

APPLICATION MOBILE CORDOVA



HTML5

CSS3

JavaScript

Cordova

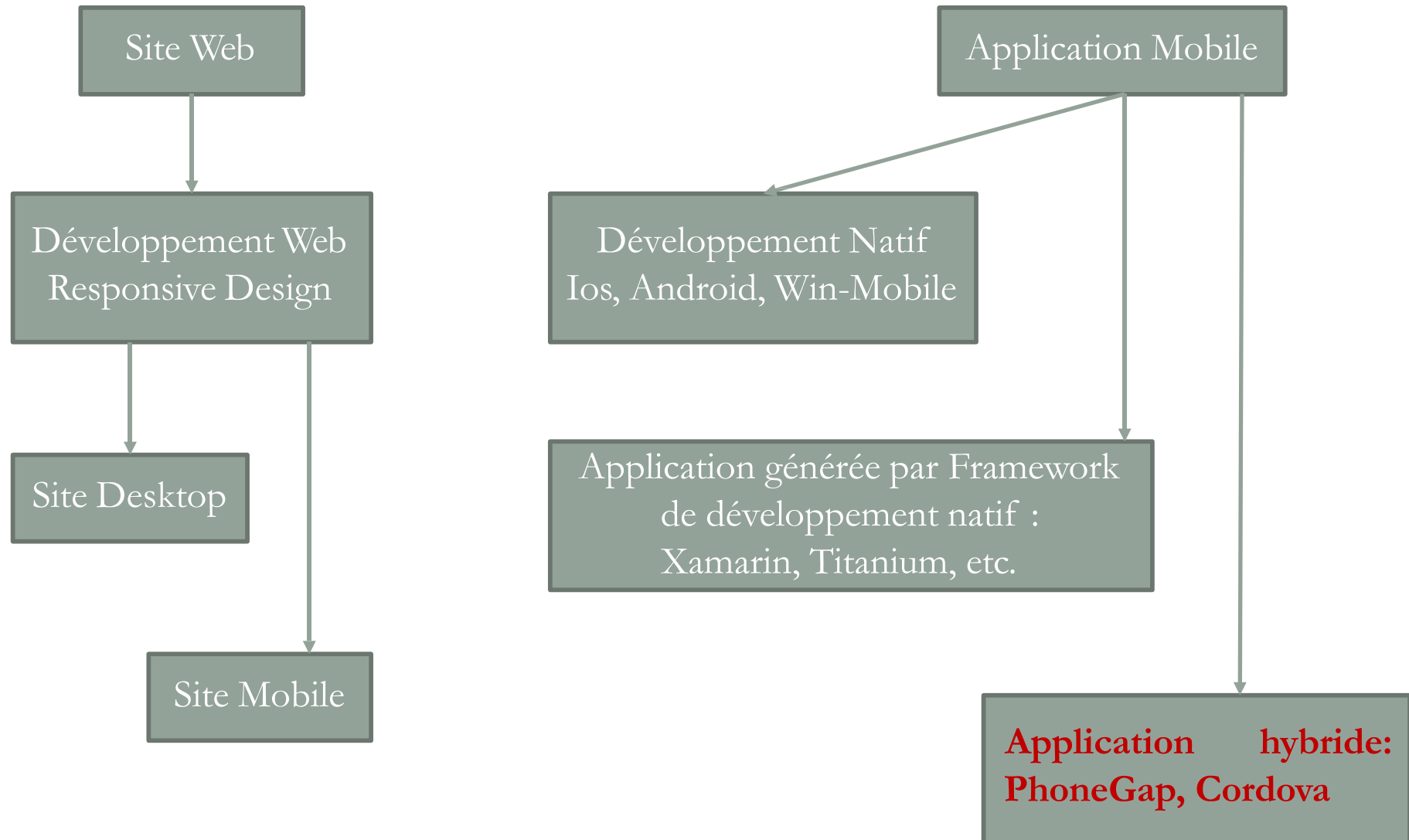




Introduction

- Développement d'applications Multi-plateformes pour appareils mobiles:
 - Une tâche complexe
 - Les différents systèmes sont incompatibles en portage d'applications.
- Alternative de **développement multiplate-forme mobile**:
 - **Cordova** : HTML5, CSS3 et JavaScript.
 - **Cordova** est un conteneur pour interfacer l'application Web avec les **fonctionnalités natives de l'appareil mobile**. Il existe une bonne vingtaine de plugins propres au Framework et bien d'autres sur GitHub.
 - La caméra, le GPS, le système de fichiers, la géolocalisation, etc.
 - Le site de référence du **Framework Cordova** :
<https://cordova.apache.org/docs/en/latest/>

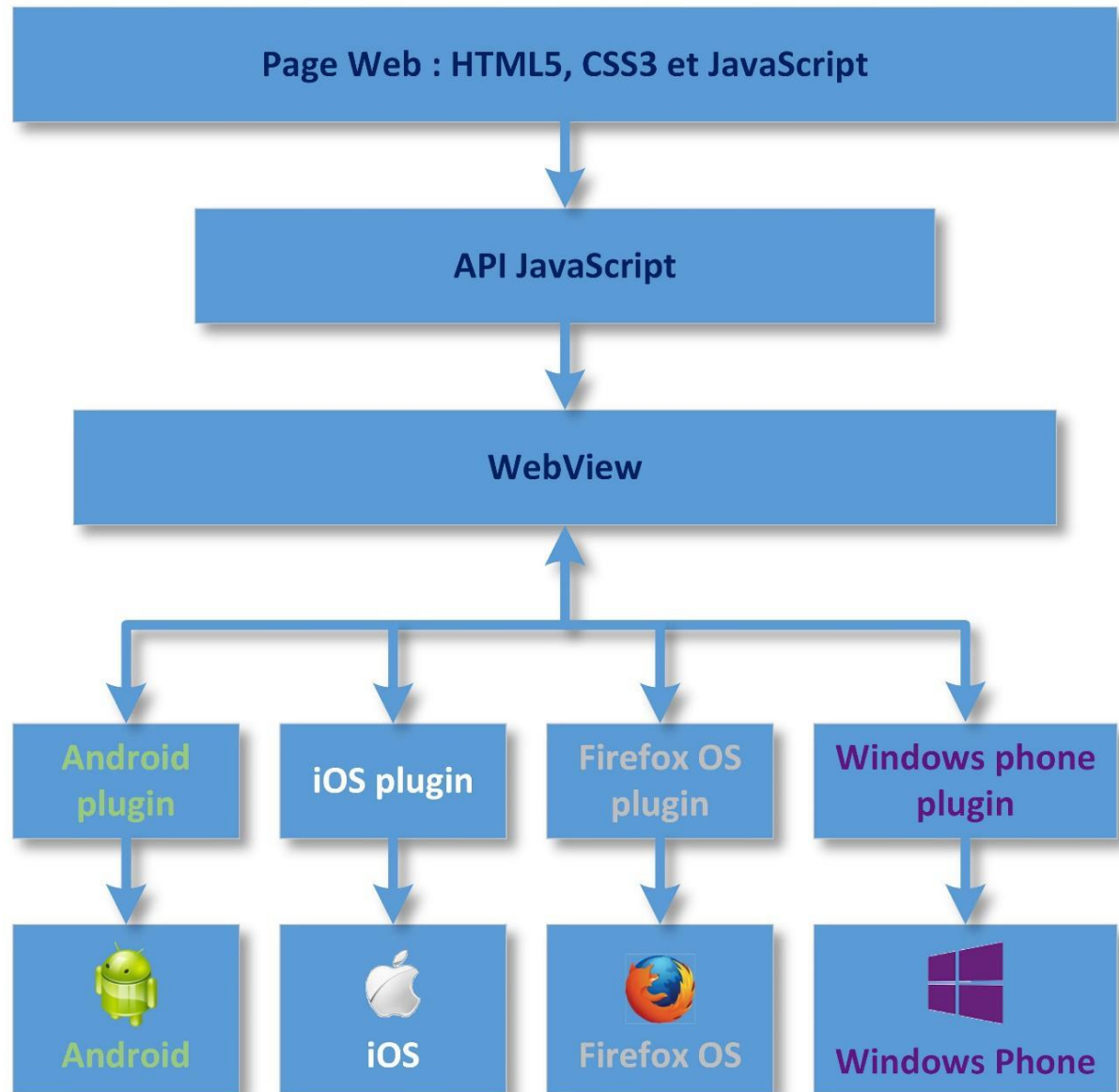
Application Web hybride et native



Principe de fonctionnement



- **Cordova** est une **API JavaScript**.
- Cordova sert de *wrapper* pour rendre cohérente l'application mobile avec les dispositifs de l'appareil.
- Systèmes concernés:
 - iOS, Android, Windows Phone, Firefox OS, Ubuntu Phone.



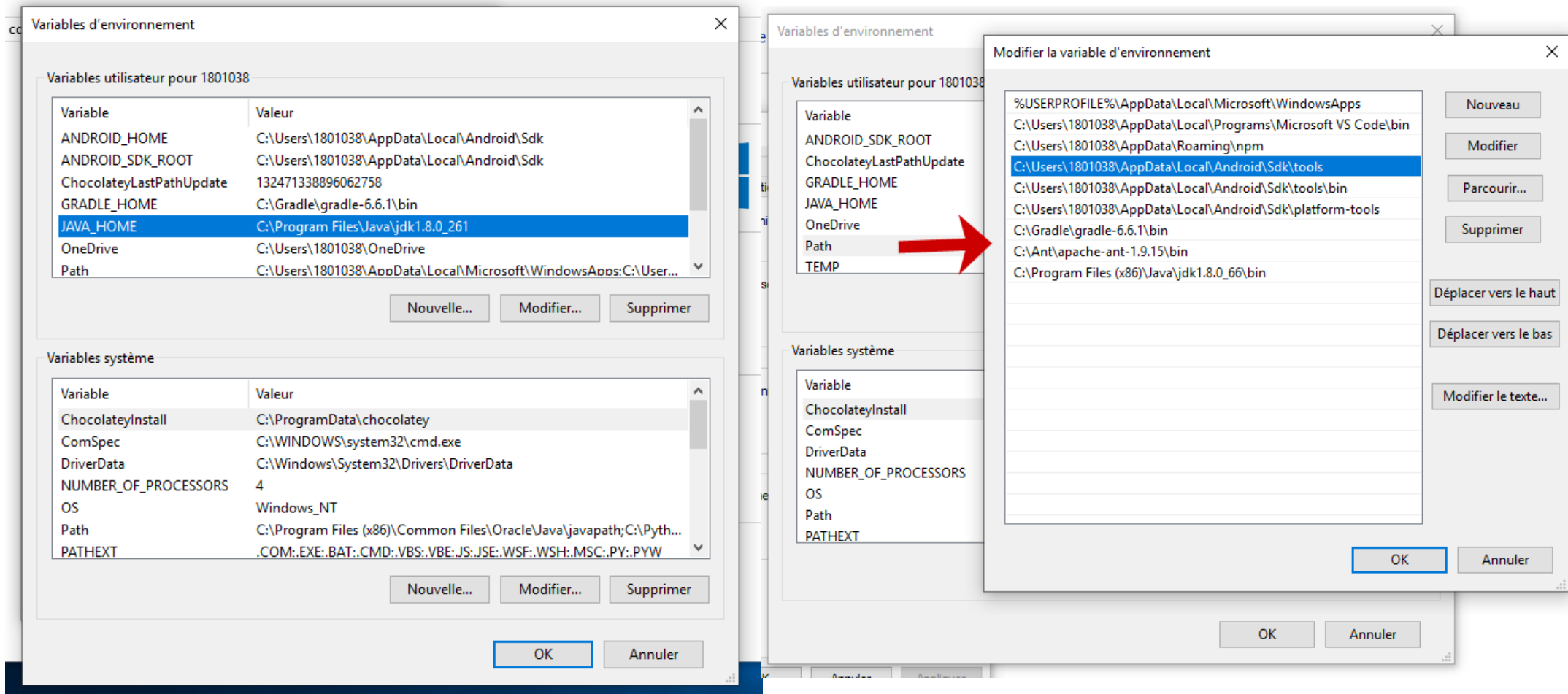
Installer et configurer Cordova



1. Installer **NodeJS** (<https://nodejs.org/>).
2. Installer **Cordova** à partir de l'invité de commandes (cmd - admin):

```
>npm install -g cordova@latest
```
3. Installer la version 8 du **Java Development Kit** à partir du site
<http://www.oracle.com/technetwork/java/javase/downloads/>
Déclarer la variable d'environnement JAVA_HOME pointant vers la racine du dossier Java, Ex: C:\Program Files\Java\jdk1.8.0_261
4. Installer le **SDK Android** (<http://developer.android.com/sdk/>)
 - Ajouter les dossiers tools, tools/bin et platform-tools dans le PATH des variables d'environnement.
 - Installer **Gradle** et ajouter le dossier C://Gradle/bin au PATH et aux variables d'environnement.

Installer et configurer Cordova – Variables env. utilisateur



Créer une application Cordova



- On peut créer le projet "*bonjour*" avec la commande :

```
>cordova create bonjour
```

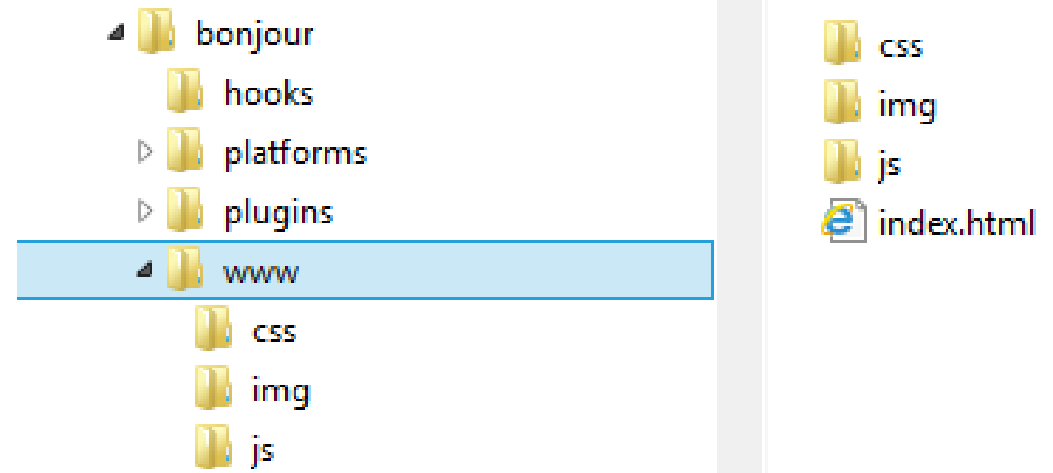
- On accède au répertoire du projet :

```
>cd bonjour
```

- On ajoute la plateforme cible :

```
>cordova platform add android --save
```

- La configuration du projet est ainsi sauvegardée dans le fichier *bonjour/config.xml*. Le fichier contient le **point d'entrée de l'application** `<content src="index.html" />`. L'élément `<name>` spécifie le nom de l'application.
- Le fichier *bonjour/www/index.html* contient le squelette d'une application qu'on peut modifier ou remplacer.



Code d'une application Cordova



Présentation du fichier **index.html**, point d'entrée de votre application :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap:
      https://ssl.gstatic.com 'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src *;
      img-src 'self' data: content:; ">
    <meta name="format-detection" content="telephone=no">
    <meta name="msapplication-tap-highlight" content="no">
    <meta name="viewport" content="initial-scale=1, width=device-width, viewport-fit=cover">
    <link rel="stylesheet" type="text/css" href="css/index.css">
    <title>Hello World</title>
  </head>
  <body>
    <div class="app">
      <h1>Apache Cordova</h1>
      <div id="deviceready" class="blink">
        <p class="event listening">Connecting to Device</p>
        <p class="event received">Device is Ready</p>
      </div>
    </div>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
  </body>
</html>
```


Contrôle de la configuration générale et de la configuration du projet



```
Administrateur : Invite de commandes

C:\WINDOWS\system32>node -v
v12.19.0

C:\WINDOWS\system32>npm -v
6.14.8

C:\WINDOWS\system32>javac -version && java -version
javac 1.8.0_261
java version "1.8.0_261"
Java(TM) SE Runtime Environment (build 1.8.0_261-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

C:\WINDOWS\system32>gradle -v

-----
Gradle 6.6.1
-----

Build time:   2020-08-25 16:29:12 UTC
Revision:     f2d1fb54a951d8b11d25748e4711bec8d128d7e3

Kotlin:       1.3.72
Groovy:        2.5.12
Ant:           Apache Ant(TM) version 1.10.8 compiled on May 10 2020
JVM:           1.8.0_261 (Oracle Corporation 25.261-b12)
OS:            Windows 10 10.0 amd64

C:\WINDOWS\system32>
```

```
Administrateur : Invite de commandes

C:\wamp64\www\TP_DWWM\Front\Cordova\bonjour>cordova requirements android

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed true
Android target: installed android-30,android-29
Gradle: installed C:\Gradle\gradle-6.6.1\bin\gradle.BAT

C:\wamp64\www\TP_DWWM\Front\Cordova\bonjour>
```

Mise en route d'une application Cordova



- Pour compiler le projet :
`>cordova build`
- Pour tester l'application on peut lancer le simulateur intégré dans le SDK Android :
`>cordova emulate android`
- ou sur un appareil mobile par connexion USB :
`>cordova run android`



Comment ça marche Cordova



- La balise `<meta>` de la section `<head>` dans le fichier `index.html` spécifie le codage du document avec l'attribut `charset` et son adaptabilité vis-à-vis les différentes tailles et résolutions d'écrans mobiles avec l'attribut `viewport`.
- La bibliothèque *JavaScript* nécessaire au fonctionnement de *Cordova* est introduite dans la balise `<script>` : `<script src="cordova.js"> </script>`
- Les fonctionnalités de la bibliothèque ne sont pas disponibles immédiatement, mais seulement après la compilation du projet. Le script **index.js** est alors sollicité au chargement du fichier **index.html** dans la **WebView**.
Ce sera dans ce script que vous pourrez ajouter de nouvelles fonctionnalités en **JavaScript**.

Les plugins Cordova



- Un **plugin Cordova** est un code d'extension (*add-on*) qui fournit l'interface **JavaScript** pour **communiquer** avec les **composants natifs de l'appareil mobile**.
- Le **plugin** permet à l'application d'utiliser les **capacités natives du périphérique** au-delà de ce qui est disponible pour une application Web classique.
- Pour **ajouter** la dernière version du **plugin pour la caméra** :

```
>cordova plugin add cordova-plugin-camera@latest --save
```
- Pour **enlever** le **plugin de la caméra** :

```
>cordova plugin rm cordova-plugin-camera --save
```

L'API Cordova : Device



- L'objet **Device** fournit des d'informations sur l'unité mobile et sur l'application en cours.
- L'API s'installe par le plugin *cordova-plugin-device*.
>cordova plugin add cordova-plugin-device@latest --save
- Dans le code on fait appel à la fonction *anonyme* lorsque l'événement *deviceready* est déclenché au chargement de la page HTML.
- L'attribut *innerHTML* remplace le contenu identifié par *devInfo* sur la page HTML.

L'API Cordova : Device



```
document.addEventListener("deviceready",function() {  
    var tmpStr="";  
  
    tmpStr+='<ul class="w3-ul">';  
    tmpStr+='<li>Version de Cordova: '+device.cordova+'</li>';  
  
    tmpStr+='<li>Systeme: '+device.platform+'</li>';  
    tmpStr+='<li>Version de l'OS: '+device.version+'</li>';  
    tmpStr+='<li>Modèle de l'appareil: '+device.model+'</li>';  
    tmpStr+='<li>Identifiant unique universel: '+device.uuid+'</li>';  
    tmpStr+='</ul>';  
    document.getElementById('devInfo').innerHTML=tmpStr;  
  
}, true);
```

L'API Cordova : Device



Information sur l'API Cordova

L'application fait appel à cordova-plugin-device

Cordova Version: 5.2.2

Operating System: Android

OS Version: 6.0.1

Device Model: Nexus 5

Universally Unique Identifier: 59cb2a62a5a387e7



L'API Cordova : Geolocation



- L'API *Geolocation* est basée sur la spécification de W3C.
- L'objet *navigator.geolocation* permet l'accès aux données de localisation du capteur GPS (*Système de Positionnement Global*) de l'appareil ou déduites des signaux du réseau mobile
 - les IDs cellulaires GSM/CDMA
 - Wi-Fi (l'adresse IP, RFID, les adresses MAC)
 - Bluetooth.
- Les **données de géolocalisation** sont considérées comme **sensibles**. Ces données révèlent l'endroit d'utilisation du GPS.
- Une application **devrait afficher une notice** avant d'accéder aux données par **respect de confidentialité** et ainsi permettre de recueillir l'**autorisation de l'utilisateur** de poursuivre.

L'API Cordova : Geolocation



- La fonctionnalité demande plugin *cordova plugin add cordova-plugin-geolocation@latest --save*
- Les autorisations nécessaires à la géolocalisation sont apportées au fichier : *platforms/android/res/xml/config.xml*

```
<feature name="Geolocation">  
  <param name="android-package"  
    value="org.apache.cordova.GeoBroker" />  
</feature>
```

et au fichier : *platforms/android/AndroidManifest.xml*

```
<uses-permission android:name =  
  "android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name=  
  "android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission android:name=  
  "android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"  
/>
```

L'API Cordova : Geolocation



- Pour détecter la position de l'appareil on fait appel à la fonction asynchrone `getCurrentPosition(par1,[par2],[par3])` dans la fonction *anonyme* passée au gestionnaire sur l'évènement *deviceready*.
- La fonction renvoie la position de l'appareil sous la forme d'objet :
`navigator.geolocation.getCurrentPosition(onSuccess,
onError);`
- La fonction *onSuccess* est appelée si les mesures de géoposition sont transmises. Elle prend en paramètre l'objet de la localisation pour exposer les coordonnées à travers la propriété *coords*.

```
function onSuccess(pos) {  
    var latitude = pos.coords.latitude;  
    var longitude = pos.coords.longitude;  
}
```

- La fonction *onError* est appelée si aucune donnée n'est transmise.

L'API Cordova : Geolocation



- L'objet géolocalisation fournit des informations sur la mesure d'un ensemble de propriétés incluant :

- Latitude,
- Longitude,
- Altitude,
- Accuracy,
- Altitude,
- Accuracy,
- Heading,
- Speed,
- Timestamp.

| Property | Returns |
|-------------------------|---|
| coords.latitude | The latitude as a decimal number (always returned) |
| coords.longitude | The longitude as a decimal number (always returned) |
| coords.accuracy | The accuracy of position (always returned) |
| coords.altitude | The altitude in meters above the mean sea level (returned if available) |
| coords.altitudeAccuracy | The altitude accuracy of position (returned if available) |
| coords.heading | The heading as degrees clockwise from North (returned if available) |
| coords.speed | The speed in meters per second (returned if available) |
| timestamp | The date/time of the response (returned if available) |

L'API Cordova : Geolocation



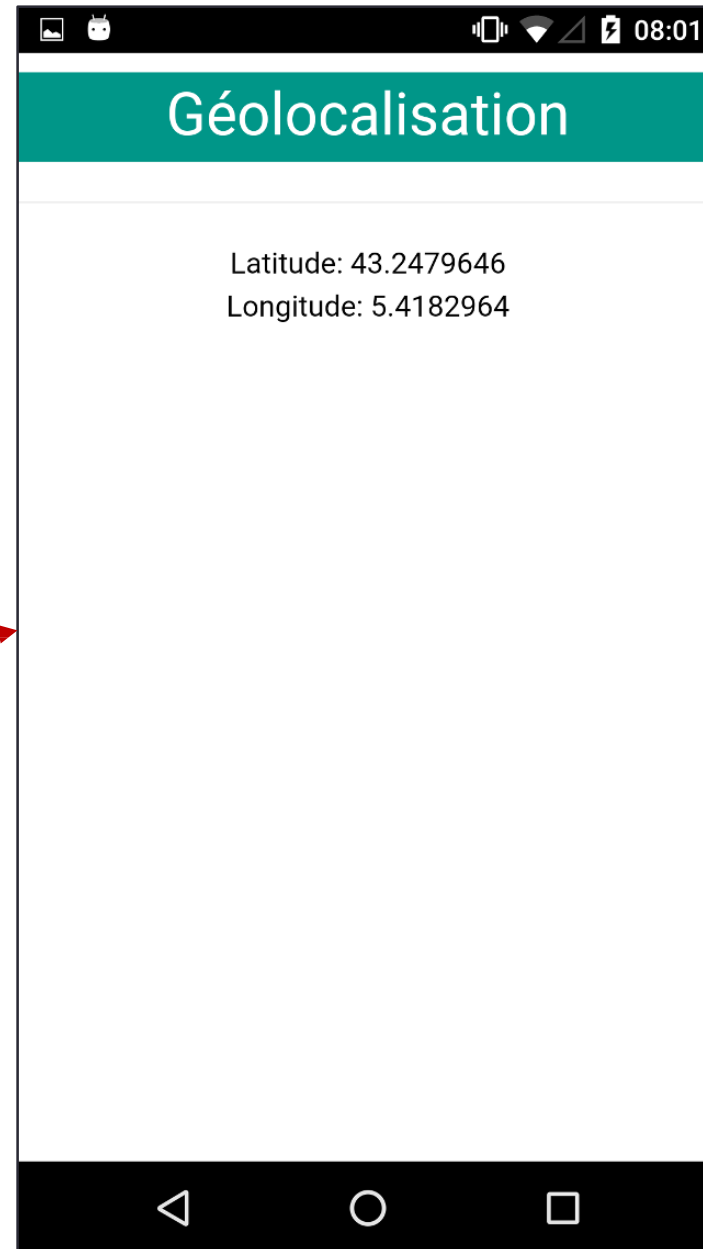
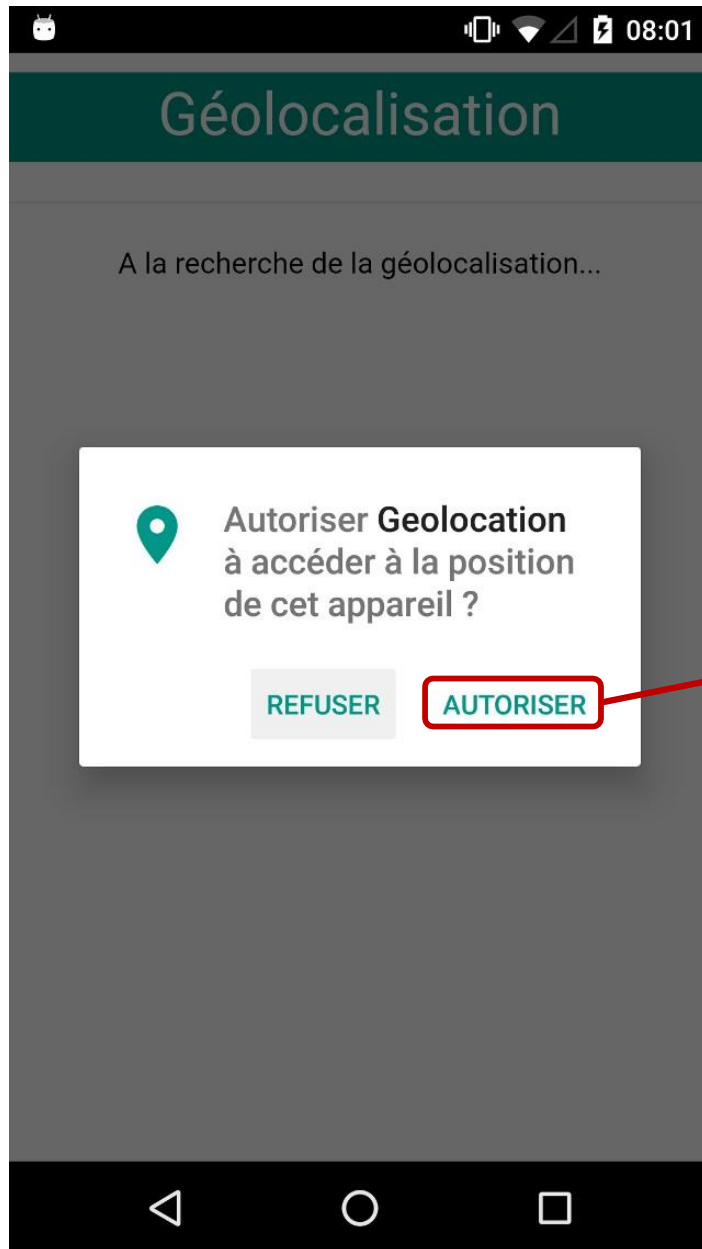
```
document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {
    navigator.geolocation.getCurrentPosition
        (onSuccess, onError);
}

function onSuccess(pos) {
    document.getElementById('geoloc').innerHTML = 'Latitude: ' +
        pos.coords.latitude + '<br />' +
        'Longitude: ' + pos.coords.longitude;
}

function onError(error) {
    document.getElementById('geoloc').innerHTML = 'code: ' +
        error.code + '<br />message: ' + error.message;
}
```

L'API Cordova : Geolocation



L'API Cordova : Camera



- L'API *Camera* peut prendre des photos avec la caméra de l'appareil mobile, ou peut choisir des images à partir d'une bibliothèque.
- Pour implémenter ses fonctionnalités il faut ajouter le plugin *cordova-plugin-camera* dans le projet.
- L'objet *navigator.camera* donne l'accès aux fonctions de la caméra.
- L'application devrait afficher une notice avant d'accéder aux fonctionnalités de l'API Camera par respect de la confidentialité et ainsi permettre de recueillir l'autorisation de l'utilisateur de poursuivre.

L'API Cordova : Camera

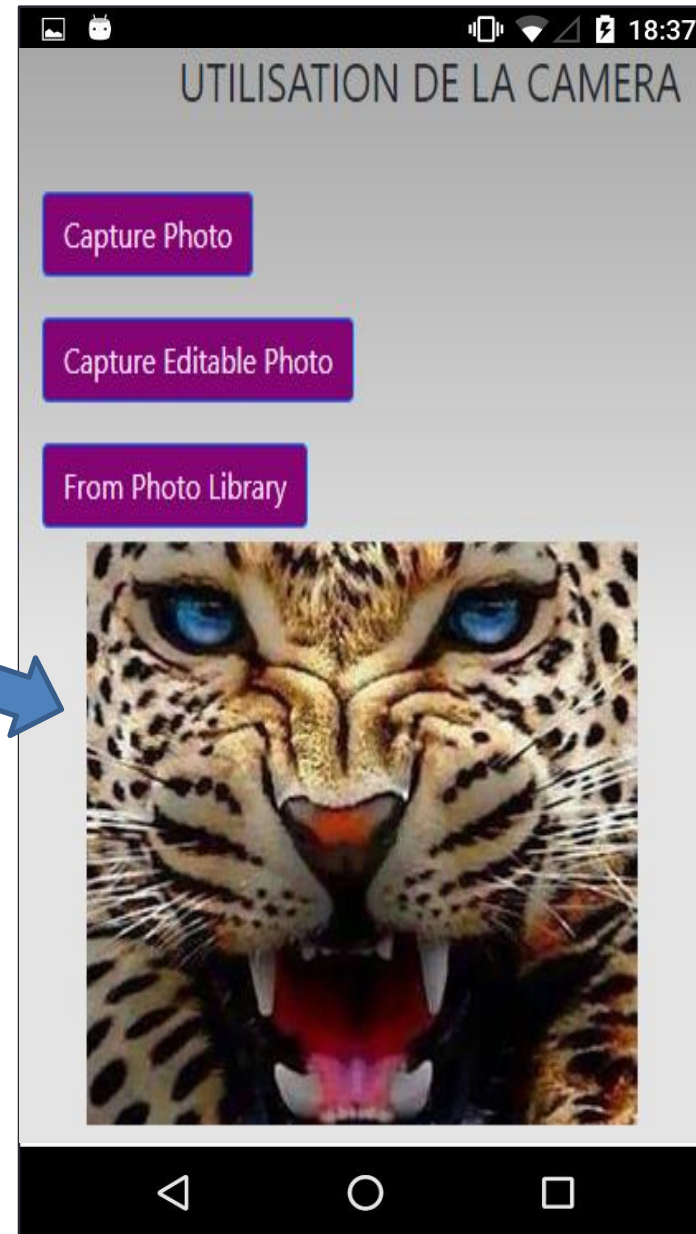
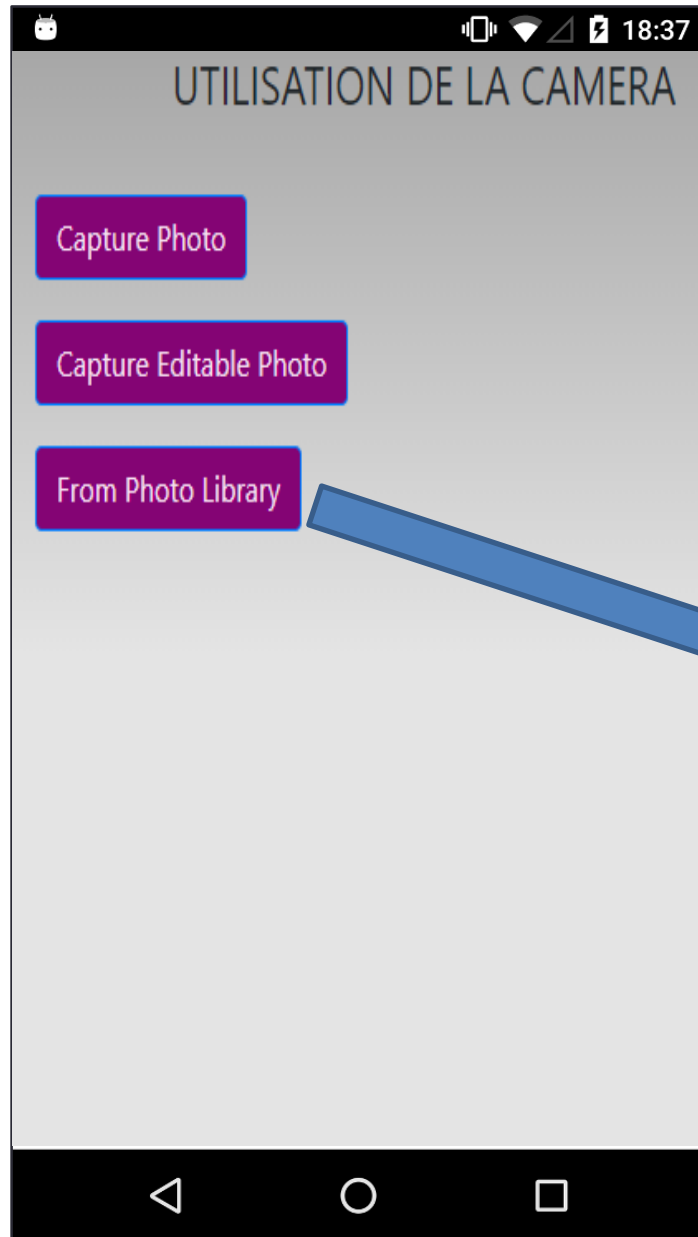


- On fait appel à la fonction *getPictures* pour ouvrir l'application par défaut de l'appareil mobile permettant à l'utilisateur de prendre des photos ou de consulter la galerie photos.

navigator.camera.getPicture(onSuccess,onFail,[Options]);

- Trois options sont disponibles :
 1. *Camera.PictureSourceType.CAMERA* active l'application caméra pour la prise de photos;
 2. *Camera.PictureSourceType.PHOTOLIBRARY*, ou
 3. *Camera.PictureSourceType.SAVEDPHOTOALBUM*, ouvre une boîte de dialogue pour accéder à une image existante.
- La fonction passe au "callback" (*onSuccess*) l'image comme une chaîne de caractères encodée en *base64* ou l'*URI* du fichier image.

L'API Cordova : Camera



L'API Cordova : Camera



```
<!DOCTYPE html>
<html>
  <head>
    <title>Capture Photo</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/index.css">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>Utilisation de la Camera</h1>
    <br><br>
    <div class="container">
      <button id="btnCapture" class="btn btn-primary monbouton">Capture Photo</button><br><br>
      <button id="btnCaptureEdit" class="btn btn-primary monbouton">Capture Editable Photo</button> <br><br>
      <button id="btnPhotoLibrary" class="btn btn-primary monbouton">From Photo Library</button><br><br>
      <img style="display:none;width:60px;height:60px;" id="smallImage" src="" />
      <img style="display:none;" id="largeImage" src="" />
    </div>
    <script src="js/jquery-3.4.1.js"></script>
    <script src="js/bootstrap.js"></script>
    <script src="cordova.js"></script>
    <script src="js/index.js"></script>
  </body>
</html>
```

L'API Cordova : Camera début du code index.js



```
var pictureSource;
var destinationType;

var btnCapture      = document.getElementById("btnCapture");
var btnCaptureEdit  = document.getElementById("btnCaptureEdit");
var btnPhotoLibrary = document.getElementById("btnPhotoLibrary");

document.addEventListener("deviceready",onDeviceReady,false);

function onDeviceReady() {
    pictureSource = navigator.camera.PictureSourceType;
    destinationType = navigator.camera.DestinationType;
}

btnCapture.addEventListener("click", function(){

    navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50,
    destinationType: destinationType.DATA_URL });

});

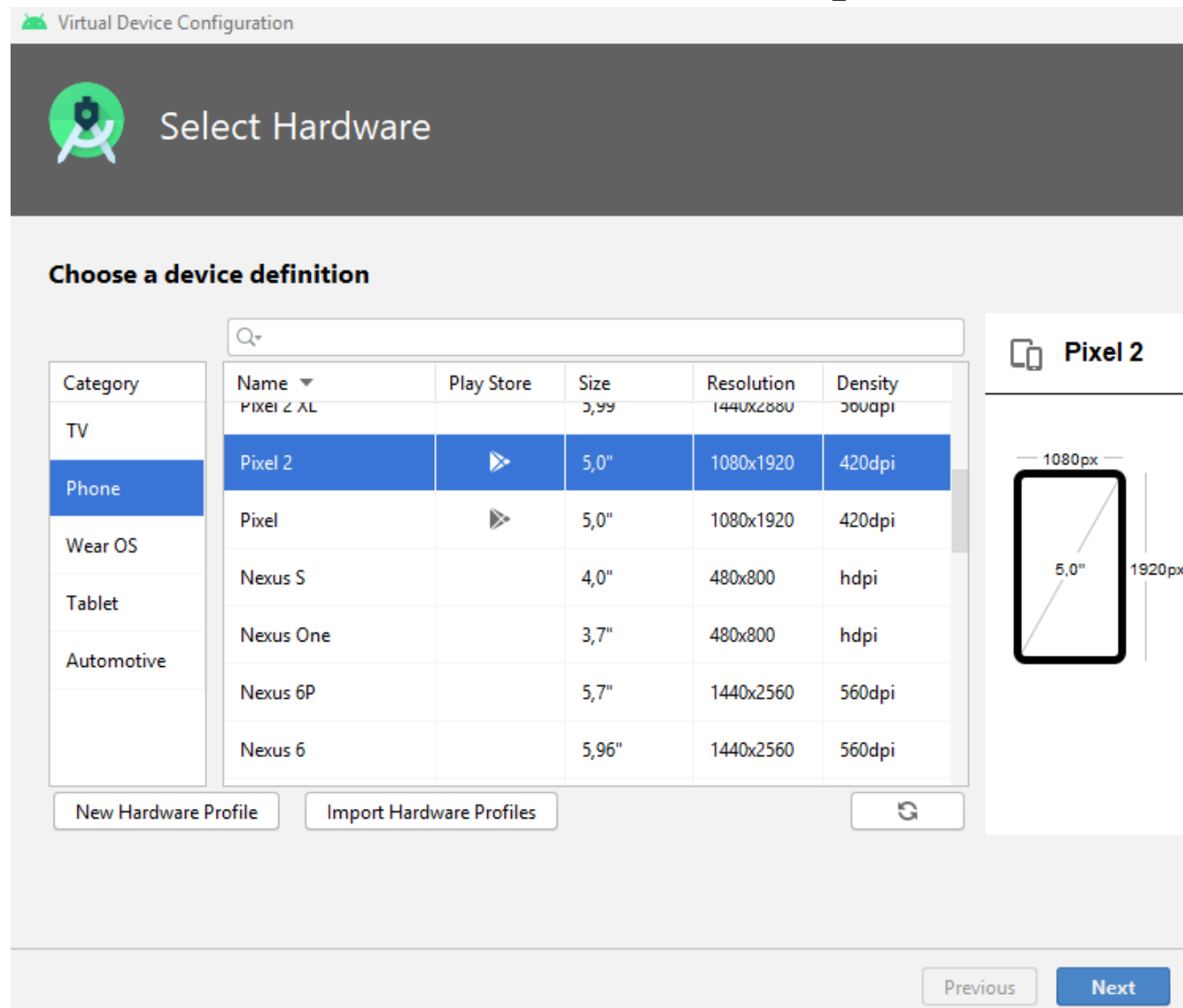
function onPhotoDataSuccess(imageData) {
    var smallImage = document.getElementById('smallImage');
    smallImage.style.display = 'block';
    smallImage.src = "data:image/jpeg;base64," + imageData;
}

// A vous d'écrire les différents gestionnaires d'évènements pour btnCaptureEdit et btnPhotoLibrary
```

Configurer un émulateur Android



- Le SDK d'Android ne fournit aucune instance d'émulateur par défaut.
- L'outil AVD Manager (*Android Virtual Device*) fournit l'interface pour la création d'unités virtuelles d'émulation.
- On peut choisir un élément du *Dispositif de définitions* dans la boîte de dialogue



Configurer un émulateur Android




Virtual Device Configuration

System Image

Select a system image

Recommended x86 Images Other Images

| Release Name | API Level ▾ | ABI | Target |
|---------------------------------|-------------|-----|-----------------------------|
| R Download | 30 | x86 | Android 10.0+ (Google Play) |
| Q Download | 29 | x86 | Android 10.0 (Google Play) |
| Pie Download | 28 | x86 | Android 9.0 (Google Play) |
| Oreo Download | 27 | x86 | Android 8.1 (Google Play) |
| Oreo Download | 26 | x86 | Android 8.0 (Google Play) |
| Nougat Download | 25 | x86 | Android 7.1.1 (Google Play) |
| Nougat Download | 24 | x86 | Android 7.0 (Google Play) |



API Level
30

Android
10.0+

Google Inc.

System Image
x86

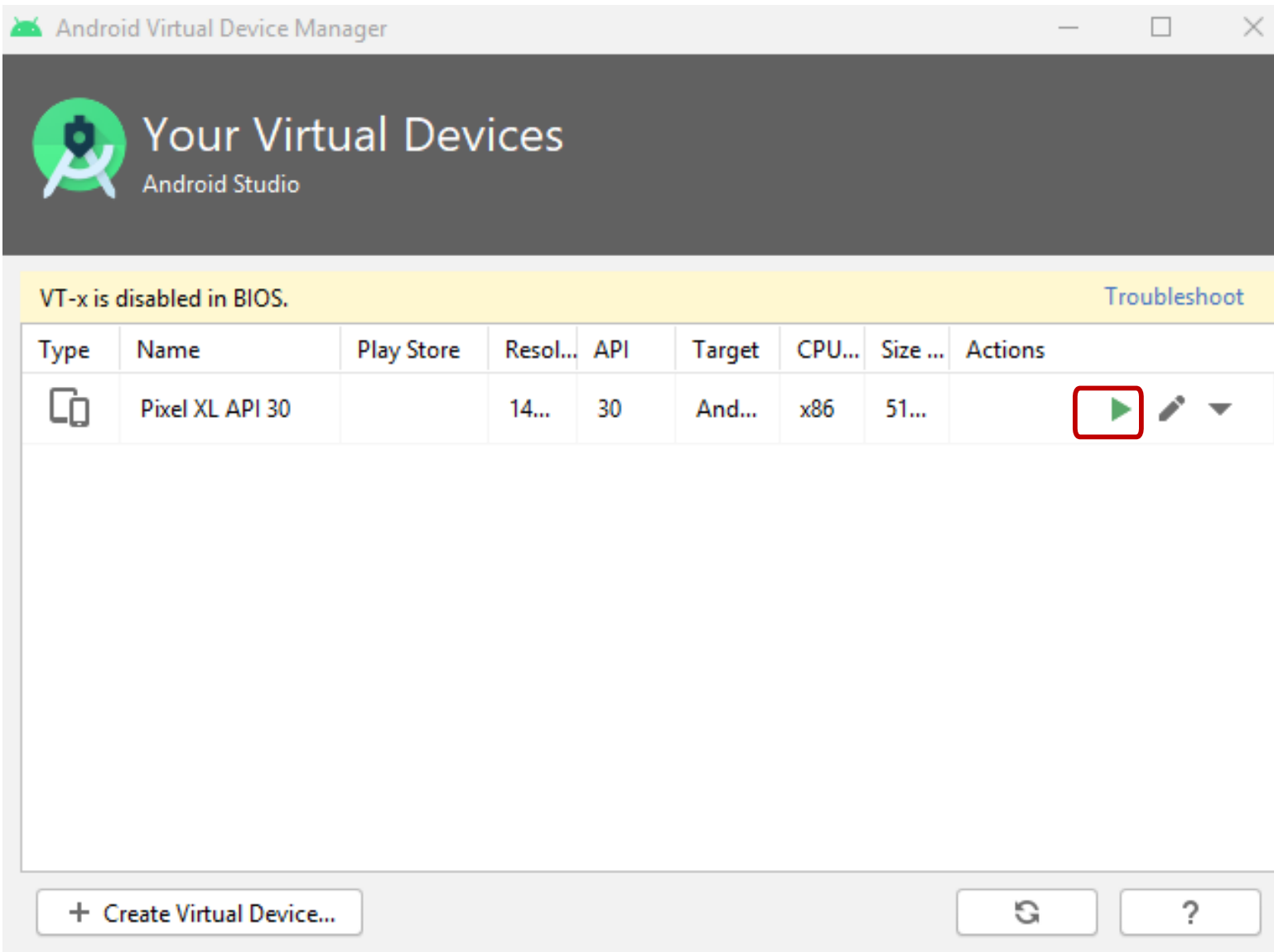
Recommendation
VT-x is disabled in BIOS.
[Troubleshoot](#)

! A system image must be selected to continue.

Previous Next **Cancel** Finish Help

La configuration peut être sauvegardée et modifiée à tout moment

Configurer un émulateur Android



L'AVD apparaît dans la liste des périphériques virtuels Android. Il faut le lancer par le bouton "Start" pour le rendre disponible pour l'application Cordova.

Configurer un émulateur Android



L'émulateur lancé, l'application Cordova prend l'unité active pour visualiser le résultat de l'application à l'écran

Version embarquée d'un site Web



- Un **site Web adaptatif**, répondant aux règles du **responsive design** pour smartphones et tablettes peut être rendu accessible au travers d'une application **Cordova**.
- Ainsi, un **site responsive** qui s'adapte à tout type de support peut se faire **embarquer par une application Cordova**. On se sert de l'objet *window* de JavaScript pour ouvrir une fenêtre avec sa méthode *open* dans le navigateur.
- 3 paramètres sont à renseigner :

window.open(URL [,nom] [,options]);

- *URL* contient l'adresse de la page à afficher.
- *nom* de la fenêtre.
- *options* à paramétrer pour la fenêtre



Publier son appli Android Cordova sur Google Play

<https://developer.android.com/studio/publish/app-signing.html#signing-manually>

<https://loicknuchel.fr/blog/2015/03/26/publier-son-application-cordova-sur-google-play>

<https://www.wolwx.net/creation-dune-application-android-webapp-utiliser-phonegap-cordova-avec-android-studio/>