# 

Table of Contents

**App2App**

**Guide for patching an existing solution**

[1 Introduction 4](#_Toc500949437)

[2 Overview 4](#_Toc500949438)

[3 Components 4](#_Toc500949439)

[3.1 SPPIframe.tpl 4](#_Toc500949440)

[3.2 SPServletFilter 4](#_Toc500949441)

[3.3 In-app Browser WKWebView 5](#_Toc500949442)

[3.3.1 Configuration 5](#_Toc500949443)

[3.3.2 Installation 5](#_Toc500949444)

[3.3.3 How to use 7](#_Toc500949445)

[4 UXP installation 7](#_Toc500949446)

[5 How to run the project 7](#_Toc500949447)

[5.1 Run the project 7](#_Toc500949448)

[5.2 Avoid SSL errors 7](#_Toc500949449)

[5.3 Certificates 8](#_Toc500949450)

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Version** | **Date** | **Details** |
| Horia Popescu | 1.0 | 12.01.2018 | Creation |
| Arun Vamadevan | 1.1 | 26.06.2018 | Plugin configurations are updated |

**Comments:**

|  |
| --- |
|  |
|  |
|  |

# Introduction

The purpose of this document is to provide a description of what steps need to be done in order to patch an existing application with the app2app-related files.

# Overview

This patch is comprised of 3 components. They should be provided along with this document.

1. SPPIframe.tpl (and SPPIframe3.tpl for PADES signature authentication)
2. luxTrustSPServletFilter.jar – in the /lib folder
3. WKWebView plugin in the form of a folder named “app2App-message-relay-plugin”

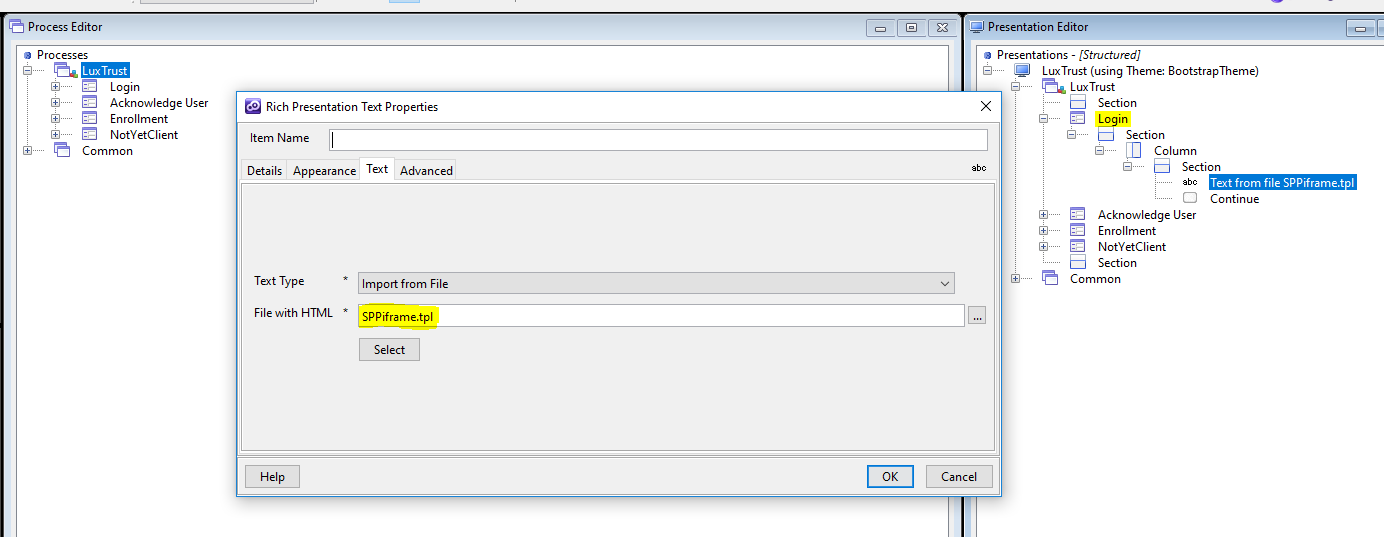
After the patch, any existing functionality should not be changed. This only aims to add functionality that is specific to IOS.

# Patching an existing solution

To patch an existing solution, the components mentioned earlier need to be installed.

## SPPIframe.tpl

This needs to be loaded in a DisplayText. The DisplayText item has to be placed in the Phase that displays the login page. See below an example of how this might look in a project.



This template works by determining whether the page is displayed by an IOS platform or not. If it is not, then startLuxtTrustAuth.html page is loaded. It is recommended that this file contains an iFrame which loads the LuxTrust-provided content but it can contain anything dictated by the scope of the project. Otherwise a WKWebView is opened and this loads LuxTrust-provided content. Note that this WebView will only contain information provided by the LuxTrust server.

## luxTrustSPServletFilter.jar

It is just a Servlet filter that adds the required javascript that closes the opened WKWebView. It only does this addition to the configured <url-pattern> requests. It should target the URL for the response of the authentication request made to Orely. To add it to an existing UXP project, two steps need to be done:

1. The following tag needs to be added to the web.xml file (located in the WEB-INF folder of the Solution):

<filter>

<filter-name>SPServletFilter</filter-name>

<filter-class>com.edgeipk.servlet.SPServletFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>SPServletFilter</filter-name>

<url-pattern>/saml/response/\*</url-pattern>

</filter-mapping>

Note that the <filter> tag needs to be added before any <listener> tags.

1. The luxTrustSPServletFilter.jar file needs to be added to the Solution’s lib folder (located in the WEB-INF folder)

## WKWebView

This plugin needs to be included in the Solution project. Its inclusion will only have an impact when a hybrid app is built because the build process will also include this plugin. You can find it in a folder called “temenos-plugin-inappbrowserwkwebview” which should be included with this document.

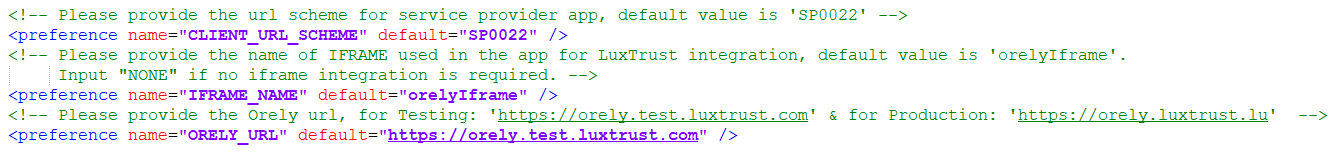
### Configuration

We need certain configuration before we install this plugin with hybrid solution. These

configurations are part of plugin.xml file. Below are the details.

1. **CLIENT\_URL\_SCHEME** – This is issued by LuxTrust for the service provider. It is mandatory and used during the communication between the two apps.
2. **IFRAME\_NAME** – when the LTMOB responds to SPMOB with the OTP, a message containing this OTP will post from an iframe with this name. Please note that we can implement the integration without an iframe too. In this case, the value must be “NONE” so that the app will understand the integration does not use an iframe.
3. **ORELY\_URL** – this is the URL to access the Orely server. For test and production environment, they use different urls.

Below is the screen print from plugin.xml, which shows the details



### Installation

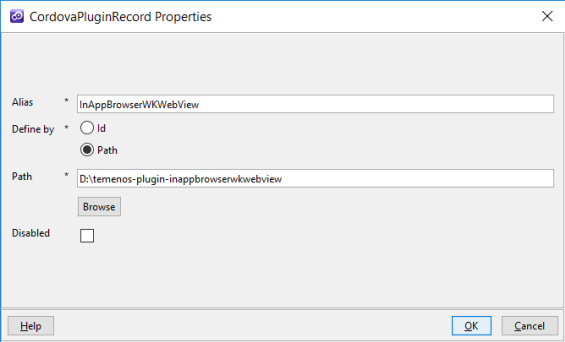
4. Add Cordova plugin via edgeConnect developer IDE. Installation will happen during the build

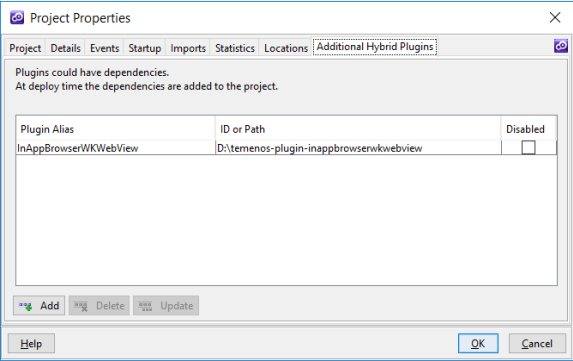
process. Below are the steps.

4.1 Copy temenos-plugin-inappbrowserwkwebview in your local computer (e.g. D drive)

4.2 In edgeConnect developer IDE, go to Project -> Properties… -> Additional Hybrid plugins

4.3 Click Add button. In Plugin Properties Window, give a desired alias and then choose Path radio button. Navigate to the temenos-plugin-inappbrowserwkwebview folder on D drive you copied earlier. Please find the below screen shot for reference.





### How to use

We can open WKWebView from java script by using below method.

*cordova.InAppBrowserWKWebView.open(authUrl, '\_blank', 'location=no')*

*authUrl – Is the url to open in WKWebView*

*location=no – Decides whether to show location/parh in the user interface.*

## IOS app

After an app that includes the WKWebView plugin is built this needs to be distributed to the clients. So they need to be provided an update via the app store.

## Android app

There are three parameters that need to be set in the plugin.xml file in order to be able to build a working .apk file:

1. **CLIENT\_URL\_SCHEME** – This is issued by LuxTrust for the service provider. It is mandatory and used during the communication between the two apps.
2. **IFRAME\_NAME** – when the LTMOB responds to SPMOB with the OTP, a message containing this OTP will post from an iframe with this name. Please note that we can implement the integration without an iframe too. In this case, the value must be “NONE” so that the app will understand the integration does not use an iframe.
3. **ORELY\_URL** – this is the URL to access the Orely server. For test and production environment, they use different urls.

