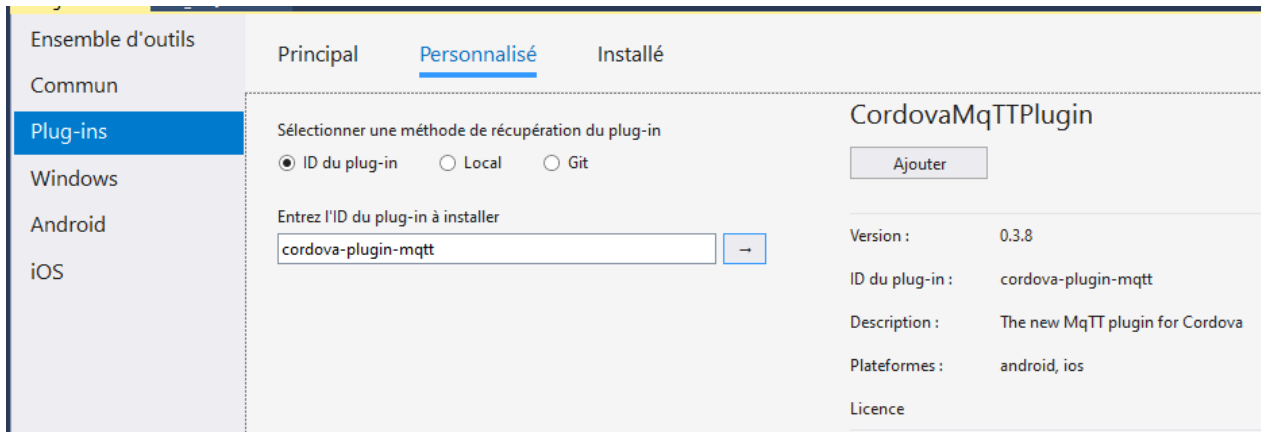


## Minimum pour utiliser MQTT avec Cordova

### 1. VISUAL STUDIO 2017

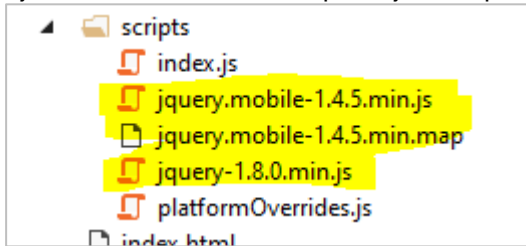
1.1. Créer ou ouvrir un projet apache cordova

1.2. Ouvrir le fichier config .xml, choisir Plug-ins dans la colonne de gauche rechercher et installer le plugin ayant l'ID suivant **cordova-plugin-mqtt**



1.3. Une fois installé on peut l'utiliser (il faut cependant que l'appareil mobile ait accès à internet)

1.4. Ajouter dans le dossier script les javascripts jquery.mobile-1.4.5.min.js et jquery-1.8.0.min.js



et dans le dossier css **jquery.mobile-1.4.5.min.css**

1.5. Page index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <meta name="format-detection" content="telephone=no">
  <meta name="msapplication-tap-highlight" content="no">
  <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">
  <link href="css/jquery.mobile-1.4.5.min.css" rel="stylesheet" />
  <link rel="stylesheet" type="text/css" href="css/index.css">

  <title>TestMQTT</title>
</head>
<body>
  <div role="main" class="ui-content">
    <h1>MQTT Test</h1>
    <form method="post" action="#" id="formulaire">
      <input id="message" type="text" placeholder="saisir un message" value="Test" />
    </form>
    <button id="btn-envoi" data-role="button">Envoi</button>
    <div id="activity"></div>
  </div>
  <footer data-role="footer" data-position="fixed">
    <div id="activity"></div>
  </footer>
  <script src="scripts/jquery-1.8.0.min.js"></script>
  <script src="scripts/jquery.mobile-1.4.5.min.js"></script>
  <script type="text/javascript" src="cordova.js"></script>
  <script type="text/javascript" src="scripts/platformOverrides.js"></script>
  <script type="text/javascript" src="scripts/index.js"></script>
</body>
</html>
```

## 1.6. Le script index.js

```
// Pour obtenir une présentation du modèle Vide, consultez la documentation suivante :
// http://go.microsoft.com/fwlink/?LinkID=397704
// Pour déboguer du code durant le chargement d'une page dans cordova-simulate ou sur les appareils/émulateurs
// Android, lancez votre application, définissez des points d'arrêt,
// puis exécutez "window.location.reload()" dans la console JavaScript.

var URL_BROKER = 'broker.hivemq.com';
var PORT_BROKER = 1883;
var USER_BROKER = '';
var PASS_BROKER = '';
var TOPIC = 'ecran-jeremy_didier';
var connect = false; // variable indiquant si on est connecté au broker

(function () {
    "use strict";

    document.addEventListener( 'deviceready', onDeviceReady.bind( this ), false );

    function onDeviceReady() {
        // Gérer les événements de suspension et de reprise Cordova
        document.addEventListener( 'pause', onPause.bind( this ), false );
        document.addEventListener( 'resume', onResume.bind( this ), false );

        console.log('ready');

        // on se connecte au broker MQTT
        mqttConnect();
        // gestion du clic sur le bouton
        $("#btn-envoi").click(function () {
            var message = $("#message").val();
            var topic = TOPIC;
            publish(topic,message) // publication du message
        })
    };

    // fonction permettant de se connecter au broker
    function mqttConnect() {
        /*
        definition de tous les paramètres de la connexion
        */
        let myConnection = {
            url: "tcp://" + URL_BROKER,
            port: PORT_BROKER,
            isBinaryPayload: false,
            success: function (s) {
                connect = true;
                console.log(JSON.stringify(s));
                $('#activity').html("connecté à : " + URL_BROKER + ":" + PORT_BROKER);
                $('#btn-envoi').removeClass('ui-disabled');
            },
            error: function (e) {
                connect = false;
                $('#btn-envoi').addClass('ui-disabled');
                $('#activity').html("erreur de connexion à : " + URL_BROKER + ":" + PORT_BROKER);
                clearInterval(isRun);
                //alert("err!! something is wrong. check the console")
                console.log(e);
            },
            onConnectionLost: function () {
                connect = false;
                $('#btn-envoi').addClass('ui-disabled');
                clearInterval(isRun);
            }
        }
        if (USER_BROKER != "") myConnection.username = USER_BROKER
        if (PASS_BROKER != "") myConnection.password = PASS_BROKER

        cordova.plugins.CordovaMQTTPlugin.connect(myConnection);
    };

    // fonction permettant de publier une donnée sur le topic
    function publish(topic, message) {

        var json = {}; // fabrication d'un objet json
        json.message = message; // affectation de la donnée message à une propriété de json

        console.log("topic : " + topic)
        console.log("message : " + message)
        console.log("json" + json)
    }
}
```

```

if (!connect) {
    alert("First establish connection then try to publish")
} else {
    cordova.plugins.CordovaMQTTPlugin.publish({
        topic: topic,
        payload: json,
        qos: 0,
        retain: false,
        success: function (s) {
            //console.log(JSON.stringify(s));
        },
        error: function (e) {
            document.getElementById("activity").innerHTML += "--> Error: something is wrong, " + e + "<br>";
            console.log(e);
        }
    });
}

};

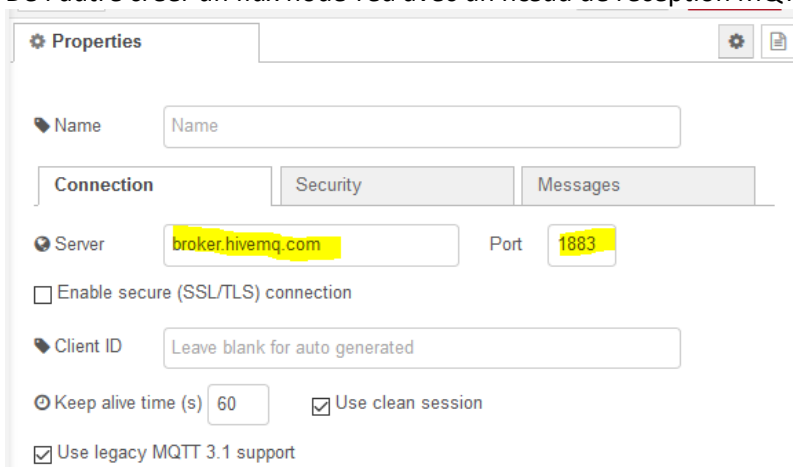
function onPause() {
    // TODO: cette application a été suspendue. Enregistrez l'état de l'application ici.
};

function onResume() {
    // TODO: cette application a été réactivée. Restaurez l'état de l'application ici.
};

} )();

```

De l'autre créer un flux node-red avec un nœud de réception MQTT le broker doit être le suivant :



**Properties**

Name:

Connection | Security | Messages

Server:  Port:

☐ Enable secure (SSL/TLS) connection

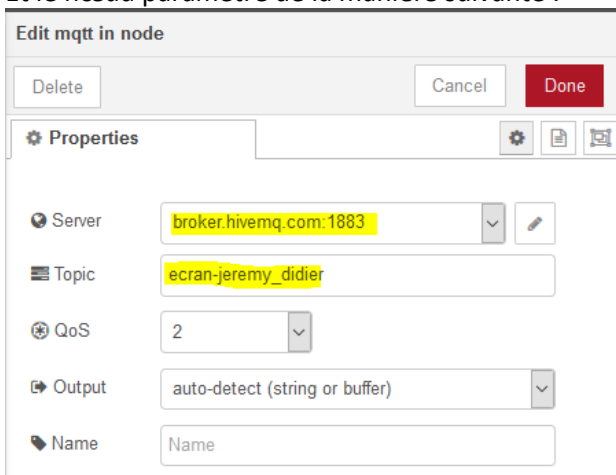
Client ID:

Keep alive time (s):  ☒ Use clean session

☒ Use legacy MQTT 3.1 support

Et le nœud paramétré de la manière suivante :

on obtient alors :



**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server:

Topic:

QoS:

Output:

Name:



```

13/05/2020 à 10:02:12 node: e88f3dca.ae419
ecran-jeremy_didier : msg.payload : Object
{ message: "Test" }

13/05/2020 à 10:03:02 node: e88f3dca.ae419
ecran-jeremy_didier : msg.payload : Object
{ message: "Test" }

13/05/2020 à 10:03:06 node: e88f3dca.ae419
ecran-jeremy_didier : msg.payload : Object
{ message: "Test" }

```