

verifytreatment™

Code Document

Versions 2.0.0



Table of Contents

Code for VisualForce Page	3
Code for Trigger Helper Class	29
Code for Creating Trigger Client Status Update	32

Code for VisualForce Page

```
<apex:page standardController="Lead"
extensions="VerifyTx.CallVerificationMethodUpdate"
lightningStyleSheets="true" sidebar="false" id="page">
  <apex:slds />
  <div class="slds-scope">
    <apex:form id="form">
      <article class="slds-card">
        <div class="slds-card__header slds-grid slds-assistive-text">
          <header class="slds-media slds-media_center
slds-has-flexi-truncate">
            <div class="slds-media__body">
              <h2 class="slds-card__header-title">
                <a href="javascript:void(0);"
class="slds-card__header-link slds-truncate" title="Card Header">
                  <span>Card Header</span>
                </a>
              </h2>
            </div>
          </header>
        </div>
        <div style="padding-top: 20px;text-align: center;
background-color: #f0f0f0;">
          <apex:image id="theImage1"
value="{!$Resource.VerifyTx__Logo}" width="200" height="200"
alt="Description of image here"/>
        </div>
        <div class="slds-card__body slds-card__body_inner hide">
          <br/>
          <div style="padding-top: 20px;text-align: center;">
            <div style="font-size:150%;padding:2%; ">Are you sure
you want to submit for verification?</div>
          </div>
          <table style="width:100%">
            <tr >
              <td style="text-align: right;font-size:100%; width:50%
```

```

; padding : 1%; padding-bottom : 0px; padding-right: 19px;"> Patient First
Name      </td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px; padding-right:
19px;"><b>{!patientFirstName}</b></td>
          </tr>
          <tr style="padding-top : 0px; padding-bottom : 0px;">
          <td style="text-align: right;font-size:100%; width:50%
; padding : 1% ; padding-bottom : 0px; padding-right: 21px;"> Patient
Last Name  </td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px;"><b>{!patientLatName}</b></td>
          </tr>
          <tr style="padding-top : 0px">
          <td style="text-align: right;font-size:100%; width:50%
; padding : 1%; padding-bottom : 0px; padding-right: 25px;"> Patient
Birth Date </td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px;"><b>{!patientDOB}</b></td>
          </tr>
          <tr style="padding-top : 0px">
          <td style="text-align: right;font-size:100%; width:50%
; padding : 1%; padding-bottom : 0px; padding-right: 63px;"> Member Id
</td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px;"><b>{!patientMemId}</b></td>
          </tr>
          <tr style="padding-top : 0px">
          <td style="text-align: right;font-size:100%; width:50%
; padding : 1%; padding-bottom : 0px; padding-right: 10px;"> Insurance
Company    </td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px;"><b>{!patientPayer}</b></td>
          </tr>
          <tr style="padding-top : 0px">
          <td style="text-align: right;font-size:100%; width:50%
; padding : 1%; padding-bottom : 0px; padding-right: 90px;"> Facility
</td>
          <td style="text-align: left; width:50% ; padding:1%;
font-size:100%; padding-bottom : 0px;"><b>{!patientFacility}</b></td>
          </tr>
</table>
<div style="padding-top: 20px;text-align: center;">
  <apex:commandbutton style=" background-color:

```

```

#f5650f; color:#ffffff; border-color: #f5650f;"
action="{!PassVerificationIdUpdate}" rerender="VerificationPopup,
MainSection, ErrorPopup, jsonRes, jsonResponse" value="Verify"
styleClass="slds-button slds-button_success" onclick="hide();"
oncomplete="show();"></apex:commandbutton>
    <apex:commandButton style=" background-color:
#F1F1F1; color:#333333; border-color: #F1F1F1;" immediate="true"
html-formnovalidate="formnovalidate" action="{!myActionInController}"
styleClass="slds-button slds-button_destructive" value="Cancel"/>
    </div>

</div>
    <apex:outputPanel id="MainSection">
        <div class="show" style="display:none;padding-top:
20px;text-align: center;">
            <apex:outputPanel rendered="{!DisplayPopUp1}"><br/>
                <b>Please wait we are verifying your
information.</b></apex:outputPanel>
            </div>
        </apex:outputPanel>
        <apex:outputPanel id="redirect">
            <div class="redirect" style="display:none;padding-top:
20px;text-align: center;">
                <apex:outputPanel ><br/>
                    <b>Please wait we are redirecting you on record
detail page.</b></apex:outputPanel>
            </div>
        </apex:outputPanel>

        <div class="slds-modal__container" style="padding-top:
10px;">
            <apex:outputpanel id="VerificationPopup" >
                <apex:outputpanel layout="block"
rendered="{!DisplayPopup}" styleclass="popupBackground">
                    <div style="font-size:150% ;padding-top:
10px;text-align: center; ">Verification Complete.</div>
                    <section role="dialog" tabindex="-1"
aria-labelledby="modal-heading-01" aria-modal="true"
aria-describedby="modal-content-id-1" class="slds-modal
slds-fade-in-open">
                        <div class="slds-modal__container"
style="padding-top: 0px;">
                            <!-- Modal/Popup Box Header Starts here-->

```

```

        <!--Modal/Popup Box Body Starts here-->
        <div class="slds-modal__content
slds-p-around_medium"
id="modal-content-id-1" >
        style="width: 500px;padding-left: 0;"
        <table style="width:100%">
            <tr >
                <td style="padding-top: 2%;">
Verification Status </td>
                <td style="padding-top: 2%;"><span
style="display: inline-block; background-color: #82e671; color: #0a0909;
font-size: inherit; font-style: normal;"><b>Verified on {!response}</b>
</span></td>
            </tr>
            <tr style="padding-top : 2%;">
                <td style="padding-top: 2%;">
padding-bottom : 0px;">
Coverage </td>
                <td style="padding-top:
2%;"><b>{!coverage}</b></td>
            </tr>
            <tr style="padding-top : 2%">
                <td style="padding-top: 2%;">
Reference Number </td>
                <td style="padding-top:
2%;"><b>{!reference}</b></td>
            </tr>
            <tr style="padding-top : 2%">
                <td style="padding-top: 2%;">
Insurance Type </td>
                <td style="padding-top:
2%;"><b>{!insType}</b></td>
            </tr>
            <tr style="padding-top : 2%">
                <td style="padding-top: 2%;">
Product </td>
                <td style="padding-top:
2%;"><b>{!plan_product}</b></td>
            </tr>
        </table>

```

```

        </div>
        <!--Modal/Popup Box Footer Starts here-->

        <footer class="slds-modal__footer">
            <apex:inputHidden value="{!deductibleStr}"
id="deductible">
                </apex:inputHidden>

                <apex:inputHidden value="{!colnsStr}"
id="colns">
                </apex:inputHidden>

                <apex:inputHidden value="{!coPayStr}"
id="coPays">
                </apex:inputHidden>

                <apex:inputHidden
value="{!outOfPocketStr}" id="outOfPockets">
                </apex:inputHidden>

                <apex:inputHidden value="{!limitationStr}"
id="limitations">
                </apex:inputHidden>

                <apex:inputHidden
value="{!planProductStr}" id="planProducts">
                </apex:inputHidden>

                <apex:inputHidden value="{!insContactstr}"
id="insContacts">
                </apex:inputHidden>

                <apex:commandbutton style="
background-color: #f5650f; color:#ffffff; border-color: #f5650f;"
action="{!ClosePopup}" rerender="redirect" value="Show Details"
styleClass="slds-button slds-button_brand disableBtn"
onclick="DisableBtn();">
                </apex:commandbutton>

```

```

</footer>

</div>
</section>

</apex:outputpanel>

<apex:outputpanel layout="block"
rendered="{!DisplayError}" styleclass="popupBackground">
    <section role="dialog" tabindex="-1"
aria-labelledby="modal-heading-01" aria-modal="true"
aria-describedby="modal-content-id-1" class="slds-modal
slds-fade-in-open">
        <div class="slds-modal__container">
            <!-- Modal/Popup Box Header Starts here-->
            <header class="slds-modal__header">

                </header>
                <!--Modal/Popup Box Body Starts here-->
                <div class="slds-modal__content
slds-p-around_medium" id="modal-content-id-1">
                    <p><b>Verification Status : <span
style="display: inline-block;color: red;font-size:inherit;font-style: normal;
">{!response}
                        </span>
                    </b>
                </p>
            </div>
            <!--Modal/Popup Box Footer Starts here-->
            <footer class="slds-modal__footer">
                <apex:commandbutton style="
background-color: #f5650f; color:#ffffff; border-color: #f5650f;"
action="{!myActionInController}" rerender="VerificationPopup"
value="Close" styleClass="slds-button slds-button_brand">
            </apex:commandbutton>
        </footer>
    </div>
</section>

```



```

        </apex:outputpanel>

    </apex:outputpanel>

    <apex:outputpanel id="ErrorPopup" >
        <apex:outputpanel layout="block"
rendered="{!messages}" styleclass="popupBackground">
            <section role="dialog" tabindex="-1"
aria-labelledby="modal-heading-01" aria-modal="true"
aria-describedby="modal-content-id-1" class="slds-modal
slds-fade-in-open">
                <div class="slds-modal__container">

                    <!--Modal/Popup Box Body Starts here-->
                    <div class="slds-modal__content
slds-p-around_medium" id="modal-content-id-1">
                        <p>
                            <b>Verification Status :
                            <span style="display: inline-block;color:
red;font-size:inherit;font-style: normal;">
                                Required Fields are missing : {!error}
                            </span>
                            </b>
                        </p>
                    </div>
                    <!--Modal/Popup Box Footer Starts here-->
                    <footer class="slds-modal__footer">
                        <apex:commandbutton style="
background-color: #f5650f; color:#ffffff; border-color: #f5650f;"
action="{!myActionInController}" rerender="VerificationPopup"
value="Close" styleClass="slds-button slds-button_brand">
                            </apex:commandbutton>
                        </footer>
                    </div>
                </section>

            </apex:outputpanel>

        </apex:outputpanel>
    </div>

```

```

</article>

</apex:form>
</div>

<style type="text/css">
.customPopup {
background-color: white;
border-style: solid;
border-width: 2px;
left: 20%;
padding: 10px;
position: absolute;
z-index: 9999;
/* These are the 3 css properties you will need to tweak so the pop
up displays in the center of the screen. First set the width. Then set
margin-left to negative half of what the width is. You can also add
the height property for a fixed size pop up.*/
width: 500px;
top: 20%;
}

.disabledTextBox {
background-color: white;
border: 1px solid;
color: black;
cursor: default;
width: 90px;
display: table;
padding: 2px 1px;
text-align:right;
}

.closeButton {
float: right;
}

.text{
align : center;
}

.slds-card {
margin:23px auto;
border: 1px solid #D6D6D6;

```

```
border-radius: 5px;
width: 836px;
font-family: 'Arial', Arial, sans-serif;margin-bottom:100px;
}
</style>
<script>

var i = 0;
function hide(){
    var x = document.getElementsByClassName("hide");
    x[0].style.display = "none";
    var y = document.getElementsByClassName("show");
    y[0].style.display = "block";}
function show(){
    var y = document.getElementsByClassName("show");
    y[0].style.display = "none";
}
function DisableBtn(){
    var z = document.getElementsByClassName("disableBtn");
    z[0].style.visibility = 'hidden';
    document.getElementsByClassName("disableBtn").disabled = true;
    var x = document.getElementsByClassName("show");
    x[0].style.display = "none";
    var y = document.getElementsByClassName("redirect");
    y[0].style.display = "block";

}
</script>

<apex:outputPanel id="jsonRes">
    <script>

    var fullResponseJson = {!fullResponseJson};
    var resultReclId = {!resultReclId};
    var dedSF = [];
    var lnsContacts = [];
    var colnsSF = [];
    var coPays = [];
    var limitations = [];
    var outOfPockets = [];
    var PlanProduct = [];
    var benefitsList = fullResponseJson.message.cache.benefits;
```

```

    // if(benefitsList.hasOwnProperty('no_network')){
    //   Deductible(fullResponseJson.message.cache.benefits.no_network,
    'No Network',resultReclD);
    // }
    if(benefitsList.hasOwnProperty('in_network')){
      Deductible(fullResponseJson.message.cache.benefits.in_network, 'In
Network', resultReclD);
    }
    if(benefitsList.hasOwnProperty('out_of_network')){

Deductible(fullResponseJson.message.cache.benefits.out_of_network,
'Out Of Network', resultReclD);
    }
    if(benefitsList.hasOwnProperty('not_applicable_network')){

Deductible(fullResponseJson.message.cache.benefits.not_applicable_net
work, 'Not Applicable Network', resultReclD);
    }

    function Deductible(netStr, ntwrkType, resultReclD){
      let dedArray = netStr;
      for(let i=0; i<dedArray.length; i++) {
        if(dedArray[i].hasOwnProperty('deductibles') &&
dedArray[i].deductibles.length > 0){
          for(let j=0; j<dedArray[i].deductibles.length; j++){
            var auth;

if(dedArray[i].deductibles[j].hasOwnProperty('authorizationRequired')){
if(dedArray[i].deductibles[j].authorizationRequired==null){
              auth = 'NA';
            }
            else
if(dedArray[i].deductibles[j].authorizationRequired){
              auth = 'Yes';
            }
            else if(!
dedArray[i].deductibles[j].authorizationRequired){
              auth = 'No';
            }
          }
        }
      }
    }
    else {
      auth = 'NA';
    }
  }

```

```

    }
    var payerNote = "";
if(dedArray[i].deductibles[j].hasOwnProperty('payerNotes')){
    var payerNotes =
dedArray[i].deductibles[j].payerNotes;
    for(let k=0; k<payerNotes.length; k++){
        if(payerNote == null || payerNote == ""){
            payerNote = payerNote+" "+payerNotes[k];
        }
        else{
            payerNote = payerNote+', '+payerNotes[k];
        }
    }
}
let ded = {
    "VerifyTx__Auth_Required__c": auth,
    "VerifyTx__Network__c" : ntwrkType,
    "VerifyTx__Service_Type__c" : dedArray[i].name,
    "VerifyTx__Status__c": dedArray[i].status,
    "VerifyTx__Payer_Notes__c" : payerNote,
    "VerifyTx__Result__c" : resultReclId.resultReclId
};

if(dedArray[i].deductibles[j].hasOwnProperty('amountTimePeriod')){
    ded.VerifyTx__Amount_Time_Period__c =
dedArray[i].deductibles[j].amountTimePeriod;
}

if(dedArray[i].deductibles[j].hasOwnProperty('benefitBeginDate')){
    ded.VerifyTx__Benefit_Begin_Date__c =
dedArray[i].deductibles[j].benefitBeginDate;
}

if(dedArray[i].deductibles[j].hasOwnProperty('benefitEndDate')){
    ded.VerifyTx__Benefit_End_Date__c =
dedArray[i].deductibles[j].benefitEndDate;
}

if(dedArray[i].deductibles[j].hasOwnProperty('coverageEndDate')){
    ded.VerifyTx__Coverage_End_Date__c =
dedArray[i].deductibles[j].coverageEndDate;
}

```

```

if(dedArray[i].deductibles[j].hasOwnProperty('coverageStartDate')){
    ded.VerifyTx__Coverage_Start_Date__c =
dedArray[i].deductibles[j].coverageStartDate;
}

if(dedArray[i].deductibles[j].hasOwnProperty('level')){
    ded.VerifyTx__Level__c =
dedArray[i].deductibles[j].level;
}

if(dedArray[i].deductibles[j].hasOwnProperty('notApplicableNetwork')){
    ded.VerifyTx__Not_Applicable_Network__c =
dedArray[i].deductibles[j].notApplicableNetwork;
}

if(dedArray[i].deductibles[j].hasOwnProperty('placeOfService')){
    ded.VerifyTx__Place_of_Service__c =
dedArray[i].deductibles[j].placeOfService;
}

if(dedArray[i].deductibles[j].hasOwnProperty('remaining')){
    ded.VerifyTx__Remaining__c =
parseFloat(dedArray[i].deductibles[j].remaining);
}

if(dedArray[i].deductibles[j].hasOwnProperty('amount')){
    ded.VerifyTx__Amount_Met__c =
parseFloat(dedArray[i].deductibles[j].amount);
}

if(dedArray[i].deductibles[j].hasOwnProperty('PlanEndDate')){
    ded.VerifyTx__Plan_End_Date__c =
dedArray[i].deductibles[j].PlanEndDate;
}

if(dedArray[i].deductibles[j].hasOwnProperty('planStartDate')){
    ded.VerifyTx__Plan_Start_Date__c =
dedArray[i].deductibles[j].planStartDate;
}

if(dedArray[i].deductibles[j].hasOwnProperty('remainingTimePeriod')){
    ded.VerifyTx__Remaining_Time_Period__c =
dedArray[i].deductibles[j].remainingTimePeriod;
}

```

```

    }
    if(dedArray[i].deductibles[j].hasOwnProperty('total')){
        ded.VerifyTx__Total__c =
parseFloat(dedArray[i].deductibles[j].total);
    }if(dedArray[i].deductibles[j].hasOwnProperty('totalTimePeriod')){
        ded.VerifyTx__Total_Time_Period__c =
dedArray[i].deductibles[j].totalTimePeriod;
    }

    dedSF.push(ded);
}
}
    if(dedArray[i].hasOwnProperty('colInsurance') &&
dedArray[i].colInsurance.length > 0){
    for(let j=0; j<dedArray[i].colInsurance.length; j++){
        var auth;
        if(dedArray[i].colInsurance[j].hasOwnProperty('authorizationRequired')){
            if(dedArray[i].colInsurance[j].authorizationRequired==null){
                auth = 'NA';
            }
            else
            if(dedArray[i].colInsurance[j].authorizationRequired){
                auth = 'Yes';
            }
            else if(!
dedArray[i].colInsurance[j].authorizationRequired){
                auth = 'No';
            }
        }
        else{
            auth = 'NA';
        }
        var payerNote = "";
        if(dedArray[i].colInsurance[j].hasOwnProperty('payerNotes')){
            var payerNotes =
dedArray[i].colInsurance[j].payerNotes;
            for(let k=0; k<payerNotes.length; k++){

```

```

        if(payerNote == null || payerNote == ""){
            payerNote = payerNote+" "+payerNotes[k];
        }
        else{
            payerNote = payerNote+', '+payerNotes[k];
        }
    }
}

let colns = {
    "VerifyTx__Auth_Required__c": auth,
    "VerifyTx__Network__c" : ntwrkType,
    "VerifyTx__Service_Type__c" : dedArray[i].name,
    "VerifyTx__Status__c": dedArray[i].status,
    "VerifyTx__Payer_Notes__c": payerNote,
    "VerifyTx__Result__c" : resultReclId.resultReclId
};

if(dedArray[i].colInsurance[j].hasOwnProperty('amountTimePeriod')){
    colns.VerifyTx__Amount_Time_Period__c =
dedArray[i].colInsurance[j].amountTimePeriod;
}

if(dedArray[i].colInsurance[j].hasOwnProperty('amount')){
    colns.VerifyTx__Amount__c =
parseFloat(dedArray[i].colInsurance[j].amount)*100;
}

if(dedArray[i].colInsurance[j].hasOwnProperty('benefitBeginDate')){
    colns.VerifyTx__Benefit_Begin_Date__c =
dedArray[i].colInsurance[j].benefitBeginDate;
}

if(dedArray[i].colInsurance[j].hasOwnProperty('benefitEndDate')){
    colns.VerifyTx__Benefit_End_Date__c =
dedArray[i].colInsurance[j].benefitEndDate;
}

if(dedArray[i].colInsurance[j].hasOwnProperty('coverageEndDate')){
    colns.VerifyTx__Coverage_End_Date__c =
dedArray[i].colInsurance[j].coverageEndDate;
}

if(dedArray[i].colInsurance[j].hasOwnProperty('coverageStartDate')){

```



```

                colns.VerifyTx__Coverage_Start_Date__c =
dedArray[i].colInsurance[j].coverageStartDate;
            }

if(dedArray[i].colInsurance[j].hasOwnProperty('level')){
                colns.VerifyTx__Level__c =
dedArray[i].colInsurance[j].level;
            }

if(dedArray[i].colInsurance[j].hasOwnProperty('notApplicableNetwork')){
                colns.VerifyTx__Not_Applicable_Network__c =
dedArray[i].colInsurance[j].notApplicableNetwork;
            }

if(dedArray[i].colInsurance[j].hasOwnProperty('placeOfService')){
                colns.VerifyTx__Place_of_Service__c =
dedArray[i].colInsurance[j].placeOfService;
            }

if(dedArray[i].colInsurance[j].hasOwnProperty('PlanEndDate')){
                colns.VerifyTx__Plan_End_Date__c =
dedArray[i].colInsurance[j].PlanEndDate;
            }

if(dedArray[i].colInsurance[j].hasOwnProperty('planStartDate')){
                colns.VerifyTx__Plan_Start_Date__c =
dedArray[i].colInsurance[j].planStartDate;
            }
        }
        colnsSF.push(colns);
    }
}

                if(dedArray[i].hasOwnProperty('coPayment') &&
dedArray[i].coPayment.length > 0){
                    for(let j=0; j<dedArray[i].coPayment.length; j++){
                        var auth;

if(dedArray[i].coPayment[j].hasOwnProperty('authorizationRequired')){
if(dedArray[i].coPayment[j].authorizationRequired==null){
                    auth = 'NA';
                }
                else
if(dedArray[i].coPayment[j].authorizationRequired){
                    auth = 'Yes';
                }
            }
        }
    }

```

```

    }
    else if(!
dedArray[i].coPayment[j].authorizationRequired){
        auth = 'No';
    }
}
else{
    auth = 'NA';
}
var payerNote = "";

if(dedArray[i].coPayment[j].hasOwnProperty('payerNotes')){
    var payerNotes =
dedArray[i].coPayment[j].payerNotes;

    for(let k=0; k<payerNotes.length; k++){
        if(payerNote == null || payerNote == ""){
            payerNote = payerNote+" "+payerNotes[k];
        }
        else{
            payerNote = payerNote+', '+payerNotes[k];
        }
    }
}

let coPay = {
    "VerifyTx__Auth_Required__c": auth,
    "VerifyTx__Service_Type__c" : dedArray[i].name,
    "VerifyTx__Payer_Notes__c": payerNote,
    "VerifyTx__Status__c": dedArray[i].status,
    "VerifyTx__Network__c": ntwrkType,
    "VerifyTx__Result__c" : resultReclId.resultReclId
};

if(dedArray[i].coPayment[j].hasOwnProperty('amountTimePeriod')){
    coPay.VerifyTx__Amount_Time_Period__c =
dedArray[i].coPayment[j].amountTimePeriod;
}

if(dedArray[i].coPayment[j].hasOwnProperty('amount')){
    coPay.VerifyTx__Amount__c =
parseFloat(dedArray[i].coPayment[j].amount);
}

```

```

if(dedArray[i].coPayment[j].hasOwnProperty('benefitBeginDate')){
    coPay.VerifyTx__Benefit_Begin_Date__c =
dedArray[i].coPayment[j].benefitBeginDate;
}

if(dedArray[i].coPayment[j].hasOwnProperty('benefitEndDate')){
    coPay.VerifyTx__Benefit_End_Date__c =
dedArray[i].coPayment[j].benefitEndDate;
}

if(dedArray[i].coPayment[j].hasOwnProperty('amountTimePeriod')){
    coPay.VerifyTx__Coverage_End_Date__c =
dedArray[i].coPayment[j].coverageEndDate;
}

if(dedArray[i].coPayment[j].hasOwnProperty('coverageStartDate')){
    coPay.VerifyTx__Coverage_Start_Date__c =
dedArray[i].coPayment[j].coverageStartDate;
}

if(dedArray[i].coPayment[j].hasOwnProperty('level')){
    coPay.VerifyTx__Level__c =
dedArray[i].coPayment[j].level;
}

if(dedArray[i].coPayment[j].hasOwnProperty('placeOfService')){
    coPay.VerifyTx__Place_of_Service__c =
dedArray[i].coPayment[j].placeOfService;
}

if(dedArray[i].coPayment[j].hasOwnProperty('planEndDate')){
    coPay.VerifyTx__Plan_End_Date__c =
dedArray[i].coPayment[j].planEndDate;
}

if(dedArray[i].coPayment[j].hasOwnProperty('planStartDate')){
    coPay.VerifyTx__Plan_Start_Date__c =
dedArray[i].coPayment[j].planStartDate;
}
    coPays.push(coPay);
}
}

if(dedArray[i].hasOwnProperty('outOfPocket') &&

```

```

dedArray[i].outOfPocket.length > 0){
    for(let j=0; j<dedArray[i].outOfPocket.length; j++){
        var auth;

if(dedArray[i].outOfPocket[j].hasOwnProperty('authorizationRequired')){
if(dedArray[i].outOfPocket[j].authorizationRequired==null){
            auth = 'NA';
        }
        } else
if(dedArray[i].outOfPocket[j].authorizationRequired){
            auth = 'Yes';
        }
        } else if(!
dedArray[i].outOfPocket[j].authorizationRequired){
            auth = 'No';
        }
        }
        } else{
            auth = 'NA';
        }
        }
        var payerNote = "";

if(dedArray[i].outOfPocket[j].hasOwnProperty('payerNotes')){
            var payerNotes =
dedArray[i].outOfPocket[j].payerNotes;

            for(let k=0; k<payerNotes.length; k++){
                if(payerNote == null || payerNote == ""){
                    payerNote = payerNote+" "+payerNotes[k];
                }
                } else{
                    payerNote = payerNote+', '+payerNotes[k];
                }
            }
        }

        let outOfPocket = {
            "VerifyTx__Auth_Required__c": auth,
            "VerifyTx__Network__c": ntwrkType,
            "VerifyTx__Service_Type__c": dedArray[i].name,
            "VerifyTx__Status__c": dedArray[i].status,
            "VerifyTx__Payer_Notes__c": payerNote,
            "VerifyTx__Result__c" : resultReclId.resultReclId

```

```

};

if(dedArray[i].outOfPocket[j].hasOwnProperty('amountTimePeriod')){
    outOfPocket.VerifyTx__Amount_Time_Period__c =
    dedArray[i].outOfPocket[j].amountTimePeriod;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('benefitBeginDate')){
    outOfPocket.VerifyTx__Benefit_Begin_Date__c =
    dedArray[i].outOfPocket[j].benefitBeginDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('benefitEndDate')){
    outOfPocket.VerifyTx__Benefit_End_Date__c =
    dedArray[i].outOfPocket[j].benefitEndDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('coverageEndDate')){
    outOfPocket.VerifyTx__Coverage_End_Date__c =
    dedArray[i].outOfPocket[j].coverageEndDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('coverageStartDate')){
    outOfPocket.VerifyTx__Coverage_Start_Date__c =
    dedArray[i].outOfPocket[j].coverageStartDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('level')){
    outOfPocket.VerifyTx__Level__c =
    dedArray[i].outOfPocket[j].level;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('planEndDate')){
    outOfPocket.VerifyTx__Plan_End_Date__c =
    dedArray[i].outOfPocket[j].planEndDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('planStartDate')){
    outOfPocket.VerifyTx__Plan_Start_Date__c =
    dedArray[i].outOfPocket[j].planStartDate;
}

if(dedArray[i].outOfPocket[j].hasOwnProperty('remaining')){
    outOfPocket.VerifyTx__Remaining__c =

```

```

parseFloat(dedArray[i].outOfPocket[j].remaining);
    }

    if(dedArray[i].outOfPocket[j].hasOwnProperty('remainingTimePeriod')){
        outOfPocket.VerifyTx__Remaining_Time_Period__c =
        dedArray[i].outOfPocket[j].remainingTimePeriod;
    }

    if(dedArray[i].outOfPocket[j].hasOwnProperty('total')){
        outOfPocket.VerifyTx__Total__c =
        parseFloat(dedArray[i].outOfPocket[j].total);
    }

    if(dedArray[i].outOfPocket[j].hasOwnProperty('totalTimePeriod')){
        outOfPocket.VerifyTx__Total_Time_Period__c =
        dedArray[i].outOfPocket[j].totalTimePeriod;
    }

    if(dedArray[i].outOfPocket[j].hasOwnProperty('amount')){
        outOfPocket.VerifyTx__Amount_Met__c =
        parseFloat(dedArray[i].outOfPocket[j].amount);
    }

    if(dedArray[i].outOfPocket[j].hasOwnProperty('amountTimePeriod')){
        outOfPocket.VerifyTx__Not_Applicable_Network__c =
        dedArray[i].outOfPocket[j].notApplicableNetwork;
    }

        outOfPockets.push(outOfPocket);
    }
}

        if(dedArray[i].hasOwnProperty('limitations') &&
dedArray[i].limitations.length > 0){
            for(let j=0; j<dedArray[i].limitations.length; j++){
                var auth;

                if(dedArray[i].limitations[j].hasOwnProperty('authorizationRequired')){
                    if(dedArray[i].limitations[j].authorizationRequired==null){
                        auth = 'NA';
                    }
                }
            }
        }
    }
}

```

```

else
if(dedArray[i].limitations[j].authorizationRequired){
    auth = 'Yes';
}
else if(!
dedArray[i].limitations[j].authorizationRequired){
    auth = 'No';
}
else{
    auth = 'NA';
}
var payerNote = "";
if(dedArray[i].limitations[j].hasOwnProperty('payerNotes')){
    var payerNotes =
dedArray[i].limitations[j].payerNotes;
    for(let k=0; k<payerNotes.length; k++){
        if(payerNote == null || payerNote == ""){
            payerNote = payerNote+" "+payerNotes[k];
        }
        else{
            payerNote = payerNote+', '+payerNotes[k];
        }
    }
}
let limitation = {
    "VerifyTx__Service_Type__c": dedArray[i].name,
    "VerifyTx__Network__c": ntwrkType,
    "VerifyTx__Payer_Notes__c": payerNote,
    "VerifyTx__Result__c" : resultReclد.resultReclد
};
if(dedArray[i].limitations[j].hasOwnProperty('level')){
    limitation.VerifyTx__Level__c =
dedArray[i].limitations[j].level;
}
if(dedArray[i].limitations[j].hasOwnProperty('quantity')){
    limitation.VerifyTx__Quantity__c =
dedArray[i].limitations[j].quantity;
}

```

```

if(dedArray[i].limitations[j].hasOwnProperty('quantityQualifier')){
    limitation.VerifyTx__Quantity__Qualifier__c =
dedArray[i].limitations[j].quantityQualifier;
}

if(dedArray[i].limitations[j].hasOwnProperty('amount')){
    limitation.VerifyTx__Amount__c
=parseFloat(dedArray[i].limitations[j].amount);
}

if(dedArray[i].limitations[j].hasOwnProperty('remaining')){
    limitation.VerifyTx__Remaining__c =
parseFloat(dedArray[i].limitations[j].remaining);
}

if(dedArray[i].limitations[j].hasOwnProperty('remainingTimePeriod')){
    limitation.VerifyTx__Remaining__Time__Period__c
= dedArray[i].limitations[j].remainingTimePeriod;
}

if(dedArray[i].limitations[j].hasOwnProperty('amountTimePeriod')){
    limitation.VerifyTx__Amount__Time__Period__c =
dedArray[i].limitations[j].amountTimePeriod;
}
limitations.push(limitation);
}
}

if(dedArray[i].hasOwnProperty('contacts') &&
dedArray[i].contacts.length > 0){

    for(let j=0; j<dedArray[i].contacts.length; j++){
        var phone = "";
        var website = "";
        if(dedArray[i].contacts[j].info != null){

            for(let l=0; l<dedArray[i].contacts[j].info.length ; l++){

if(dedArray[i].contacts[j].info[l].hasOwnProperty('phone')){
    phone = dedArray[i].contacts[j].info[l].phone;
}

if(dedArray[i].contacts[j].info[l].hasOwnProperty('website')){
    website = dedArray[i].contacts[j].info[l].website;
}
}
}
}
}

```



```

    }
  }
}

let InsContact = {
  "VerifyTx__Network__c": ntwrkType,
  "VerifyTx__Service_Type__c": dedArray[i].name,
  "VerifyTx__phone__c": phone,
  "VerifyTx__website__c": website,
  "VerifyTx__Result__c" : resultReclId.resultReclId
};
if(dedArray[i].contacts[j].hasOwnProperty('name')){
  InsContact.Name = dedArray[i].contacts[j].name;
}
if(dedArray[i].contacts[j].hasOwnProperty('type')){
  InsContact.VerifyTx__Type__c =
dedArray[i].contacts[j].type;
}
if(dedArray[i].contacts[j].hasOwnProperty('category')){
  InsContact.VerifyTx__Category__c =
dedArray[i].contacts[j].category;
}
InsContacts.push(InsContact);
}
}

}

var cache = fullResponseJson.message.cache;
if(cache.hasOwnProperty('plans') && cache.plans.length > 0){
let plansJson = cache.plans;
for(let i=0; i<plansJson.length ; i++){
  if(plansJson[i].hasOwnProperty('plan_details')){
    var planDetails = plansJson[i].plan_details;
    var label = "";
    var payerNote = "";
    if(planDetails.length > 0){
      if(planDetails[0].hasOwnProperty('label')){
        label = planDetails[0].label;
      }
    }
  }
}
}

```

```

    }
    if(planDetails[0].hasOwnProperty('notes')){
    var payerNotes = planDetails[0].notes;
    if(payerNotes != null){
    for(let k=0; k<payerNotes.length; k++){
    if(payerNote == null || payerNote == ""){
    payerNote = payerNote+" "+payerNotes[k];
    }
    else{
    payerNote = payerNote+', '+payerNotes[k];
    }
    }}
    }
}
let plan = {
  "VerifyTx__Payer_Notes__c": payerNote,
  "VerifyTx__Insurance_Type__c": label,
  "VerifyTx__Result__c": resultRecId.resultRecId
};

if(plansJson[i].hasOwnProperty('authorizationRequiredUnknown')){
    plan.VerifyTx__Auth_Req_Unknown__c =
plansJson[i].authorizationRequiredUnknown;
}

if(plansJson[i].hasOwnProperty('authorizationRequiredUnknown')){
    plan.VerifyTx__Auth_Required__c =
plansJson[i].authRequired;
}
    if(plansJson[i].hasOwnProperty('coverage_dates')){
    if(plansJson[i].coverage_dates.hasOwnProperty('start')){
    plan.VerifyTx__Coverage_Start_Date__c =
plansJson[i].coverage_dates.start;
    }
    if(plansJson[i].coverage_dates.hasOwnProperty('end')){
    plan.VerifyTx__Coverage_End_Date__c =
plansJson[i].coverage_dates.end;
    }
    }
    if(plansJson[i].hasOwnProperty('benefitBeginDate')){
    plan.VerifyTx__Benefit_Begin_Date__c =
plansJson[i].benefitBeginDate;
}
}

```

```

        if(plansJson[i].hasOwnProperty('benefitEndDate')){
            plan.VerifyTx__Benefit_End_Date__c =
plansJson[i].benefitEndDate;
        }
        if(plansJson[i].hasOwnProperty('description')){
            plan.VerifyTx__Plan_Product__c =
plansJson[i].description;
        }
        if(plansJson[i].hasOwnProperty('plan_number')){
            plan.VerifyTx__Plan_Number__c =
plansJson[i].plan_number;
        }
        if(plansJson[i].hasOwnProperty('name')){
            plan.VerifyTx__Service_Type__c = plansJson[i].name;
        }
        if(plansJson[i].hasOwnProperty('status')){
            plan.VerifyTx__Status__c = plansJson[i].status;
        }
    }

    PlanProduct.push(plan);
}
}

```

```

        var ColnsField =
document.getElementById("page:form:deductible");
ColnsField.value = JSON.stringify(dedSF);

var ColnsField = document.getElementById("page:form:colns");
ColnsField.value = JSON.stringify(colnsSF);

var CoPaysField = document.getElementById("page:form:coPays");
CoPaysField.value = JSON.stringify(coPays);

        var OutOfPocketField =
document.getElementById("page:form:outOfPockets");
OutOfPocketField.value = JSON.stringify(outOfPockets);

        var LimitationField =
document.getElementById("page:form:limitations");
LimitationField.value = JSON.stringify(limitations);

```

```
var planProductsField =
document.getElementById("page:form:planProducts");
planProductsField.value = JSON.stringify(PlanProduct);

var insContactsField =
document.getElementById("page:form:insContacts");
insContactsField.value = JSON.stringify(InsContacts);
</script>
</apex:outputPanel>

</apex:page>
```

Code for Trigger Helper Class

```
/* Now no need of Generating refresh token and access token in this class*/
public class ChangeStatusHelperBatch implements
Database.Batchable<Object>,
Database.Stateful,Database.AllowsCallouts {

    List<VerifyTx__Cred__c> credential;
    List<Object> finallist = new List<Object>();
    list<Lead> leadData = new list<Lead>();
    list<Lead> oppList = new list<Lead>();
    Map<ID, VerifyTx__VTX_Facility__c> facilityRecordListMap;

    public ChangeStatusHelperBatch(List<Id> leadRecordIds, Set<Id>
facilityIds)
    {
        finallist = [Select
Id,VTX_Facility__c,Client_Status__c,Vob_Id__c,Patient_First_Name__c
from Lead where Id in:leadRecordIds];
        facilityRecordListMap = new Map<ID,
VerifyTx__VTX_Facility__c>([select Id,VerifyTx__Facility_Id__c from
VerifyTx__VTX_Facility__c]);
    }
    public list<object> start(Database.BatchableContext BC)
    {
        return finallist;
    }
    public void execute(Database.BatchableContext BC, List<object>
scope)
    {
        // regenerating acces token if it has been expired
        String url="";
        String refreshToken;
        String access_token;
        credential = [Select
id,VerifyTx__Access_Token_Date__c,VerifyTx__Refresh_Token__c,Verif
yTx__Access_Token__c,VerifyTx__SF_User_Email__c,VerifyTx__User_R
elated__c from VerifyTx__Cred__c Limit 1];
        String eAT="";
        String eRT="";
        access_token =
```

```

VerifyTx.DecryptAccessToken.DecryptMethod((String)credential[0].Verify
Tx__Access_Token__c);
    system.debug('access_token'+access_token);
refreshToken =
VerifyTx.DecryptAccessToken.DecryptMethod((String)credential[0].Verify
Tx__Refresh_Token__c);
    // API callouts for "VerifyTx__Practice_Group__c" records whose
client_status has been changed
    leadData = (list<Lead>)scope;
    for(integer i=0;i<leadData.size();i++)
    {

        Http h = new Http();
        HttpRequest req = new HttpRequest();
        string url1 = 'https://v2.api.gateway.verifytx.com/vobs/update';
        string url2 = 'https://api.verifytx.com/vobs/update';
        req.setEndpoint(url2);
        req.setMethod('PUT');
        req.setHeader('Authorization','Bearer '+access_token);
        req.setHeader('Content-Type','application/json');
        req.setTimeout(60000);

        String Facility = "";
        VerifyTx__VTX_Facility__c facilityId =
facilityRecordListMap.get(leadData[i].VTX_Facility__c);
        Facility = facilityId.VerifyTx__Facility_Id__c;

        string clientStatus ;
        if(leadData[i].Client_Status__c != null){
            clientStatus = leadData[i].Client_Status__c.toLowerCase() ;
        }

        if(clientStatus == null){
            clientStatus = 'Prospect';
            clientStatus = clientStatus.toLowerCase();
        }

        String ReqBody =
{"client_type":""+clientStatus+"","_id":""+leadData[i].Vob_Id__c+"","facility":
""+Facility+"","first_name":""+leadData[i].Patient_First_Name__c+""};
        req.setBody(ReqBody);

        if(Test.isRunningTest())
        {
            System.debug('Nothing to do');
        }
    }

```

```
    }
    else
    {
        HttpResponse res = h.send(req);

        if(res.getStatus()=='OK' && res.getStatusCode()==200)
        {
            leadData[i].Client_Status__c = clientStatus;
        }
    }
}

}

public void finish(Database.BatchableContext BC)
{
    update leadData;
}

}
```

Code for Creating Trigger Client Status Update

```
trigger ChangeStatus on Lead (after Update) {  
  
    list<Id> leadRecordData = new list<Id>();  
    Set<Id> facilityIds = new Set<Id>();  
    Map<Id,Id> mapOfFacilityIds = new Map<Id,Id>();  
  
    for(integer i=0 ;i<trigger.new.size();i++)  
    {  
        if(Trigger.New[i] != Trigger.old[i])  
        {  
            leadRecordData.add(Trigger.New[i].Id);  
            if(!facilityIds.contains(Trigger.New[i].VTX_Facility__c))  
                facilityIds.add(Trigger.New[i].VTX_Facility__c);  
        }  
    }  
    if(leadRecordData.size()>0)  
  
        Database.executeBatch(new  
ChangeStatusHelperBatch(leadRecordData,facilityIds),1);  
}
```