**Zeamster Payment SDK (Cordova Plugin) User Guide for Developers**

# **Table of Content**

# Introduction ……………………………………………………………………………. 3

# Cordova Payment SDK Plugin Uses ………………….…..…………………… 3

# Introduction

This document is to give a detailed overview on the how to use the Zeamster Cordova Payment SDK Plugin for Mobile Application.

# Cordova Payment SDK Plugin Uses in Mobile Apps:

Step-1: Download the latest Cordova Payment SDK plugin from following location:

Android:

https://bitbucket.org/fortispay/zeamster-android/src/master/PaymentSDKPlugInCordova/

iOS:

https://bitbucket.org/fortispay/zeamster-ios/src/master/PaymentSDKPluginIOS/

Step-2: Open project’s root directory in command prompt or terminal.

Step-3: Add the plugin in project by running following command in the command prompt.

cordova plugin add <path to plugin>

Step-4: Run the following command to install the plugin in the project

plugman install --platform [android|ios] --project /path/to/my/project/www --plugin /path/to/my/plugin

Step-5 Once plugin is installed run the following command to verify if plugin has been included in the project.

cordova plugin list

Step-6: Now run the following command to build and run the project

cordova run [android|ios]

Step-7: Once plugin is included successfully in the project. Here are the following endpoint functions provided by cordova Payment SDK plugin to perform EMV or non-EMV transaction as user needs:

1. performTransaction : function (arg0, success, error)  
     
   This function used to perform non-EMV transaction, it takes 3 parameters which are:
   1. Transaction payload json with all required details
   2. Success callback function
   3. Error callback function
2. performEMVTransaction: function (arg0, success, error)  
     
   This function used to perform EMV transaction, it takes 3 parameters which are:
   1. Transaction payload json with all required details
   2. Success callback function
   3. Error callback function
3. cancelEMVTransaction: function (arg0, success, error)  
     
   This function used to cancel the current EMV transaction, we need to pass only 2 parameters which are:
   1. Success callback function
   2. Error callback function
4. setDeviceType: function (arg0, success, error)  
     
   This function used to set the given device type for EMV transaction, so that if users want to change the default device type then they can use this function. It takes 3 parameters which are:
   1. Device Type
   2. Success callback function
   3. Error callback function
5. scanForDevices: function (arg0, success, error)  
     
   This function used to scan the Bluetooth devices and connect it for EMV transaction, it takes 3 parameter which are:
   1. Transaction payload json which **MUST** have the device type defined in it, otherwise default VP3300BT device will be connected if available.
   2. Success callback function
   3. Error callback function
6. stopScanForDevices: function (arg0, success, error)  
     
   This function used to stop the scanning of Bluetooth devices, , we need to pass only 2 parameters which are:
   1. Success callback function
   2. Error callback function
7. connectDeviceByName: function (arg0, success, error)  
     
   This function used to connect specific device by its name to perform EMV transaction, it takes 3 parameters which are:
   1. Device Name
   2. Success callback function
   3. Error callback function
8. initialDeviceSetup: function (arg0, success, error)  
     
   This function used to perform initial device setup for the IDTech card reader device, it takes 3 parameters which are:
   1. Payload json with all required details to perform initial device setup.
   2. Success callback function
   3. Error callback function
9. disconnectDevice: function (arg0, success, error)  
     
   This function used to disconnect the currently connected IDTech card reader device, we need to pass only 2 parameters which are:
   1. Success callback function
   2. Error callback function

**Note**: User needs to implement below SDK callback function in mobile app where the plugin is used:

1. deviceConnected:

deviceConnected: function(message) {

//Add code to show on screen that device is connected

},

1. deviceDisconnected:

deviceDisconnected: function(message) {

//Add code to show on screen that device is disconnected

},

1. deviceMessage:

deviceMessage: function(message) {

//Add code to show the device message on screen

},

1. outputLogs:  
   outputLogs: function(message) {

//Add code to show transaction output log on the screen

},

**Follow below mentioned steps to perform a EMV or non-EMV transaction:**

1. **For non-EMV transaction:**
   1. Prepare the following JavaScript function for map objects as per need and as following:

createPayload: function() {

//Mandatory section -- Starts

var headerParams = new Object();

headerParams.protocol = "https";

headerParams.hostname = "api.sandbox.zeamster.com";

headerParams.apiEndpoint = "/v2/transactions";

headerParams.developerId = "\*\*\*\*\*\*\*\*\*\*";

headerParams.userId = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userAPIKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userHashKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.username = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.password = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.domain = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.authType = "apikey";

//Mandatory section -- Ends

//Optional as per use

headerParams.transactionId = "";

//Optional section (as per use) -- Starts

var queryParams = new Object();

queryParams.fieldName1 = "Value1";

queryParams.fieldName2 = "Value2";

queryParams.fieldName3 = "Value3";

queryParams.fieldName4 = "Value4";

//Optional section -- Ends

//Optional section (as per use) -- Starts

var bodyParams = new Object();

bodyParams.action = "sale";

bodyParams.account\_number = "5454545454545454";

bodyParams.payment\_method = "cc";

bodyParams.exp\_date = "1223";

bodyParams.transaction\_amount = "63.42";

//Optional section -- Ends

//Mandatory section -- Starts

var params = new Object();

params.header = headerParams;

// params.queryParams = queryParams; //optional, add as per need.

params.body = bodyParams;

return JSON.stringify(params);

//Mandatory section – Ends  
},

* 1. Call the following Plugin method to perform a specific task, provide appropriate payload values for the same.

cordova.plugins.PaymentSDKPlugin.performTransaction(app.createPayload(), function(response){

//Add code to show success

}, function(error){

//Add code to show failure

});

1. **For EMV transaction:**
   1. Prepare the following JavaScript function for map objects as per need and as following:

createPayloadForEMV: function() {

//Mandatory section -- Starts

var headerParams = new Object();

headerParams.protocol = "https";

headerParams.hostname = "api.sandbox.zeamster.com";

headerParams.apiEndpoint = "/v2/transactions";

headerParams.developerId = "\*\*\*\*\*\*\*\*\*\*";

headerParams.userId = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userAPIKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userHashKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.username = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.password = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.domain = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.authType = "apikey";

//Mandatory section -- Ends

//Optional as per use

headerParams.transactionId = "";

headerParams.deviceType = "IDTECH-VP3300BT";

//Optional section (as per use) -- Starts

var queryParams = new Object();

queryParams.fieldName1 = "Value1";

queryParams.fieldName2 = "Value2";

queryParams.fieldName3 = "Value3";

queryParams.fieldName4 = "Value4";

//Optional section -- Ends

//Optional section (as per use) -- Starts

var bodyParams = new Object();

bodyParams.action = "sale"; //mandatory for MSR & EMV

bodyParams.transaction\_amount = "24.32"; //mandatory for MSR & EMV

bodyParams.terminal\_id = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; //mandatory for MSR & EMV

// bodyParams.product\_transaction\_id = "<value>";

//Optional section -- Ends

//Mandatory section -- Starts

var params = new Object();

params.header = headerParams;

// params.queryParams = queryParams; //optional, add as per need.

params.body = bodyParams;

return JSON.stringify(params);

//Mandatory section -- Ends

},

* 1. Call scan and connect the device for EMV transaction:

cordova.plugins.PaymentSDKPlugin.scanForDevices(app.createPayloadForEMV(), function(response){

var parts = response.split("!@#$");

if (parts[0].indexOf("deviceScanResponse") >= 0) {

app.outputLogs("Device Found: " + parts[2] + " -- " + parts[1]);

if (parts[2].indexOf("92231") >= 0) {

//here compare with the device, which needs to be connected.

if (app.isConnecting) {

return;

} else {

app.connectDeviceByName(parts[2]);

// Alternatively next line can also be used if required to connect by name.

// app.connectDeviceById(parts[1]);

}

}

} else if (parts[0].indexOf("deviceMessage") >= 0) {

app.deviceMessage(parts[1]);

} else if (parts[0].indexOf("outputLogs") >= 0) {

app.outputLogs(parts[1]);

} else {

responseContainer.textContent += response + " ---- ";

}

}, function(error){

//Add code to show failure

});

* 1. If user want to set device type then here is the sample code:  
       
     cordova.plugins.PaymentSDKPlugin.setDeviceType("IDTECH-VP3300BT", function(response){

//Add code to show success

}, function(error){

//Add code to show failure

});

* 1. If users want to connect device by its name, so they can use below code:  
       
     cordova.plugins.PaymentSDKPlugin.connectDeviceByName(deviceName, function(response){

//Add code to show success

}, function(error){

//Add code to show failure

});

* 1. If users want to stop device scan, so they can use below sample code:  
       
     cordova.plugins.PaymentSDKPlugin.stopScanForDevices( function(response){

//Add code to show success

}, function(error){

//Add code to show failure

});

* 1. Now, call the EMV transaction function to perform the transaction:  
       
     cordova.plugins.PaymentSDKPlugin.performEMVTransaction(app.createPayloadForEMV(), function(response){

var parts = response.split("!@#$");

if (parts[0].indexOf("deviceMessage") >= 0) {

app.deviceMessage(parts[1]);

} else if (parts[0].indexOf("outputLogs") >= 0) {

app.outputLogs(parts[1]);

} else if (parts[0].indexOf("transactionResponse") >= 0) {

app.outputLogs(parts[1]);

} else {

responseContainer.textContent += response + " ---- ";

}

}, function(error){

//Add code to show failure

});

* 1. If user wants to cancel the current EMV transaction, so here the is the sample code:

cordova.plugins.PaymentSDKPlugin.cancelEMVTransaction( function(response){

var parts = response.split("!@#$");

if (parts[0].indexOf("deviceMessage") >= 0) {

app.deviceMessage(parts[1]);

} else if (parts[0].indexOf("outputLogs") >= 0) {

app.outputLogs(parts[1]);

} else if (parts[0].indexOf("transactionResponse") >= 0) {

app.outputLogs(parts[1]);

} else {

responseContainer.textContent += response + " ---- ";

}

}, function(error){

//Add code to show failure

});

* 1. To disconnect the current IDTech device, so use below code:

cordova.plugins.PaymentSDKPlugin.disconnectDevice( function(response){

var parts = response.split("!@#$");

if (parts[0].indexOf("deviceMessage") >= 0) {

app.deviceMessage(parts[1]);

} else if (parts[0].indexOf("outputLogs") >= 0) {

app.outputLogs(parts[1]);

} else if (parts[0].indexOf("transactionResponse") >= 0) {

app.outputLogs(parts[1]);

} else {

responseContainer.textContent += response + " ---- ";

}

}, function(error){

//Add code to show failure

});

1. **For Initial Device Setup:**
   1. Prepare the following JavaScript function for map objects as per need and as following:

createPayloadForDevice: function () {

//Mandatory section -- Starts

var headerParams = new Object();

headerParams.protocol = "https";

headerParams.hostname = "api.sandbox.zeamster.com";

headerParams.apiEndpoint = "/devices/idtechinitialize";

headerParams.developerId = "\*\*\*\*\*\*\*\*\*\*";

headerParams.userId = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userAPIKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.userHashKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.username = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.password = "\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.domain = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

headerParams.authType = "apikey";

//Mandatory section -- Starts

//Optional as per use

headerParams.transactionId = "";

headerParams.deviceType = "IDTECH-VP3300BT";

//Mandatory section -- Starts

var params = new Object();

params.header = headerParams;

return JSON.stringify(params);

},

* 1. Call function start the initial device setup:

cordova.plugins.PaymentSDKPlugin.initialDeviceSetup(app.createPayloadForDevice(), function(response){

var parts = response.split("!@#$");

if (parts[0].indexOf("deviceDisconnected") >= 0) {

app.deviceDisconnected(parts[1]);

} else if (parts[0].indexOf("deviceMessage") >= 0) {

app.deviceMessage(parts[1]);

} else if (parts[0].indexOf("outputLogs") >= 0) {

app.outputLogs(parts[1]);

} else {

responseContainer.textContent += parts + " ---- ";

}

}, function(error){

//Add code to show failure

});