

Tellus Justifier fields and data source specification

Monday, June 09, 2014 2:41 PM

Justifier Integration Related Fields

http://tellusdev/index_justifier2_rollout_testOk.php

System (field)

Values: Tellus, Online, Atlas, etc.

When value is "Tellus", it does not query for Justifier builds.

Server query URL to populate this field:

qJustifierSystem.php

Data source:

\$url = "[http://iswebstga03.pc.factset.com:8090/justifier/api/BuildSystems/?\\$filter=active%20eq%20true&\\$select=Id,Name&\\$orderby=Order&\\$format=json](http://iswebstga03.pc.factset.com:8090/justifier/api/BuildSystems/?$filter=active%20eq%20true&$select=Id,Name&$orderby=Order&$format=json)";

Example data:

```
[{"$id": "1", "Id": 0, "Name": "Online", "Order": 2}, {"$id": "2", "Id": 11, "Name": "Atlas", "Order": 3}, {"$id": "3", "Id": 10, "Name": "FI - FI/PA Analytics Testing", "Order": 6}, {"$id": "4", "Id": 5, "Name": "FIA - Release Testing", "Order": 7}, {"$id": "5", "Id": 4, "Name": "FI - Weekly Analytics", "Order": 8}, {"$id": "6", "Id": 8, "Name": "Prebuild", "Order": 9}, {"$id": "7", "Id": 14, "Name": "FIM - Instant Messenger", "Order": 10}, {"$id": "8", "Id": 2, "Name": "Real Time Client", "Order": 11}, {"$id": "9", "Id": 13, "Name": "Real Time DataFeed", "Order": 12}, {"$id": "10", "Id": 3, "Name": "Real Time Server", "Order": 13}, {"$id": "11", "Id": 9, "Name": "Real Time Server Automation", "Order": 14}, {"$id": "12", "Id": 7, "Name": "Real Time Ticker Plant", "Order": 15}, {"$id": "13", "Id": 12, "Name": "Web Application Testing", "Order": 16}]
```

Example get name of specific system ID (0 for Online):

[http://is.factset.com/justifier/api/BuildSystems/?\\$filter=Id%20eq%200&\\$select=Id,Name](http://is.factset.com/justifier/api/BuildSystems/?$filter=Id%20eq%200&$select=Id,Name)

Justifier Build (field):

This is the "Justifier Build" field in job entry form.

When you select a "System" that is not "Tellus", it will populate this dropdown list.

Server query URL to populate this field:

"calendar/php/qJustifierBuild.php?sid=" + systemDrop

Data source:

"[http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/?\\$filter=active%20eq%20true%20and%20Template/BuildSystem/Id%20eq%20](http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/?$filter=active%20eq%20true%20and%20Template/BuildSystem/Id%20eq%20)" . \$sid . "&\$select=Id,displayname&\$orderby=ID%20DESC&\$format=json"

Product

To get product ID 3673:

[http://is.factset.com/Justifier/api/FactSetProducts/?\\$filter=Id%20eq%203673](http://is.factset.com/Justifier/api/FactSetProducts/?$filter=Id%20eq%203673)

Output:

```
<ArrayOfFactSetProduct>
  <FactSetProduct
    $id="i1">
      <Id>3673</Id>
      <EntityKey>3327301</EntityKey>
      <PathKey>3645-299-3673</PathKey>
      <Name>PA 3.0</Name>
      <Description nil="true"/>
      <Notes nil="true"/>
      <Path>Analytics > Portfolio Analytics > PA 3.0</Path>
      <lastUpdatedDate>2016-08-10T11:47:05.357</lastUpdatedDate>
      <UpdatedBy>cwerner</UpdatedBy>
      <Deleted>false</Deleted>
    </FactSetProduct>
  </ArrayOfFactSetProduct>
```

Library Name (field):

This is the "Library Name" field.

When you select a "Justifier Build", it will populate this dropdown list.

Server query URL to populate this field:

calendar/php/qJustifierBuildLibrary_v2.php?sid=" + justifierBuild

Data source:

\$url = "<http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/>" . \$sid .
"/BuildTestCases(TestCaseHistory.Automated%20eq%20true)/LibraryHistory/Library?
\\$select=Id,Name&\\$orderby=Name%20&\\$format=json";

Test Plan Name (field):

This is the "Test Plan" field on UI form.

When you select a "Library Name", it will populate this dropdown list.

Server query URL to populate this field:

"calendar/php/qJustifierBuildTestPlan_v2.php?tid1=" + justifierBuild + "&tid2=" + libraryName

where justifierBuild is the selected value of "Justifier Build" above, and libraryName is that of "Library Name".

Data source:

\$url = "<http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/>" . \$tid1 .
"/BuildTestCases(TestCaseHistory.Automated%20eq%20true%20and%
20LibraryHistory.Library.Id%20eq%20" . \$tid2 . ")/TestPlanHistory/?
\\$select=Id,Name&\\$orderby=ID%20DESC&\\$format=json";

Test Cases (field):

This field's data is actually not from Justifier Web service (TO BE CONFIRMED).

This field's values are initialized based on selected Test Plan name.

<http://iswebstga03.pc.factset.com:8090/justifier/api/TestCases>

Server query URL to populate this field:

"calendar/php/CalendarGetTestCases.php?q=" + encodeURIComponent(testPlan)

Data source:

The data source is Tellus database tables: SchedulerTestCategories (test plans) and SchedulerTests (test cases), and SchedulerTestCatAssoc (mapping test plans to test cases).

Automated (flag)

//is.factset.com/...

Steps

Tellus does not need to use this.

Associations

Steps can be associated with multiple test cases.

A test case can be associated with multiple plans.

Library will becomes Product.

Mapping

Tellus-Justifier many-to-many

Do not need to depend on Justifier plan or other hierarchy.

Push to Justifier

build

plan

//is.factset.com/justifier/api/testplan

Justifier test plan hierarchy

Monday, June 02, 2014 12:49 PM

1. Justifier test plan organization (external, IS)

- a. **Template**
- b. **Build System**
For example, Online, Atlas
- c. **Library**
- d. **Test plan**
- e. **Test case**
- f. **Test step**

2. Comments

- a. New naming convention can have embedded space.

Justifier REST web service location

Tuesday, June 24, 2014 12:57 PM

Justifier test case REST Web service URL

a. **QA:**

iswebqa.pc.factset.com/justifier/api

Staging:

iswebstg.pc.factset.com/justifier/api

Both have separate databases and are not tied to production in any way.

b. Justifier test plan data source:

<http://iswebstga03.pc.factset.com:8090/justifier/api/TestPlans>

c. Test case schema: <http://iswebstga03.pc.factset.com:8090/justifier/api/TestCases>

d. Example:

- i. [All Test Cases of Test Plan 32232](#) (Includes deleted Test Cases)
- ii. [All Test Case Names of Test Plan 32232](#) (Includes deleted Test Cases)
- iii. [Distinct list of all the Test Case Updaters of Test Plan 32232](#) (Excludes deleted Test Cases)
- iv. [Distinct list of all the Test Case Names of Test Plan 32232](#) (Excludes deleted Test Cases AND only Includes Test Cases that are Primary Links)

Justifier Integration Info

Monday, July 07, 2014 9:30 AM

From: Chris Werner

The API requires Authentication, which is already working fine. Any internal users will automatically be authenticated to their user accounts. Then it uses this information (auth user) to look up if you have proper permissions in Justifier to do updates (POST, PUT or DELETE).

I have added you as a 'manager' level user in Justifier (on QA, STG, and Prod) and now it should work for you without a problem.

Hi all,

I'm so sorry for how long it has taken me to get this out. Below are some emails I sent Frank with various examples of using the API and answering various questions. These mostly cover informational things, getting lists of Libraries or SWPs or only Automated Test Cases, etc. Hopefully it's helpful.

The very last email was to Dan Felson and that covers how to POST back Test Run result data. This is the current process and how it works, but you'll see lots of mention about Test Plan *History* and Test Case *History*. This is how Justifier works behind the scenes, but I am planning to try to make a wrapper for this and make it a bit cleaner.

Eventually this should be as easy as giving the permanent, non-changing Test Plan Id and Build Id and it will find the correct Test Plan History Id, to start a new Run. Same with Test Cases, you (and most Automation I would gather) will probably not care about 'steps' so much, it might be easier to just hardcode the permanent Test Case Id and let the api figure out the correct 'History' Id and update all of the Steps in that Case as Pass or Fail. Anyway, I will keep you updated about that and try to get a better primer for that change when it happens.

Feel free to ask any questions or assign meetings to run through examples or whatever you need.

Chris

From: Chris Werner

Sent: Friday, January 17, 2014 2:46 PM

To: Frank Yin; Robert L. Groce

Subject: Justifier Staging Change

Frank,

I have added this change on Staging so you can give it a try:

<http://iswebstg.pc.factset.com/justifier/api/testCases> I have not added the AutomatedSystems end point yet, but currently created two values, ID 1 is 'Tellus – TestComplete', and ID 2 is 'Tellus – Selenium'.

EDIT: /api/TestCases shows schema (structure) of the test cases.

EDIT: To read data fields of test case #73695:

<http://iswebstg.pc.factset.com/justifier/api/TestCases/73695>

You can update some Test Cases like this: POST to:

<http://iswebstg.pc.factset.com/justifier/api/TestCases/73695>

```
{AutomatedSystem:{Id:2}}
```

I have populated some, so for instance, now to get the Automated Test Cases for a certain build you'd do this:

[http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases\(TestCaseHistory.TestCase.AutomatedSystem.Id%20eq%201%20or%20TestCaseHistory.TestCase.AutomatedSystem.Id%20eq%202\)/TestCaseHistory/?\\$select=TestCase/Id,Name,TestCase/AutomatedSystem](http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases(TestCaseHistory.TestCase.AutomatedSystem.Id%20eq%201%20or%20TestCaseHistory.TestCase.AutomatedSystem.Id%20eq%202)/TestCaseHistory/?$select=TestCase/Id,Name,TestCase/AutomatedSystem)

Eventually you'd query only TestCaseHistory.AutomatedSystem, but I didn't update the History Items, so for now I'm querying the Test Case directly.

Previously I gave you a query to pull all Automated eq true instead:

[http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases\(TestCaseHistory.Automated%20eq%20true%20and%20TestPlanHistory.TestPlan.Id%20eq%2072021\)/TestCaseHistory/?\\$select=TestCase/Id,Name](http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases(TestCaseHistory.Automated%20eq%20true%20and%20TestPlanHistory.TestPlan.Id%20eq%2072021)/TestCaseHistory/?$select=TestCase/Id,Name)

Frank,

Automated is flagged at the Test Case level or Test Plan level.

If you want all Libraries that have at least 1 Automated item in it then I'd do this:

[http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases\(TestCaseHistory.Automated%20eq%20true\)/LibraryHistory/?\\$select=Library/Id,Name](http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases(TestCaseHistory.Automated%20eq%20true)/LibraryHistory/?$select=Library/Id,Name)

Then after they pick a Library do this:

[http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases\(TestCaseHistory.Automated%20eq%20true%20and%20LibraryHistory.Library.Id%20eq%2022182\)/TestPlanHistory/?\\$select=TestPlan/Id,Name](http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases(TestCaseHistory.Automated%20eq%20true%20and%20LibraryHistory.Library.Id%20eq%2022182)/TestPlanHistory/?$select=TestPlan/Id,Name)

Then lastly for the cases themselves use the Test Plan Id instead:

[http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases\(TestCaseHistory.Automated%20eq%20true%20and%20TestPlanHistory.TestPlan.Id%20eq%20990\)/TestCaseHistory/?\\$select=TestCase/Id,Name](http://iswebstg03.pc.factset.com:8090/justifier/api/Builds/9976/BuildTestCases(TestCaseHistory.Automated%20eq%20true%20and%20TestPlanHistory.TestPlan.Id%20eq%20990)/TestCaseHistory/?$select=TestCase/Id,Name)

The inline filters work best here.

If they don't want 'only' Automated, then just leave that inline filter out.

Also note that I got the historic Names, where you were getting the current Names before, it should not really matter, just noting the difference.

From: Chris Werner

Sent: Friday, August 16, 2013 1:17 PM

To: Frank Yin

Cc: Robert L. Groce

Subject: Justifier API some examples - RE: Authentication issue solved! Working on Paring Justifier API URL path

Frank,

All documentation is here: <http://iswebstg.pc.factset.com/justifier/api>

If you keep scrolling down all Entities available (build, etc) are listed at the bottom. If you click on any of them it will show you which properties are available: <http://iswebstg.pc.factset.com/justifier/api/Builds>

So for instance, you could see that in Builds there is a property BuildTestCases (of type Array[BuildTestCase]), if you click on BuildTestCase you will see its properties, for instance TestPlanHistory and LatestTestRun, each of these have properties, like Name and Id (respectively). So using this information you can start to see what is available.

The API lets you enumerate through as many properties as you want. You have two options:

1) you are allowed to follow the path before the ?, so in this way it ignore your chosen end point and looks for the distinct result of the children:

<http://iswebstg.pc.factset.com/justifier/api/Builds/8955/BuildTestCases/TestCaseHistory/QALead/UserName>

For instance, in this example even though you're querying for Builds where Build.Id is 8955, you are driving down to get QA Leads for each Test Case for each Build Test Case. The above returns the distinct list of User Names that are Leads on the given build (8955).

You can also stop on an entity and do the normal selecting of properties expected from OData, ie:

[http://iswebstg.pc.factset.com/justifier/api/Builds/8955/BuildTestCases/TestCaseHistory/QALead/?\\$select=Id,UserName,Employee/FullName_FNF](http://iswebstg.pc.factset.com/justifier/api/Builds/8955/BuildTestCases/TestCaseHistory/QALead/?$select=Id,UserName,Employee/FullName_FNF)

This 'stops' at QA Lead level, and then I can ask for QA Lead/Id, QA Lead/UserName and even continue to walk the path and get QA Lead/Employee/FullName_FNF.

PS: After you specify a child entity you cannot ask for the parent info, so as above, the \$select=Id was the User's Id, you cannot get back to the Build's ID even though the root or base entity is the Build (ie. api/Builds)

Because of this I added the ability to specify a simple (WHERE CLAUSE) along this path, so for instance: [http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks/\\$count](http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks/$count) This is the number of Test Cases inside of Test Plan 178

[http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks\(Deleted%20eq%20true\)/\\$count](http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks(Deleted%20eq%20true)/$count) This is the number of Test Cases with Deleted eq true (which is 0 for this Test Case).

The point is so I can always ask for only Deleted eq false children, but keep going:

[http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks\(Deleted%20eq%20false\)/TestCase/QALead/UserName](http://iswebstg.pc.factset.com/justifier/api/TestPlans/178/TestCaseLinks(Deleted%20eq%20false)/TestCase/QALead/UserName) So this is all the QA Leads for all undeleted Test Case in Test Plan 178

2) You can stop at the root level, which means you pick one entity and can either query the entire thing, or specify and integer to be it's ID. So /api/Builds will give you all of the builds in the system, but /api/Builds/8955 will give only Build Id eq 8955.

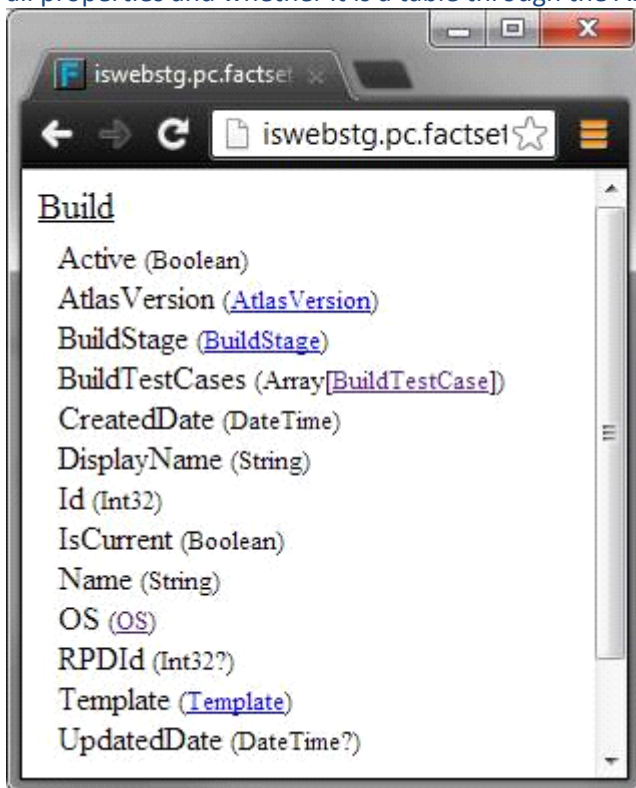
Filtering and selecting can be done as typical OData, like in your example below:

[http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/?\\$filter=active%20eq%20true%20and%20iscurrent%20eq%20true%20and%20Template/BuildSystem/Id%20eq%2011&\\$select=Id,displayname&\\$format=json](http://iswebstga03.pc.factset.com:8090/justifier/api/Builds/?$filter=active%20eq%20true%20and%20iscurrent%20eq%20true%20and%20Template/BuildSystem/Id%20eq%2011&$select=Id,displayname&$format=json)

This says to \$filter all Builds where Build/Active eq true and Build/IsCurrent eq true and Build/Template/BuildSystem/Id eq 11 then \$select only Build/Id and Build/DisplayName columns.

As for more examples, \$filter and \$select should work just like SQL Where and Select, you can think of an Entity as a 'table' and everything else as a Column, except when the 'column' or property is actually another Table. In which case, linking is handled automatically and seamlessly, and like I said you can see

all properties and whether it is a table through the API documentation:



Click on Template ->



Click on BuildSystem ->



You can tell that Id is an actual Property (Int32) and not another table.

By browsing this you can 'know' that therefore you can reach this property if you want from Build like Build/Template/BuildSystem/Id (as in the example you provided).

POST TEST RUN DATA BACK TO API

From: Chris Werner
Sent: Tuesday, June 03, 2014 2:17 PM
To: Alisa C. Berry; Dan Felson
Cc: Kristin E. Bedoukian; Robert L. Groce
Subject: RE: ECS Fixed Income use of the Justifier API

So, this is an email I sent as a primer a while ago, but may help.

All documentation is here: <http://is.factset.com/justifier/api/>

Feel free to email or IM me with any questions.

The first few parts just explain using the API, the main part you probably care about is at the bottom where I discuss how to create test plans and replace
<http://iswebqa.pc.factset.com/justifier/service/justifierfixedincomeservice.asmx?op=AddTestPlanComment>

To get the current Active Online builds: [http://iswebqa.pc.factset.com/justifier/api/builds?\\$filter=template/buildsystem/id%20eq%200%20and%20iscurrent%20eq%20true&\\$select=Id,Name,BuildStage&\\$expand=BuildStage](http://iswebqa.pc.factset.com/justifier/api/builds?$filter=template/buildsystem/id%20eq%200%20and%20iscurrent%20eq%20true&$select=Id,Name,BuildStage&$expand=BuildStage)

This lists all of them, that's why I expanded the BuildStage option.

Typically I am thinking you would want the Active QA Online build, so we simply specify that we want QA: [http://iswebqa.pc.factset.com/justifier/api/builds?\\$filter=template/buildsystem/id%20eq%200%20and%20iscurrent%20eq%20true&\\$select=Id,Name,BuildStage&\\$expand=BuildStage](http://iswebqa.pc.factset.com/justifier/api/builds?$filter=template/buildsystem/id%20eq%200%20and%20iscurrent%20eq%20true&$select=Id,Name,BuildStage&$expand=BuildStage)

[20and%20iscurrent%20eq%20true%20and%20BuildStage/Id%20eq%201&\\$select=id,name](http://iswebqa.pc.factset.com/justifier/api/testplans/?$filter=library/name%20eq%20%27futures%20montage%27&$select=id,name)

I just specified BuildStage/Id eq 1 which won't change but you could do BuildStage/Name eq 'QA' too. It's hard to see with the url encoding, but it's pretty straightforward to specify the \$filter.

And lastly, you can use Library Id or Name to get its list of Test Plans:

[http://iswebqa.pc.factset.com/justifier/api/testplans/?\\$filter=library/name%20eq%20%27futures%20montage%27](http://iswebqa.pc.factset.com/justifier/api/testplans/?$filter=library/name%20eq%20%27futures%20montage%27)

This is all Test Plans that have a Library/Name of 'futures montage'. Now, OData doesn't support a 'like' command, but does support startswith, endswith, indexof and substring of, so if you want to get all test plans that have a Library Name like 'atlas %':

[http://iswebqa.pc.factset.com/justifier/api/testplans/?\\$filter=startswith\(library/name,'atlas%20'\)&\\$select=id,name&\\$orderby=name](http://iswebqa.pc.factset.com/justifier/api/testplans/?$filter=startswith(library/name,'atlas%20')&$select=id,name&$orderby=name)

(the result was actually a bit large for a web browser to handle, should be fine in JSON, the SQL executed fast, but I included the select of just id and name because that's all we really need and it's much faster for demonstration)

((Also I threw in \$orderby so you could see it in action))

Now for the meat. You can create items if you send a POST instead of a GET and you can update items if you send a PUT. Right now only a couple of the entities support this, I am adding them as I see a need to, but the skeleton is there and it's pretty easy to add something if you have a use case to add/update and item.

Feel free to test any of these out, it's just dev data and you cannot hurt it. You can use either Fiddler:

<http://www.fiddler2.com/fiddler2/> Or I prefer POSTMAN (go to Chrome Web Store and search for 'postman', it's called Postman – REST Client and is an add on for Chrome).

To create a new Template:

POST TO <http://iswebqa.pc.factset.com/justifier/api/templates/>

```
{'name':'CJW test', 'Active':true, 'BuildSystem':{'Id':2}, 'testplans': [{'id':176},{'id':294}]}
```

(if you set content-type to application/json, you 'should' be able to send in any format, if you have a particular format I don't currently support just let me know, it's easy to add)

The response gives you the full object and linked testplans so you can get the new ID, when I just tested this I got ID = 363

Add more testplans to the Template:

<http://iswebqa.pc.factset.com/justifier/api/templatetestplans/>

(if you set content-type to application/json)

```
{'testplanid':123456,'templateid':363},  
{'testplanid':1234567,'templateid':363}]
```

These are clearly bogus testplans, but if I check the DB table JustifierProd.qae.TemplateTestPlans I now have 4 listed items.

I'm not sure if it's better to keep these Ids exposed like this or not, you can also post like this:

```
{'testplan':{'id':123},'template':{'id':363}}, .... ]
```

The main item that Fixed Income is using now is this:

<http://iswebqa.pc.factset.com/justifier/service/justifierfixedincomeservice.asmx?op=AddTestPlanComment>

You're specifying build_id, testplan_id, description, status_id, numbugs.

This will be replaced by creating a Test Run for a certain BuildId, then updating each step as Pass or Fail. Most of your Cases probably only have 1 Test Step.

Lastly, I have incorporated some of our logic into the API, so you cannot POST to /api/TestRunSteps because this does not make sense in our business logic. Similarly you cannot call DELETE on TestRunSteps.

When you POST to /api/TestRuns it will call our CreateNewTestRun SP and will create the TestRun and its TestRunSteps and return the entire TestRun so you can see the new TestRunId and all the new TestRunStepIds.

This is the way we do it in our system because TestRunSteps can only exist with a TestRun and this way all TestRunSteps always exists and will have an initial value of no result (result=null).

POST to <http://iswebqa.pc.factset.com/justifier/api/testruns/>:

```
{'testplanHistory':{'id':57020}, 'os':{'id':8}, 'atlasversion': {'id':1720} , 'stage': {'id':1}, 'tester': {'id':290}, build: {id: 11123} }
```

This gave me a TestRunId of 762712 and a handful of TestRunSteps.

Now we can update each TestRunStep as we pass/fail.

We can update a few at once like this:

PUT to <http://iswebqa.pc.factset.com/justifier/api/testrunsteps/>

```
{id: 23577873, result:{id:1}},  
{id: 23577874, result:{name:'fail'}}
```

(You can specify either the Result Id shown here: <http://iswebqa.pc.factset.com/justifier/api/results?Stop=5> Or specify the 'name' 'pass'/'fail' or 'skip')

Or we can update 1 at a time like this:

PUT to <http://iswebqa.pc.factset.com/justifier/api/testrunsteps/23577874>
{result:{id: 2}}

You can see the results of this here:

<http://iswebqa.pc.factset.com/justifier/TestManager/TestRun/762712#/23577874>

1 important change is that Test Run requires a TestPlanHistoryId and will not take the Test Plan Id that you're used to giving.

So for instance, my example so far has been with Test Plan Id:

72176 <http://iswebqa.pc.factset.com/justifier/TestManager#/61/72176>

For a given Build Id and Test Plan Id I can look up the proper History Id this way:

[http://iswebqa.pc.factset.com/justifier/api/builds/11123/buildtestcases/TestPlanHistory\(TestPlan.Id%20eq%2072176\)/Id](http://iswebqa.pc.factset.com/justifier/api/builds/11123/buildtestcases/TestPlanHistory(TestPlan.Id%20eq%2072176)/Id)

That's using an inline filter to get the distinct IDs, I could also go directly to BuildTestCases with a filter:

[http://iswebqa.pc.factset.com/justifier/api/buildtestcases/?\\$select=TestCaseHistory/Name,TestCaseHistory/Id,TestPlanTestcaseLinkHistoryId,TestPlanHistory/Id,TestPlanHistory/Name&\\$filter=Build/Id%20eq%2011123%20and%20TestPlanHistory/TestPlan/Id%20eq%2072176](http://iswebqa.pc.factset.com/justifier/api/buildtestcases/?$select=TestCaseHistory/Name,TestCaseHistory/Id,TestPlanTestcaseLinkHistoryId,TestPlanHistory/Id,TestPlanHistory/Name&$filter=Build/Id%20eq%2011123%20and%20TestPlanHistory/TestPlan/Id%20eq%2072176)

(this gets all Test Cases for that Plan)

These 'history' ids only change when an item is edited in the Justifier, so I don't know how you want to handle this. I think it would be best to keep using the Test Plan Id and look up the proper 'History' id, or

maybe we can override Test Run to be smart about using a Test Plan Id if a Build Id is given.

The big benefit you will get from this refactor is that you can create a Run at a Test Case level. Years ago Mike Kessler was an employee and wanted to try to refactor all of the FI Test Plan names to be Test Cases, but the web service available and SWP limited that.

Now you can use that last link above to get the proper Test Plan Test Case Link History Ids and specify these and create a separate Test Run per Case and SWP will do the proper thing:

POST to <http://iswebqa.pc.factset.com/justifier/api/testruns/>:

```
[{'testplanHistory':{'id':57020}, testrunsteps: [{ testplantestcaselinkhistory: { Id: 328509 } }],  
  'os':{'id':8}, 'atlasversion': {'id':1720}, 'stage': {'id':1}, 'tester': {'id':290}, build: {id: 11123},  
    StageNotes: ' This is a test of just a single Test Case'  
}]
```

Here's the example I've been using so far to see the SWP: <http://iswebqa.pc.factset.com/justifier/#11123?filtertext=news-online>

Justifier Documentation

Monday, July 07, 2014 2:46 PM

Documentation URL:

<http://iswebstg.pc.factset.com/justifier/api>

Justifier API AutomatedSystems

Monday, October 09, 2017 3:21 PM

<http://iswebstg.pc.factset.com/Justifier/api/AutomatedSystems>

1	TestComplete	true
2	Selenium	true
3	ORCA	
4	PUB	
5	BATCH_CALC	
6	Monster Chart	
7	PANTS	
9	FIA	
10	Selenium/Protractor	true
11	Quant(AT3,PS2)	
12	CIE(CS/BDF)	
13	Security Scanner	
15	SIT	true
16	Appium	true
17	PC Profiler Performance	
18	Test Builder	
19	SoapUI	true

QAI SWP calling Tellus API

Thursday, March 08, 2018 7:57 AM

"EmailRecipient": "qai.tellus.results@factset.com"

Example QAI SWP Request:

```
{
  "instance": {
    "scheduledBy": "GREENWICH\\ldefelice",
    "title": "Activist Profile - Devel",
    "fdsAv2Stage": "QA",
    "fdsAv2Override": null,
    "testEnvironment": "FPE",
    "scheduleType": "RunNow",
    "runNow": "1",
    "testBuildID": "-1",
    "testBuildVersion": "-1",
    "browserType": "CEFP",
    "browserVersion": "Any",
    "testPlans": [{
      "justifierTestPlanID": 79413,
      "justifierTestPlanName": "Activist Profile - Devel",
      "justifierConfigID": 29118,
      "justifierConfigName": "CEFP",
      "duration": 4,
      "testCases": [{
        "justifierTestCaseID": "123689",
        "testPlanTestCaseLinkHistoryId": 695156
      }],
      "justifierTestPlanHistoryID": 170210
    }],
    "duration": 4,
    "emailRecipient": "qai.tellus.results@factset.com",
    "stage": "QA",
    "vmsUserID": "FDSQAR_C",
    "caccessCode": "LARGESETACCESS",
    "buildID": "12074",
    "buildVersion": "\\2018_V1\\atlas_snapshot_2016.6.227\\2016.6.227.066",
    "windowsVersion": "Win 7x64",
    "bitVersion": "64bit",
    "officeVersion": "Office 2013",
    "languageName": "English (US)",
    "justifierSystemID": "0",
    "justifierBuildID": "32288",
    "justifierProductID": "5034",
    "fixedTime": "0",
    "tickerListRequired": "0",
```



```
"tickerList": "-1",  
"htmlReports": "Devel",  
"htmlReportsOS": "DEFAULT",  
"testCaseStage": "Live",  
"isRegressionJob": "0",  
"isLateRegressionJob": "0",  
"startDate": "2018-03-02",  
"startTime": "11:28",  
"qaiServerName": "is.factset.com"  
}  
}
```