



**BUILDING BLOCKS / UI Designer / UI component types / Collection /  
collection\_prototype**

# Contents

- BUILDING BLOCKS / UI Designer / UI component types / Collection / Collection Prototype
  - Description
  - Configurable properties:
  - Example
  - Adding elements with UI Actions
    - Step 1 - Defining the Node Action
    - Step 2 - Adding the Button & UI Action
    - Result

## BUILDING BLOCKS / UI Designer / UI component types / Collection / Collection Prototype

### Description

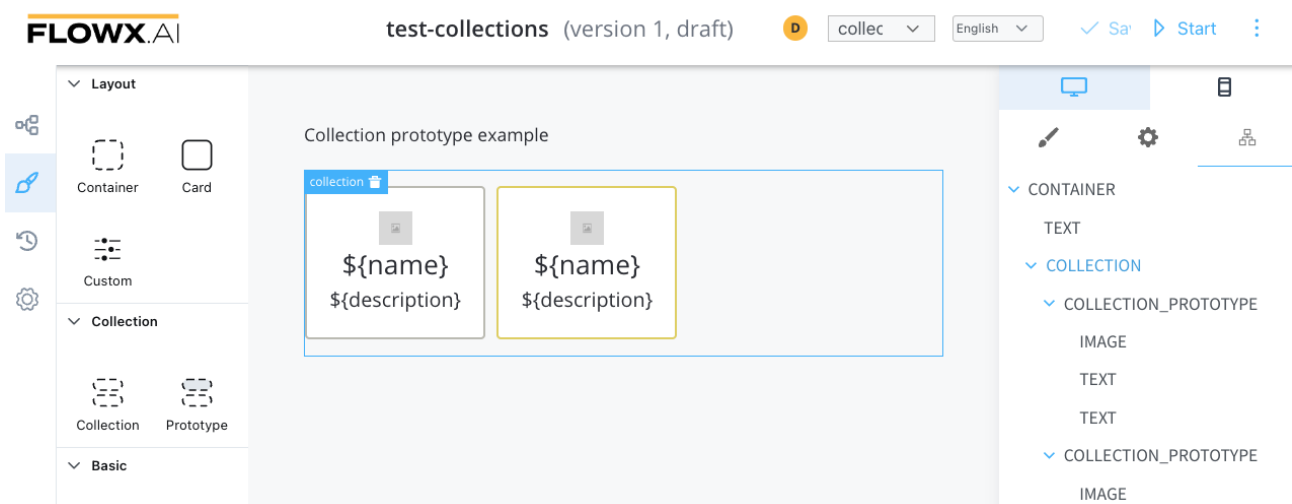
This additional container type is needed to allow multiple prototypes to be defined for a single **Collection**. This allows elements from the same collection to be displayed differently.

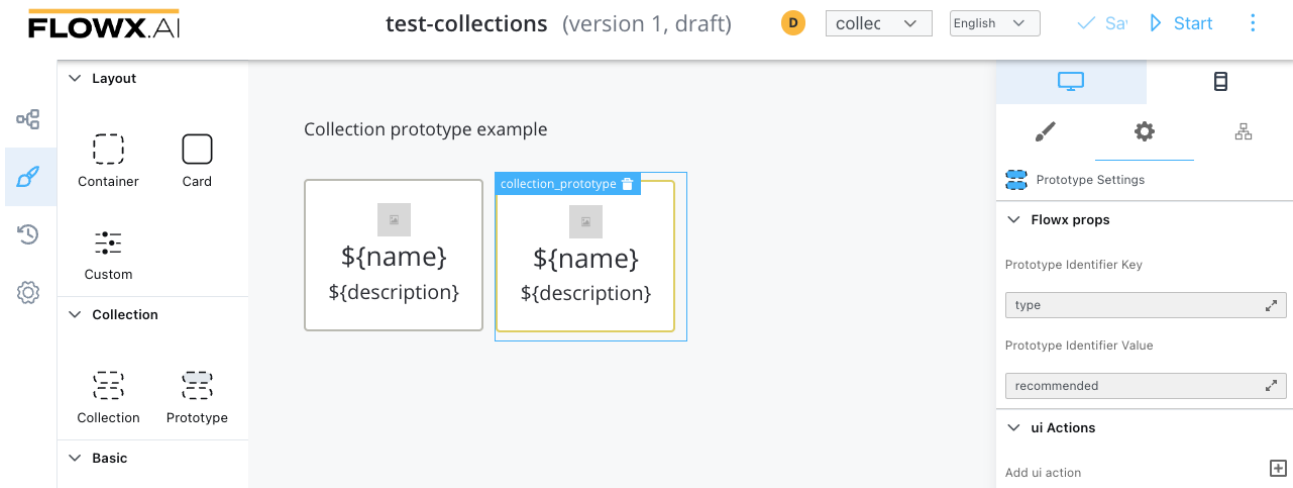
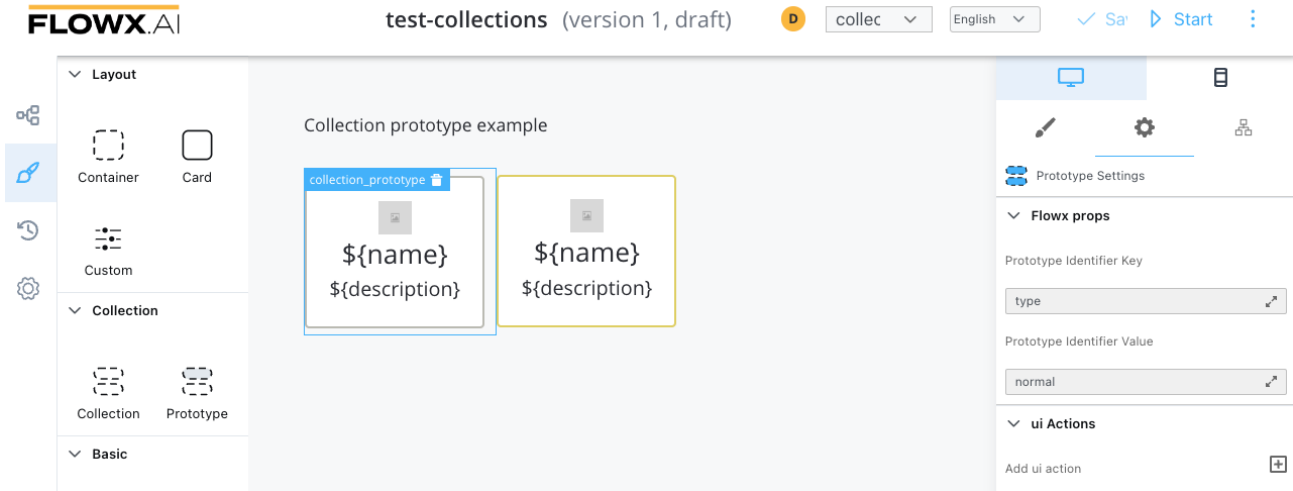
For example, suppose you are creating a piece of UI in which the user is presented a list of possible products from which to choose, but you want one of the products to be highlighted as the recommended one. This example requires a collection with two **collection prototypes** (one for the normal product and one for the recommended one).

## Configurable properties:

1. **Prototype Identifier Key** - the key where to look in the iterated object to determine the prototype to be shown - in the below example the key is "type"
2. **Prototype Identifier Value** - the value that should be present at the **Prototype Identifier Key** when this `COLLECTION_PROTOTYPE` should be displayed - in the below example the value is "normal" or "recommended"

## Example



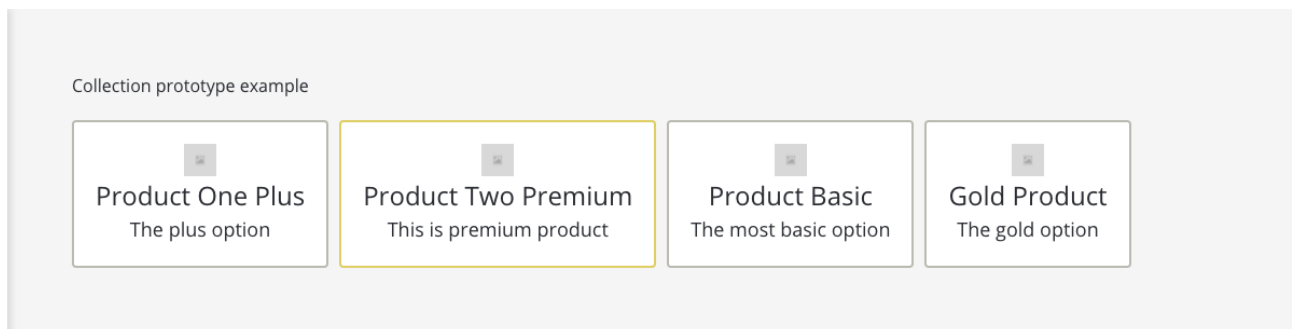


Source collection data example for products:

```
products: [  
  {  
    name: 'Product One Plus',  
    description: 'The plus option'  
    type: 'normal'  
  },  
  {  
    name: 'Product Two Premium',
```

```
    description: 'This is premium product'
    type: 'recommended',
  },
  {
    name: 'Product Basic',
    description: 'The most basic option'
    type: 'normal'
  },
  {
    name: 'Gold Product',
    description: 'The gold option'
    type: 'normal',
  }
]
```

The above configuration will render:





## Adding elements with UI Actions

There are a few differences you need to take into consideration when configuring elements that make use of **UI Actions** inside a **Collection Prototype**.

To showcase these differences, we'll use the next example:

(Some) FlowX employees:

 Mihai Saru	<button>Select</button>	<button>Save Name</button>
 Andra Popazu	<button>Select</button>	<button>Save Name</button>

We have a **Collection** with two employees and we want to provide the user with the option of selecting one of the employees (eg. to allow for further processing in the next steps of the process).

## Step 1 - Defining the Node Action


To select one employee from the list, we first must add an **Action** to the **User Task Node** this UI is attached to:


Node: **collection-1** (ID: 431461) ⌵ ✕


Node Config

Actions

Actions +

save-item + 

send-item-name + 

stop + 

Action Edit

ID: 431905

Name

save-item

Order

1

Timer Expression

Save Data ⌵

☐ Automatic

☒ Manual

☐ Mandatory


☒ Optional

☒ Repeatable

Autorun Children? ☐

Allow BACK on this action? ☐

Data to send

selectedEmployee 

Add Key

Save

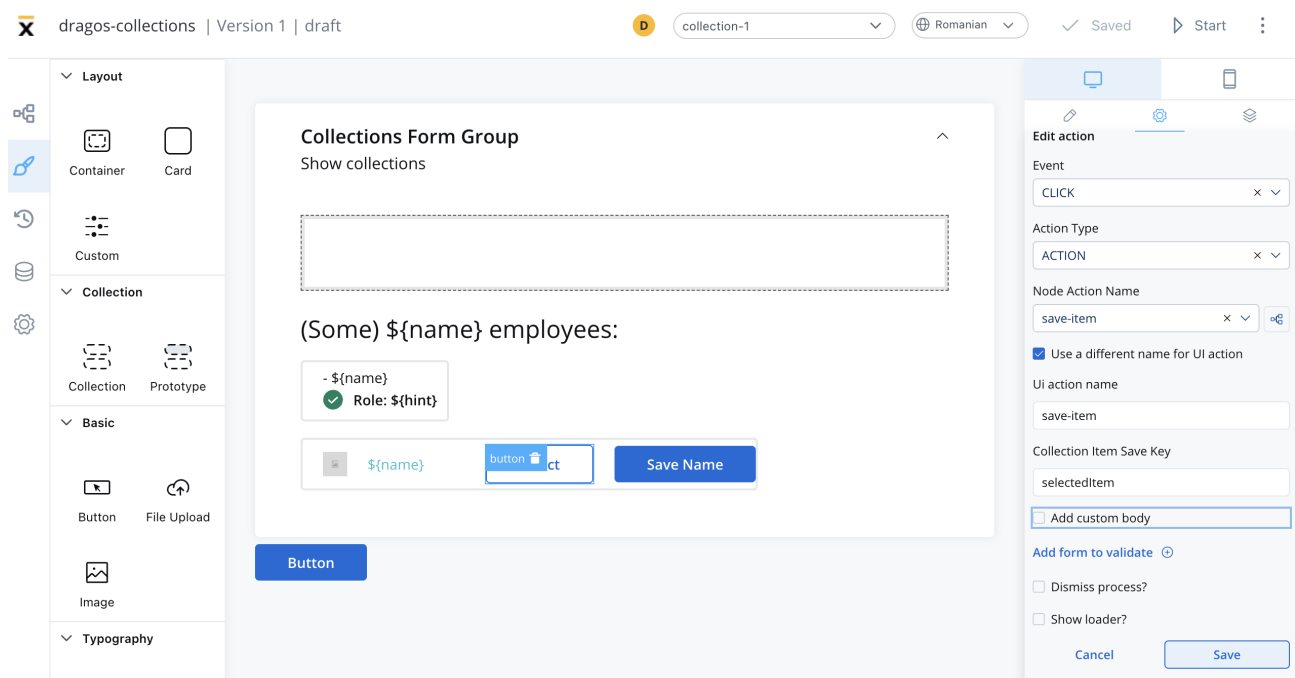
This **save-item** action is **manual** (since it will be triggered by the user) and **optionally** (since selecting an employee is not a requirement to go to the next **Node** in the process).

To allow the user to change his mind about the selected employee, this action is also marked as **Repeatable**.

Keep in mind to check the **Data to send** section. Here we are telling the platform where we want the selected employee (for which the user pressed the **Select** button) to be saved in the **process data**. In this example, we want it to be saved under the `selectedEmployee` key.

## Step 2 - Adding the Button & UI Action

Now that we have a **Node Action** defined, we can go ahead and add the **Select** button in the UI of the **User Task** which contains the Employees Collection.



**Collection Item Save Key** field has an important role in the UI Action configuration of the **Select** button. This field represents how we pass the value of the **Employee** that the user has selected to the **Node Action** that we created in **Step 1**, named *save-item*.

In our example, we set **Collection Item Save Key** to be `selectedEmployee`.

### DANGER

**IMPORTANT:** `selectedEmployee` key is how we expose the data from the **Collection** to the Node Action. It is **imperative** that the name in the **Collection Item Save Key** is the same as the one used in the **Data to send** input in the Node Action.

The button and UI action are mostly configured as any other Button and UI Action would be configured.

## Result

This is how the process data looked before we pressed the **Select** button for an employee:

### Process status

Data:

```
processInstanceId: 483001
processData: Object ({"companies":[{"employees":[{"name":"Mihai Saru","imageSrc":"https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Svelte_Logo.svg/1702px-Svelte_Logo.svg.png","type":"colore
tokenId: 483051
tokenUid: "ae109887-e5aa-46f9-ba18-668ccce33ce8"
websocketPath: "/ws/updates/process"
processInstanceId: "01d5b64d-7c61-4d98-a590-d23336f89f95"
websocketAddress: "ws://public.qa.flowxai.dev/01d5b64d-7c61-4d98-a590-d23336f89f95"

```

Tokens

Token uid	Token Status	Status Current Node	Date updated
ae109887-e5aa-46f9-ba18-668ccce33ce8	ACTIVE	EXECUTED_PARTIAL	04 May 2022, 12:09 PM

This is how the process data looks after we selected an employee from the list (notice the new field `selectedEmployee`):



Process status

```
Data:
  processInstanceId: 483001
  processData: Object {"companies":[{"employees":[{"name":"Mihai Saru","imageSrc":"https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Svelte_Logo.svg/1702px-Svelte_Logo.svg.png","type":"colored"}]}]}
  tokenId: 483051
  selectedEmployee:
    name: "Mihai Saru"
    imageSrc: "https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Svelte_Logo.svg/1702px-Svelte_Logo.svg.png"
    type: "colored"
  tokenUid: "ae109887-e5aa-46f9-ba18-668cce33ce8"
  websocketPath: "/ws/updates/process"
  processInstanceId: "01d5b64d-7c61-4d98-a590-d23336f89f95"
  websocketAddress: "wss://public.qa.flowxai.dev/01d5b64d-7c61-4d98-a590-d23336f89f95"
```

Tokens

Token uuid	Token Status	Status Current Node	Date updated
ae109887-e5aa-46f9-ba18-668cce33ce8	ACTIVE	EXECUTED_PARTIAL	04 May 2022, 12:09 PM

Was this page helpful?