

# Camunda

Automation of recurring workflows

JUNE 2021

# Introduction

Camunda Platform is a light-weight, open-source platform for Business Process Management. This Platform ships with tools for creating workflow and decision models, operating deployed models in production, and allowing users to execute workflow tasks assigned to them.

---

## Features

Workflows in Camunda are defined in BPMN which is basically an XML file. It can be graphically modeled using the *Camunda Modeler*. Camunda can not only call services right away (Push-Principle) using some built-in connectors, but also put work items into a kind of built-in queue. Then a worker can fetch work items via REST, do the work and let Camunda know of the completion. So first you fetch tasks and lock them for you (as other workers might fetch at the same time to scale your system) and once done, tell Camunda the worker has completed its work.

---

## Use Case

Camunda can be used to efficiently complete jobs and tasks within a team or group. In our use case of insertion into database, Camunda would provide the following workflow:

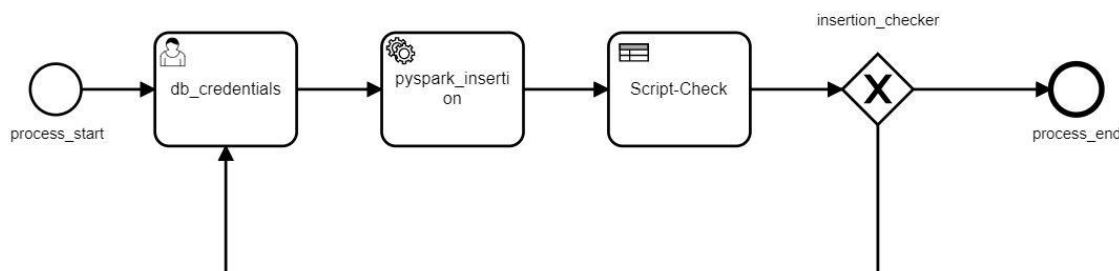
- Design a workflow that does dataframe creation, insertion, and checking the inserted data for correctness.
- The Camunda Admin (team head) can then create workflow instances for the various insertion jobs, and these jobs can be pre assigned to users(team members), or made open and can be claimed by users.
- Once a user(employee) is assigned a user task, he can work on that specific task. We can include timed deadlines. Once the user is done with his task, he can fill the task parameters and mark his task complete.

- Next, Camunda allows a service task that will take as input the data that was provided by the user in the previous task and use it to perform the insertion. If the data provided is correct, the script would run fine and we would get response OK, else Error.
  - If the response is fine, the user task for the user is gone (finished), and the job is done correctly. Else, the control goes back to the user task where the user has to again enter his script/data.
- 

## Comparison with PySpark

- Pyspark SQL Framework works as a distributed SQL query engine, and provides very efficient query performance.
  - Pyspark is a diverse framework that provides efficient components like Streaming, MLlib that can be integrated with Pyspark SQL for future uses.
  - Camunda can serve well for automation of the workflow, but we can't comment on the throughput and efficiency without deploying and testing for varying sized tables of a database.
  - To take advantage of Pyspark, we can use the Pyspark SQL engine in the Query Processing task of Camunda while designing the workflow. This would make the query processing faster, and also support automation.
- 

## Using PySpark with Camunda



## Overview:

- For every instance of workflow created by admin (Eg. Mr. Sonu), a user task is the first element, where the user (assigned employee/Apurv) will enter the Database details like DatabaseName, ServerName, TableName, Credentials etc. He/She will also provide his created Pyspark Dataframe as a parameter.
  - The external [service task](#) will perform the insertion in the specified location and return a response parameter. We will use the External Service Task with [camunda-external-task-client-python3](#) to create a python script that will just take as input a PySpark dataframe, perform the insertion in a try-catch block and return the response as SUCCESS or FAILURE.
  - The next task will check the response parameter to check if the insertion task was done correctly and successfully or not. We can also include additional checks which are used for checking the insertion. This will be a DMN where we can include multiple checks. Finally this task would return as output a variable which will give the final result.
  - The gateway then compares this result value. If result = 1, it takes us to the end of the process (the task is completed successfully), or back to the user task if result = 0 (Some Error).
  - If the process ends, then that instance of the workflow is gone. Now, the admin can create a fresh instance for the next job and assign it to a group member. He will receive notifications which can inform him whether the task was completed before deadline/pending.
-