Sumit Vadaviya

Camunda BPM Expert, Corporate Trainer

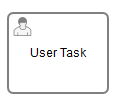
Email : sumit.vadaviya@trainosoft.com

I

n this session we will go through user task.

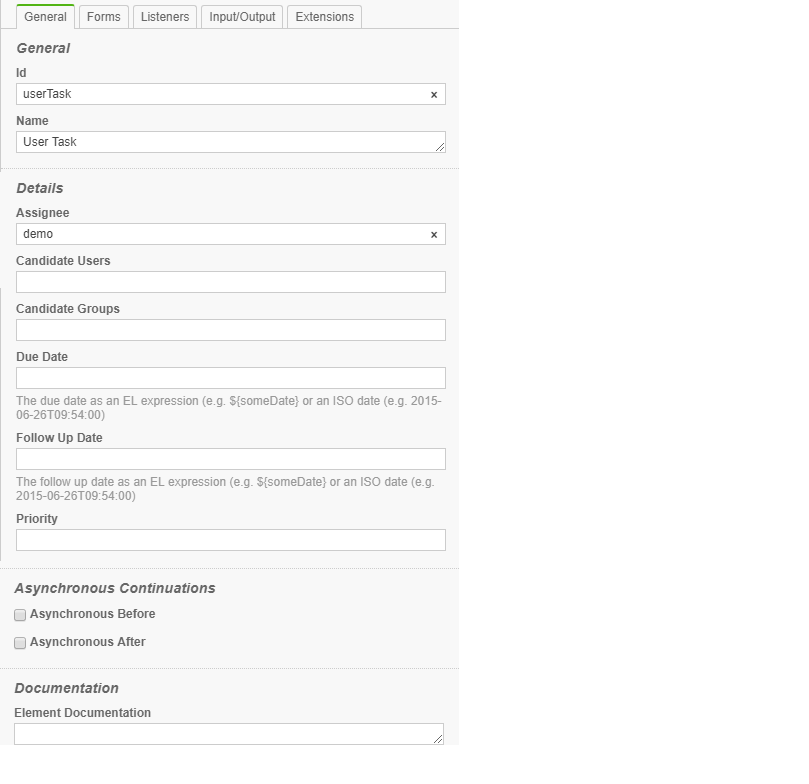
A User Task is used to model work that needs to be done by a human actor. When the process execution arrives at such a User Task, a new task is created in the task list of the user(s) or group(s) assigned to that task.

User task



**User Task Properties:**

**General:**



**Id:** Id of task

**Name:** Name of task

**Assignee:** This custom extension allows direct assignment of a User Task to a given user.

**Candidate Users:** This custom extension allows you to make a user a candidate for a task.

**Candidate Groups:** This custom extension allows you to make a group a candidate for a task.

**Due Date:** Each task has a field indicating the due date of that task. The Query API can be used to query for tasks that are due on, before or after a certain date.

The due date of a task can also be altered using the TaskService or in TaskListeners using the passed DelegateTask.

**Follow Up Date:** Each task has a field indicating the follow up date of that task. The Query API can be used to query for tasks that need to be followed up on, before or after a certain date.

**Priority:** The attribute specifies the initial priority of a [User Task](https://docs.camunda.org/manual/7.11/reference/bpmn20/tasks/user-task/) when it is created.

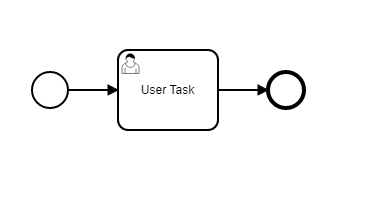
Default value is valid java long value : 50

**Asynchronous Before:** Specifies an asynchronous continuation after an activity

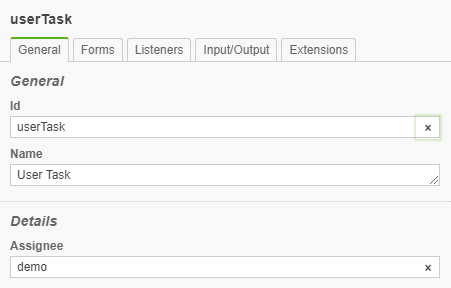
**Asynchronous After:** Specifies an asynchronous continuation before an activity

**Element Documentation:** A User Task can also have a description. In fact, any BPMN 2.0 element can have a description. A description is defined by adding the documentation element.

**Exercise User Task with Assignee:**

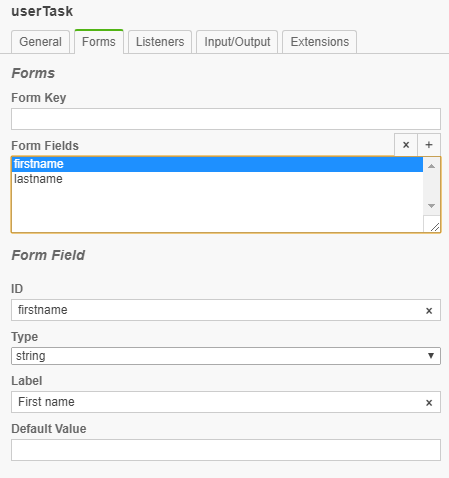


**User Task Properties:**



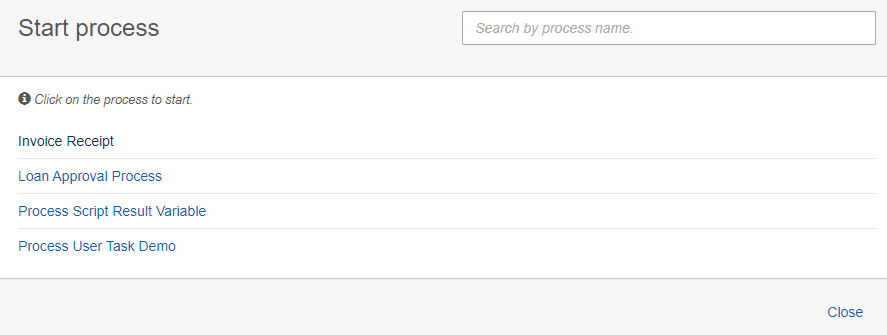
This user task is assigned to demo user.

**Task Form:**



When user task is created. Default form with field firstname and lastname will be presented to user demo.

**Deploy and Test the process**

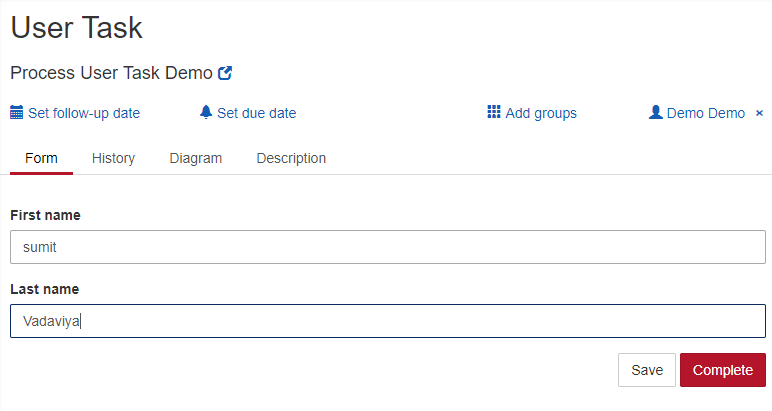


**Start Process:**

Start Process User Task Demo.

**User task:**

User task will be created and assigned to demo user as below

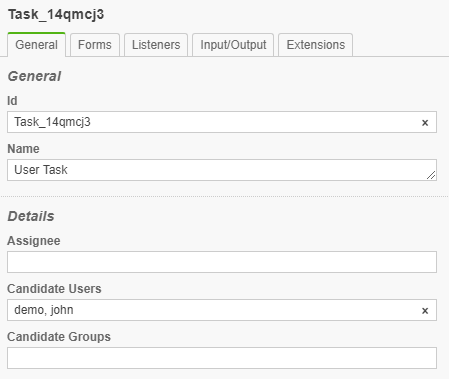
****

Click the Complete button.

Process will be completed.

**Exercise User Task with Candidate User:**

**User Task Properties:**



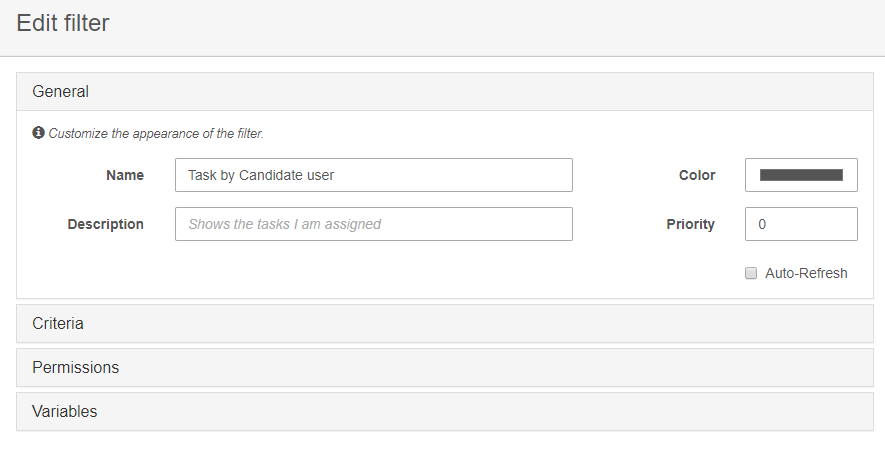
Demo and john user will be candidate for this user task to work on.

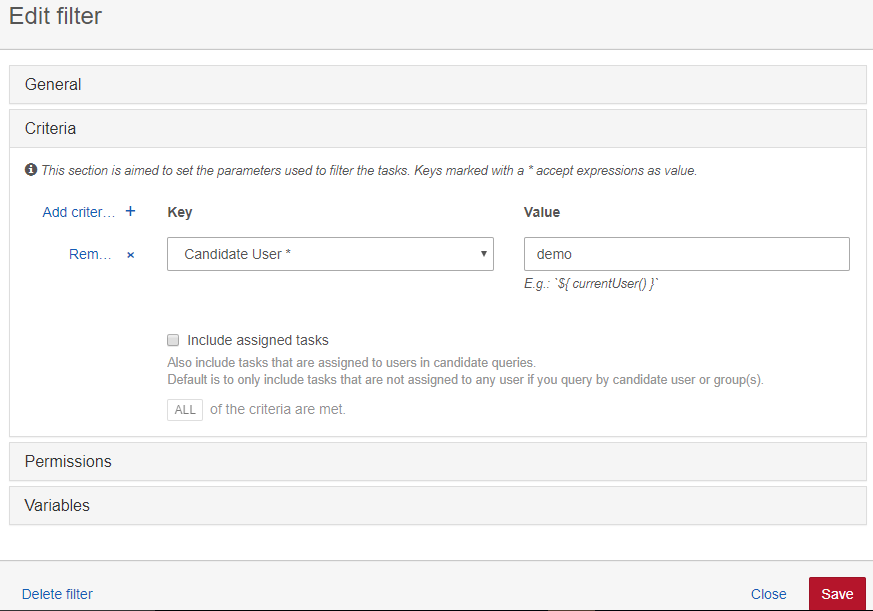
**Deploy the process**

Login as demo user to Camunda TaskList application.

**Create Filter:**

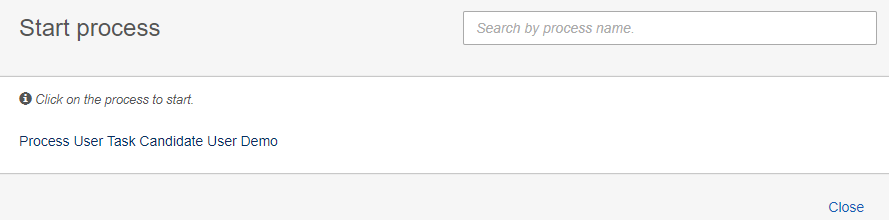
Create filter to list user task which demo user can work on.



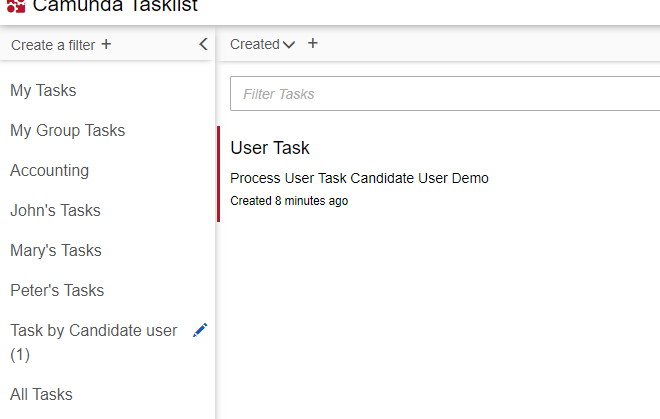


Save the filter configuration.

**Start the Process**

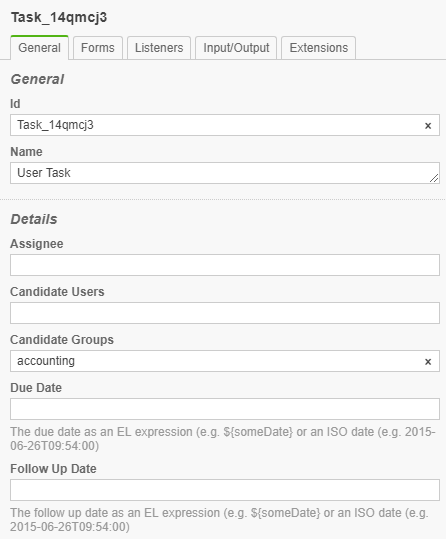


**Claim and complete user task:**



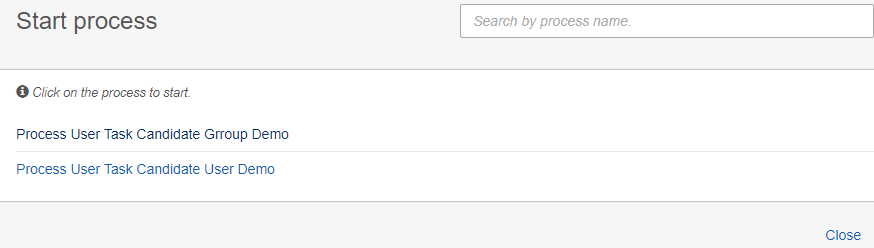
You can see filter Task by Candidate user has on task which demo user can claim and complete.

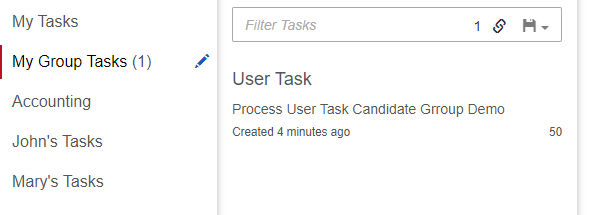
**User Task with Candidate Groups:**



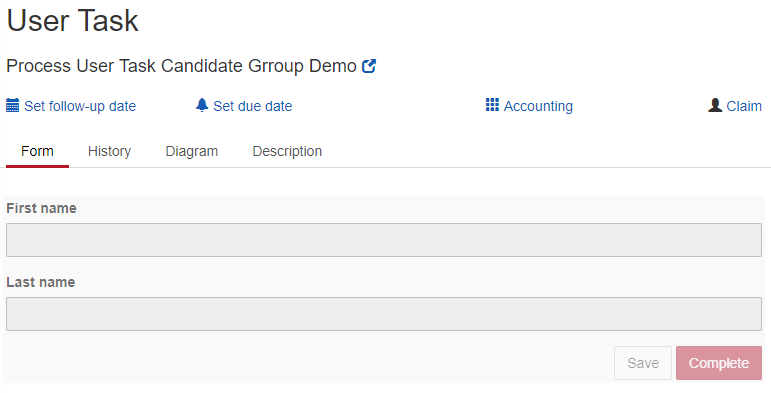
**Deploy the process**

**Start the process**

****

****

**Task Detail**



User task is assigned to accounting group.

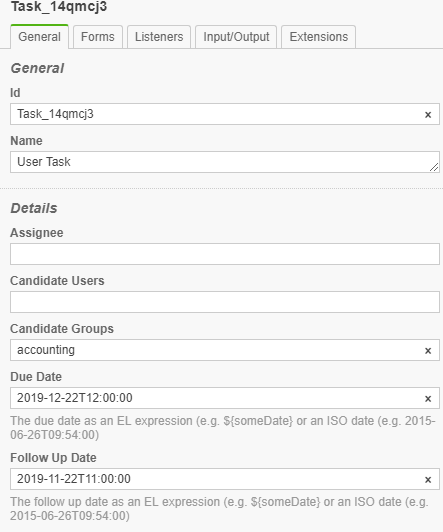
User can claim and complete the task

**User Task Due Date:**

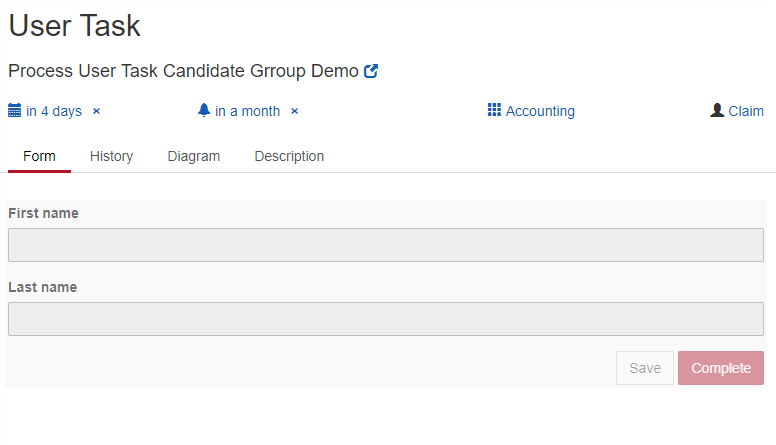
Set Due Date to 2019-12-22T12:00:00

**User Task Follow Up Date:**

Set Follow Up Date to 2019-11-22T11:00:00



**Deploy the process and Start process**

****

**Deploy User Task Form Field Validation**

Validation can be used for specifying frontend and backend validation of form fields. Camunda BPM provides a set of built-in form field validators and an extension point for plugging in custom validators.

The following built-in validators are supported out of the box:

|  |  |
| --- | --- |
| **Validator** | **Explanation** |
| required | Applicable to all types. Validates that a value is provided for the form field. Rejects 'null' values and empty strings.  eg. <camunda:constraint name="required" /> |
| minlength | Applicable to string fields. Validates minlength of text content. Accepts 'null' values.  eg. <camunda:constraint name="minlength" config="4" /> |
| maxlength | Applicable to string fields. Validates maxlength of text content. Accepts 'null' values.  eg. <camunda:constraint name="maxlength" config="25" /> |
| min | Applicable to numeric fields. Validates the min value of a number. Accepts 'null' values.  eg. <camunda:constraint name="min" config="1000" /> |
| max | Applicable to numeric fields. Validates the max value of a number. Accepts 'null' values.  eg. <camunda:constraint name="max" config="10000" /> |
| readonly | Applicable to all type. Makes sure no input is submitted for given form field.  eg. <camunda:constraint name="readonly" /> |