Crux: BRE/WFE - Task #17854

WSC: WFinstanceNew()

15/02/2024 09:40 AM - Sachin Divekar

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Kanchan Barve	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Sprint 29 Mar		

Description

This creates a new instance of a workflow. This is called when one entity (*i.e.* one object instance) begins its journey through a workflow.

Request

```
{
    "slice": 7,
    "entityid": "0eb8da50-aece-11ee-b168-3b192f7cd2b6",
    "entity": {
        "class": "salesvoucher",
        "voucherdate": "2024-05-20T00:00:00Z",
        "branch": "belampally",
        "voucheramt": "84213"
    },
    "app": "retailsales",
    "workflow": "customeronboarding",
    "trace": 0,
    "parent": 952
}
```

where:

- slice: integer, specifies a realm-slice in whose context the workflow engine must operate
- entityid: mandatory, string, a unique ID for this specific entity instance which is about to start its workflow traversal. Note that Crux will not ensure the uniqueness of this value in its own tables -- this value is for the application code, and is supposed to help it map the entity with the workflow traversal stages it is going through.
- entity: a JSON object with a set of fields and values. These fields must have names defined in the workflow schema and the values are the ones which the workflow engine will use to traverse the workflow. All the fields shown above are purely examples -- they are not specs of Crux. They illustrate how a sample entity's attributes may be specified. The names of the fields, like voucherdate, branch, *etc* must match the attributes in the for this class. One field is mandatory and reserved: the class. The value of class here must match the value of class of the workflow specified. Here are more details of how an entity is represented.
- app: the application ID to which this workflow belongs
- workflow: the name of the workflow where the workflow engine will start the traversal
- trace: optional, integer, 0 means no trace, 1 means high level trace, and 2 means detailed trace of the workflow engine's processing. If the parameter is omitted, it means the caller wants no trace (trace = 0).
- parent: optional, integer, an ID of some other workflow instance, returned from an earlier call toWFinstanceNew(). If the request includes this, it is indicating to Crux that this new instance is a child of an earlier instance, *i.e.* this instance is being initiated because another instance indicated that one of its steps was in fact a sub-workflow (see the stepworkflow table).

Processing

Any authenticated user may make this call, from one of the permitted IP addresses.

The call will verify

- the entity must conform to the structure given in this section
- all attributes of entity match the names in the schema
- no attributes are missing from entity
- all values of the fields in entity match types of the attributes as specified in the schema
- the value of app specified in the request matches the app ID with which this workflow is associated
- the workflow named has is_active == true and internal == false

- the class of the workflow must match that of entity
- there is no record in the wfinstance table with the same values for the tuple (slice,app,workflow,entityid). One entity cannot traverse one workflow multiple times in parallel -- it's a logical inconsistency.

If any of these conditions fail, the call will return with a suitable error immediately. Else it will add two attributes to the entity object:

```
"step": "START",
"stepfailed": false
```

Then it will check whether all the rulesets and schema belonging to the slice specified in the request are loaded into the in-memory cache. If they are not present, the call will load them.

Then it will call doMatch() with:

- entity: the entity submitted in the request
- ruleset: a pointer to the in-cache data structure holding the workflow
- actionset: an empty action set data structure
- seenruleset: an empty array
- trace: integer, as supplied in the request

When doMatch() returns, the call will check whether doMatch() has returned with a critical error. If yes, then it sends an error response to the client with the appropriate details. If there is no critical error, then it will inspect the tasks and properties of the actionset returned. The properties object of actionset must always contain either an attribute called done or an attribute called nextstep. If neither is present, or if a third attribute is present, then the call must log a critical error and return this error in its response. If all this is ok, then it will insert records in the wfinstance table as per the following scheme:

- if there is an attribute done in properties of actionset, this means that the workflow has been traversed to its end, and nothing needs to be done. In that case, no record will be inserted in wfinstance.
- If there is only one name in tasks, then there is at this point only one step to be performed. Therefore, one record will be entered into the table, as detailed below.
- if the actionset.tasks contains multiple step names, then this means that these steps may be executed concurrently. In that case, multiple records will be inserted, one for each task.

New records in the wfinstance table will have the following values

- instanceid: the entityid from the request
- slice: from the request
- app: from the request
- class: from the class field of the ruleset being executed
- workflow: from the request
- step: the value of one of the tasks returned in the tasks array of actionset. If only one task is returned in this array, then one record will be created. If multiple tasks were returned in the tasks array, then the corresponding number of records will be created, one for each task.
- nextstep: the value of the nextstep attribute in the properties object in actionset. If only one task was returned in tasks, then the value of nextstep will be equal to that task name. If multiple tasks were returned, then nextstep will have some other name which may have no connection with the task names listed.
- doneat: set to null
- parent: will be set to null, thereby indicating that this workflow instance is not a sub-workflow of any other workflow.

It will then look up the stepworkflow table to see if any of the task values obtained from actionset.properties.tasks are specified in stepworkflow. (There could be multiple values in the tasks set.) The stepworkflow table specifies if one of the steps, say S1, returned by workflow X is itself to be executed by traversing another multi-step workflow, *e.g.* workflow Y. In that case, the name Y is specified in stepworkflow against step S1. If this is the case, then this information needs to be returned to the caller.

Response

If all goes well, and it is a simple case where there is just one next step to be executed, the response will be:

```
{
    "tasks": [ {"passportchk": 8734} ],
    "loggedat": "2025-09-04T19:05:34Z",
    "subflows": {
        "passportchk": "dopassportworkflow"
    },
    "tracedata": "{...}"
}
```

where everything is mandatory other than subflows, done and tracedata. The tasks array will have a set of objects, one per next step. In this example, there is only one next step, therefore, there is just one object. The name of the attribute of that object is passportchk, which is the name of the step to execute, and the value is 8734, which is the ID of the row in wfinstances corresponding to this entry. Trace data will only be present, in the form of a large JSON block, if it has been requested.

It is possible that the call returns with the indication that the workflow traversal has been completed. In this case, the response will only contain:

```
{
    "done": true
}
```

This parameter informs the caller that this entity has reached the last step in its workflow and there is nothing more to be done. It is rare in a real system for WFinstanceNew() to get a response of "done": true, since this call is just the start of a workflow traversal. In cases when done = true is being returned, tracedata may optionally be returned too, to indicate the matching engine's process to arrive at this conclusion.

If the workflow has thrown up a set of concurrent steps which need to be performed, the response will be of the form:

```
{
    "tasks": [
        {"passportchk": 8734},
        {"degreechk": 8543},
        {"updategeneralregister": 8611}
    ],
    "nextstep": "step2done",
    "loggedat": "2025-09-04T19:05:34Z",
    "subflows": {
        "passportchk": "dopassportworkflow"
    }
}
```

where everything will be mandatory and present, other than subflows, done and tracedata. The nextstep member will not be present when the tasks array has just one next step, and will always be present if tasks returns multiple steps. The tasks array has an ID against each next step.

The subflows object here is shown specifying the workflow name to call to execute the step called passportchk. The subflows block will only be present if there is a record for passportchk in the stepworkflow table. This attribute has no connection with whether tasks specifies one or multiple next steps.

It is expected that the application code will take note of this sub-workflow indication and will trigger a separate workflow for this entity for dopassportworkflow by making a separate call to WFinstanceNew() and using the same instance ID which it had used before.

Associated revisions

Revision 6d03dfa6 - 20/03/2024 07:07 PM - Kanchan

refs : #17854 - DoMatch() Integration completed.

Revision c21837c7 - 20/03/2024 07:10 PM - remiges-kanchan

Merge pull request #151 from remiges-kanchan/wsc

refs : #17854 - DoMatch() Integration completed.

Revision ffaf4aa5 - 21/03/2024 06:22 PM - Kanchan

refs :#17854 - validation for patternschema against entity added

Revision 527bd576 - 21/03/2024 06:23 PM - remiges-kanchan

Merge pull request #155 from remiges-kanchan/wsc

refs :#17854 - validation for patternschema against entity added

Revision 80d8cf32 - 05/04/2024 10:41 AM - Kanchan

refs : #17854 - Error handling added in AddTask method

Revision 96c2754e - 05/04/2024 10:42 AM - remiges-kanchan

Merge pull request #167 from remiges-kanchan/wsc

refs : #17854 - Error handling added in AddTask method

Revision 0a7b4ede - 29/04/2024 06:59 PM - remiges-kanchan

refs : #17854 - changed seed data and modified test cases for WFInstanceNew added

Revision 5891a994 - 29/04/2024 07:00 PM - remiges-kanchan

Merge pull request #193 from remiges-kanchan/wsc

refs : #17854 - changed seed data and modified test cases for WFInstanceNew added

Revision 9472f993 - 14/05/2024 02:49 PM - remiges-kanchan

refs : #17854 - modified WFinstancen'ew API test cases

Revision 1bcee601 - 14/05/2024 02:50 PM - remiges-kanchan

Merge pull request #199 from remiges-kanchan/wsc

refs : #17854 - modified WFinstancen'ew API test cases

History

#1 - 17/02/2024 05:26 PM - Shuvam Misra

- Assignee set to Kanchan Barve

#2 - 17/02/2024 05:26 PM - Shuvam Misra

- Status changed from New to In Progress

#3 - 17/02/2024 05:26 PM - Shuvam Misra

- Target version set to Sprint 29 Feb

#4 - 19/02/2024 07:18 PM - Kanchan Barve

- % Done changed from 0 to 80

- Added test cases

#5 - 21/02/2024 11:03 AM - Kanchan Barve

- Status changed from In Progress to On Hold

#6 - 27/02/2024 07:09 PM - Kanchan Barve

- Added data change logs
- Added validation to request parameters

#7 - 08/03/2024 05:31 PM - Shuvam Misra

- Target version changed from Sprint 29 Feb to Sprint 15 Mar

#8 - 15/03/2024 12:01 PM - Kanchan Barve

- Status changed from On Hold to In Progress

#9 - 15/03/2024 09:08 PM - Kanchan Barve

- integration with Matching engine

#10 - 18/03/2024 06:37 PM - Kanchan Barve

- facing issues while calling DoMatch() :- can't get ruleset for particular combination even if data is present still getting error.
- All rulesets and schema's are accessible.

#11 - 19/03/2024 12:56 PM - Kanchan Barve

- Status changed from In Progress to On Hold

#12 - 19/03/2024 02:28 PM - Kanchan Barve

- Status changed from On Hold to In Progress

#13 - 19/03/2024 07:05 PM - Kanchan Barve

- DoMatch() integeration is completed.

#14 - 20/03/2024 07:14 PM - Kanchan Barve

- Added New method in *cacheUtils* for fetching only workflows from *rulesetCache* .
- In progress - validating entity attributes against patternschema as schema_t data struct has been changed now.

#15 - 21/03/2024 12:12 PM - Kanchan Barve

- Status changed from In Progress to Testing

#16 - 21/03/2024 06:39 PM - Kanchan Barve

- % Done changed from 80 to 90

- validating entity against patternschema completed.
- testing for single task and multiple task using single ruleset done.
- In progress testing using multiple rulesets.

#17 - 26/03/2024 07:25 PM - Shuvam Misra

- Target version changed from Sprint 15 Mar to Sprint 29 Mar

#18 - 16/04/2024 11:38 AM - Sachin Divekar

- % Done changed from 90 to 100

- Status changed from Testing to Closed

#19 - 29/04/2024 06:54 PM - Kanchan Barve

-changed seed data for wfinstancNew and modified test cases accordingly