# The Orchestrated Workflow Engine

This API application provides endpoints for creating, managing, and deleting an orchestration workflow. It allow for items (trackables)

to be submitted into and managed through the lifecycle of an orchestration workflow. To access the endpoints you will need to have a

HTTP Header key = user-key with a value of "test" in each of your API calls.

###What is an orchestration workflow?

An orchestration workflows is a structured representations of a business processes. it defines a logical flow of activities or tasks from

a start event to an end event to accomplish a specific purpose or service.

###What is an example of an orchestration workflow (OW) in action?

One common example is the process of reviewing and approving documents. In this example the OW defines the logical flow or path a

document would travel through in it's journey to be 'approved.' First a document is created, then submitted into a previously defined

OW. As different actions are performed (physical review by individuals; an amount of time has passed, a computerized event occurs),the

document is advanced (or retarded) to the next step(s) in the OW. In this example the specific purpose is to produce an "approved"

document. this may require various changes and re-reviews, all of which are tracked and facilitated by the OW.

###Do orchestration workflows support multi-directional paths?

Yes, orchestration workflows support multiple directions and multiple concurrent and parallel paths.

###Do orchestration workflows support single dimensional and multi-dimensional workflow paths?

Yes, the OW engine supports single dimensional and multi-dimensional workflow paths. Common single dimensional paths are where one task

leads to another task and another in a serial mannor. Multi-dimensional paths commonly are where one completed task can simultaneouly

spawn multiple other tasks that are managed in parellel. Additionally, in the multi-dimensional scenario a task that is yet to be queued

may require the completion of multiple unrelated tasks to be complete before that task can be attempted.

###How are orchestration workflows represented in this tool?

Workflows are created by submitting a JSON object that represent the structure and rules of the workflow. Below are the JSON Schema and

examples of both a Simple and Complex workflow. Upon submission workflows are validated. You can pre-validate by using the

API/Workflow/Validate api. Workflows must have at least one starting node and at least one ending node. Path validation is also required,

where every node must have at least one entry point and one exit point. the exceptions are, starting nodes do not need a explicitly entry

paths, and ending nodes do not need to have explicit exit paths.

Workflow JSON Schema

```JSON

{

"title": "Workflow Schema",

"type": "object",

"properties": {

"workflowId": {

"description": "The unique identifier for a product",

"type": "string"

},

"key": {

"description": "used for testing, surrogage for unique key during testing, remove",

"type": "string"

},

"workflowName": {

"description" : "human friendly name give to thw workflow",

"type": "string"

},

"nodes": {

"description": "a collection of nodes that represent a resting state in which items belong during thier journey though the workflow",

"type": "array",

"items": {

"title": "node",

"description" : "a resting state of a workflow that contains trackable items",

"type": "object",

"properties": {

"nodeName": {

"type": "string"

},

"nodeDescription": {

"type": "string"

},

"nodeId": {

"description": "node unique identifier",

"type": "string"

},

"isStart": {

"description": "is this a starting node for an orchestrated workflow",

"type": "boolean",

"optional": true

},

"isEnd": {

"description": "is this an ending node for an orchestrated workflow",

"type": "boolean",

"optional": true

}

}

}

},

"path": {

"description": "Array of available movements with the workflow",

"type": "array",

"items": {

"title": "movements",

"description": "defines a single available movement between steps in a workflow",

"type": "object",

"properties": {

"from": {

"description": "starting NodeName for movement",

"type": "string"

},

"to": {

"description": "ending NodeName for movement",

"type": "string"

},

"approveUsers":{

"description": "Array of users that can execute movement along this path"

"type": "array",

"items":{

"description": "User object"

"type": "object",

"parameters":

{

"email": {

"type":"string"

}

}

}

}

}

}

},

"required": [ "WorkflowId", "Key", "workflowName" ]

}

```

Workflow Example with simple steps and one item being tracked in first step

```JSON

{

"workflowId": "173b1e83-6d84-446b-a9bc-5c2e952e659d",

"key": "SampleWorkflow1",

"workflowName": "SampleWorkflow1",

"nodes": {

"SampleNode1": {

"nodeName": "SampleNode1",

"nodeDescription": null,

"nodeId": null,

"isStart": true,

"isEnd": false

},

"SampleNode2": {

"nodeName": "SampleNode2",

"nodeDescription": null,

"nodeId": null,

"isStart": false,

"isEnd": false

},

"SampleNode3": {

"nodeName": "SampleNode3",

"nodeDescription": null,

"nodeId": null,

"isStart": false,

"isEnd": false

},

"SampleNode4": {

"nodeName": "SampleNode4",

"nodeDescription": null,

"nodeId": null,

"isStart": false,

"isEnd": true

}

},

"path": [

{

"from": null,

"to": "SampleNode1",

"approveUsers": [

{

"email": "Sample.User@somewhere.com"

}

]

},

{

"from": "SampleNode1",

"to": "SampleNode2",

"approveUsers": [

{

"email": "Sample.User@somewhere.com"

}

]

},

{

"from": "SampleNode2",

"to": "SampleNode3",

"approveUsers": [

{

"email": "Sample.User@somewhere.com"

}

]

},

{

"from": "SampleNode3",

"to": "SampleNode4",

"approveUsers": [

{

"email": "Sample.User@somewhere.com"

}

]

}

],

```

### What can a orchestrated workflow track?

Any type item can be managed through an orchestration workflow. The actual details of the item should be stored in a system other than

the orchestration workflow system. However, the submission of an unique identifier is required for the orchestration workflow to track

and facilitate the item through the OW. It is recommended that this unique identifier be used to match the items native information

with it's location and status within the orchestration workflow. Below is a JSON schema and JSON example of a trackable item which is

required for submission to any orchestrated workflow.

Trackable Json Schema

```JSON

{

"trackables": {

"description": "array of items within the given node",

"type": "array",

"items": {

"title": "trackable",

"description": "object being tracked within this step",

"type": "object",

"properties": {

"trackableId": {

"description": "unique identifier for this item",

"type": "string"

},

"trackingName:" {

"description": "human friendly name for this item",

"type": "string"

},

"key": {

"description": "unique identifier for this item",

"type": "string"

},

"Location": {

"type"="array",

"description": "List of locations that trackable is in.",

"items": {

"type": "object",

"description": "workflowId and NodeId that designates a location of item",

"parameters": {

"workflowId": "string",

"nodeId": "string"

}

}

}

}

}

}

}

```

Trackable JSON example

```

{

"trackableId": "SampleDoc2",

"key": "SampleDoc2",

"trackingName": "SampleDoc2",

"locations": []

}

```

### How do I submit something to be managed by an orchestration workflow?

Items are submitted to an orchestrated workflow through transactions. A transaction defines a workflow, a trackable item, and the

action that is being requested. Actions include; Move, Copy, and Comment. Each action is evaluated against the rules set forth by

the defined path in the given workflow. Below is the JSon schema and a JSON example of a transaction.

Transaction JSON Schema

```JSON

{

"Transaction": {

"type":"object",

"description": "represents something done within a workflow to a trackable Item.",

"properties": {

"Key": "string",

"trackableId": "string",

"type": "string",

"comment": "string",

"previousNodeId": "string",

"newNodeId": "string",

"transactiontime": {

"type": "string",

"readonly": "true"

},

"submitterId": "string",

"workflowId": "string"

}

}

}

```

Transaction JSON Example

```

{

"key": "SampleTransaction0-2",

"trackableId": "SampleDoc2",

"type": 0,

"comment": "Moving to step 3",

"previousNodeId": null,

"newNodeId": "SampleNode1",

"submitter": {

"email": "Sample.User@somewhere.com"

},

"workflowId": "SampleWorkflow1"

}

```

### Submitting and Managing items through a orchestration Workflow

To submit an item to an orchestration workflow you simply Post a transaction JSON object to the API/transaction.

###What RESTful services are available to me

### decidedly not holding to true RESTful Textbook but make for practicality.

[GET]

API/workflows - returns all workflows managed by the application.

API/workflows/{workflowId} - return the specific workflow.

API/workflows/example - return an example workflow to get you started

## below maybe not needed / wanted

API/workflows/{workflowId}/node/{nodeId}/trackables - returns all the trackable in that workflow node

API/workflows/{workflowId}/orchestrations

API/workflow/{workflowId}/orchestration/{orchestrationId}

API/workflows/{workflowId}/nodes - return all the nodes in a specific workflow.

API/workflows/{workflowId}/nodes/{nodeId} - return all the nodes in a specific workflow.

## end of above

##API/workflow/{workflowId}/orchestration/rule?

API/trackable - returns all the trackable managed by the application

API/trackable/{trackableId} - returns that specific trackable

API/trackable/example - return an example of a trackable to get you started.

API/trackable/{trackableId}/transaction - returns the transaction history for the given trackable

##API/trackable/newId - will return a uniquely new ID for inclusion in a trackable POST

##API/trackable/availablemoves - returns the workflow NodeId's that are valid for the next moves.

API/transaction - returns all transactions

API/transaction/{transactionId} - returns a specific transacation

API/transaction/example - returns an example transaction to get you started.

\*\*API/trackables/{workflowId}/paths/{pathId}/trackables - returns all the trackables that went through that path

\*\*API/trackables/{workflowId}/nodes/{nodeId}/trackables - returns all the trackable in that workflow node

API/trackables - returns all the trackable managed by the application

API/trackables/{trackableId} - returns that specific trackable

API/trackables/{trackableId}/transaction - returns the transaction history for the given trackable

API/trackables/newId - will return a uniquely new ID for inclusion in a trackable POST

API/trackables/availablemoves - returns the workflow NodeId's that are valid for the next moves.

API/transactions - returns all transactions

API/transactions/{transactionId} - returns a specific transacation

API/transactions/example - returns an example transaction to get you started.

[POST]

API/workflow - submits the workflow definition to the OW engine.

API/trackable - submits a trackable to the OW engine

API/trackable/isunique/{trackableId} - varifies if a trackableId is new to the OW engine (used for prevalidation).

API/transaction - submits a transaction to the OW engine

API/trackable - Registers a trackable to the OW engine

API/trackable/isunique/{trackableId} - varifies if a trackableId is new to the OW engine (used for prevalidation).

API/transaction - submits a transaction to the OW engine with will manipulates a trackable within OW engine.

[PUT]

API/workflow/{workflowId} - updates an existing workflow definition

API/trackable/{trackableId} - updates an existing trackable

[Delete]

API/workflow/{workflowId} - removes an existing workflow definition from the OW engine

API/trackable/{trackableId}

As a user I want to beable to define a workflow orchestration that represents my business process so that i can use to manage items through that process.

as a user i want to be able to submit items to my defined workflow orchestration.

as a user I want to be able to move item(s) through the steps of my workflow.

as a user I want to be able to remove an item from the workflow.

Wanting search capabilities so that I can satisfy the following use cases:

As a user I want to see all the items in a particular node of a given workflow so that i can understand how many things are waiting review/approval of a particular step

As a user I want to see what node a particular trackable is within a given workflow so that I can understand what needs to happen next for a particular trackable

As a user i want to see the history of a particular item has been through or going through a given Workflow so that i can see the progress an item has made.

As a user i want to see all the items that have passed through a particular node in a workflow in a given timeframe so that see how much work has been accomplished by that step in the process

As a user I want to see all the items that a particular user has approved or denied in given workflow for a given timeframe so that i can understand the amount of work an individual has done.

As a user i want to see how long a given item has spent in a particular (node) in a workflow so that I can understand if there are bottlenecks

As a user I want to see how long a given item has spent in a particular workflow so that i can understand how long something has been under review.

API/Search/q=workflow&id=2

API/Search?select={selectKeywork}&where={whereKeyword}={value}

Key words for Search

Select Key words determine the document that is returned:

* Workflows (API/Search?select=workflows)
  + Returns all the workflows
* Trackables (API/Search?select=trackables)
  + Returns all the trackables
* Transactions (API/Search?select=transactions)

Where key words

* WorkflowId (API/Search?select=transaction&workflowId=SampleWorkflow1)
  + Returns all the transactions applying to workflow with Id ‘SampleWorkflow
* TrackableId (API/Search?select=Node&workflowId=SampleWorkflow1)
* TransactionId (API/Search?select=transaction&workflowId=SampleWorkflow1)
* NodeId (API/Search?select=trackables&nodeId=SampleNode1)
  + Returns all the trackables in the give Node.
* Start (API/Search?select=transaction&start=01-01-2017)
  + Returns all transactions after that occurred after 01-01-2017
* End

Search parameters

| Argument | Example | Description |

|----------|----------------------------|------------------------------|

| t | t=wf | Type of object you are wanting [Worflow; orchestration; Trackable;

| id | id=workflowId |

| nane |