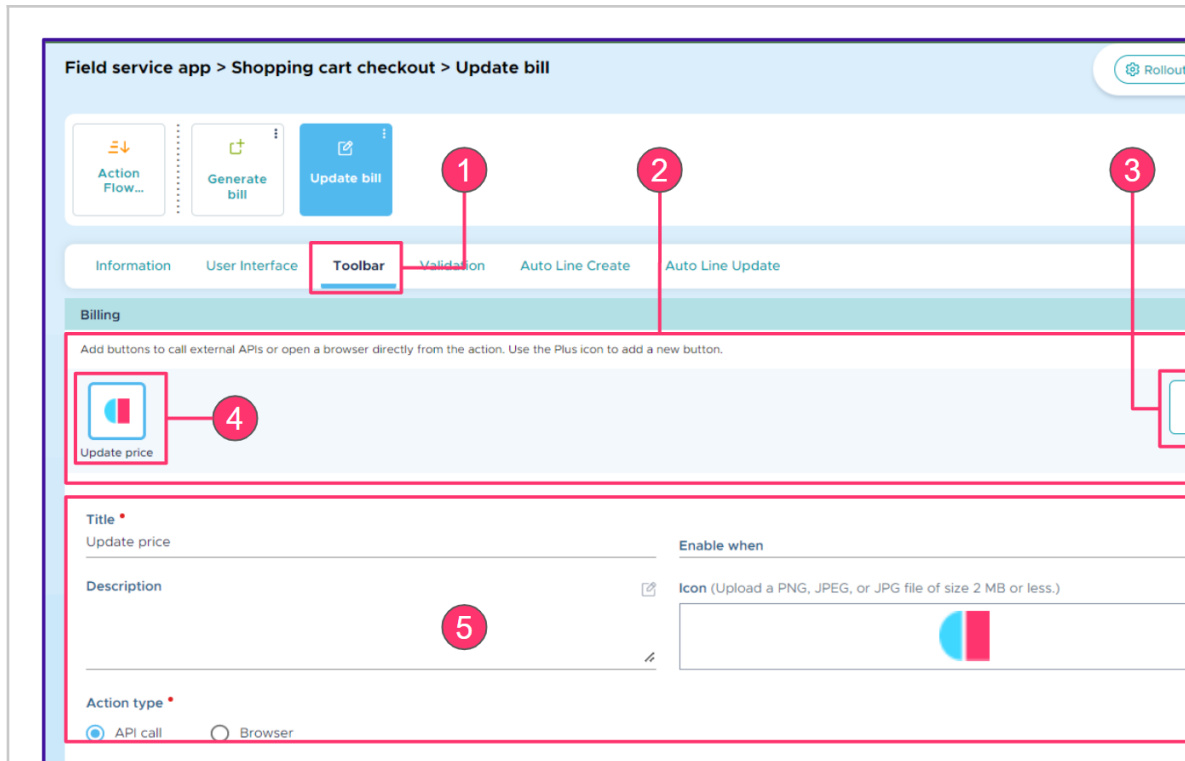
In this type of action, you can enter a URL and add query parameters to be sent as part of the request when the user clicks the button. The asset monitoring example can be set up using this type of action. Depending on the scheme you specify, the URL may open in the relevant app such as the Browser, Gmail, WhatsApp, or Dialler.

## Accessing custom button settings

Look for the new 'Toolbar' tab in the action designer. The settings for creating a custom button are available under this tab.

### Steps to add a custom button for an action:

1. Open the designer for the action where you need to add the custom button.
2. Click the Toolbar tab. The custom button toolbar appears below.
3. Click the Plus icon on the toolbar. A new button appears on the toolbar.
4. Click the newly added button; the settings page for the custom button appears below.



Accessing custom button settings		
1. Toolbar tab	3. Add new button	5. Custom button settings page
2. Toolbar	4. Newly added custom button	

## Custom button common settings

Custom button settings that are common across action types are listed below.

- Title: Button title. This text will appear below the button.
- Hide when: You can set criteria here to hide the button when it is not needed. For example, you can hide the 'Update price list' button in Generate bill action when there are no billable items.
- Icon: Upload an icon to customize the appearance of the button.
- Action type: Choose one of two options - Make an API call or Open in the Browser to open any URL when the button is clicked. Default type: Browser.

The UI fields for common settings

1. Title	4. Description	5. Custom button settings page
2. Enable when	5. Newly added custom button	

## Creating a custom button for opening a URL

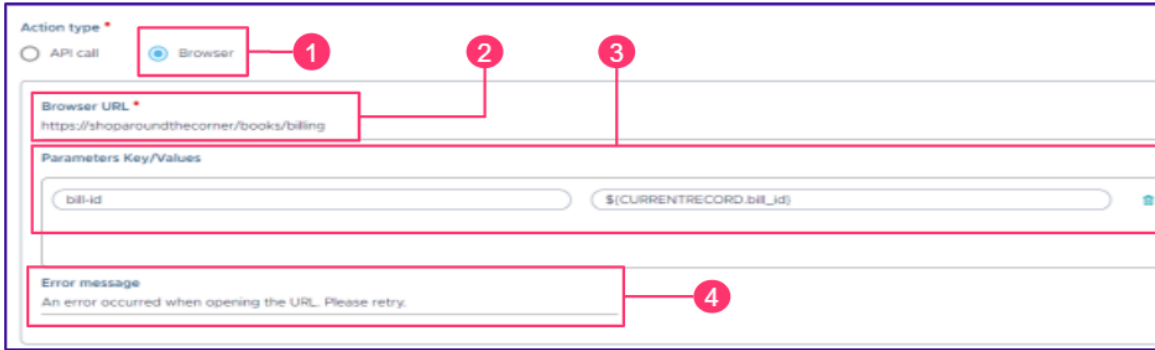
To add a custom button for opening a URL, use the Open in browser action type. You can add query parameters of the URL. You can also send field values from the current record as query parameters by setting the value with the `${CURRENTRECORD}` literal.

### Steps to add a custom button for action are below:

1. Open the settings page or the custom button.
2. Choose Browser from the Action type options.
3. Enter the URL that should open when the user clicks the button.
4. Add query parameters to the request. To send the current record information to the request, you can set the query parameter value using the `${CURRENTRECORD}` literal. For example, you can send the field value of `bill_id` using the literal `${CURRENTRECORD.order_id}`, where `order_id` is the field name.
5. Enter the message that should appear if the URL fails to open.

## Settings available for the URL action are described below.

1. Browser URL: URL to open when the user clicks the button.
2. Query parameters: Values to send to the URL as query parameters. You can use the `#{CURRENTRECORD}` literals.
3. Error message: The error message you enter here will appear on the app if the URL fails to open when the user clicks the button.



The screenshot shows the configuration interface for a browser action. It includes a radio button for 'API call' and a selected radio button for 'Browser'. The 'Browser URL' field contains the text 'https://shoparoundthecorner/books/billing'. Below this is a 'Parameters Key/Values' section with a table containing one row: 'bill-id' and ' #{CURRENTRECORD.bill\_id} '. At the bottom, the 'Error message' field contains the text 'An error occurred when opening the URL. Please retry.' Red callout boxes with numbers 1 through 4 point to the 'Browser' option, the 'Browser URL' field, the 'Parameters Key/Values' section, and the 'Error message' field respectively.

The UI fields for browser action settings	
1. Action type - Browser option	3. Query parameters
2. URL	4. Error message

## Creating a custom button for calling an API

To add a custom button for calling an API, use the Make an API call action type. This type of action is available in Create, Edit, and Fork actions. You can add authentication, request headers, query parameters, and field values from the current record to the API call.

### Note:

Action type 'Make an API call' is not available in view actions. Since the records



information is read-only in view actions, you can't invoke APIs to update the values of the record's field.

The UI available for setting up API call action type is described below.

The screenshot shows the configuration interface for an API call action type. It includes sections for Action type, Authentication scheme, Computation API setup, REST endpoint, HTTP method, request headers, query parameters, in-progress and success messages, failure message setup, objects to include, and fields to send. Red circles with numbers 1 through 12 highlight specific UI elements.

The UI fields for API call action type settings

1. API call option	5. Request header	9. Failure message setup
2. Authentication scheme	6. Query parameters	10. Objects to include
3. REST endpoint	7. In-progress message	11. Fields to send
4. HTTP method	8. Success messages	12. Record selection criteria

Steps to add a custom button for API call action:

1. From the action's toolbar, open the settings page for the custom button.
2. From the Action type options, choose 'API Call'. The API settings section opens below.

### Steps to enable authentication:

1. Choose a suitable scheme for your API from the Authentication scheme dropdown. All authentication schemes listed here are set up globally at the platform level.
2. After successful authentication, the response header values can be accessed in the subsequent call to the button's API using the `#{AUTHAPI}` literal.

#### Note:

If the Authentication scheme dropdown is empty, it could mean that you don't have any authentication schemes set up yet. Please reach out to your support contact at Appify to help you with setting it up.

### Steps to set up the API request properties:

1. Enter the endpoint for the REST API you need to call when the user clicks the button.
2. Add request headers such as Content-type if needed.
3. To send authentication tokens in the request header, set the value using the `#{AUTHAPI}` literals. For example, to pass the auth-token received in the authentication call set the header value to `#{AUTHAPI.auth-token}`
4. Add query parameters to the URL if needed.

### Steps to set up API status messages:

1. Enter the text that should appear in the app when the API is in-progress.
2. Enter the text that should appear in the app when the API completes successfully.

3. Set up the failure message using values from the response body of the failed API call. See the example below for mapping the response fields to the failure settings.

Failure response body sample:	Failure message settings using failed response fields:
<pre>{   "errorTitle": "No response",   "errorMessage": "Server failed to   respond",   "errorCode": "500",   "responseMsg": "Move over, here   comes error" }</pre>	<ul style="list-style-type: none"><li>● Field to use for error code: <b>errorCode</b></li><li>● Field to use for error title: <b>errorTitle</b></li><li>● Field to use for error message: <b>errorMessage</b></li></ul>

## Including objects for the API call

1. All the sections from the action's layout are listed in the 'Objects to include' panel. Select the sections to include in the API.
2. You can use an alias to refer to the object in the API call. To set the object alias, click the edit icon on the section name in the panel. The properties panel opens on the right side. Enter the alias here.

## Adding fields to the API call

1. To add fields from the selected sections, click the section on the Objects to include panel and use the Add fields button. The field picker panel opens on the right side. Select all the fields to send.
2. You can use an alias to refer to the field name in the API call. Enter the alias in the Alias name column for the field.
3. If you need to update the record value with the value returned in the API response, check the Update flag for the field.

## Filtering the records for the API call

1. To limit the number of records sent in the API, set the filter criteria from the Records to filter field. The filter criteria panel opens on the right side.
2. Set the condition as needed.

## Payload compliance for API calls

The structure of the request / response payload of the external API should be compatible with that of the payload generated by the custom button. The example below illustrates what the payload structure looks like and how the fields are mapped. Reach out to your Appify contact for more information regarding the API payload compliance.

For instance, let us say you have a custom button 'Price' to fetch the line prices in an order management flow. The action includes a header section - Bill and a child section - Items containing two items for pricing. Refer to the image below.

bill-000001  
Last modified by Karteek Joshi on 4/1/22 07:41 pm

Save Ca

Billing

Price New Used Green Cash Bill Gift Wallet Cart Barcode

Bill

Bill date Bill status Pending

ITEMS

	Actions	Filter	Filter
<input type="checkbox"/>		Item name	Item price
<input type="checkbox"/>		HDD	
<input type="checkbox"/>		Monitor	

Records per Page: Show 5 < 1 2 > Showing 1 - 4

Records in the header & child sections

1. Custom button - Price	2. Header section	3. Child section
--------------------------	-------------------	------------------

The request payload generated when the user clicks the custom button will contain a record ID field in addition to the selected header and child records. The record ID field uniquely identifies the record. The payload structure and sample request body for the above example is given below:

The request payload structure looks like this:

JSON

```
{
  "<header_section_name>":
  [
    {
      "<record_id_field_name>":"<ID_field_value>",
      "<header_field_name>":null,
      "<child_section_name>":
      [
        {
          "<child_field_name>":"<value>",
          "<recordid_field_name>":"<value>"
        }
      ]
    }
  ]
}
```

A sample request payload for the above record:

JSON

```
{
  "Bill":
  [
```

```
{
  "recordid":"00011959094297712427008",
  "Bill date":null,
  "Items":
  [
    {
      "Item name":"HDD",
      "recordid":"00012959094299432091648"
    },
    {
      "Item name":"Monitor",
      "recordid":"00012959453893043978240"
    }
  ]
}
```

## App user notes

When you add custom buttons to actions, and the buttons are enabled, the toolbar appears below the action title at the top of the screen.

- When the user clicks the custom button, if the button's action type is to open an URL, a web page opens with the URL set for the button.
- In the case of API call action type, the API is called when the user clicks the button, and a popup appears on the screen indicating the status.
- If the update option is enabled for any fields in the Send fields settings, the field values are automatically updated when the API completes successfully.
- The latest data available on the action screen is sent, even if the data is not yet saved yet.
- The button action is applicable to all the records that qualify for the record selection criteria, including records that are yet to be loaded in the child section.
- If the button action fails, an error message popup appears on the screen.

## Web clients

- The button's action is applicable to the header and all the selected child and grandchild records.
- Users can collapse the toolbar using the collapse icon on the top right corner of the toolbar.

Field service app shop v2.0

bill-000002  
Last modified by Kartik Joshi on 4/22/22 10:56 am

Billing

Price New Used Green Cash Bill Gift Wallet Cart Barcode

Bill

Bill date: 3/30/22  
Bill status: Pending

	Actions	Filter	Filter
		Item name	Item price
<input type="checkbox"/>		Dell	\$20
<input type="checkbox"/>		Greenback	\$20
<input type="checkbox"/>		HP	\$20
<input type="checkbox"/>		Leatherback	\$20
<input type="checkbox"/>		Lenovo	\$20

Records per Page: Show 5 < 1 2 3 > Showing 1 - 5

Custom button toolbar - Web client

1. Toolbar title	2. Custom buttons toolbar	3. Collapse button
------------------	---------------------------	--------------------

## Mobile clients (iPad, iPhone, and Android)
































- The button's action is applicable to the header and all the selected child records. If any grandchild sections are selected, the records are **NOT** included in the payload.
- Users can use the more / less buttons to expand / collapse the toolbar.

# Child section enhancements

The following enhancements are now available in the child and grandchild sections of Create, Edit, and Fork actions:

- The filter row is now located above the child section's header row, and it comes with built-in options to filter blank rows.
- Columns are resizable, and you can adjust the width by clicking and dragging the column border.
- Directly type the value in the field to edit it inline. There is no need for double-clicking for edit mode in editable sections.
- You can now use the 'Select all' option for bulk operations to include child records that are yet to be loaded into the child section UI.



ITEMS (13 of 13 records) <span>Select all 13 Records</span> <span>1</span>				
			Filter	Filter
<input checked="" type="checkbox"/>	 		Filter	Item Price
<input checked="" type="checkbox"/>			Show blanks	\$20
<input checked="" type="checkbox"/>			Show non blanks	
<input checked="" type="checkbox"/>			Dell	\$20
<input checked="" type="checkbox"/>			HP	\$20
<input checked="" type="checkbox"/>			Lenovo	\$20
<input checked="" type="checkbox"/>			Playstation	\$546
<input checked="" type="checkbox"/>			Playstation	\$546
<input checked="" type="checkbox"/>			Playstation	\$546
<input checked="" type="checkbox"/>			joystick	\$546
<input checked="" type="checkbox"/>			Playstation	\$546
<input checked="" type="checkbox"/>			joystick	\$546
<input checked="" type="checkbox"/>			Olive ridley	\$20
<input checked="" type="checkbox"/>			Leatherback	\$20
<input checked="" type="checkbox"/>			Greenback	\$20

Records per Page: Show 5

Child section enhancements

1. Select all option

1. Filter options


# Rollout enhancements

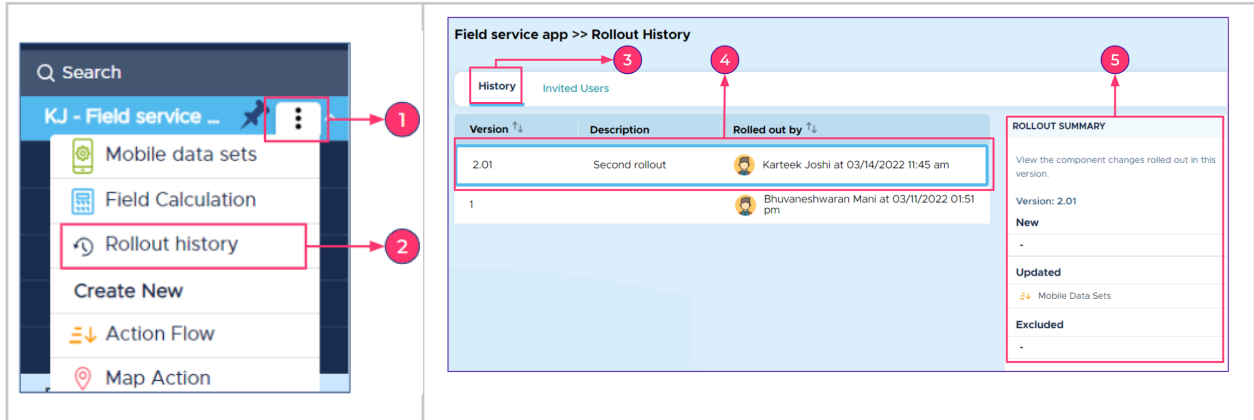
Rollout enhancements in this release include the following:

- Rollout history is now available per app and is accessible from the app menu.
- There is a new UI for inviting app users.
- The improved UI with guided navigation provides a smoother rollout experience.
- Starting with this release, you can rollout to profiles directly in addition to individual users.

## New home for rollout history

Rollout history is now available at the app level, and you can access it directly from the app menu on the left navigation bar.

1. Click the more icon  on the app menu in the left navigation bar to expand the menu.
2. Click Rollout history.
3. The Rollout history page opens on the right.
4. Click on the entry in the history section to see the details.
5. The rollout summary panel appears on the right. The rollout changes are available [here](#).



(fig 1) Accessing rollout history

(fig 2) Rollout history page

1. More icon on app menu
2. Rollout history option

3. Rollout history tab
4. Rollout history entries

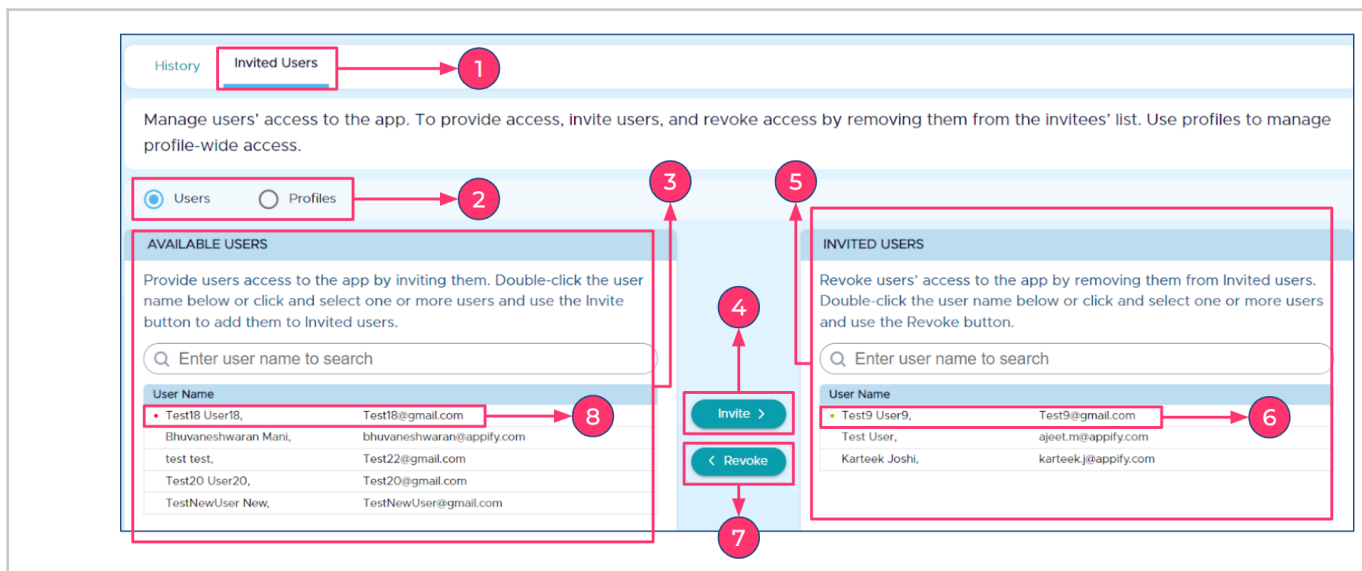
5. Rollout summary

## New UI for inviting and revoking app users

You can invite users to an app or revoke users' access to an app in the following two ways:

- At the time of rollout in step 2 - Invite users section.
- From the Invite users tab in the rollout history page at any time.

The new UI for inviting / revoking users looks like this:



(fig 3) Invite app users			
1. Invite users tab 2. User / Profiles	3. Available users list 4. Invite button	5. Invited users list 6. Newly invited user	7. Revoke button 8. Newly revoked user

### Steps for inviting users to an app:

1. Go to the invite users page
2. Choose Users. The user selection section appears below.
3. Select one or more users to invite from the Available users by clicking the name and using the Invite button.
4. The selected users are moved to Invited users. A green dot indicates pending rollout.
5. You can also select and invite individual users by double-clicking the name.
6. Confirm the rollout.

An email notification containing the sign-in link is sent to the invited users.

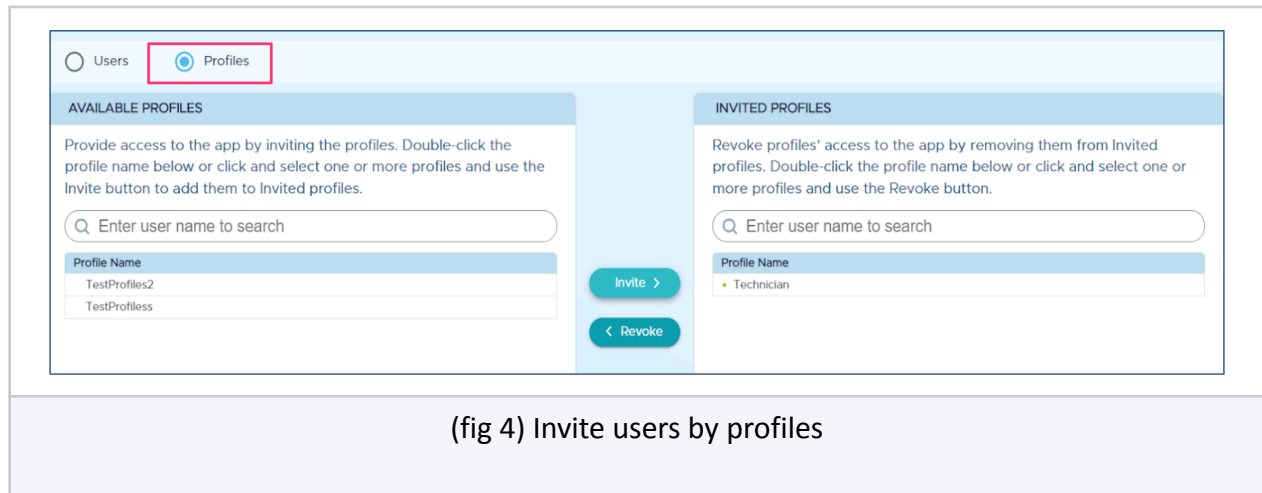
### Steps for revoking users' access to an app:

1. Go to the invite users page
2. Choose Users. The user selection section appears below.
3. Select one or more users to revoke from the Invited users by clicking the name and using the Revoke button.
4. The selected users are moved to Available users. A red dot indicates pending rollout.
5. You can also select and revoke individual users by double-clicking the name.
6. Confirm the rollout.

Revocation is immediate. Users won't be able to sign in to their apps.

## Rollout to profiles

You now can invite users by profiles to a rollout instead of selecting users individually. Use the Profiles option on the Invite users page.



With the introduction of this feature, when you create new users, they will automatically have access to the app rolled out to their profile. Any user-level invitation takes precedence over that of the profile level.