# Process.json Explanation

**Description:**
Process.json is used to capture the types of data in each tool and their workflow so that you can define what types of data needs to be entered into the ledger for each tool.

**General structure:**
These are mandatory fields. If you want to capture all types of data, then use the below template changing the Id and Tool accordingly.

```
{
    "Id": "102",
    "Types": "",
    "Tool": "Git",
    "SubworkflowFlag": false,
    "insertIntoLedger": true
  }
```

**Types:**

If the tool has types such as types of issues in jira, and you want to control their insertion into ledger, define them as follows:

```
{
    "Id": "101",
    "Tool": "Jira",
    "Types": [
     {
      "Id": "201",
      "Entity": {"issueType" : "Story"},
      "SubworkflowFlag": true,
      "Subworkflow": {
       "Entity": "toolstatus",
       "Steps": [
        {
         "Entity": "To Do",
         "insertIntoLedger": true
        },
        {
         "Entity": "In Progress",
         "insertIntoLedger": true
        },
        {
         "Entity": "Done",
         "insertIntoLedger": true
        }
       ]
      }
     }
  ]
}
```

- Each type has an Id field. The *Entity* field has to define the type. Example: The jira issue types that are captured in the field “issueType” according to datamodel.json.
- If the *SubworkflowFlag* is set to false, there is no need to define the *Subworkflow* field.
- If there is no subworkflow field, the *inserIntoLedger* field has to be defined for the type alone.

Example:

```
{
 "Id": "101",
 "Tool": "Jira",
 "Types": [
  {
   "Id": "201",
   "Entity": {
    "issueType": "Story"
   },
   "SubworkflowFlag": false,
   "inserIntoLedger": true
  }
 ]
}
```

**SubworkFlow:**

Subworkflow is used to have workflow level control over the data entered. **Subworkflow can exist within or without the *Types* field**. For example, each jira issue can have a workflow like *To Do, In Progress, Resolved, Done, etc* and each state can be defined to be entered into ledger of not.

Example:

```
"Subworkflow": {
 "Entity": "toolstatus",
 "Steps": [
  {
   "Entity": "To Do",
   "insertIntoLedger": true
  },
  {
   "Entity": "In Progress",
   "insertIntoLedger": true
  },
  {
   "Entity": "Done",
   "insertIntoLedger": true
  }
 ]
}
```

- The *Entity* field has to define the fieldname that captures the workflow change in this case *toolstatus* which changes from *To Do* to *In Progress* to *Done*.