



# **Workflow Engine**

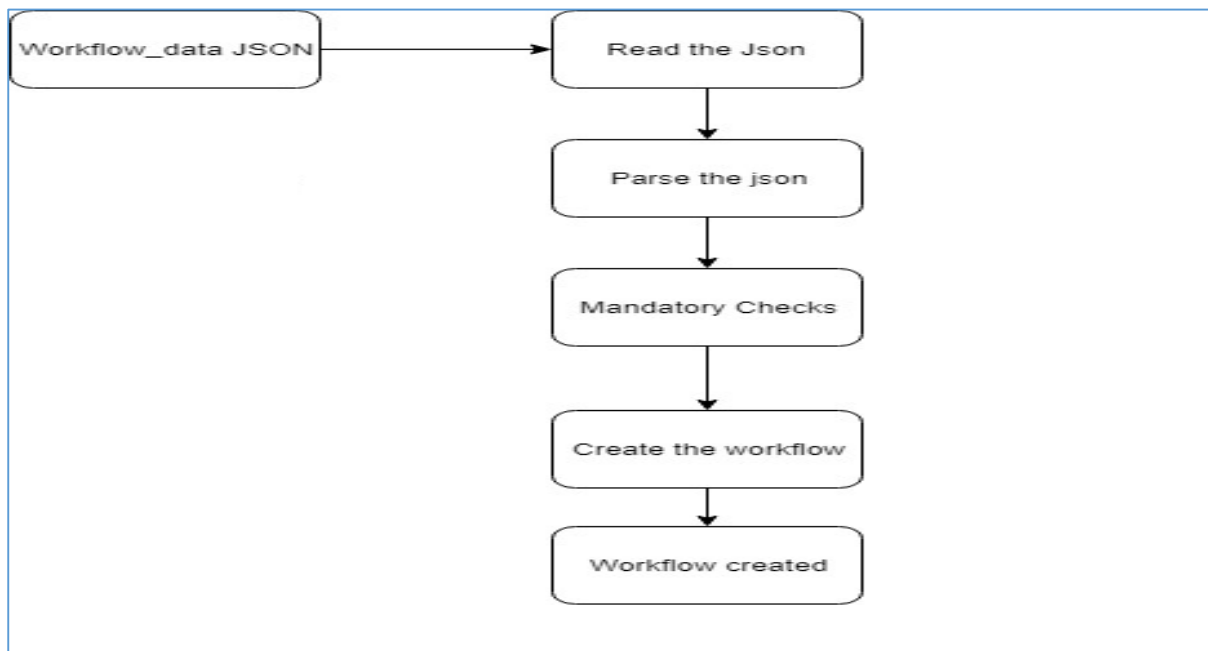
**Version 1.0**

Prepared by: SyRA Team

Dated: Jan 14, 2019

State: Draft

## Workflow Engine



The above diagram shows the flow of the workflow engine.

The workflow\_data.json will work as the configuration of the workflows. On start of the application the it will read the json, parse it, validate it and create the workflow. Each workflow will have an unique id. In order to execute the workflow, external application can either do a rest call with the flow id, or the whole project could be imported as jar and the service could call the workflow service in the jar.

## Workflow Engine Configuration Guide

### Workflow Engine configuration: -

For consuming generic Workflow Engine, we need to configure 3 files, which are mentioned below:

1. workflow\_data.json.
2. Create Job tasks by implementing WorkflowAction.
3. Create Decision tasks by implementing DecisionAction.
- 4.Using the Workflow context to share the data between the workflow

**Workflow\_data.json:** - This is the main configuration file for the workflows. It consists of following fields

1. **workflows:** it's a collection of workflow.
2. **flowId :** This field is mandatory .This field identifies the workflow. Its value should be unique.
3. **startNode :** This field is mandatory. It tells the workflow which is the starting node. The name should be same as one of the **jobName** field.
4. **flowDescription:** This field is optional. It's a description of what the workflow does.
5. **flowNodes:** It's a collection of workflow nodes.
6. **type:** This field is mandatory for each node. Its values are **PROCESS** or **DECISION**
7. **jobName:** This field is mandatory. It corresponds to the task name that will to be performed on workflow execution. There should be a corresponding JobTasks code with the same name. Also this name should be unique in the all the workflows
8. **nodeDescription :** This field is optional .It gives a brief overview of the task that is performed by the node.
9. **normalExit :** This field is conditional. It is mandatory if the **type** of the node is **PROCESS**. This name should be same as the **jobName** in the same workflow. It describes the next node the workflow should execute after executing the current node
10. **position :** This is a mandatory field .It determines the node position. It can be **START,MIDDLE,END**. A workflow should have a start ,middle and end.
11. **decisionTask :** It describes the action to be undertaken to come to decision flow. There should be a corresponding decision Job in the code. This field is conditional. If type of the node is **DECISION**, then this field is mandatory.
12. **conditionalTrue :** This field determines which flow node to execute if the **DECISION** is true. This field is conditional .If the type is **DECISION** then this field is mandatory.
13. **conditionalFalse :** This field determines which flow node to execute if the **DECISION** is false. This field is conditional .If the type is **DECISION** then this field is mandatory.



workflow\_data.json

## Sample json

```
{
  "workflows": [
    {
      "flowId": "id-delete",
      "startNode": "TestJob",
      "flowDescription": "Testing workflow 1",
      "flowNodes": [
        {
          "type": "PROCESS",
          "jobName": "TestJob",
          "nodeDescription": "Perform Task A",
          "normalExit": "TestJob2",
          "position": "START"
        },
        {
          "type": "DECISION",
          "jobName": "TestJob2",
          "nodeDescription": "Perform Task B",
          "decisionTask": "DecisionA",
          "conditionalTrue": "TestJob3",
          "conditionalFalse": "TestJob4",
          "position": "MIDDLE"
        },
        {
          "type": "PROCESS",
          "jobName": "TestJob3",
          "nodeDescription": "Perform Task C",
          "normalExit": "TestJob5",
          "position": "MIDDLE"
        }
      ]
    }
  ]
}
```

### Create Job tasks by implementing WorkflowAction : -

This job corresponds to the task a node should perform. The name should be unique and the name should match with one of the jobName in the nodes .

### Create Decision tasks by implementing DecisionAction : -

This job corresponds to the decision task that needs to be undergone once decision needs to be undertaken, its return type is Boolean. Based upon Boolean value the workflow will determine if conditionalTrue task needs to be executed or conditionalFalse.

### Using the Workflow context to share the data between the workflow :-

Workflowcontext is the memory context for the workflows. Its memory space is unique for individual workflows. i.e each workflow is allocated individual memory. Its generally a hashtable or dictionary memory where individual can generate or consume data generated by other nodes.