CRV How-To Guide: Custom URL Component For Deep Linking

Audience: CRV Team member seeking to understand how to leverage reusable custom URL to deep link to servicing workflows

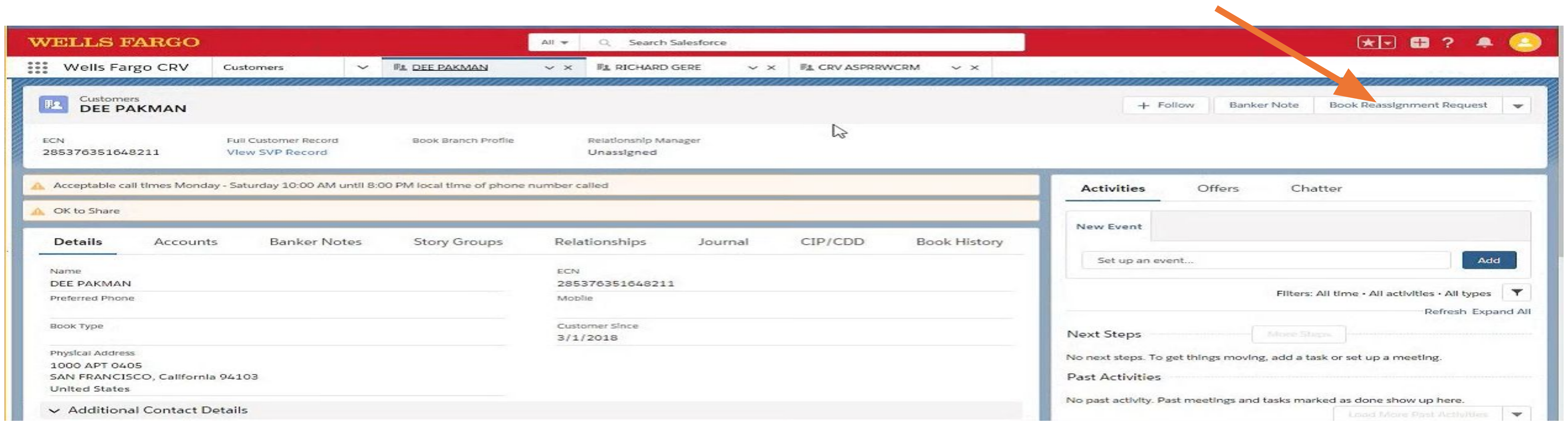
Problem Statement: CRV needs consistent and dynamic URL for deep linking to servicing workflows which would be launched from quick actions in CRV.

Solution Summary: A reusable custom URL component that can be used across board in CRV for deep liking to service workflows

Contents:

Where is Servicing Workflow Invoked?

Initiation of Workflow



WELLS FARGO | All | Search Salesforce

Wells Fargo CRV | Customers | DEE PAKMAN | RICHARD GERE | CRV ASPRRWCRM

Customers
DEE PAKMAN

+ Follow | Banker Note | Book Reassignment Request

ECN	Full Customer Record	Book Branch Profile	Relationship Manager
285376351648211	View SVP Record		Unassigned

Acceptable call times Monday - Saturday 10:00 AM until 8:00 PM local time of phone number called

OK to Share

Details | Accounts | Banker Notes | Story Groups | Relationships | Journal | CIP/CDD | Book History

Name: DEE PAKMAN	ECN: 285376351648211
Preferred Phone	Mobile
Book Type	Customer Since: 3/1/2018
Physical Address: 1000 APT 0405, SAN FRANCISCO, California 94103, United States	

Additional Contact Details

Activities | Offers | Chatter

New Event

Set up an event... | Add

Filters: All time • All activities • All types

Refresh Expand All

Next Steps | More Steps

No next steps. To get things moving, add a task or set up a meeting.

Past Activities

No past activity. Past meetings and tasks marked as done show up here.

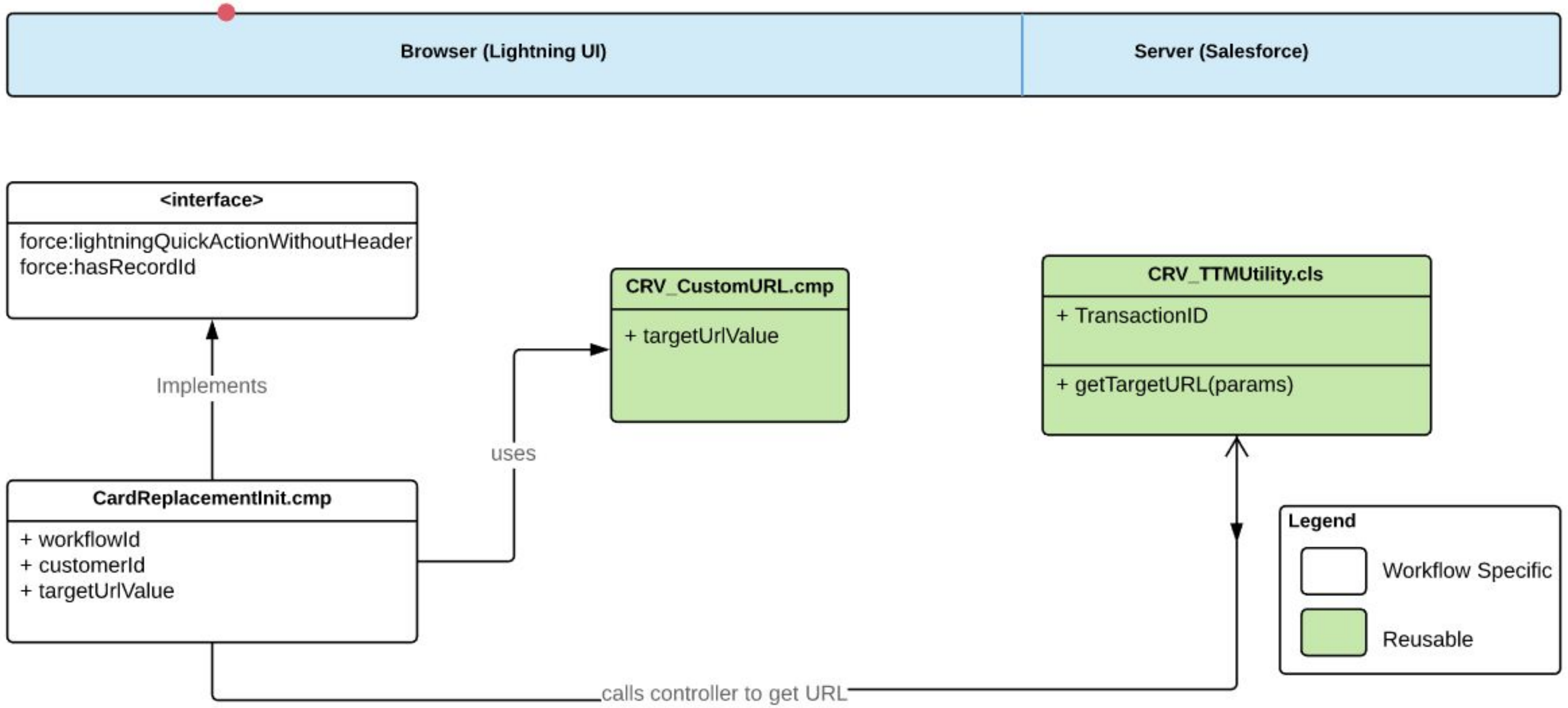
Load More Past Activities

Each Workflow has a QuickAction component so it can be placed in the Customer page actions (see above).

Naming Standards: <Name>WorkflowNNNInit.cmp

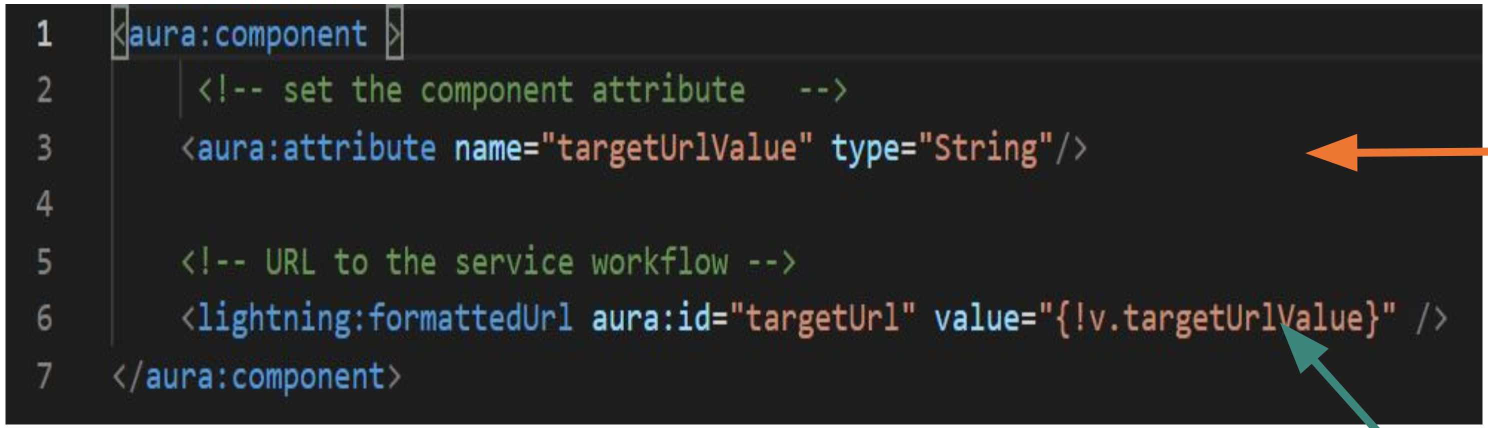
Overview of Components

CRV_CustomURL.cmp can be used by deep linking servicing workflows launched from CRV



The CRV_CustomURL placed inside a parent component will get the URL value passed to it for the deep linking to servicing workflow which then automatically opens up

CRV_CustomURL.cmp



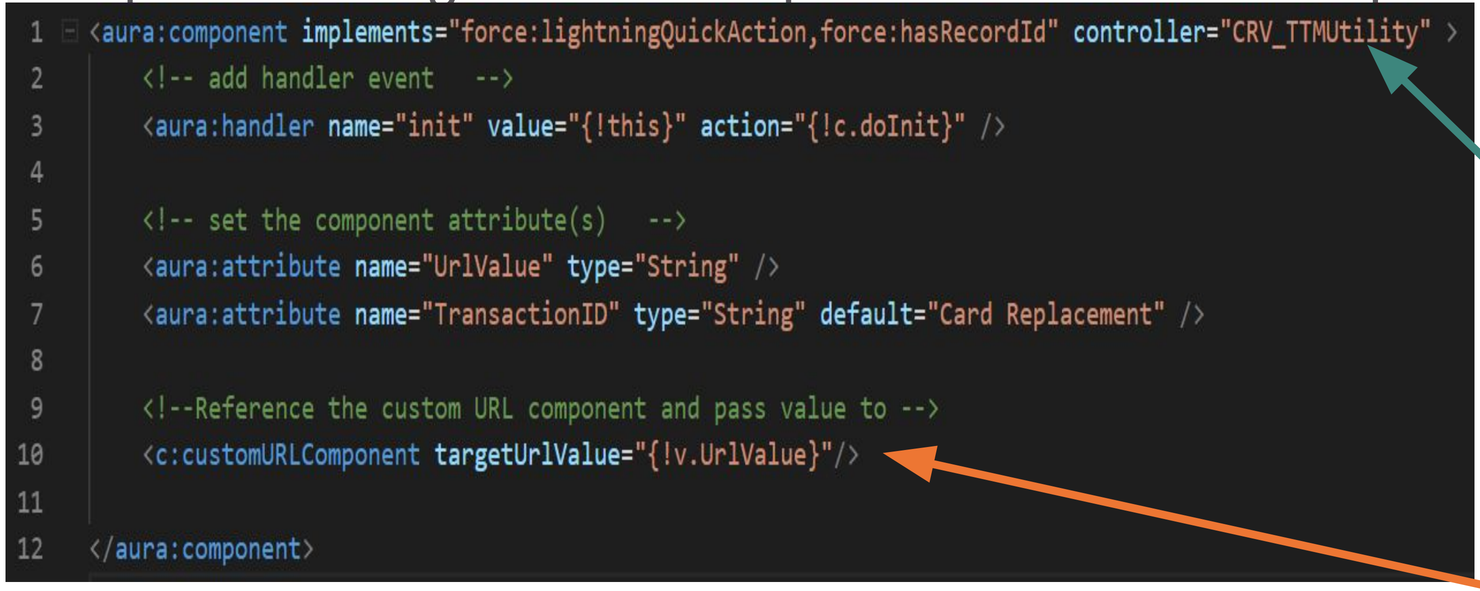
```
<aura:component >
    <!-- set the component attribute   -->
   <aura:attribute name="targetUrlValue" type="String"/>

   <!-- URL to the service workflow -->
   <lightning:formattedUrl aura:id="targetUrl" value="{!v.targetUrlValue}" />
</aura:component>
```

1. The component attribute that stores the value of the target URL. The value would be passed from the parent component (e.g. Card Replacement servicing workflow)

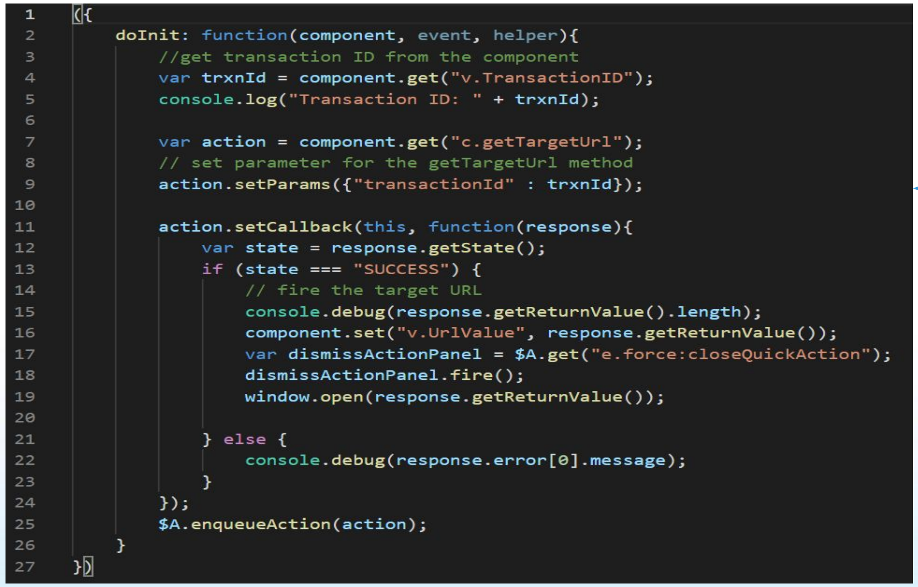
2. The value referenced in the formatted URL

How Servicing Workflows Leverage CRV_CustomURL.cmp?

Sample Servicing workflow implementation: Card Replacement



```
<aura:component implements="force:lightningQuickAction,force:hasRecordId" controller="CRV_TTMUtility" >
    <!-- add handler event   -->
    <aura:handler name="init" value="{!this}" action="{!c.doInit}" />

    <!-- set the component attribute(s)   -->
    <aura:attribute name="UrlValue" type="String" />
    <aura:attribute name="TransactionID" type="String" default="Card Replacement" />

    <!--Reference the custom URL component and pass value to -->
    <c:customURLComponent targetUrlValue="{!v.UrlValue}"/>

</aura:component>
```



```
({
    doInit: function(component, event, helper){
        //get transaction ID from the component
        var trxnId = component.get("v.TransactionID");
        console.log("Transaction ID: " + trxnId);

        var action = component.get("c.getTargetUrl");
        // set parameter for the getTargetUrl method
        action.setParams({"transactionId" : trxnId});

        action.setCallback(this, function(response){
            var state = response.getState();
            if (state === "SUCCESS") {
                // fire the target URL
                console.debug(response.getReturnValue().length);
                component.set("v.UrlValue", response.getReturnValue());
                var dismissActionPanel = $A.get("e.force:closeQuickAction");
                dismissActionPanel.fire();
                window.open(response.getReturnValue());

            } else {
                console.debug(response.error[0].message);
            }
        });
        $A.enqueueAction(action);
    }
})
```

CardReplacementInit.cmp

1. Utility class for TTM with aura enabled getTargetUrl(transactionId) method. Return URL would be concatenated with any query string parameters.

2. Include CRV_CustomURL.cmp in CardReplacementInit.cmp

3. Provide a callback function to handle the response from getTargetUrl(transactionId), passed the response to CRV_CustomURL.cmp and launch the servicing workflow.

THANK YOU