
Market Trials: Dispatch Service via the Application Programming Interface (API)

Sept 16, 2024

Table of Contents

Table of Contents	1
Introduction	2
Test Cases	3
DS_API_TC01: Retrieving Dispatches - Dispatches can be retrieved using multiple filters through an API	3
DS_API_TC02: Actioning Dispatches - New dispatches can either be accepted or rejected through an API	12
DS_API_TC03: Retrieving Commitments - START/DECOM/EXTEND Commitments can be retrieved through an API	16
DS_API_TC04: Actioning START Commitments - New START commitments can either be accepted or rejected through an API	29
DS_API_TC05: Actioning DECOM Commitments - New DECOM commitments can be accepted or rejected through an API	33
Additional Resources	38

Introduction

This document outlines the test cases to be executed in Dispatch Service via the Application Programming Interface (API) as part of Market Trials Phase 3. The primary objective of Market Trials Phase 3 is to confirm that certain functions of the new Dispatch Service are working as intended.

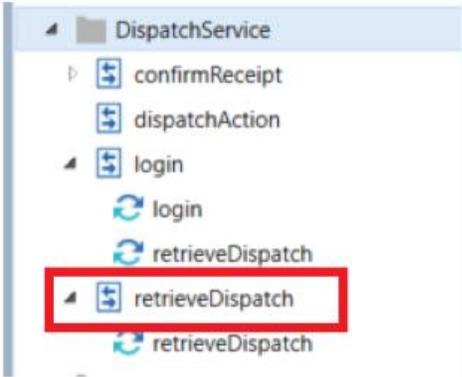
Market Participants should submit any defects discovered during testing via qTest or in the Market Trials Issue Log Template. Doing so will contribute significantly to the overall improvement of the IESO tools. For more information about submitting defects via qTest, refer to the [Quick Take: Submission of Test Case Results and Defects in qTest](#) on the [Market Participant Testing](#) web page.

Preconditions:

1. Access to <https://gateway-sbx.ieso.ca/>
2. Access to the IESO's Dispatch Service sandbox environment
3. User needs to be assigned to any of the following contact roles in Online IESO
 - Dispatch Instruction Viewer
 - Dispatch Instruction Operator
4. Perform the test cases under Connectivity Testing: Dispatch Service Application Programming Interface (API):
 - TC01 - Access Dispatch Service through IESO's Gateway
 - TC02 - Access Dispatch Service Using Web API
5. The connectivity test cases for precondition 4 is located in: <https://www.ieso.ca/-/media/Files/IESO/Document-Library/market-renewal/Market-Participant-Testing/Test-Cases-for-Connectivity-Testing.zip>
6. The API tool used to demonstrate the test cases in this document is Tosca's API Scan. Users may choose to test with this tool or any other web service tool of their choice.

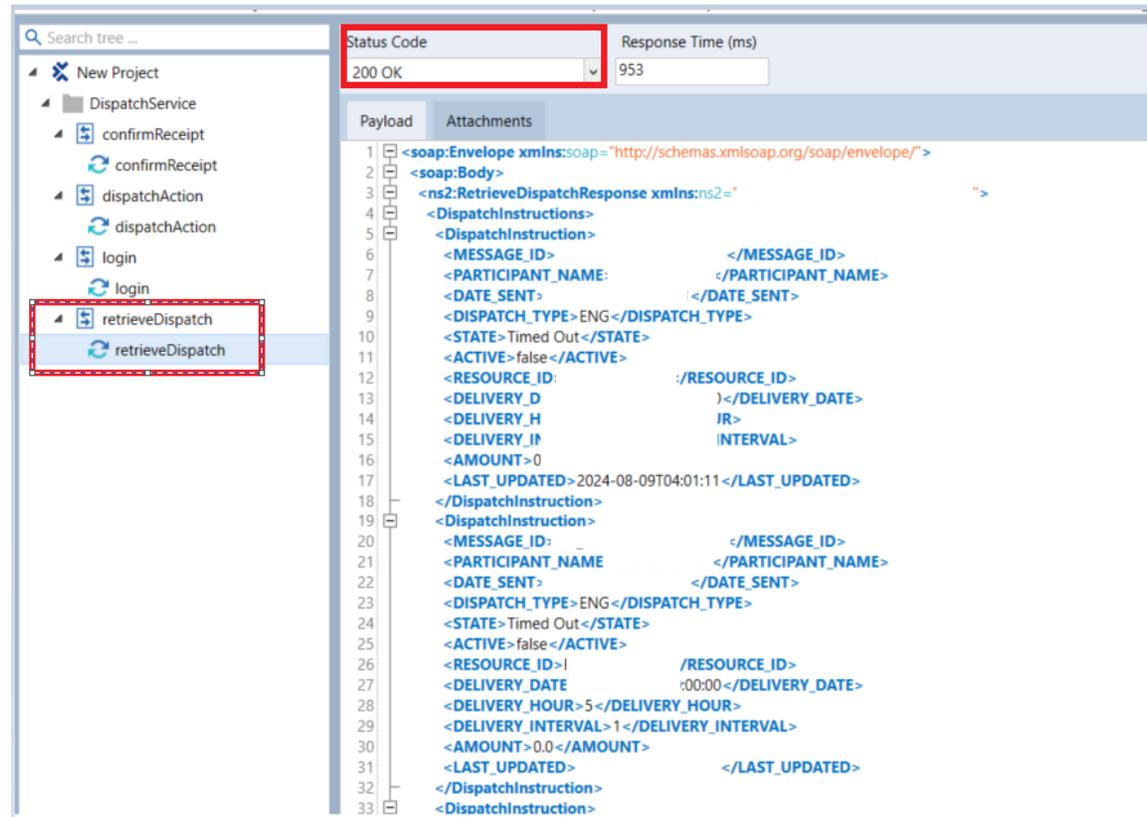
Test Cases

DS_API_TC01: **Retrieving Dispatches** - Dispatches can be retrieved using multiple filters through an API

Step	User Action	Expected Result
1	Once the user connected to is connected to Dispatch Services using an API system, the user will see a "Dispatch Service" folder on the left panel containing 4 subfolders: "confirmReceipt", "dispatchAction", "login", and "retrieveDispatch". Select the retrieveDispatch subfolder.	The retrieveDispatch request payload should open. 
2	In the retrieveDispatch request payload, update the value of the dispatch type field: specify any or all of the dispatch types shown below. Note that only select dispatch types that you expect to receive. <DISPATCH_TYPE>Dispatch Type</DISPATCH_TYPE>	Only dispatches of the specified type are displayed.

Step User Action

Expected Result



- 3 In the retrieveDispatch request payload, update the value of the state field:
<STATE>Dispatch Instruction States</STATE>
Select Run to send the request payload.

Only dispatches of the specified state(s) are displayed.

Step **User Action**

Expected Result

Note: The request payload may contain multiple state tags but only one value can be set for each tag. Refer to the format below.

<STATE>New</STATE>

<STATE>Rejected</STATE>

<STATE>Accepted</STATE>

The screenshot shows a SOAP client interface. On the left, a search tree displays a project structure under 'New Project' with a 'DispatchService' folder. Inside this folder, there are two 'retrieveDispatch' entries, one of which is highlighted with a red box. On the right, the 'Payload' tab is active, showing the XML request. The XML structure is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="t" ...>
3   <s:Body>
4     <RetrieveDispatch xmlns="...">
5       <Filters xmlns="...">
6         <STATE>Accepted</STATE>
7       ...
8     </RetrieveDispatch>
9   </s:Body>
10 </s:Envelope>
```

The <STATE>Accepted</STATE> tag is highlighted with a red box in the original image.

Step User Action

Expected Result

The screenshot displays a SOAP client interface with a search tree on the left and a response view on the right. The search tree shows a project named 'New Project' containing a service 'DispatchService' with several actions: 'confirmReceipt', 'dispatchAction', 'login', and 'retrieveDispatch'. The 'retrieveDispatch' action is selected and highlighted with a red box. The response view shows a status code of '200 OK' and a response time of '795' ms, both highlighted with red boxes. The XML payload is displayed in a code editor, showing a SOAP envelope with a body containing a 'RetrieveDispatchResponse' element. The response includes several 'DispatchInstruction' elements, each with a 'STATE' attribute set to 'Accepted', which is highlighted with red boxes. The XML structure is as follows:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <ns2:RetrieveDispatchResponse xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/">
4       <DispatchInstructions>
5         <DispatchInstruction>
6           <MESSAGE_ID>...</MESSAGE_ID>
7           <PARTICIPANT_NAME>...</PARTICIPANT_NAME>
8           <DATE_SENT>...</DATE_SENT>
9           <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
10          <STATE>Accepted</STATE>
11          <ACTIVE>>false</ACTIVE>
12          <RESOURCE_ID>...</RESOURCE_ID>
13          <EFFECTIVE_TIME>...</EFFECTIVE_TIME>
14          <LAST_UPDATED>...</LAST_UPDATED>
15        </DispatchInstruction>
16        <DispatchInstruction>
17          <MESSAGE_ID>...</MESSAGE_ID>
18          <PARTICIPANT_NAME>...</PARTICIPANT_NAME>
19          <DATE_SENT>...</DATE_SENT>
20          <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
21          <STATE>Accepted</STATE>
22          <ACTIVE>>false</ACTIVE>
23          <RESOURCE_ID>...</RESOURCE_ID>
24          <EFFECTIVE_TIME>...</EFFECTIVE_TIME>
25          <LAST_UPDATED>...</LAST_UPDATED>
26        </DispatchInstruction>
27        <DispatchInstruction>
28          <MESSAGE_ID>...</MESSAGE_ID>
29          <PARTICIPANT_NAME>...</PARTICIPANT_NAME>
30          <DATE_SENT>...</DATE_SENT>
31          <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
32          <STATE>Accepted</STATE>
33          <ACTIVE>>false</ACTIVE>
34          <RESOURCE_ID>...</RESOURCE_ID>
35          <EFFECTIVE_TIME>...</EFFECTIVE_TIME>
36          <LAST_UPDATED>...</LAST_UPDATED>
37        </DispatchInstruction>
38        <DispatchInstruction>
39          <MESSAGE_ID>...</MESSAGE_ID>
40          <PARTICIPANT_NAME>...</PARTICIPANT_NAME>
41          <DATE_SENT>...</DATE_SENT>
42          <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
43          <STATE>Accepted</STATE>
44          <ACTIVE>>false</ACTIVE>
45          <RESOURCE_ID>...</RESOURCE_ID>
46          <EFFECTIVE_TIME>...</EFFECTIVE_TIME>
47          <LAST_UPDATED>...</LAST_UPDATED>
48        </DispatchInstruction>
49      </DispatchInstructions>
50    </ns2:RetrieveDispatchResponse>
51  </soap:Body>
52 </soap:Envelope>
```

Step **User Action**

Expected Result

- 4 In the retrieveDispatch request payload, update the value of the participant name field:
 <PARTICIPANT_NAME>
 Participant Name to retrieve
 </PARTICIPANT_NAME>
 Select Run to send the request.

Only dispatches of the specified participant name(s) are displayed.

The screenshot displays the SoapUI interface. On the left, the 'Project Tree' shows a 'DispatchService' with several actions. The 'retrieveDispatch' action is selected and highlighted with a red box. The main area shows the 'Payload' tab of a SOAP request. The request is a SOAP envelope with a body containing a 'RetrieveDispatch' element. The 'Filters' element within 'RetrieveDispatch' contains the text '<PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>', which is also highlighted with a red box. The response area shows a SOAP envelope with a body containing a 'RetrieveDispatch' element. The response body contains various XML elements, including 'MESSAGE_ID', 'DISPATCH_TYPE', 'STATE', 'DATE_SENT', 'DELIVERY_DATE', 'DISPATCH_TYPE', 'ALT_SYNC_TIME', 'ACTIVE', 'RESOURCE_ID', 'MLP_TIME', 'DELIVERY_HOUR', 'DELIVERY_INTERVAL', 'DELIVERY_START_TIME', 'DELIVERY_STOP_TIME', 'RESPONDER', 'LAST_UPDATED_SINCE', 'SENT_SINCE', 'HISTORY_DAYS', 'EFFECTIVE_TIME', 'MLP_TIME', 'SYNC_TIME', and 'ALT_SYNC_TIME'. The response body is truncated with '...' at the end.

Step User Action

Expected Result

The screenshot displays a SOAP API client interface. On the left, a search tree shows the 'retrieveDispatch' operation under the 'DispatchService' namespace, highlighted with a red box. The main area shows the response details for a 200 OK status with a 128ms response time. The payload is an XML document containing two dispatch entries. The 'PARTICIPANT_NAME' fields in both entries are highlighted with red boxes. The XML structure is as follows:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <ns2:RetrieveDispatchResponse xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/">
4       <DispatchInstructions>
5         <DispatchInstruction>
6           <MESSAGE ID>...</MESSAGE ID>
7           <PARTICIPANT_NAME /PARTICIPANT_NAME>
8           <DATE_SENT /DATE_SENT>
9           <DISPATCH_TYPE>RESV</DISPATCH_TYPE>
10          <STATE>Timed Out</STATE>
11          <ACTIVE>>false</ACTIVE>
12          <RESOURCE_ID>'...</RESOURCE_ID>
13          <DELIVERY_DATE /DELIVERY_DATE>
14          <DELIVERY_HOUR>8</DELIVERY_HOUR>
15          <DELIVERY_INTERVAL>1</DELIVERY_INTERVAL>
16          <AMOUNT>5.0</AMOUNT>
17          <RESERVE_CLASS>10S</RESERVE_CLASS>
18          <LAST_UPDATED>...</LAST_UPDATED>
19        </DispatchInstruction>
20        <DispatchInstruction>
21          <MESSAGE ID>...</MESSAGE ID>
22          <PARTICIPANT_NAME /PARTICIPANT_NAME>
23          <DATE_SENT>2024-07-06T07:00:10</DATE_SENT>
24          <DISPATCH_TYPE>RESV</DISPATCH_TYPE>
25          <STATE>Timed Out</STATE>
26          <ACTIVE>>false</ACTIVE>
27          <RESOURCE_ID>'...</RESOURCE_ID>
28          <DELIVERY_DATE /DELIVERY_DATE>
29          <DELIVERY_HOUR> /DELIVERY_HOUR>
30          <DELIVERY_INTERVAL>1</DELIVERY_INTERVAL>
31          <AMOUNT> /AMOUNT>
32          <RESERVE_CLASS>10S</RESERVE_CLASS>
33          <LAST_UPDATED>...</LAST_UPDATED>
34        </DispatchInstruction>

```

5 In the retrieveDispatch request payload, update the value of the delivery date field:

Only dispatches of the specified delivery date(s) are displayed.

Step **User Action**

Expected Result

<DELIVERY_DATE> Delivery Date
the dispatch instruction was
sent</DELIVERY_DATE>
Select Run to send the request.

The screenshot displays a SOAP client interface. On the left, a search tree shows a project structure with 'retrieveDispatch' selected. The main area shows a configuration for a POST request to an explicit endpoint. The 'Payload' tab is active, showing an XML request. A red box highlights the '<DELIVERY_DATE>2024-09-04</DELIVERY_DATE>' element within the '<RetrieveDispatch>' tag. The XML payload includes various fields such as MESSAGE_ID, DISPATCH_TYPE, STATE, DATE_SENT, PARTICIPANT_NAME, ALT_SYNC_TIME, ACTIVE, RESOURCE_ID, MLP_TIME, DELIVERY_HOUR, DELIVERY_INTERVAL, DELIVERY_START_TIME, DELIVERY_STOP_TIME, RESPONDER, LAST_UPDATED_SINCE, SENT_SINCE, HISTORY_DAYS, EFFECTIVE_TIME, and SYNC_TIME.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="">
3   <s:Body>
4     <RetrieveDispatch xmlns="">
5       <Filters xmlns="">
6         <DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
7       </Filters>
8     </RetrieveDispatch>
9   </s:Body>
10 </s:Envelope>
```

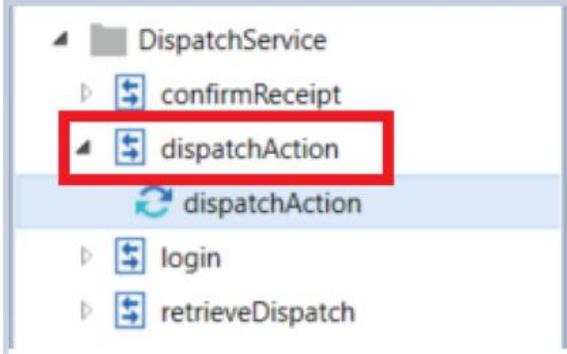
Step User Action

Expected Result

The screenshot displays a SOAP client interface with a search tree on the left and a response view on the right. The search tree shows a project named 'New Project' containing a 'DispatchService' with several actions: 'confirmReceipt', 'dispatchAction', 'login', and 'retrieveDispatch'. The 'retrieveDispatch' action is highlighted with a red box. The response view shows a status code of '200 OK' and a response time of '842' ms. The XML payload is displayed in a code editor, showing a SOAP envelope with a body containing two 'DispatchInstruction' elements. The first 'DispatchInstruction' element contains several fields, including 'DELIVERY_DATE' which is highlighted with a red box. The second 'DispatchInstruction' element is partially visible at the bottom of the screenshot.

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
2 <soap:Body>
3 <ns2:RetrieveDispatchResponse xmlns:ns2="
4 <DispatchInstructions>
5 <DispatchInstruction>
6 <MESSAGE_ID>|_ ;</MESSAGE_ID>
7 <PARTICIPANT_NAME>.....</PARTICIPANT_NAME>
8 <DATE_SENT>2024-09-04T02:00:10</DATE_SENT>
9 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
10 <STATE>Timed Out</STATE>
11 <ACTIVE>>false</ACTIVE>
12 <RESOURCE ID> :/RESOURCE ID>
13 <DELIVERY_DATE>2024-09-04T00:00:00</DELIVERY_DATE>
14 <DELIVERY_HOUR>3</DELIVERY_HOUR>
15 <DELIVERY_INTERVAL>1</DELIVERY_INTERVAL>
16 <AMOUNT </AMOUNT>
17 <LAST_UPDATED>2024-09-04T02:01:10</LAST_UPDATED>
18 </DispatchInstruction>
19 <DispatchInstruction>
20 <MESSAGE_ID> </MESSAGE_ID>
21 <PARTICIPANT_NAME> .</PARTICIPANT_NAME>
22 <DATE_SENT>2024-09-04T02:00:10</DATE_SENT>
23 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
24 <STATE>Timed Out</STATE>
25 <ACTIVE>>false</ACTIVE>
26 <RESOURCE ID> </RESOURCE ID>
27
28 <DELIVERY_HOUR>3</DELIVERY_HOUR>
29 <DELIVERY_INTERVAL>1</DELIVERY_INTERVAL>
30 <AMOUNT>516.78</AMOUNT>
31 <LAST_UPDATED>2024-09-04T02:01:10</LAST_UPDATED>
32 </DispatchInstruction>
```

DS_API_TC02: Actioning Dispatches - New dispatches can either be accepted or rejected through an API

Step	User Action	Expected Result
1	<p>Once the user connected to is connected to Dispatch Services using an API system, the user will see a "Dispatch Service" folder on the left panel containing 4 subfolders: "confirmReceipt", "dispatchAction", "login", and "retrieveDispatch". Select the dispatchAction subfolder.</p>	<p>The dispatchAction request payload should open.</p>  <p>The screenshot shows a tree view under the 'DispatchService' folder. It contains four subfolders: 'confirmReceipt', 'dispatchAction', 'login', and 'retrieveDispatch'. The 'dispatchAction' folder is selected and highlighted with a red rectangular box. Below the tree view, there is a blue bar with a refresh icon and the text 'dispatchAction'.</p>
2	<p>Using the retrieveDispatch request payload, update the below fields to retrieve new dispatches and note the ID of the new dispatch that is returned:</p> <pre data-bbox="226 1101 554 1198"><DISPATCH_TYPE>Dispatch Type</DISPATCH_TYPE> <STATE>New</STATE></pre> <p>Select Run to send the request.</p>	<p>The ID of the dispatch is noted in the retrieveDispatch response, which is enclosed within the <MESSAGE_ID> field.</p>

Step User Action

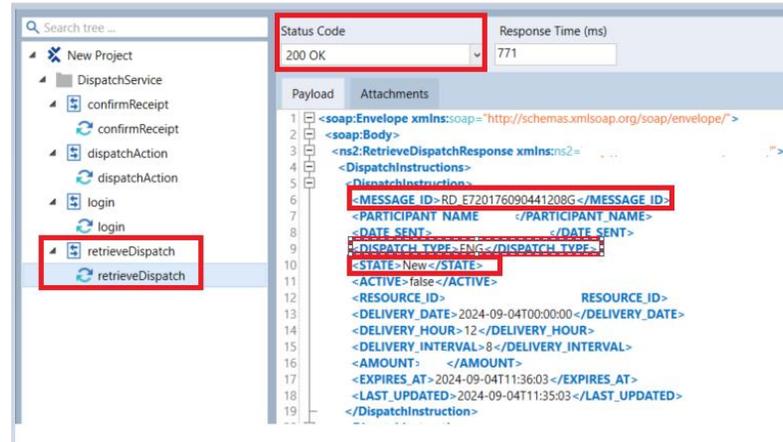
Expected Result

The screenshot displays a SOAP client interface. On the left, a search tree shows a project named 'New Project' containing a 'DispatchService' with several actions: 'confirmReceipt', 'dispatchAction', 'login', and 'retrieveDispatch'. The 'retrieveDispatch' action is highlighted with a red box. The main editor on the right shows the XML payload for this action. The payload is an XML document with the following structure:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns="1
3 <s:Body>
4 <RetrieveDispatch xmlns=""
5 <Filters xmlns=""
6
7
8 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
9 <STATE>New</STATE>
10
11 <!--
12 <MESSAGE_ID>MESSAGE_ID1</MESSAGE_ID>
13 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
14 <STATE>Accepted</STATE>
15 <DATE_SENT>1900-01-01</DATE_SENT>
16 <DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
17
18 <PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>
19
20 <ALT_SYNC_TIME>2024-07-13T15:01:00</ALT_SYNC_TIME>
21 <ACTIVE>true</ACTIVE>
22 <RESOURCE_ID>RESOURCE_ID1</RESOURCE_ID>
23 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
24 <DELIVERY_HOUR>16</DELIVERY_HOUR>
25 <DELIVERY_INTERVAL>8</DELIVERY_INTERVAL>
26 <DELIVERY_START_TIME>1900-01-01T01:01:01Z</DELIVERY_START_TIME>
27 <DELIVERY_STOP_TIME>1900-01-01T01:01:01Z</DELIVERY_STOP_TIME>
28 <RESPONDER>RESPONDER1</RESPONDER>
29 <LAST_UPDATED_SINCE>1900-01-01T01:01:01Z</LAST_UPDATED_SINCE>
30 <SENT_SINCE>1900-01-01T01:01:01Z</SENT_SINCE>
31 <HISTORY_DAYS>1472740310</HISTORY_DAYS>
32 <EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>
33 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
34 <SYNC_TIME>1900-01-01T01:01:01Z</SYNC_TIME>
35 <ALT_SYNC_TIME>1900-01-01T01:01:01Z</ALT_SYNC_TIME> -->
36 </Filters>
37 <offset xmlns="">0</offset>
38 <limit xmlns="">100</limit>
39 </RetrieveDispatch>
40 </s:Body>
41 </s:Envelope>
```

Step User Action

Expected Result



3 In the dispatchAction request payload, update the value of the below fields based on the action you wish to apply to the dispatch instruction.

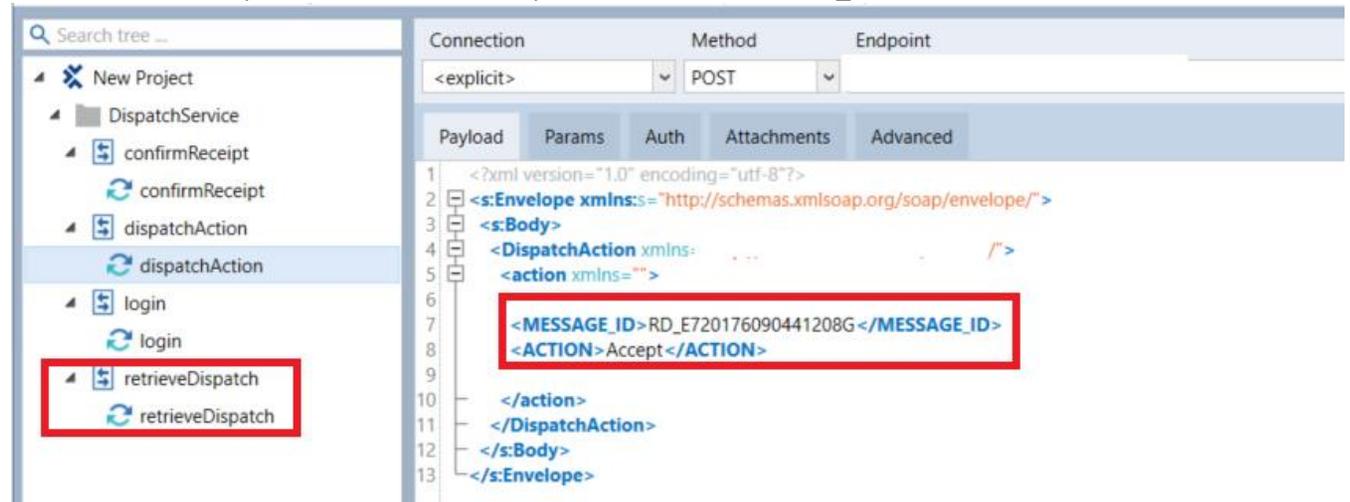
To accept the dispatch instruction:

<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>
<ACTION>Accept</ACTION>

To reject the dispatch instruction:

<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>

The file should be updated with the values specified for the MESSAGE_ID and ACTION fields.



Step User Action

Expected Result

<ACTION>Reject</ACTION>

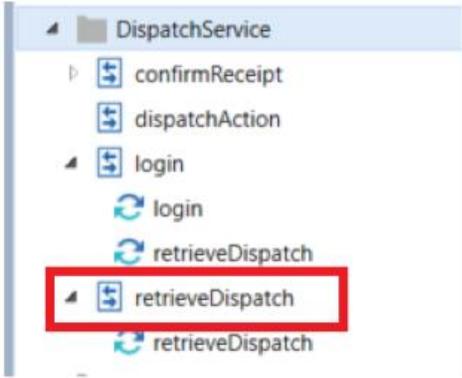
4 Select **Run** to send the request.

A response should be received with "Status Code: 200 OK."
The state of the dispatch will be updated based on the selected action: [Status] = Accepted / Rejected.

The screenshot displays a SOAP client interface. On the left, a search tree shows the project structure with 'DispatchService' expanded to show 'dispatchAction' selected and highlighted with a red box. On the right, the 'Payload' tab shows the SOAP response. At the top, the 'Status Code' is '200 OK' and the 'Response Time (ms)' is '97', both highlighted with red boxes. The XML payload is as follows:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap:Body>
3 <ns2:DispatchActionResponse xmlns:ns2="
4 <actionResponse>
5 <MESSAGE_ID>...</MESSAGE_ID>
6 <PARTICIPANT_NAME>.../PARTICIPANT_NAME>
7 <STATE>Accepted</STATE>
8 <RESPONDER>...</RESPONDER>
9 </actionResponse>
10 </ns2:DispatchActionResponse>
11 </soap:Body>
12 </soap:Envelope>
```

DS_API_TC03: Retrieving Commitments - START/DECOM/EXTEND Commitments can be retrieved through an API

Step	User Action	Expected Result
1	<p>Once the user connected to is connected to Dispatch Service using an API system, the user will see a "Dispatch Service" folder on the left panel containing 4 subfolders: "confirmReceipt", "dispatchAction", "login", and "retrieveDispatch". Select the retrieveDispatch subfolder.</p>	<p>The retrieveDispatch request payload should open.</p>  <p>The screenshot shows a tree view of folders under 'DispatchService'. The folders are 'confirmReceipt', 'dispatchAction', 'login', and 'retrieveDispatch'. The 'retrieveDispatch' folder is highlighted with a red box.</p>
2	<p>In the retrieveDispatch request, update the following field to retrieve Binding Start-Up for Commitments: <DISPATCH_TYPE>START</DISPATCH_TYPE> Select Run to send the request.</p>	<p>Only Commitments of START type are displayed.</p>

Step

User Action

Expected Result

The screenshot displays a REST client interface with a search tree on the left and a configuration panel on the right. The search tree shows a project named 'New Project' containing a service 'DispatchService' with several operations. The 'retrieveDispatch' operation is selected and highlighted with a red box. The configuration panel shows the connection method as 'POST' and the endpoint as '<explicit>'. The 'Payload' tab is active, showing an XML request body. The XML body is a SOAP envelope containing a 'RetrieveDispatch' element with a 'DISPATCH_TYPE' set to 'START'. The 'Filters' element is empty. The XML body is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="" i:"">
  <s:Body>
    <RetrieveDispatch xmlns="" i:"">
      <Filters xmlns="">
        <DISPATCH_TYPE>START</DISPATCH_TYPE>
      </Filters>
    </RetrieveDispatch>
  </s:Body>
</s:Envelope>
```

Step User Action

Expected Result

Search tree ...

- New Project
 - DispatchService
 - confirmReceipt
 - confirmReceipt
 - dispatchAction
 - login
 - login
 - retrieveDispatch
 - retrieveDispatch

Status Code: 200 OK

Response Time (ms): 873

Payload

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap:Body>
3 <ns2:RetrieveDispatchResponse xmlns:ns2=
4 <DispatchInstructions>
5 <DispatchInstruction>
6 <MESSAGE_ID>
7 <PARTICIPANT_NAME> </PARTICIPANT_NAME>
8 <DATE_SENT>2024-07-14T01:30:02</DATE_SENT>
9 <DISPATCH_TYPE>START</DISPATCH_TYPE>
10 <STATE>Timed Out</STATE>
11 <ACTIVE>>false</ACTIVE>
12 <RESOURCE_ID>L
13 <EFFECTIVE_TIME>00:00</EFFECTIVE_TIME>
14 <MLP_TIME>2024
15 <SYNC_TIME>202
16 <LAST_UPDATED>
17 </DispatchInstruction>
18 <DispatchInstruction>
19 <MESSAGE_ID>L
20 <PARTICIPANT_NAME>
21 <DATE_SENT>2024-07-14T02:30:02</DATE_SENT>
22 <DISPATCH_TYPE>START</DISPATCH_TYPE>
23 <STATE>Timed Out</STATE>
24 <ACTIVE>>false</ACTIVE>
25 <RESOURCE_ID>
26 <EFFECTIVE_TIME>:00:00</EFFECTIVE_TIME>
27 <MLP_TIME>202
28 <SYNC_TIME>20:
29 <LAST_UPDATED>
30 </DispatchInstruction>
```

Step	User Action	Expected Result
------	-------------	-----------------

3	<p>In the retrieveDispatch request, update the following fields to retrieve Extension for Commitments:</p> <pre><DISPATCH_TYPE>DECOM</DISPATCH_TYPE></pre> <p>Select Run to send the request.</p>	<p>Only commitments of DECOM type are displayed.</p>
---	--	--

The screenshot displays a SOAP client interface with a search tree on the left and a SOAP message editor on the right. The search tree shows a project structure with 'retrieveDispatch' selected. The SOAP message editor shows the following XML:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns:="" ...>
3   <s:Body>
4     <RetrieveDispatch xmlns="" ... />
5     <Filters xmlns="">
6
7
8       <DISPATCH_TYPE>DECOM</DISPATCH_TYPE>
9
10
11 <!--
12 <MESSAGE_ID>MESSAGE_ID1</MESSAGE_ID>
13 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
14 <STATE>Accepted</STATE>
15 <DATE_SENT>1900-01-01</DATE_SENT>
16   <DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
17   <STATE>New</STATE>
18   <PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>
19
20   <ALT_SYNC_TIME>2024-07-13T15:01:00</ALT_SYNC_TIME>
21   <ACTIVE>true</ACTIVE>
22   <RESOURCE_ID>RESOURCE_ID1</RESOURCE_ID>
23   <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
24   <DELIVERY_HOUR>16</DELIVERY_HOUR>
25   <DELIVERY_INTERVAL>8</DELIVERY_INTERVAL>
26   <DELIVERY_START_TIME>1900-01-01T01:01:01Z</DELIVERY_START_TIME>
27   <DELIVERY_STOP_TIME>1900-01-01T01:01:01Z</DELIVERY_STOP_TIME>
28   <RESPONDER>RESPONDER1</RESPONDER>
29   <LAST_UPDATED_SINCE>1900-01-01T01:01:01Z</LAST_UPDATED_SINCE>
30   <SENT_SINCE>1900-01-01T01:01:01Z</SENT_SINCE>
31   <HISTORY_DAYS>1472740310</HISTORY_DAYS>
32   <EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>
33   <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
34   <SYNC_TIME>1900-01-01T01:01:01Z</SYNC_TIME>
35   <ALT_SYNC_TIME>1900-01-01T01:01:01Z</ALT_SYNC_TIME> -->
36 </Filters>
37 </RetrieveDispatch>
38 </s:Body>
39 </s:Envelope>

```

Step User Action

Expected Result

The screenshot displays a SOAP client interface. On the left, a search tree shows the project structure with 'retrieveDispatch' selected. The main area shows the SOAP response details:

- Status Code: 200 OK
- Response Time (ms): 885
- Payload: XML response containing two dispatch instructions. The first instruction has a DISPATCH_TYPE of DECOM and a STATE of Timed Out. The second instruction also has a DISPATCH_TYPE of DECOM and a STATE of Timed Out.

4 In the retrieveDispatch request, update the following fields to Only commitments of EXTEND type are displayed.

Step	User Action	Expected Result
------	-------------	-----------------

retrieve Extension for Commitments:
 <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
 Select **Run** to send the request.

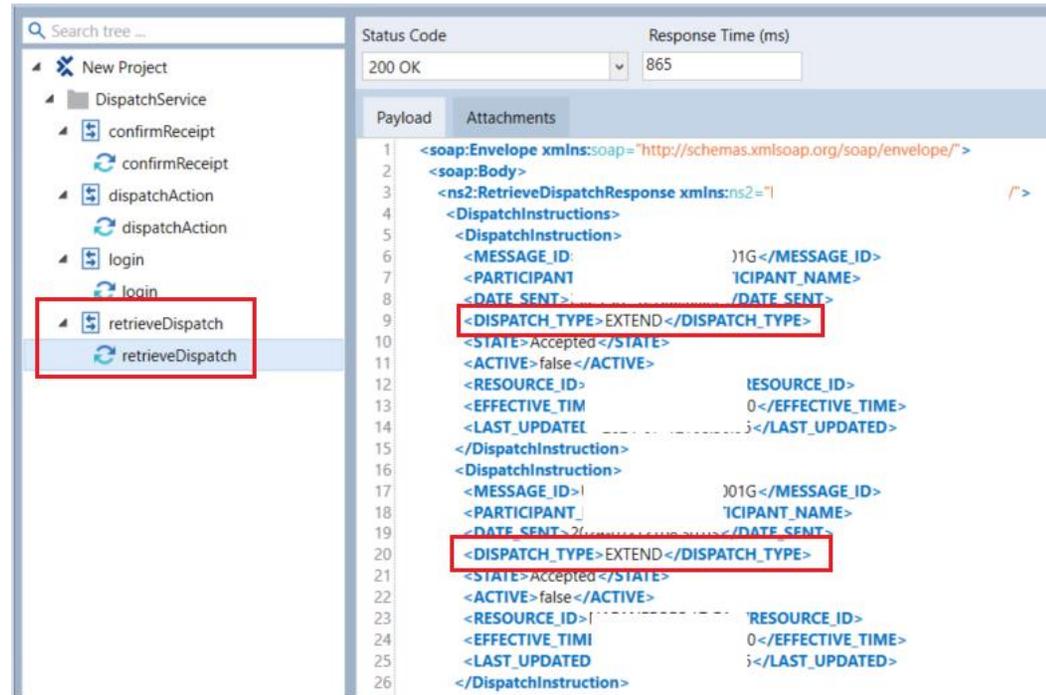
The screenshot displays a SOAP client interface. On the left, a search tree shows a project structure under 'DispatchService' with several actions. The 'retrieveDispatch' action is selected and highlighted with a red box. On the right, the 'Payload' tab is active, showing an XML message. A red box highlights the following XML element within the message body:

```
<DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
```

The full XML message shown in the editor is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="s/">
  <s:Body>
    <RetrieveDispatch xmlns="s">
      <Filters xmlns="">
        <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
      </Filters>
      <offset xmlns="">0</offset>
      <limit xmlns="">100</limit>
    </RetrieveDispatch>
  </s:Body>
</s:Envelope>
```

Step	User Action	Expected Result
------	-------------	-----------------



- | | | |
|---|---|--|
| 5 | <p>In the retrieveDispatch request payload, update the following field:</p> <pre><MLP_TIME>Specified MLP Time</MLP_TIME></pre> <p>The format should be YYYY-MM-DDTHH24:MI:SS. For example: 2020-01-27T14:30:00.</p> | <p>All dispatches with the specified MLP time are displayed.</p> |
|---|---|--|

Step **User Action** **Expected Result**

Select **Run** to send the request payload.

The screenshot shows a SOAP client interface. On the left, a search tree displays a project structure under 'DispatchService'. The 'retrieveDispatch' action is selected and highlighted with a red box. The main window shows the SOAP request details, including the connection (POST), endpoint, and payload. The payload is an XML document with the following structure:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <RetrieveDispatch xmlns="http://schemas.xmlsoap.org/soap/envelope/">
      <Filters xmlns="">
        <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
      </Filters>
      <offset xmlns="">0</offset>
      <limit xmlns="">100</limit>
    </RetrieveDispatch>
  </s:Body>
</s:Envelope>
```

Step User Action

Expected Result

The screenshot displays a SOAP client interface with the following details:

- Status Code:** 200 OK
- Response Time (ms):** 236
- Search tree:** A tree view on the left shows the project structure. The 'retrieveDispatch' action is selected and highlighted with a red box.
- Payload:** The XML response is shown in the main area. It contains two 'RetrieveDispatchResponse' elements. The first element has an 'MLP_TIME' attribute with the value '2024-07-12T15:00:10', which is highlighted with a red box. The second element has an 'MLP_TIME' attribute with the value '2024-07-12T12:00:00', also highlighted with a red box.

Step	User Action	Expected Result
------	-------------	-----------------

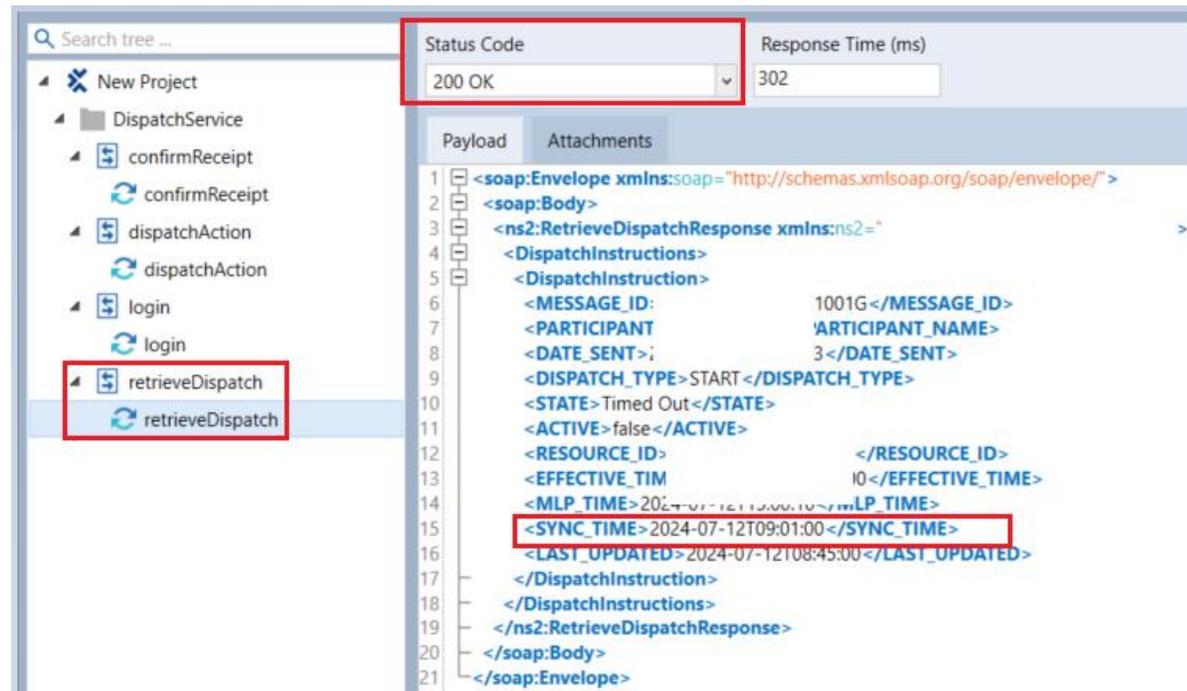
6	<p>In the retrieveDispatch request payload, update the following field: <SYNC_TIME>Resource Scheduled to synchronize time</SYNC_TIME> The format should be YYYY-MM-DDTHH24:MI:SS. For example: 2020-01-27T14:30:00. Select Run to send the request payload.</p>	<p>All dispatches with the specified SYNC time are displayed.</p>
---	---	---

The screenshot shows a SOAP client interface with a search tree on the left and a main view on the right. The search tree shows a project named 'New Project' with a service 'DispatchService' containing several actions, including 'retrieveDispatch'. The main view shows the 'Payload' tab selected, displaying the SOAP request XML. The response XML is also visible, showing a list of dispatches. A red box highlights the updated <SYNC_TIME> field in the request, and another red box highlights the same field in the response.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns="1
3 <s:Body>
4 <RetrieveDispatch xmlns=
5 <Filters xmlns="">
6
7
8 <SYNC_TIME>1900-01-01T01:01:01Z</SYNC_TIME>
9
10 <!--
11 <MESSAGE_ID>MESSAGE_ID1</MESSAGE_ID>
12 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
13 <STATE>Accepted</STATE>
14 <DATE_SENT>1900-01-01</DATE_SENT>
15 <DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
16 <STATE>New</STATE>
17 <PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>
18 <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
19 <ALT_SYNC_TIME>2024-07-13T15:01:00</ALT_SYNC_TIME>
20 <ACTIVE>>true</ACTIVE>
21 <RESOURCE_ID>RESOURCE_ID1</RESOURCE_ID>
22 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
23 <DELIVERY_HOUR>16</DELIVERY_HOUR>
24 <DELIVERY_INTERVAL>8</DELIVERY_INTERVAL>
25 <DELIVERY_START_TIME>1900-01-01T01:01:01Z</DELIVERY_START_TIME>
26 <DELIVERY_STOP_TIME>1900-01-01T01:01:01Z</DELIVERY_STOP_TIME>
27 <RESPONDER>RESPONDER1</RESPONDER>
28 <LAST_UPDATED_SINCE>1900-01-01T01:01:01Z</LAST_UPDATED_SINCE>
29 <SENT_SINCE>1900-01-01T01:01:01Z</SENT_SINCE>
30 <HISTORY_DAYS>1472740310</HISTORY_DAYS>
31 <EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>
32 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
33
34 <ALT_SYNC_TIME>1900-01-01T01:01:01Z</ALT_SYNC_TIME> -->
35 </Filters>
36 <offset xmlns="">0</offset>
37 <limit xmlns="">100</limit>
38 </RetrieveDispatch>
39 </s:Body>
40 </s:Envelope>
  
```

Step User Action Expected Result



7 In the retrieveDispatch request payload, update the following field:
<EFFECTIVE_TIME> Time associated with the product</EFFECTIVE_TIME>

All dispatches with the specified effective time are displayed.

Step**User Action****Expected Result**

The format should be YYYY-MM-DDTHH24:MI:SS. For example: 2020-01-27T14:30:00.
Select **Run** to send the request payload.

The screenshot displays the SoapUI interface for a REST client. On the left, the 'New Project' tree shows the 'DispatchService' with several actions, including 'retrieveDispatch', which is highlighted with a red box. The main window shows the configuration for a POST request to 'http://dispatchwsqa-mrp.corp-pp.intp'. The 'Payload' tab is active, showing an XML request body with a '<RetrieveDispatch>' element. The response tab shows an XML response with various fields, including '<EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>', which is highlighted with a red box.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns="http://webservices.ieso.ca/ieso-ds/">
3 <s:Body>
4 <RetrieveDispatch xmlns="http://webservices.ieso.ca/ieso-ds/">
5 <Filters xmlns="">
6
7
8 <EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>
9
10 <!--
11 <MESSAGE_ID>MESSAGE_ID1</MESSAGE_ID>
12 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
13 <STATE>Accepted</STATE>
14 <DATE_SENT>1900-01-01</DATE_SENT>
15 <DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
16 <STATE>New</STATE>
17 <PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>
18 <DISPATCH_TYPE>EXTEND</DISPATCH_TYPE>
19 <ALT_SYNC_TIME>2024-07-13T15:01:00</ALT_SYNC_TIME>
20 <ACTIVE>true</ACTIVE>
21 <RESOURCE_ID>RESOURCE_ID1</RESOURCE_ID>
22 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
23 <DELIVERY_HOUR>16</DELIVERY_HOUR>
24 <DELIVERY_INTERVAL>8</DELIVERY_INTERVAL>
25 <DELIVERY_START_TIME>1900-01-01T01:01:01Z</DELIVERY_START_TIME>
26 <DELIVERY_STOP_TIME>1900-01-01T01:01:01Z</DELIVERY_STOP_TIME>
27 <RESPONDER>RESPONDER1</RESPONDER>
28 <LAST_UPDATED_SINCE>1900-01-01T01:01:01Z</LAST_UPDATED_SINCE>
29 <SENT_SINCE>1900-01-01T01:01:01Z</SENT_SINCE>
30 <HISTORY_DAYS>1472740310</HISTORY_DAYS>
31
32 <MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
33 <SYNC_TIME>1900-01-01T01:01:01Z</SYNC_TIME>
34 <ALT_SYNC_TIME>1900-01-01T01:01:01Z</ALT_SYNC_TIME> -->
35 </Filters>
36 <offset xmlns="">0</offset>
37 <limit xmlns="">100</limit>
38 </RetrieveDispatch>
39 </s:Body>
40 </s:Envelope>
  
```

Step User Action

Expected Result

The screenshot displays a SOAP client interface. On the left, a search tree shows a project structure with a 'DispatchService' containing several actions. The 'retrieveDispatch' action is highlighted with a red box. On the right, the 'Payload' tab shows the SOAP response. The status code is 200 OK and the response time is 1750 ms, both highlighted with red boxes. The XML payload is as follows:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap:Body>
3 <ns2:RetrieveDispatchResponse xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/">
4 <DispatchInstructions>
5 <DispatchInstruction>
6 <MESSAGE_ID>01G</MESSAGE_ID>
7 <PARTICIPANT_NAME>AWEC</PARTICIPANT_NAME>
8 <DATE_SENT>2024-07-12T08:01:09</DATE_SENT>
9 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
10 <STATE>Timed Out</STATE>
11 <ACTIVE>>false</ACTIVE>
12 <RESOURCE_ID>RIPLEYSOUTH-LT.AG_T3</RESOURCE_ID>
13 <DELIVERY_DATE>2024-07-12T00:00:00</DELIVERY_DATE>
14 <DELIVERY_HOUR>9</DELIVERY_HOUR>
15 <DELIVERY_INTERVAL>1</DELIVERY_INTERVAL>
16 <AMOUNT>6.8</AMOUNT>
17 <LAST_UPDATED>2024-07-12T08:01:09</LAST_UPDATED>
18 </DispatchInstruction>
19 <DispatchInstruction>
20 <MESSAGE_ID>RD_E467526071240903G</MESSAGE_ID>
21 <PARTICIPANT_NAME>AWEC</PARTICIPANT_NAME>
22 <DATE_SENT>2024-07-12T08:10:05</DATE_SENT>
23 <DISPATCH_TYPE>ENG</DISPATCH_TYPE>
24 <STATE>Timed Out</STATE>
25 <ACTIVE>>false</ACTIVE>
26 <RESOURCE_ID>RIPLEYSOUTH-LT.AG_T3</RESOURCE_ID>
27 <DELIVERY_DATE>2024-07-12T00:00:00</DELIVERY_DATE>
28 <DELIVERY_HOUR>9</DELIVERY_HOUR>
29 <DELIVERY_INTERVAL>3</DELIVERY_INTERVAL>
30 <AMOUNT>6.2</AMOUNT>
31 <LAST_UPDATED>2024-07-12T08:11:05</LAST_UPDATED>
32 </DispatchInstruction>
```

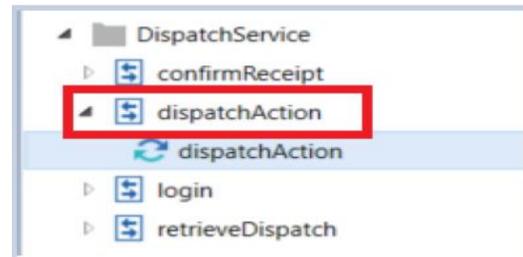
DS_API_TC04: Actioning START Commitments - New START commitments can either be accepted or rejected through an API

Step **User Action**

Expected Result

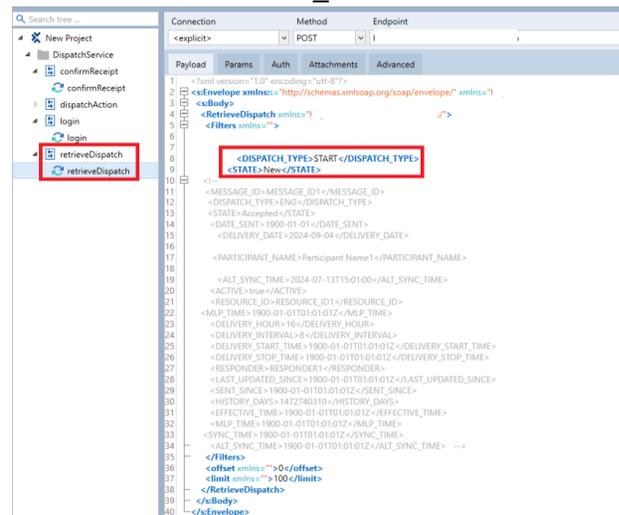
1 Once the user connected to is connected to Dispatch Service using an API system, the user will see a "Dispatch Service" folder on the left panel containing 4 subfolders: "confirmReceipt", "dispatchAction", "login", and "retrieveDispatch". Select the **dispatchAction** subfolder.

The dispatchAction request payload should open.



2 Using the retrieveDispatch request payload, update the following fields to retrieve new START commitments and note the ID of the START commitment that is returned:
 <DISPATCH_TYPE>START</DISPATCH_TYPE>
 <STATE>New</STATE>
 Select **Run** to send the request payload.

The ID of the START commitment is noted in the retrieveDispatch response, which is enclosed within the <MESSAGE_ID> field.



Step User Action

Expected Result

The screenshot shows a SOAP client interface. On the left, a search tree displays a project structure with 'retrieveDispatch' selected. On the right, the 'Payload' tab shows the SOAP response XML. The status code is '200 OK' and the response time is '1436 ms'. The XML payload includes the following highlighted fields:

```
<MESSAGE_ID>UCM000639090441601G</MESSAGE_ID>  
<DISPATCH_TYPE>START</DISPATCH_TYPE>  
<STATE>New</STATE>
```

3

In the dispatchAction request, update the following fields without using the Alt Sync Time filter.

To accept the dispatch instruction:

```
<MESSAGE_ID>Message ID of  
the dispatch instruction to  
action on</MESSAGE_ID>  
<ACTION>Accept</ACTION>
```

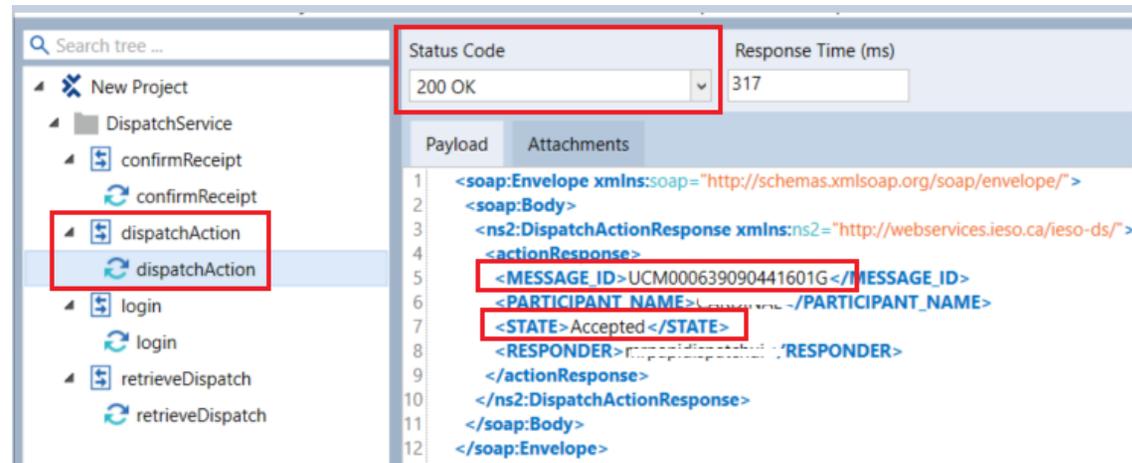
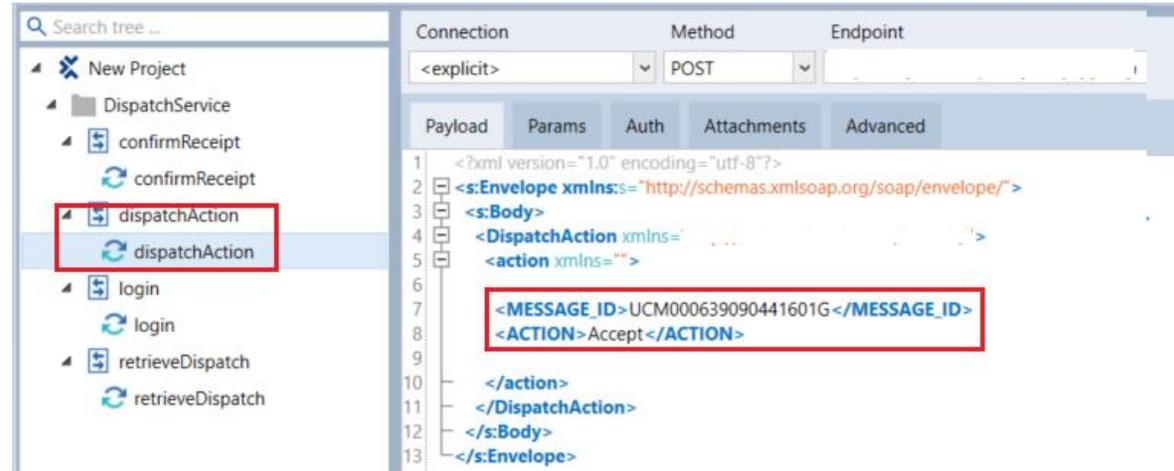
The file should be updated with the values specified for the MESSAGE_ID and ACTION fields.

Step User Action

Expected Result

To reject the dispatch instruction:
<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>
<ACTION>Reject</ACTION>

Select **Run** to send the request payload.



4 In the dispatchAction request, update the **Alt Sync Time field**

The file should be updated with the values specified for the MESSAGE_ID, ACTION, and ALTERNATE_SYNC_TIME fields.

Step User Action

Expected Result

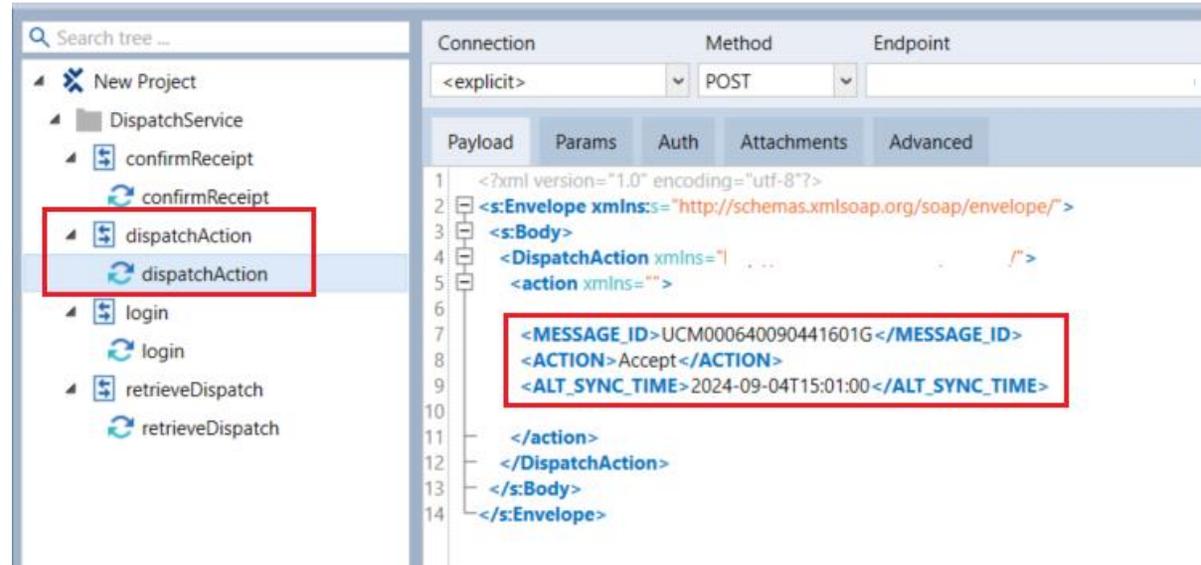
for a dispatch instruction you are accepting:

<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>
<ACTION>Accept</ACTION>
<ALT_SYNC_TIME>Alt Sync Time</ALT_SYNC_TIME>

The format should be YYYY-MM-DDTHH24:MI:SS. For example: 2020-01-27T14:30:00.

Note: the Alt Sync Time must meet the following conditions:

- [Alt Sync Time] > [Current Time]
- [Alt Sync Time] >= [Sync Time] - 1 Hour[
- Alt Sync Time] <= [Sync Time] + 1 Hour
- [Alt Sync Time] <= [MLP Time]



5 Select **Run** to send the request payload.

A response should be received with "Status Code: 200 OK."
The state of the commitment will be updated based on the selected action: [Status] = Accepted / Rejected.

Step User Action

Expected Result

The screenshot displays a SOAP client interface. On the left, a search tree shows a hierarchy: 'New Project' > 'DispatchService' > 'confirmReceipt' > 'confirmReceipt' > 'dispatchAction' (highlighted with a red box) > 'dispatchAction' > 'login' > 'login' > 'retrieveDispatch' > 'retrieveDispatch'. On the right, the 'Payload' tab is active, showing the following XML response:

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
2 <soap:Body>
3 <ns2:DispatchActionResponse xmlns:ns2="http://webservices.ieso.ca/ieso-ds/" >
4 <actionResponse>
5 <MESSAGE_ID>UCM000640090441601G</MESSAGE_ID>
6 <PARTICIPANT_NAME>...</PARTICIPANT_NAME>
7 <STATE>Accepted</STATE>
8 <ALT_SYNC_TIME>2024-09-04T15:01:00</ALT_SYNC_TIME>
9 <RESPONDER>...</RESPONDER>
10 </actionResponse>
11 </ns2:DispatchActionResponse>
12 </soap:Body>
13 </soap:Envelope>
```

Key elements in the response are highlighted with red boxes: the 'Status Code' field showing '200 OK', the 'Response Time (ms)' field showing '512', the 'MESSAGE_ID' element, and the 'STATE' element with the value 'Accepted'.

DS_API_TC05: Actioning DECOM Commitments - New DECOM commitments can be accepted or rejected through an API

Step User Action

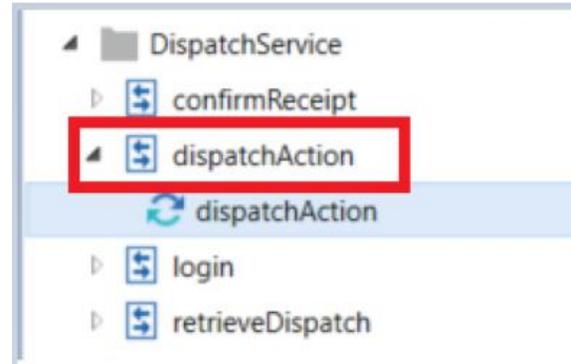
Expected Result

- 1 Once the user connected to is connected to Dispatch Service using an API system, the user will see a "Dispatch Service" folder on the left panel
- The dispatchAction request payload should open.

Step User Action

Expected Result

containing 4 subfolders:
"confirmReceipt",
"dispatchAction", "login", and
"retrieveDispatch". Select the
dispatchAction subfolder.



- 2 Using the retrieveDispatch request payload, update the following fields to retrieve the new DECOM commitments and note the ID of the DECOM commitment that is returned:
<DISPATCH_TYPE>DECOM</DISPATCH_TYPE>
<STATE>New</STATE>
Select **Run** to send the request payload.

The ID of the DECOM commitment is noted in the retrieveDispatch response, which is enclosed within the <MESSAGE_ID> field.

Step User Action

Expected Result

Connection: POST

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns="1
3 <<Body>
4 <<RetrieveDispatch xmlns=
5 <<Filters xmlns="">
6
7
8 <<DISPATCH_TYPE>DECOM</DISPATCH_TYPE>
9 <<STATE>New</STATE>
10
11
12 <<MESSAGE_ID>MESSAGE_ID1</MESSAGE_ID>
13 <<DISPATCH_TYPE>ENG</DISPATCH_TYPE>
14 <<STATE>Accepted</STATE>
15 <<DATE_SENT>1900-01-01</DATE_SENT>
16 <<DELIVERY_DATE>2024-09-04</DELIVERY_DATE>
17
18 <<PARTICIPANT_NAME>Participant Name1</PARTICIPANT_NAME>
19
20 <<ALT_SYNC_TIME>2024-07-13T15:01:00</ALT_SYNC_TIME>
21 <<ACTIVE>true</ACTIVE>
22 <<RESOURCE_ID>RESOURCE_ID1</RESOURCE_ID>
23 <<MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
24 <<DELIVERY_HOUR>16</DELIVERY_HOUR>
25 <<DELIVERY_INTERVAL>8</DELIVERY_INTERVAL>
26 <<DELIVERY_START_TIME>1900-01-01T01:01:01Z</DELIVERY_START_TIME>
27 <<DELIVERY_STOP_TIME>1900-01-01T01:01:01Z</DELIVERY_STOP_TIME>
28 <<RESPONDER>RESPONDER1</RESPONDER>
29 <<LAST_UPDATED_SINCE>1900-01-01T01:01:01Z</LAST_UPDATED_SINCE>
30 <<SENT_SINCE>1900-01-01T01:01:01Z</SENT_SINCE>
31 <<HISTORY_DAYS>1472740310</HISTORY_DAYS>
32 <<EFFECTIVE_TIME>1900-01-01T01:01:01Z</EFFECTIVE_TIME>
33 <<MLP_TIME>1900-01-01T01:01:01Z</MLP_TIME>
34 <<SYNC_TIME>1900-01-01T01:01:01Z</SYNC_TIME>
35 <<ALT_SYNC_TIME>1900-01-01T01:01:01Z</ALT_SYNC_TIME> -->
36 </Filters>
37 <<offset xmlns="">0</offset>
38 <<limit xmlns="">100</limit>
39 </RetrieveDispatch>
40 </Body>
41 </Envelope>

```

Status Code: 200 OK

Response Time (ms): 1201

```

1 <<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <<soap:Body>
3 <<ns2:RetrieveDispatchResponse xmlns:ns2=
4 <<DispatchInstructions>
5 <<DispatchInstruction>
6 <<MESSAGE_ID>UCM000645090441601G</MESSAGE_ID>
7 <<PARTICIPANT_NAME>PARTICIPANT_NAME
8 <<DATE_SENT>2024-09-04T14:30:03</DATE_SENT>
9 <<DISPATCH_TYPE>DECOM</DISPATCH_TYPE>
10 <<STATE>New</STATE>
11 <<ACTIVE>false</ACTIVE>
12 <<RESOURCE_ID>
13 <<EXPIRES_AT>2024-09-04T14:45:00</EXPIRES_AT>
14 <<EFFECTIVE_TIME>
15 <<LAST_UPDATED>
16 </DispatchInstructi...

```

Step User Action

Expected Result

3 In the dispatchAction request payload, update the following fields based on the action you wish to apply to the dispatch instruction.

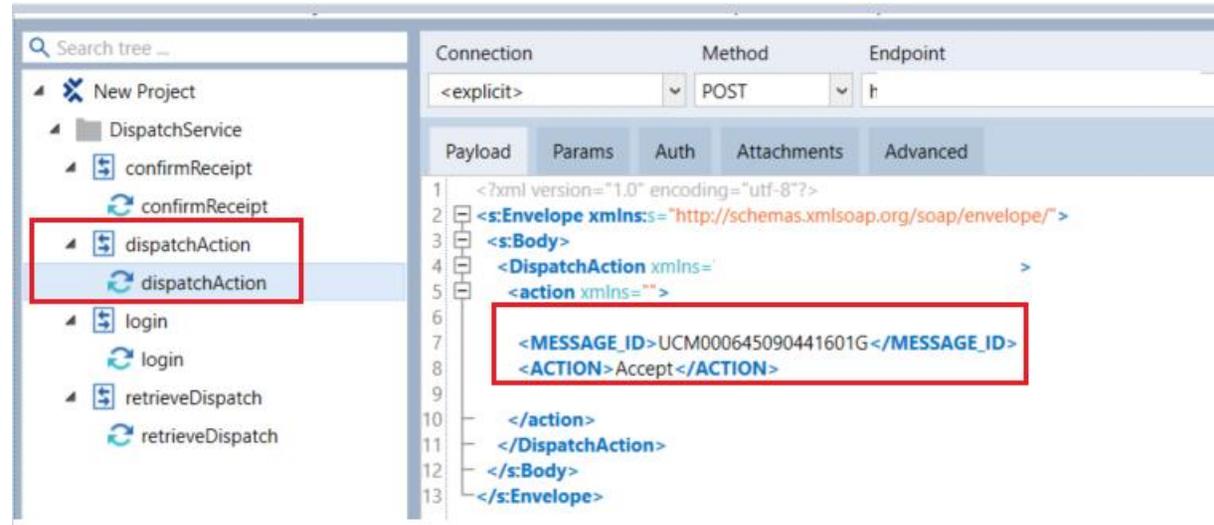
To accept the dispatch instruction:

<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>
<ACTION>Accept</ACTION>

To reject the dispatch instruction:

<MESSAGE_ID>Message ID of the dispatch instruction to action on</MESSAGE_ID>
<ACTION>Reject</ACTION>

The file should be updated with the values specified for the MESSAGE_ID and ACTION fields.



4 Select **Run** to send the request payload.

A response should be received with "Status Code: 200 OK."

Step User Action

Expected Result

The state of the commitment will be updated based on the selected action: [Status] = Accepted / Rejected.

The screenshot displays a SOAP message viewer interface. On the left, a tree view shows the message structure with 'dispatchAction' selected. The main area shows the XML payload with several elements highlighted in red boxes:

- Status Code:** 200 OK
- Response Time (ms):** 120
- XML Element:** `<MESSAGE ID>UCM000645090441601G</MESSAGE ID>`
- XML Element:** `<PARTICIPANT_NAME> .PARTICIPANT_NAME`
- XML Element:** `<STATE>Accepted</STATE>`

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2 <soap:Body>
3 <ns2:DispatchActionResponse xmlns:ns2="..." />
4 <actionResponse>
5 <MESSAGE ID>UCM000645090441601G</MESSAGE ID>
6 <PARTICIPANT_NAME> .PARTICIPANT_NAME
7 <STATE>Accepted</STATE>
8 <RESPONDER> /RESPONDER>
9 </actionResponse>
10 </ns2:DispatchActionResponse>
11 </soap:Body>
12 </soap:Envelope>
```



Additional Resources

- [MRP Market Trials Test Plan](#)
- [Dispatch Notification Service Web Service Design Specifications](#)
- [Dispatch Service Web Service Design Specifications](#)
- [Web Based Dispatch Service, Market Participant's Guide](#)
- [Test Cases for Connectivity Testing \(.zip\)](#)
- [Quick Take: Submission of Test Case Results and Defects in qTest](#)

**Independent Electricity
System Operator**

1600-120 Adelaide Street West
Toronto, Ontario M5H 1T1

Phone: 905.403.6900

Toll-free: 1.888.448.7777

E-mail: customer.relations@ieso.ca

ieso.ca



[@IESO_Tweets](https://twitter.com/IESO_Tweets)

linkedin.com/company/IESO