HTML-based Android Apps

Lecture contents

- Why make an HTML-based app?
- Intro to PhoneGap/Cordova
- Basics of deploying an app
- Most important library calls

Phonegap / Cordova
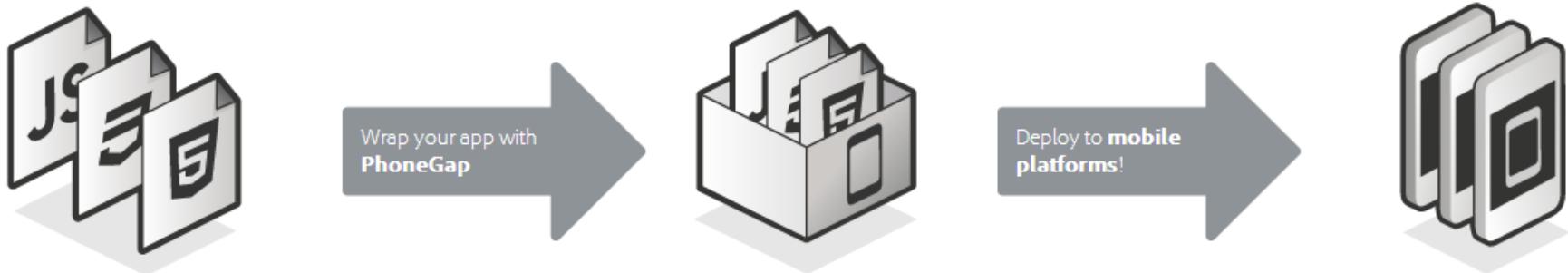
- Apache Cordova is a tool for packaging HTML/JS/CSS-based apps into mobile apps
- Apps can be deployed to multiple platforms
- Libraries to access phone's internal functions
- Phonegap is Adobe's distribution of Cordova
- Phonegap has online build tools

HTML as an app

Server

App

Server:
MVC
REST
Controller

App:
HTML

Java
Script

JSON

Example Android app
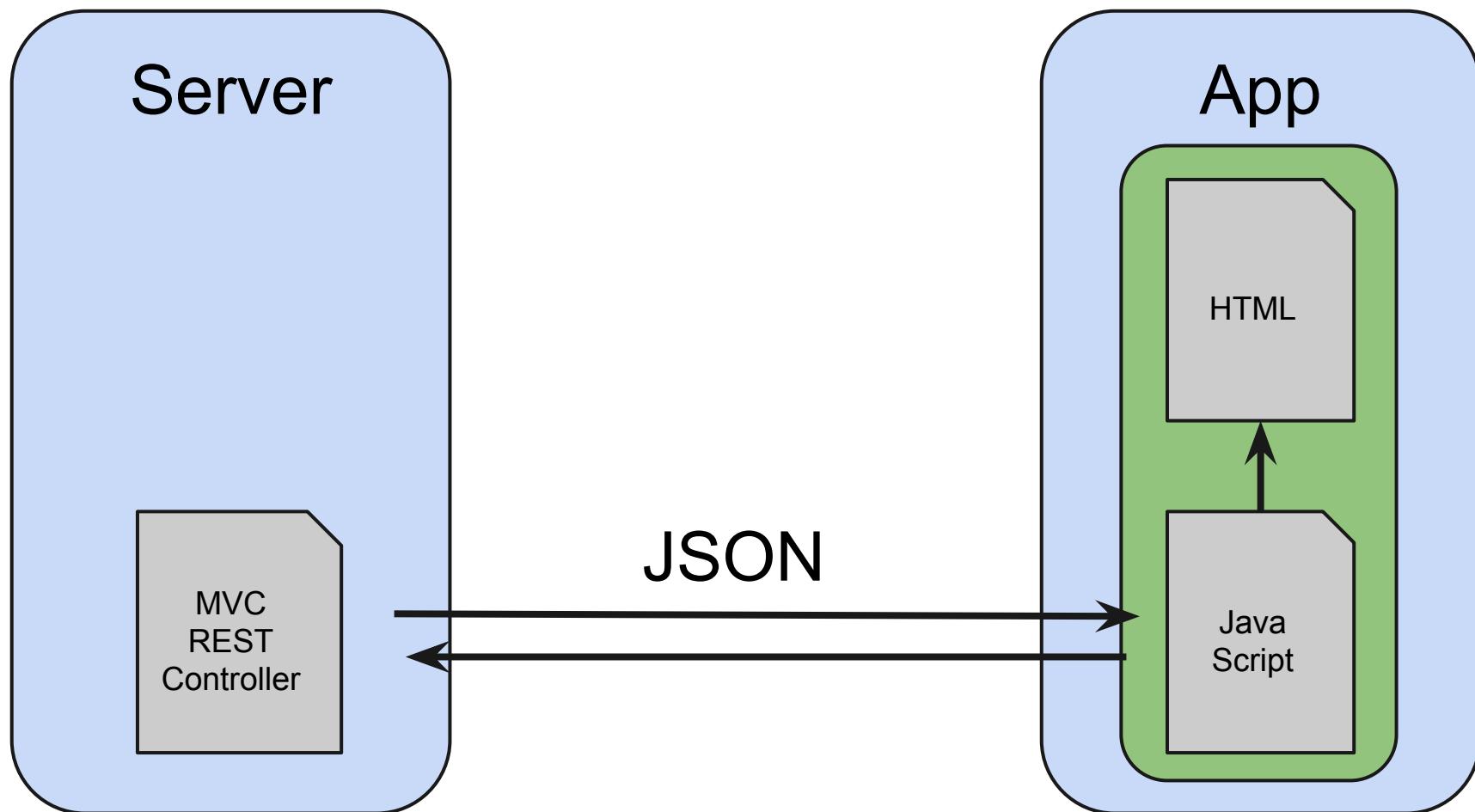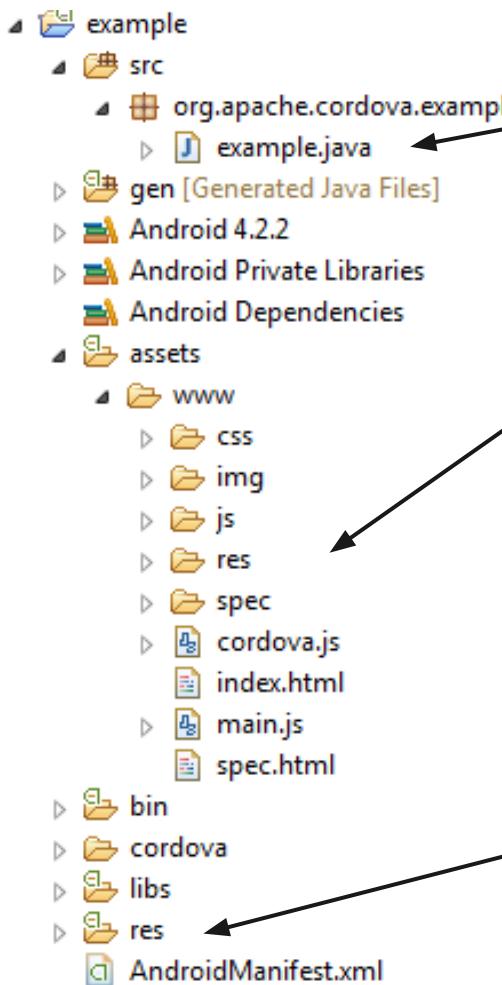
```
▲ 📁 example
  ▲ 📁 src
    ▲ 🔶 org.apache.cordova.example
      ▷ J example.java                          ←———— Embedded web browser
  ▷ 📁 gen [Generated Java Files]
  ▷ 📗 Android 4.2.2
  ▷ 📗 Android Private Libraries
    📗 Android Dependencies
  ▲ 📁 assets
    ▲ 📂 www
      ▷ 📂 css                                  Web files
      ▷ 📂 img
      ▷ 📂 js
      ▷ 📂 res                         ←————————
      ▷ 📂 spec
      ▷ 📄 cordova.js
        📄 index.html
      ▷ 📄 main.js
        📄 spec.html
  ▷ 📁 bin
  ▷ 📂 cordova                                  Resource files, including a
  ▷ 📁 libs                                     cordova config xml file
  ▷ 📁 res                         ←————————
    📄 AndroidManifest.xml
```

Pre-requisites

- The SDK of the platform (Android, iOS, Blackberry, Windows Phone, WebOS)
- Phonegap allows you to upload and build your project, without a local SDK (costs $$)
- Node.js (A Javascript runtime)
- Installing Cordova: (in theory)
  - *$ sudo npm install -g cordova*
  - *$ cordova create hello com.example.hello HelloWorld*
- On Windows, the above is difficult!
  - Instead, simpler to download and unpack the an already built example (seems this is now sort of fixed

Command line examples

$ sudo npm install -g cordova

$ cordova create hello com.example.hello HelloWorld

$ cd hello

$ cordova platform add ios (Mac only)

$ cordova platform add android

$ cordova platforms ls

$ cordova build

$ cordova build android

$ cordova emulate android

$ cordova run android

config.xml

- The config.xml file sets up Cordova behaviour
- Specifies APIs, plugins and platform fns

```
<widget id="com.example.hello" version="0.0.1">
    <name>HelloWorld</name>
    <description>A sample application</description>
    <author email="lars@roland.bz" href="http://dhis2.org">Mobilars</author>
    <content src="index.html" />
    <access origin="*" />
    <preference name="Fullscreen" value="true" />
    <preference name="WebViewBounce" value="true" />
</widget>
```

config.xml 2

- The feature tag enabled device-level APIs

```xml
<feature name="Plugin" value="PluginID" />
```

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

App plugin

Accessing some basic app features.

- navigator.app.clearCache
- navigator.app.show
- navigator.app.loadUrl('http://dhis2.org/');
- navigator.app.cancelLoadUrl
- navigator.app.clearHistory
- navigator.app.backHistory
- navigator.app.overrideButton
- navigator.app.overrideBackButton
- navigator.app.exitApp

Device plugin

Access device information (hardware and firmware) from Javascript:

```
var element = document.getElementById('deviceProperties');
   element.innerHTML = 'Device Model: '    + device.model    + '<br />' +
          'Device Cordova: '  + device.cordova  + '<br />' +
          'Device Platform: ' + device.platform + '<br />' +
          'Device UUID: '     + device.uuid     + '<br />' +
          'Device Version: '  + device.version  + '<br />';
```

Geolocation plugin

- ● Provides location access from Javascript

```
function onDeviceReady() {
    navigator.geolocation.getCurrentPosition(onSuccess, onError);
}
function onSuccess(position) {
    var element = document.getElementById('geolocation');
    element.innerHTML = 'Latitude: '      + position.coords.latitude      + '<br />' +
                'Longitude: '      + position.coords.longitude      + '<br />' +
                'Altitude: '      + position.coords.altitude      + '<br />' +
                'Accuracy: '      + position.coords.accuracy      + '<br />';
}
function onError(error) {
    alert('code: '    + error.code    + '\n' +
        'message: ' + error.message + '\n');
}
```

Passing additional geolocal parameters

- You can also pass additional parameters

navigator.geolocation.getCurrentPosition(

location_found, // method called when location is found

location_error, // method called when location !found

{frequency:5000,

maximumAge: 0, // Accept a cached position with age x

timeout: 10000, // ms before timing out

enableHighAccuracy:true});

More info [here](#) and [here](#).

Small print: Android 2.x emulators do not return a geolocation result unless the `enableHighAccuracy` option is set to `true`.

Accelerometer plugin

- [Access accelerometer data](#)

```javascript
navigator.accelerometer.getCurrentAcceleration(onSuccess, onError);

function onSuccess(acceleration) {
    alert('Acceleration X: ' + acceleration.x + '\n' +
        'Acceleration Y: ' + acceleration.y + '\n' +
        'Acceleration Z: ' + acceleration.z + '\n' +
        'Timestamp: '      + acceleration.timestamp + '\n');
};
function onError() {
    alert('onError!');
};
```

Acceleration 2

```
var options = { frequency: 3000 };
// Update every 3 seconds

var watchID = navigator.accelerometer.watchAcceleration(
    onSuccess, onError, options);
```

Camera plugin

● [Access the phone's camera](#)

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

```
function onSuccess(imageData) {
    var image = document.getElementById('myImage');
    image.src = "data:image/jpeg;base64," + imageData;
}
function onFail(message) {
    alert('Failed because: ' + message);
}
```

Storage plugin

- <u>Storing data on the phone</u>

  .

  window.localStorage.setItem("key", "value");
  window.localStorage.setItem("key2", "value2");

  var value = **window.localStorage.getItem("key");**

  window.localStorage.removeItem("key");
  window.localStorage.clear();

Media plugin

- [Access the media player functions](#)

```
playAudio("http://audio.ibeat.org/content/p1rj1s/p1rj1s_-_rockGuitar.mp3");
var my_media = null;
function playAudio(src) {
        // Create Media object from src
        my_media = new Media(src, onSuccess, onError);

        // Play audio
        my_media.play();
…
}
```

Other plugins

- [Events - pick up key presses and more](#)
- [Connection - info about network status](#)
- [Compass](#)
- [Contacts](#)
- [File](#)
- [Globalization - locale/timezone](#)
- [InAppBrowser](#)
- [Notification](#)
- [SplashScreen](#)