

User task

In 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 Properties:

General:

General	Forms	Listeners	Input/Output	Extensions
---------	-------	-----------	--------------	------------

General

Id

Name

Details

Assignee

Candidate Users

Candidate Groups

Due Date

The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Follow Up Date

The follow up date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Priority

Asynchronous Continuations

☐ Asynchronous Before
 ☐ Asynchronous After

Documentation

Element Documentation

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 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:

userTask

General

Forms

Listeners

Input/Output

Extensions

General

Id
userTask x

Name
User Task

Details

Assignee
demo x

This user task is assigned to demo user.

Task Form:

userTask

General

Forms

Listeners

Input/Output

Extensions

Forms

Form Key

Form Fields

x

+

firstname

lastname

Form Field

ID

firstname

x

Type

string

Label

First name

x

Default Value

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

Click on the process to start.

Invoice Receipt

Loan Approval Process

Process Script Result Variable

Process User Task Demo

Close

Start Process:

Start Process User Task Demo.

User task:

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

User Task

Process User Task Demo

Set follow-up date

Set due date

Add groups

Demo Demo

Form

History

Diagram

Description

First name

sumit

Last name

Vadaviya

Save

Complete

Click the Complete button.

Process will be completed.

Exercise User Task with Candidate User:

User Task Properties:

Task_14qmcj3

General

Forms

Listeners

Input/Output

Extensions

General

Id

Task_14qmcj3

Name

User Task

Details

Assignee

Candidate Users

demo, john

Candidate Groups

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.

Edit filter

General

i Customize the appearance of the filter.

Name

Task by Candidate user

Color



Description

Shows the tasks I am assigned

Priority

0

☐ Auto-Refresh

Criteria

Permissions

Variables

Edit filter

General

Criteria

i This section is aimed to set the parameters used to filter the tasks. Keys marked with a * accept expressions as value.

Add criter... +

Key

Value

Rem... x

Candidate User *

demo

E.g.: `\${currentUser()}`

☐ Include assigned tasks

Also include tasks that are assigned to users in candidate queries.

Default is to only include tasks that are not assigned to any user if you query by candidate user or group(s).

of the criteria are met.

Permissions

Variables

[Delete filter](#)

[Close](#)

[Save](#)

Save the filter configuration.

Start the Process

Start process

Click on the process to start.

Process User Task Candidate User Demo

Close

Claim and complete user task:

Camunda tasklist

Create a filter + <

Created ▾ +

My Tasks

My Group Tasks

Accounting

John's Tasks

Mary's Tasks

Peter's Tasks

Task by Candidate user (1)

All Tasks

Filter Tasks

User Task

Process User Task Candidate User Demo

Created 8 minutes ago

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

User Task with Candidate Groups:

Task_14qmcj3

General

Forms

Listeners

Input/Output

Extensions

General

Id

Task_14qmcj3

Name

User Task

Details

Assignee

Candidate Users

Candidate Groups

accounting

Due Date

The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Follow Up Date

The follow up date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Deploy the process

Start the process

Start process

Search by process name.

Click on the process to start.

Process User Task Candidate Group Demo

Process User Task Candidate User Demo

Close

My Tasks

My Group Tasks (1)

Accounting

John's Tasks

Mary's Tasks

Filter Tasks

1

User Task

Process User Task Candidate Group Demo

Created 4 minutes ago

50

Task Detail

User Task

Process User Task Candidate Group Demo

Set follow-up date

Set due date

Accounting

Claim

Form

History

Diagram

Description

First name

Last name

Save

Complete

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

Task_14qmcj3

General

Forms

Listeners

Input/Output

Extensions

General

Id

Task_14qmcj3

Name

User Task

Details

Assignee

Candidate Users

Candidate Groups

accounting

Due Date

2019-12-22T12:00:00

The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Follow Up Date

2019-11-22T11:00:00


The follow up date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-26T09:54:00)

Deploy the process and Start process

User Task

Process User Task Candidate Group Demo [🔗](#)

 in 4 days ×

 in a month ×

 Accounting

 Claim

Form History Diagram Description

First name

Last name

Save

Complete

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. <code><camunda:constraint name="required" /></code>
minlength	Applicable to string fields. Validates minlength of text content. Accepts 'null' values. eg. <code><camunda:constraint name="minlength" config="4" /></code>

maxlength	<p>Applicable to string fields. Validates maxlength of text content. Accepts 'null' values.</p> <p>eg. <camunda:constraint name="maxlength" config="25" /></p>
min	<p>Applicable to numeric fields. Validates the min value of a number. Accepts 'null' values.</p> <p>eg. <camunda:constraint name="min" config="1000" /></p>
max	<p>Applicable to numeric fields. Validates the max value of a number. Accepts 'null' values.</p> <p>eg. <camunda:constraint name="max" config="10000" /></p>
readonly	<p>Applicable to all type. Makes sure no input is submitted for given form field.</p> <p>eg. <camunda:constraint name="readonly" /></p>