



7/2/2014

fmPilot 2

Service Provider API

Testing URLs and data

FACILITY SOURCE

Contents

Changes log	1
Testing tool	1
Initial setup	1
GET Requests	2
Multiple Work Order polling.....	2
Individual Work Order polling.....	2
Work Order Status polling	2
Comment polling.....	2
POST Requests	2
Work Order change.....	2
PUT Requests	3
Comment addition	3
Filtering options	3

Changes log

Version	Created/updated	Date	Details
0.1	Joel Etherton		Initial draft
0.2	Carlos Chaves	07/09/2014	Document format
0.3	Carlos Chaves	07/16/2014	Change WO Status update for WO update
0.4	Carlos Chaves	08/05/2014	Filtering options changed. The explanation of filtering options was added. New content hierarchy on the available actions.

Testing tool

One of the simplest tools for testing is the Chrome Extension “Advanced REST Client” which can be added from the Chrome Web Store. We will be referencing this tool mostly in this document, but any API testing tool can be used with similar configuration and results.

Initial setup

For executing all test in current document it is needed to have 3 headers to get any data returned:

```
authenticationToken: c921b269-e1ef-4ce2-99f0-5e8c7e41ee7b
callingClient: sasops
actingDomain: RiteAid
```

The response from the Vendor API by default will be formatted using json, in order to change it to xml and test the content negotiation, you can add an additional header test:

accept: application/xml

GET Requests

Multiple Work Order polling

The basic URL structure is:

<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders>

The user has the ability of adding filtering parameters. Please check the Filtering Options section for more information.

For testing the next actions, it is recommended to select one work order for the response of the multiple work order polling and save its WorkOrderId and WorkOrderNumber. This is necessary as a parameter for all the actions.

Individual Work Order polling

The basic URL structure is:

<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/3280198>
<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/WO-0500002>

Both URLs should return the same result, since the WorkOrderId and the WorkOrderNumber used in the example both are related to the same Work Order. In all the examples shown in this document the same premise will be valid.

Work Order Status polling

The basic URL structure is:

<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/3280198/WorkOrderStatus/>
<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/WO-0500002/WorkOrderStatus/>

Comment polling

The basic URL structure is:

<http://demo-api.fmpilot2.com/Vendor/api/WorkOrder/3280198/Comments/>
<http://demo-api.fmpilot2.com/Vendor/api/WorkOrder/WO-0500002/Comments/>

POST Requests

Work Order change

In this action, it is needed to change the method to POST. The basic URL structure is:

<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/3280198/>
<http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/WO-0500002/>

Additionally, it is required to send any of the following information:

```
{
  "StatusCode": "INPRG",
  "ScheduledStartDateTime": "/Date(1403978100000)/",
  "ScheduledCompleteDateTime": "/Date(1403978100000)/",
  "ActualStartDateTime": "/Date(1403978100000)/",
  "ActualCompleteDateTime": "/Date(1403978100000)/",
  "CauseID": "123",
  "RemedyID": "123",
  "EquipmentID": "123"
}
```

They should echo the Work Order result. The data should reflect the change.

The date time format used in json is a timespan in milliseconds. The sample used above represents the date time "6/28/2014 5:55:00 PM". If you need to convert any date time into a timespan of milliseconds, you can use any tool available on the internet. For testing purposes of this document the following page was used: <http://www.epochconverter.com/#>.

PUT Requests

Comment addition

In this action, it is needed to change the method to PUT. Since edition is not allowed, each operation will create a new comment. The basic URL structure is:

```
http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/3280198/Comments/
http://demo-api.fmpilot2.com/Vendor/api/WorkOrders/WO-0500002/Comments/
```

Additionally, it is required to send, as minimum, the following information:

```
{
  "Subject": "Subject test",
  "Body": "Body test",
  "OwnerUserId": 93424,
  "AllowClient": true,
  "AllowVendor": true,
  "AllowInternal": true
}
```

This action should echo the new Comment fragment.

Filtering options

In the multiple Work Order polling, you have the ability of adding a number of filtering options. The available options are defined on the QueryBy enumeration, and needs to be built into a collection of filter clauses. Then, this collection must be serialized into a single string in order to be added to the query string. For testing purposes this sound very complicated, that's why we have added a filtering UI page that provides a very simple user interface for selecting the filtering options and, when ready for

testing, just copy the final output string. It can be found directly on <http://demo-api.fmpilot2.com/Vendor/FilteringUI/Serializer>.

A very simple example can be select all Work Orders with a specific status: "In Progress". So, the Filtering UI will looks similar to this:

The screenshot shows a web form for filtering work orders. It has the following fields and controls:

- Query By:** A dropdown menu with "WorkOrder_StatusCode" selected.
- Keyword:** An empty text input field.
- Operation:** A dropdown menu with "Equal" selected.
- Value:** A text input field containing "INPRG". Below it is a date time picker with the date "08/06/2014" selected. A note below the picker says "(select a date in the date time picker below and it will be copied in the Value text automatically.)".
- Serialize:** A button to generate the filter string.
- Output:** A text area containing the generated filter string: "?filterClauses=0Bt000Af000CdsatAE0Af000EINPRG".
- Clear:** A button to reset the form.

In case you need to select all Work Orders that has been created in a specific date, you can use the date time picker. The date will be populated automatically (in string format) for your convenience. The Filtering UI will looks similar to this:

The screenshot shows the same filtering UI as above, but with different values:

- Query By:** A dropdown menu with "WorkOrder_EnteredDate" selected.
- Keyword:** An empty text input field.
- Operation:** A dropdown menu with "Equal" selected.
- Value:** A text input field containing "08/05/2014". Below it is a date time picker with the date "08/06/2014" selected. A note below the picker says "(select a date in the date time picker below and it will be copied in the Value text automatically.)".
- Serialize:** A button to generate the filter string.
- Output:** A text area containing the generated filter string: "?filterClauses=0ATAJ0Af000J08/05/2014".
- Clear:** A button to reset the form.

The keyword operation is composed by two combo boxes, the first one corresponds to a negation value. If the text is empty (as it is on the above images), the operation is exactly what is indicated in the second combo box. But, if you select the option Not, the operation is negated. For instance Not Equal.

You can add as many filters as you need, just by selecting a new QueryBy combination and clicking into Serialize button again. The output string will be reflect immediately all the filtering options added. In case you need to start over a new filtering you can click on the Clear button.

Output

?filterClauses=0BtAD0Af000CP48tAH0At000Dnull