**WICI Enrol-DeEnrol (Device Admin) Solution – v0.7**

**[1402]: [Device Admin - Refactor] Rename XMLPayloadHelper.java to PayloadHelper.java**

**Technical Description:**

The purpose of this technical task is to rename the XMLPayloadHelper to PayloadHelper. There will be a new story created where the PayloadHelper will also parse JSON payloads, so this renaming is apt and necessary

**Technical Details:**

Rename XMLPayloadHelper.java to PayloadHelper.java

Rename any reference of XMLPayloadHelper to PayloadHelper

**[1403]: [Device Admin - Refactor] Rename XMLPayloadHelperTestCases.java to PayloadHelperTestCases.java**

**Technical Description:**

There will be a new story created where the PayloadHelper will also parse JSON payloads, so this renaming is apt and necessary

**Technical Details:**

A.

Rename XMLPayloadHelperTestCases.java to PayloadHelperTestCases.java

Rename test\_that\_xmlpayloadhelper\_can\_set\_and\_get\_the\_payload to test\_that\_payloadhelper\_can\_set\_and\_get\_an\_xml\_payload

Rename test\_that\_xmlpayloadhelper\_can\_set\_a\_payload\_then\_get\_eventtype to test\_that\_payloadhelper\_can\_set\_an\_xml\_payload\_then\_get\_eventtype

Rename test\_that\_xmlpayloadhelper\_can\_set\_a\_payload\_then\_get\_devicefriendlyname to test\_that\_payloadhelper\_can\_set\_an\_xml\_payload\_then\_get\_devicefriendlyname

Rename test\_that\_xmlpayloadhelper\_can\_set\_a\_payload\_then\_get\_serialnumber to test\_that\_payloadhelper\_can\_set\_an\_xml\_payload\_then\_get\_serialnumber

Rename test\_that\_xmlpayloadhelper\_can\_determine\_if\_a\_payload\_is\_a\_wici\_payload to test\_that\_payloadhelper\_can\_determine\_if\_an\_xml\_payload\_is\_a\_wici\_payload

B.

Rename any reference of XMLPayloadHelper with the tests to PayloadHelper

C.

At the beginning of each test, add the following

systemUnderTest = new PayloadHelper(“text/xml”);

D.

Remove the @Before public void initialize() method

**[1404]: [Device Admin - Refactor] Refactor PayloadHelperTest.java to add tests for JSON payloads**

**Technical Description:**

Refactor the PayloadHelperTestCases.java to add new unit tests for handling JSON payloads

**Technical Details:**

A.

For the following tests at the beginning of each test, add the following

systemUnderTest = new PayloadHelper(“application/json”);

Add the following tests:

1. test\_that\_payloadhelper\_can\_set\_and\_get\_a\_json\_payload

Add new Test Data: (you may need to escape the double quotes)

String jsonPayload = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

Details:

1. Call the setPayload method passing in the String jsonPayload above
2. Assert that the value of the getPayload method call matches the above String jsonPayload
3. test\_that\_payloadhelper\_can\_set\_a\_json\_payload\_then\_get\_eventtype

Test Data:

String jsonPayload = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

Details:

1. Call the setPayload method passing in the String jsonPayload above
2. Assert that the value of getEventType() returns the String “DeviceEnterpriseWipeRequested”
3. test\_that\_payloadhelper\_can\_set\_a\_json\_payload\_then\_get\_devicefriendlyname

Test Data:

String jsonPayload = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

Details:

1. Call the setPayload method passing in the String jsonPayload above
2. Assert that the value of the getDeviceFriendlyName method returns String “WIC-41062e0d324290d9”
3. test\_that\_payloadhelper\_can\_set\_a\_json\_payload\_then\_get\_serialnumber

Test Data:

String jsonPayload = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

Details:

1. Call the setPayload method passing in the String xmlPayload above
2. Assert that the value of the getSerialNumber method returns String “41062e0d324290d9”
3. test\_that\_payloadhelper\_can\_determine\_if\_a\_json\_payload\_is\_a\_wici\_payload

Test Data:

String jsonPayloadWICI = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

String jsonPayloadNONWICI = “{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"ZAZA-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”

Details:

1. Call the setPayload method passing in the String jsonPayloadWICI above
2. Assert that the value of the isWICIEvent method returns Boolean true
3. Call the setPayload method passing in the String jsonPayloadNONWICI above
4. Assert that the value of the isWICIEvent method returns Boolean false

**[1405]: [Device Admin Refactor] Refactor PayloadHelper.java to handle JSON payloads**

**Technical Description:**

Refactor PayloadHelper class to handle both XML and JSON payloads

Inputs:

Only one type of JSON Payloads:

1. DeEnrol:

“{"EventId":229,"EventType":"DeviceEnterpriseWipeRequested","DeviceId":280,"DeviceFriendlyName":"WIC-41062e0d324290d9","EnrollmentEmailAddress":"michael.kirton@cantire.com","EnrollmentUserName":"WIC","EventTime":"\/Date(1391182403986)\/","EnrollmentStatus":"EnterpriseWipePending","CompromisedStatus":"","CompromisedTimeStamp":"\/Date(1391182409025)\/","ComplianceStatus":"Compliant","PhoneNumber":"","Udid":"System.Byte[]","SerialNumber":"41062e0d324290d9","MACAddress":"System.Byte[]"}”DeEnrol:

﻿

**Technical Details:**

1. Create the constructor
   * public PayloadHelper ( String argContentType)
2. Alter the class to check the argContentType via a private data member
   * If argContentType == “text/xml”, then it handles XML payloads as it does today
   * If argContentType == “application/json” then it handls JSON payloads

**[1406]: [Device Admin Refactor] Refactor WICIServletMediator.java to parse the content-type from the Request Header**

**Technical Description:**

Refactor WICIServletMediator.java to get the content type from the request header

Technical Details:

1. Create a new public method getContentType()
   1. Return the content-type attribute from the Request Header object.

**[1407]: [Device Admin] Create DeviceAdminServlet.java**

**Technical Description:**

A new servlet called DeviceAdminServlet will be called from the Airwatch interface and pass in a payload to the request object

**Technical Details:**

1. Create a new servlet “DeviceAdminServlet”
2. Inputs/Params:
   1. payload, a string in XML format or JSON format
3. Extend WICIServlet following the same pattern we use for other servlets involving the WICIServletMediator
4. In the handleRequest method:
   1. Instantiate a new DeviceAdminHelper() passing the servletMediator as an argument
      1. DeviceAdminHelper deviceAdminHelper = new DeviceAdminHelper( servletMediator );
   2. Call the deviceAdminHelper.processPayload() method

**[1408]: [Device Admin] Create DeviceAdminHelper.java to process the payload from the DeviceAdminServlet**

**Technical Description:**

Create a helper class to take action on the parsed field from the XML/JSON Payload. This class will be used directly within the new servlet DeviceAdminServlet

**Technical Details:**

1. Create a new class DeviceAdminHelper.java
2. Create a private member, servletMediator of type ServletMediator
   1. ServletMediator servletMediator = null;
3. Create a private member, payloadHelper of type PayloadHelper
   1. PayloadHelper payloadHelper = null;
4. Create overloaded constructor:
   1. Public DeviceAdminHelper( ServletMediator argServletMediator)
      1. servletMediator = argServletMediator;
      2. payloadHelper = new PayloadHelper(servletMediator.getContentType);
5. Create a private method isWICIEvent:
   1. Private Boolean isWICIEvent();
      1. Returns payloadHelper.isWICIEvent();
6. Create a private method enrolRequired:
   1. Private Boolean enrolRequired();
      1. Returns payloadHelper.getEventType()== “CompromisedStatusChanged”;
7. Create a private method deEnrolRequired:
   1. Private Boolean deEnrolRequired();
      1. Returns payloadHelper.getEventType() == “BreakMDMConfirmed” || “DeviceEnterpriseWipeRequested”;
8. Create a processPayload method that returns a String value
   1. Parse the request object in the servletMediator to get the payload parameter:
      1. String payload = servletMediator.getPayload();
   2. Set the payload:
      1. payloadHelper.setPayload( payload )
   3. Implement the following logic:
      1. If isWICIEvent()
         1. if enrolRequired()
            1. new WICIDBHelper().enrolWICIDevice( payloadHelper.getSerialNumber());
         2. else if deEnrolRequired()
            1. new WICIDBHelper().deEnrolWICIDevice( payloadHelper.getSerialNumber());

**[1409]: [Device Admin Refactor] Refactor WICIDBHelper to add method enrolWICIDevice**

**Technical Description:**

Create a method on the WICIDBHelper to enroll a WICI Device

**Technical Details:**

1. In WICIDBHelper, add the following public method:
   1. Public void enrolWICIDevice( String argSerialNumber );
      1. If serial number exists in WICI\_WHITELIST, then UPDATE WICI\_WHITELIST SET AUTHORIZED=’1’ WHERE AUTHFIELD\_VALUE=argSerialNumber
      2. If serial number does not exist in WICI\_WHITELIST, then INSERT INTO WICI\_WHITELIST (AUTHFIELD\_VALUE,AUTHORIZED) VALUES (argSerialNumber,1)

**[1410]: [Device Admin Refactor] Refactor WICIDBHelper to add method deEnrolWICIDevice**

**Technical Description:**

Create a method on the WICIDBHelper to enroll a WICI Device

**Technical Details:**

1. In WICIDBHelper, add the following public method:
   1. Public void deEnrolWICIDevice( String argSerialNumber );
      1. UPDATE WICI\_WHITELIST SET AUTHORIZED=’1’ WHERE AUTHFIELD\_VALUE=argSerialNumber

**[1411]: [Device Admin Refactor] Refactor WICIDBHelper to add method getAirwatchDFNSearchPrefix**

**Technical Description:**

Create a method on the WICIDBHelper to get the Airwatch Prefix that identifies WICI-related events.

**Technical Details:**

1. In WICIDBHelper, add the following public method:
   1. Public String getAirwatchDFNSearchPrefix();
      1. SELECT CONFIG\_VALUE FROM WICI\_CONFIG WHERE CONFIG\_NAME=’AirwatchDeviceFriendlyNameSearchPrefix’
      2. Return the String result

**[1412]: [Device Admin Test] Create new Project Device Admin Simulator (similar to BRBEcomSimulatedSite)**

**Technical Description:**

Create a new Static Web Project in eclipse to be used as a test harness for the Device Admin feature

**Technical Details:**

1. In eclipse, create a new Static Web Project, called “DeviceAdminSimulator”
2. Copy the index.html, sample.local.properties, build.properties,local.properties and build.xml from the BRB Ecom Simulated Site project and modify for this project
3. Index.html should have the following text fields:
   1. Serial number
   2. AirwatchDeviceFriendlyNameSearchPrefix
4. Index.html should have the following buttons:
   1. CompromisedStatusChanged
   2. BreakMDMConfirmed
   3. DeviceEnterpriseWipeRequested
5. When CompromisedStatusChanged is clicked do the following:
   1. Construct an xml payload, like the example ones in the unit tests, but modify the following tags:
      1. <EventType> CompromisedStatusChanged</EventType>
      2. <SerialNumber>Value from the text field in 3a. </SerialNumber>
      3. <DeviceFriendlyName>Value from the text field in 3b. plus “-“ plus value from the textfield in 3a.</DeviceFriendlyName>
      4. Send the request to DeviceAdminServlet using POST
6. When BreakMDMConfirmed is clicked do the following:
   1. Construct an xml payload, like the example ones in the unit tests, but modify the following tags:
      1. <EventType> BreakMDMConfirmed </EventType>
      2. <SerialNumber>Value from the text field in 3a. </SerialNumber>
      3. <DeviceFriendlyName>Value from the text field in 3b. plus “-“ plus value from the textfield in 3a.</DeviceFriendlyName>
      4. Send the request to DeviceAdminServlet using POST
7. When DeviceEnterpriseWipeRequested is clicked do the following:
   1. Construct a json payload, like the example ones in the unit tests, but modify the following fields:
      1. “EventType” : “DeviceEnterpriseWipeRequested”
      2. “SerialNumber” : “<Value from the text field in 3a.>”
      3. “DeviceFriendlyName” : “<Value from the text field in 3b. plus “-“ plus value from the textfield in 3a.>”
      4. Send the request to DeviceAdminServlet using POST

**[1413]: [Device Admin Test] As an Admin of Devices, I should be able to Enrol a WICI Device with XML**

**Technical Description:**

Use the DeviceAdmin Simulator page to send an Enrol Request to the DeviceAdmin Servlet

**Technical Details:**

**Case 1: Enrol a new device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that does not exist in the WICI\_WHITELIST table

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the CompromisedStatusChanged button

Then

---the WICI\_WHITELIST table should have a row with the following:

---AUTHFIELD\_VALUE = serial number

---AUTHORIZED=1

---DATE\_ADDED should be the current date/time

---DATE\_UPDATED should be blank

**Case 2: Enrol an existing device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that currently exists in the WICI\_WHITELIST table where AUTHORIZED=1 and DATE\_ADDED currently exists

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the CompromisedStatusChanged button

Then

---the WICI\_WHITELIST table should have a row with the following:

---AUTHFIELD\_VALUE = serial number

---AUTHORIZED=1

---DATE\_ADDED value is unchanged

---DATE\_UPDATED value is current date/time

**[1414]: [Device Admin Test] As an Admin of Devices, I should be able to DeEnrol a WICI Device with XML**

**Technical Description:**

Use the DeviceAdmin Simulator to send a DeEnrol request in XML format to the DeviceAdmin Servlet

**Technical Details:**

**Case 1: DeEnrol an existing device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that currently exists in the WICI\_WHITELIST table where AUTHORIZED=1 and DATE\_ADDED currently exists or DATE\_UPDATED currently exists

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the BreakMDMConfirmed button

Then

---the WICI\_WHITELIST table should have a row with the following:

---AUTHFIELD\_VALUE = serial number

---AUTHORIZED=0

---DATE\_ADDED value is unchanged

---DATE\_UPDATED should be current date/time

**Case 2: DeEnrol a new device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that does not exist in the WICI\_WHITELIST table

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the BreakMDMConfirmed button

Then

---the WICI\_WHITELIST table should be unchanged

**[1415]: [Device Admin Test] As an Admin of Devices, I should be able to DeEnrol a WICI Device with JSON**

**Technical Description:**

Create a new Static Web Project in eclipse to be used as a test harness for the Device Admin feature

**Technical Details:**

**Case 1: DeEnrol an existing device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that currently exists in the WICI\_WHITELIST table where AUTHORIZED=1 and DATE\_ADDED currently exists or DATE\_UPDATED currently exists

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the DeviceEnterpriseWipeRequested button

Then

---the WICI\_WHITELIST table should have a row with the following:

---AUTHFIELD\_VALUE = serial number

---AUTHORIZED=0

---DATE\_ADDED value is unchanged

---DATE\_UPDATED should be current date/time

**Case 2: DeEnrol a new device**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter serial number that does not exist in the WICI\_WHITELIST table

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of “WIC”

And

---I click on the DeviceEnterpriseWipeRequested button

Then

---the WICI\_WHITELIST table should be unchanged

**[1416]: [Device Admin Test] As an Admin of Devices, WICI\_WHITELIST table should not change if AirwatchDeviceFriendlyNameSearchPrefix is not “WIC”**

**Technical Description:**

Create a new Static Web Project in eclipse to be used as a test harness for the Device Admin feature

**Technical Details:**

Given

---I have launched the DeviceAdmin Simulator

When

---I enter any serial number

And

---I enter a value for AirwatchDeviceFriendlyNameSearchPrefix of something that is NOT “WIC”

And

---I click on either DeviceEnterpriseWipeRequeste, BreakMDMConfirmed or CompromisedStatusChanged button

Then

---the WICI\_WHITELIST table should NOT change