How do I debug the Web Service v2 connections in Autoform DM?

Ross Glover - 2022-11-04 - Comments (0) - Autoform DM FAQs

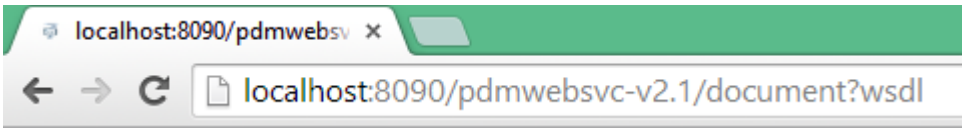
Sometimes, there may be issues with certain processes carried out within the DM software, for example, incorrect data is returned. This article will show users how to debug these issues.

Accessing the WSDL page

It is important to ensure that Web Services are enabled and working correctly. To do this, you would need to access the WSDL page. To access the WSDL page, follow these steps:

1. Open a web browser and navigate to the following address where DMSERVER is the server name and PORT is the port number DM runs on:

```
http://DMSERVER:PORT/pdmwebsvc-v2.1/document?wsdl
```



You should get XML returned describing the endpoint (in this case the Document endpoint):

This shows that Web Services are deployed and responding.

Enabling Web Service debugging in the DM log files

To enable Web Service debugging within the DM log files, follow these steps:

1. Navigate to **OS(C:)** > **Program Files** > **EFS Technology** > **AUTOFORM DM** > **Server xx** > **wildfly xx** > **standalone** > **configuration** > **standalone.xml**

2. Copy everything between the XML tag `size-rotating-file-handler`

3. Make a copy on the line below and change it to match the following:

CODE

4. Find the `org.apache.cxf` (Web Service V2) and `org.apache.axis.EXCEPTIONS` (Web Service V1) logging categories

5. Under the logging name add in the following lines for both entries:

CODE

CODE

6. Save the log and restart DM.

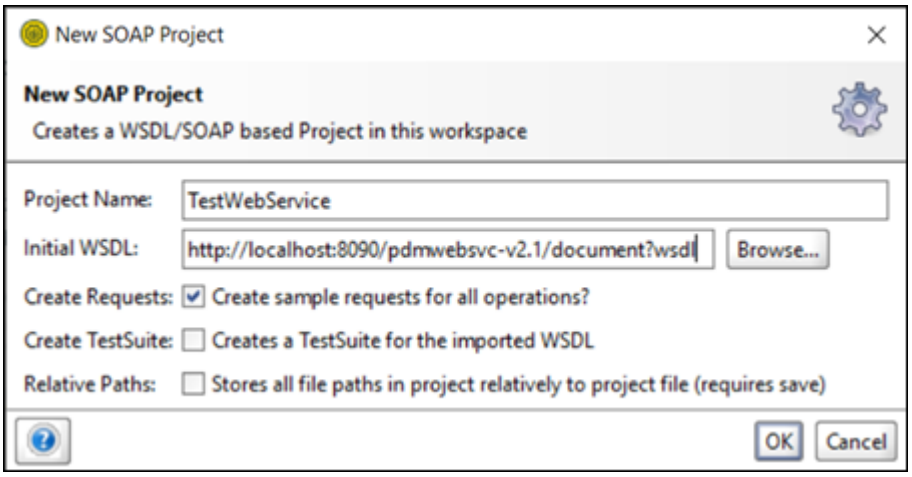
You should now have a new log-filled webservicev2.log containing the Web Service logging. To have the Web Service logging appear in server.log remove everything in the tags and restart DM.

Use SoapUI to make a Web Service request

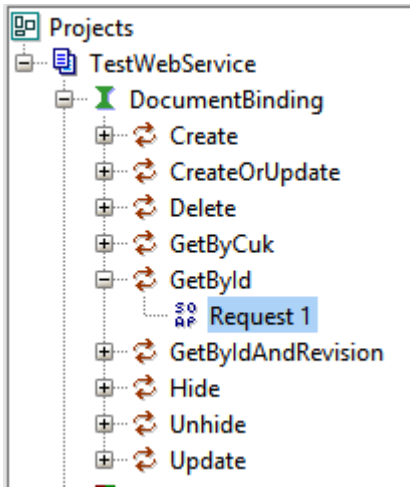
The next stage is to ensure you make a web service request by following these steps:

1. Download and install SoapUI (open source).

2. Create a new **SoapUI Project**, giving it a memorable name and pointing it at the WSDL file as described earlier.



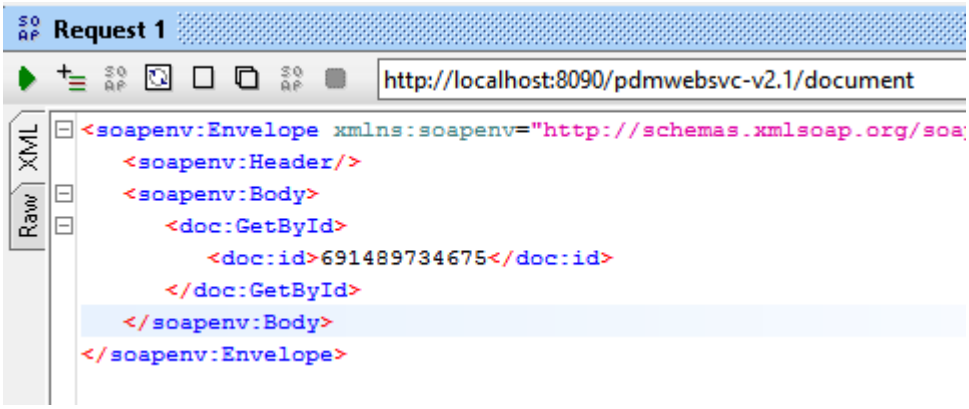
3. Click **OK** and you will get some example requests:



4. Expand GetByID and select **Request 1**.

5. Use a DocID of an existing document and remove the doc:fetchHints section.

6. Re-type the full name of the endpoint above the request.



7. In the bottom left, set the *Request Properties* as follows:

Username: use DM credentials

Password: use DM credentials

WWS-Password Type: set to *PasswordText*

Enable MTOM: set to *true*

Force MTOM: set to *true*

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	363
Encoding	UTF-8
Endpoint	http://localhost:809...
Timeout	
Bind Address	
Follow Redirects	true
Username	admin
Password	**********
Domain	
Authentication Type	No Authorization
WSS-Password Type	PasswordText
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false
Enable MTOM	true
Force MTOM	true
Inline Response Atta...	false
Expand MTOM Atta...	false
Disable multiparts	true
Encode Attachments	false
Enable Inline Files	false
Strip whitespaces	false
Remove Empty Con...	false
Entitize Properties	false
Pretty Print	true
Dump File	
Max Size	0
WS-Addressing	false
WS-Reliable Messag...	false

8. Click the green **Submit** button above the request and wait for a response. A response similar to the image below shows a successful request:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns4:GetByIdResponse xmlns:ns6="http://efstech.com/integration_v2_1/mod
         <ns4:document>
            <name>Delivery Document</name>
            <type>Delivery Document</type>
            <customerUniqueKey>LM19583767001  ?HDR_ASN</customerUniqueKey>
            <properties>
               <id>691489734675</id>
               <owner>jfinder</owner>
               <version>1</version>
               <updated>2014-07-01T12:02:21.780+01:00</updated>
               <hidden>false</hidden>
            </properties>
            <metadata/>
         </ns4:document>
      </ns4:GetByIdResponse>
   </soap:Body>
</soap:Envelope>
```

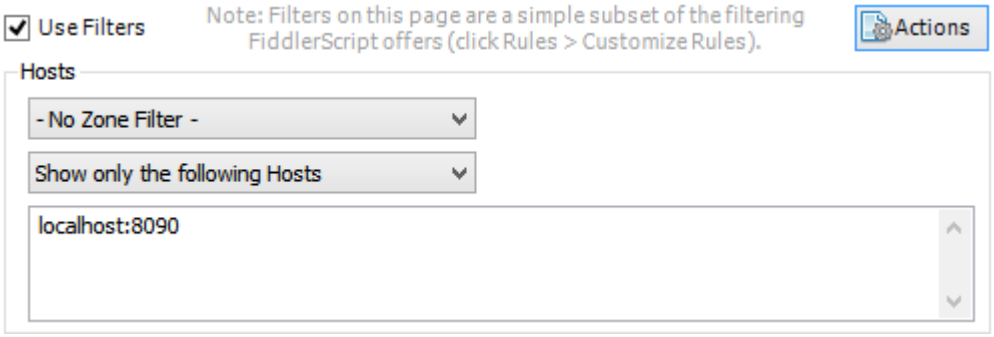
Use Fiddler to record a Web Service Request

Running Fiddler can block or monitor all network traffic. We recommend checking with your IT before installing.

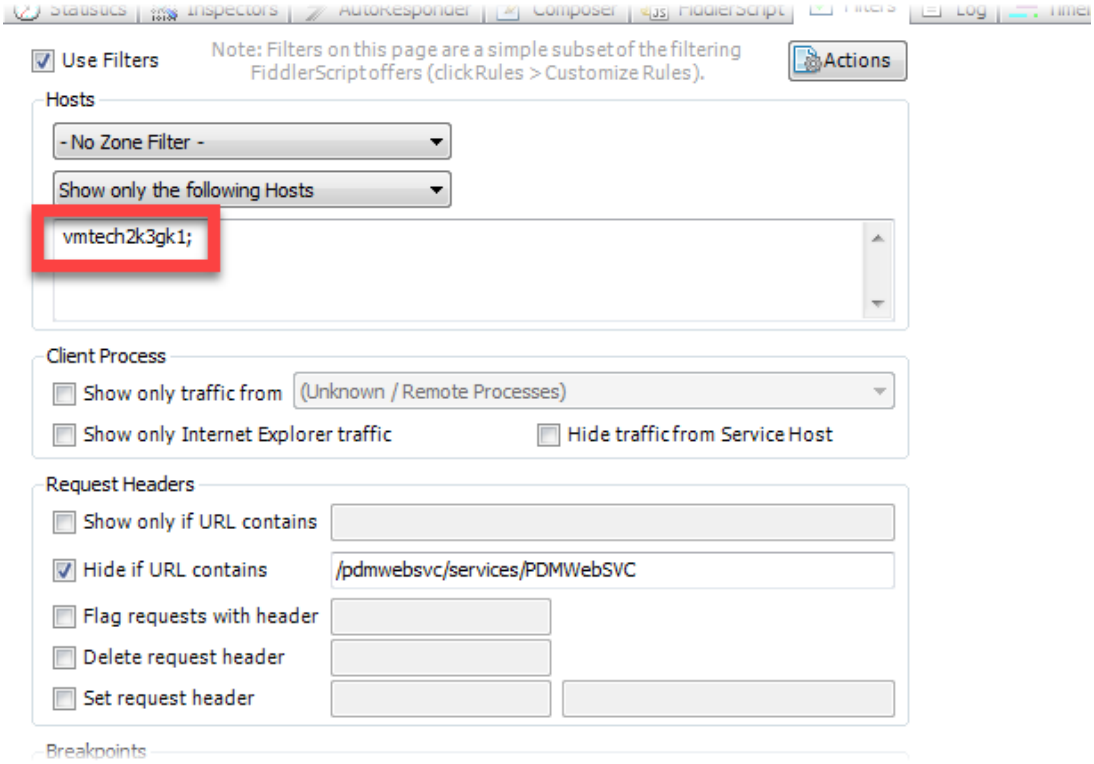
1. Download and install Fiddler 4 from Telerik.

On the left pane, you will see the Web Service Requests coming in.

2. To filter for just the DM requests, open **Filters** on the right-hand pane and use appropriate settings.

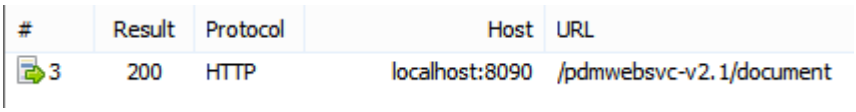


You can also limit the traffic from the Client to ignore all pings to keep the connection alive by selecting the **Hide if URL contains** checkbox and entering the string seen here:



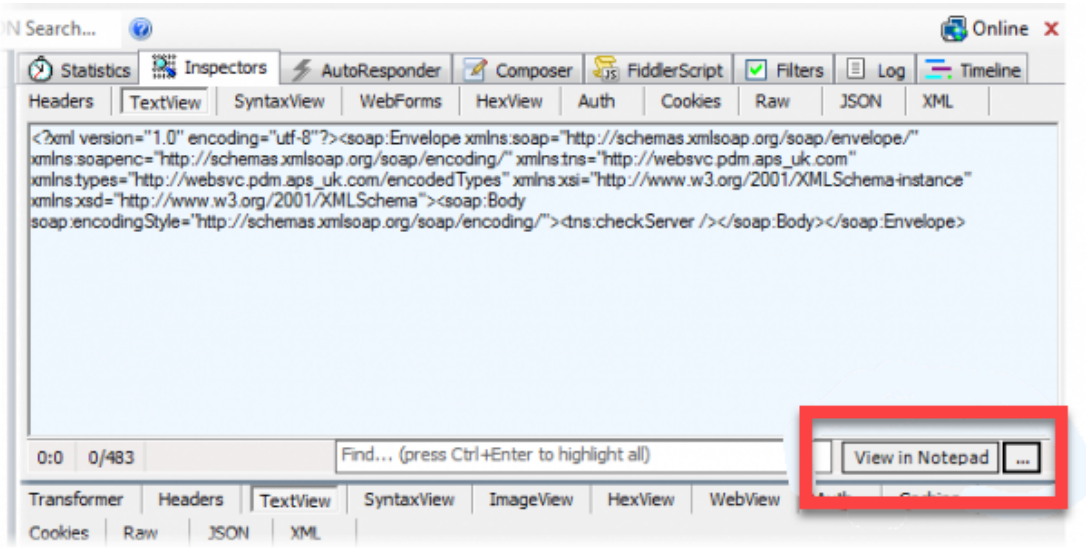
3. Click **Actions** and select **Run Filterset Now** to enable this filter.

4. Make a Web Service request and it will appear on the left using your DM server as the Host and the endpoint as the URL.



5. Click the relevant request and go to the **Inspectors** tab to view the request details.

6. Click the **View in Notepad** button to open the request in any application (for example Notepad++ or EditPad Pro via the Tools>Fiddler Options>Tools option from the menu bar).

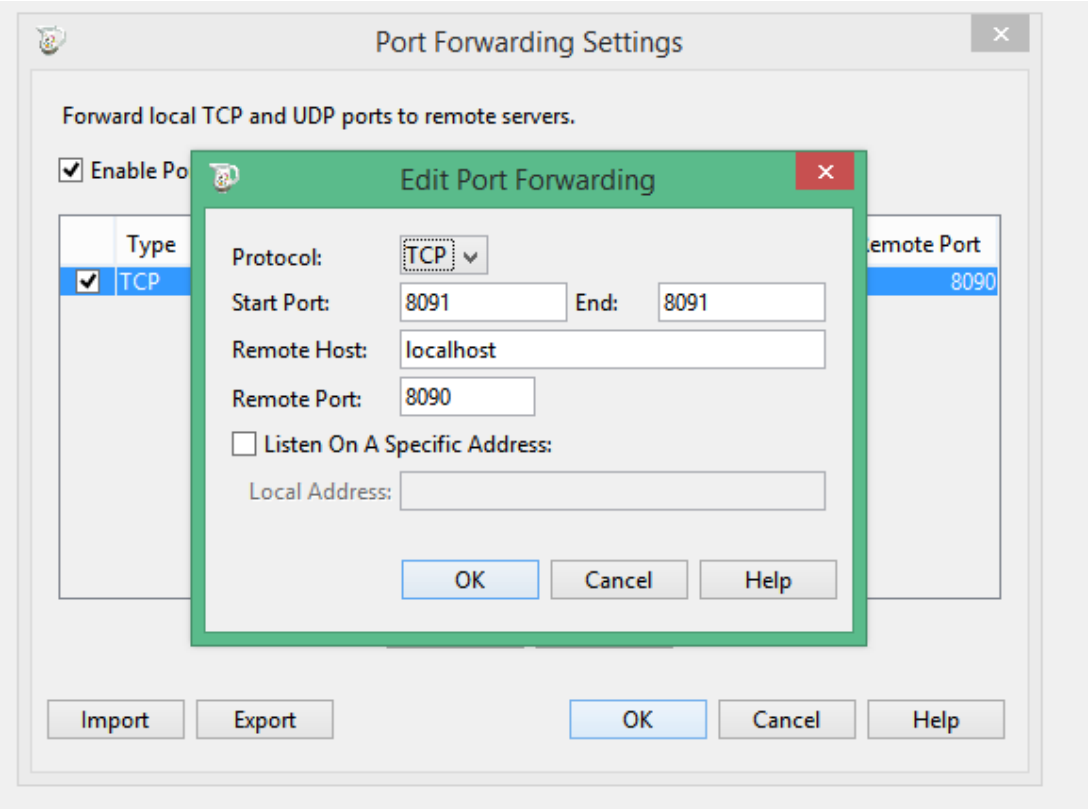


7. This request can either be debugged manually or copied to SoapUI for debugging.

Use Charles to record a Web Service Request

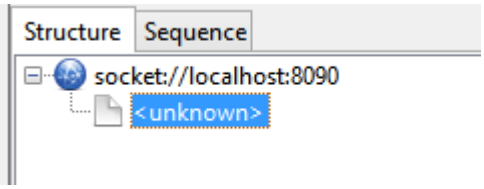
Running Charles can block or monitor all network traffic. We recommend checking with your IT department before installing.

1. Download and install Charles Web Debugging Proxy from Telerik.
2. Under *Proxy* at the top, select **Port Forwarding**.
3. Select the **Enable Port Forwarding** checkbox and click **Add**.
4. Use **TCP** as the Protocol and set **Start** and **End** ports to a currently unused port.
5. Set **Remote Host** to the server name of DM server (or localhost if DM is local).
6. Set **Remote Port** to the DM Port.



7. Change the application performing the Web Service to the port specified in the Port Forwarding (in this case use DM on 8091 rather than 8090)

8. Make the Web Service request and it will display in the Charles application.



9. Click the request and you can see the XML request in the **Request** tab and the XML response in the **Response** tab.

These XML requests can be copied to an XML formatter to improve readability.

10. This request can either be debugged manually or copied to SoapUI for debugging.