JS Bridge to Native & Web

Plugins newly add during discover tab dev

Network request DONE

please note that the func "fetch offers data" should be replace by "http request"

for the "http request" need 3 paramaters:

paramaters[0]:required, JSONString

paramaters[1]:optional, JSONString, default is ""

paramaters[2]:optional, String, default is "Post"

data fetch

```
// http request for getCustomerOffers, icmId will be dynamic, can get in sessionStorage 'icmpid'
window.cordova.exec(
    function (response) { //Successfully obtaining data will trigger this callback
        alert("Success: " + JSON.stringfy(response));// the success value will be a json string
    },
    function (error) {  //Failed to retrieve data will trigger this callback
        alert("Error: " + error);
    },
    "gqlplugin",//plugin name
    "request", //function name
    [JSON.stringify({
        "operationName": "getCustomerOffers",
        "variables": {
          "placeholder": "discover_all",
          "icmId": "7020000002298039",
          "templateMerge": false
        },
        "query": "query getCustomerOffers($placeholder: String!, $icmId: String, $templateMerge: Boolean) {
offers(placeholder: $placeholder, icmId: $icmId, templateMerge: $templateMerge) { __typename placeholder offers
{ __typename id title message images { __typename ...imageSet } primarycta { __typename ...ctaLink }
secondarycta { __typename ...ctaLink } feedback { __typename name group interactionId issue rank placement }
index fields { __typename name value } } } } fragment imageSet on ImageSet { __typename small medium large }
fragment ctaLink on CtaLink { __typename label link deeplink }"
}),JSON.stringify({"X-APOLLO-OPERATION-NAME": "getCustomerOffers"}),"POST"] //body(JSONString),header
(JSONString),requestMethod
);


// http request for updateMultipleImpression, 'input' will be dynamic, can get in offers item 'feedback',the
header in HK should add 'icm-id'
window.cordova.exec(
    function (response) { //Successfully obtaining data will trigger this callback
        alert("Success: " + JSON.stringfy(response));// the success value will be a json string
    },
    function (error) {  //Failed to retrieve data will trigger this callback
        alert("Error: " + error);
    },
    "gqlplugin",//plugin name
    "request", //function name
    [JSON.stringify({
        "operationName": "updateMultipleImpression",
        "variables": {
          "input": [
            {
              "name": "EasyPay",
              "group": "CreditCards",
              "interactionId": "4",
              "issue": "Sales",
              "placement": "timeline_transaction",
              "rank": 2,
              "outcome": "Impression"
            }
          ]
        },
        "query": "mutation updateMultipleImpression($input: [CustomerOfferFeedbackInput!]!, $cuId: String) {
updateMultipleImpression(feedbacks: $input, cuId: $cuId) { __typename status message } }"
      }),JSON.stringify({"X-APOLLO-OPERATION-NAME": "updateMultipleImpression"}),"POST"] //body(JSONString),
header(JSONString),requestMethod
);

// http request for updateImpression, 'input' will be dynamic, can get in offers item 'feedback',the header in
HK should add 'icm-id'
window.cordova.exec(
    function (response) { //Successfully obtaining data will trigger this callback
        alert("Success: " + JSON.stringfy(response));// the success value will be a json string
    },
    function (error) {  //Failed to retrieve data will trigger this callback
        alert("Error: " + error);
    },
    "gqlplugin",//plugin name
    "request", //function name
    [JSON.stringify({
          "operationName": "updateImpression",
          "variables": {
            "input": {
              "name": "EasyPay",
              "group": "CreditCards",
              "interactionId": "4",
              "issue": "Sales",
              "placement": "timeline_transaction",
              "rank": 2,
              "outcome": "Clicked" //Clicked/Dismissed               }
          },
          "query": "mutation updateImpression($input: CustomerOfferFeedbackInput!) { updateImpression(feedback:
$input) { __typename status message } }"
      }),JSON.stringify({"X-APOLLO-OPERATION-NAME": "updateImpression"}),"POST"] //body(JSONString),header
(JSONString),requestMethod
);
```

sample json in callback success value: discover_all_data.txt

Web page hide / visible

This is not a signal Plugin, **'document.visibilityState'** is a common web API.

webView can use the common Page Visibility API to check page visibility state.

JS

```
document.addEventListener('visibilitychange', function() {
     if (document.visibilityState === 'visible') {
     // action for page visible
     } else if (document.visibilityState === 'hidden') {
     // action for page hide
   }
});
```

Default tab DONE

native can set default tab to web, because the native end may redirect to different tab pages through DeepLink.

use '**document.deeplink**' redirect to default page. the value will be **Accounts / Cards / Loans / Mortgages**. default is 'null' to L1 page, can Ref: https://confluence.global.standardchartered.com/display/DSMHIVE/Discover+tab+Deeplink

Pre load with 'Web page hide / visible' & 'Default tab'

This is not a signal Plugin

Native end init webview and loadurl in homepagebut because discover tab is not visibleso visibility flag '**document.visibilityState**' is 'hiden'. this time, the web does not trigger requests for offers, only display loading. when user change tab to discover, native end will trigger 'visible' event, web end check native passport tab flag '**document.deeplink**' and data state, if no offers data, trigger plugin to fetch offers data and render.

Tab & StatueBar hide/visible DONE

Because the bar needs to be hidden on the details page, So it is necessary to control the display and hiding.

Control the visible or hide of UI for native component:

data fetch

```
window.cordova.exec(
        function (success) {},
    function (error) {},
        "nativeuiplugin",//plugin name
    "hide", //function name  visible/hide
    ["bottomTabBar"] //parameter 'bottomTabBar'/'titleBar'
);
```

HK BCP mode passport TODO

Get app version info DONE

data fetch

```
window.cordova.exec(
        function (success) {
        alert("Success: " + success);
    },
    function (error) {
        alert("Error: " + error);
    },
        "appversion",//plugin name
    "getAppName", //function name
    [] //parameter
);
```

this plugin support functions:

1. getAppNameSC Mobile
2. getPackageNamecom.sc.Breeze
3. getVersionNumberappStore version (21.0.0)
4. getVersionCode: buildNumber (20240829.01)