

Phonegap

Sebastian Viereck & Mathias Münscher

I. Worum geht es?

Worum geht es?

- Phonegap ist ein Framework
- für Rich Internet Applications
- auf Mobilien Systemen

Geschichte

- Ursprünglich ein Open-Source Projekt einer Web-Agentur
- 2009 People's Choice Award (O'Reilly Media's Web 2.0 Conference)
- 2011 von Adobe gekauft und weiter als freies Projekt gefördert
- Gleichzeitig als Apache Cordova Projekt übergeben worden (PG ist eine Distribution von Cordova)
- Mitte 2012 Release 2.0

Philosophie

Die Phonegap Entwickler sind überzeugt:

1. Web-Technologie wird die universelle Plattform
2. Alle Technik wird irgendwann wieder ausgemustert

Philosophie

.. und daraus leiten sie Ihre Ziele ab:

1. Web-Technologie als “first class”
Entwicklungs-Plattform
2. Phonegap soll irgendwann überflüssig sein
3. [Inoffiziell:] Keeping proprietary plattformen
out

Welche Rolle spielt PG?

- Größte Verbreitung bei mobilevision.com Befragung 2012 (Knapp vor Sencha Touch, 10% Vorsprung vor Appcelerator Titanium)
- Appcelerator Titanium geht anderen Weg
- Kampf zwischen Web und Native: Facebook wechselt zu Native

2. Übersicht

Spezifikation

- Alle großen Plattformen (und weitere)
 - iOS
 - Android
 - Blackberry
 - Windows

Spezifikation

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	X

Features

- Online Doc: <http://docs.phonegap.com/>
- Bewegung
 - Accelerometer
 - Compass
 - Geolocation

Features

- Persistieren
 - File Access
 - SQL-DB

Features

- Multi-Media
 - Foto (kein Video!)
 - Audio Rec + Playback

Features

- Contacts (lesen/schreiben)
- Connection (Wifi, 3G,..)
- Device-Identifikation
- Events (PG loaded, Device-Lock,..)
- Notification (Alert, Sound, Vibration)

Plugins

- Cloud-Storage
- Barcode Scanner
- Bannerwerbung
- Login via Social Platform

Plugins sind *teilweise kostenpflichtig!*

3. Praxis

Was zeigen wir?

- Einbindung von PG in Javascript
- Eigene Native Funktionen einbauen
- Kleiner Einblick in die Phonegap Library

GPS Query



```
function onSuccess (position) {
  alert(
    'Latitude: '      + position.coords.latitude +
    'Longitude: '    + position.coords.longitude +
    'Altitude: '     + position.coords.altitude +
    'Accuracy: '     + position.coords.accuracy +
    'Altitude Accuracy: ' + position.coords.altitudeAccuracy +
    'Heading: '      + position.coords.heading +
    'Speed: '        + position.coords.speed +
    'Timestamp: '    + new Date(position.timestamp) +
  );
};

function onError(error) {
  alert('code: ' + error.code + '\n' +
    'message: ' + error.message + '\n');
}

navigator.geolocation.getCurrentPosition(onSuccess, onError);
```

Demonstration



Upload Photo



In JS: Eigenes natives Plugin starten

```
cordova.exec( uploadSuccess, uploadFail, "UploadImage", "upload",  
    [  
        imageUrl  
    ]  
);
```

Upload Photo



Eigener Nativer Code

```
public class UploadImage extends Plugin {
    @Override
    public PluginResult execute(String action, JSONArray data, String callbackId)
    {
        if (action.equals("upload")) {
            String imageUrl = data.getString(0);

            upload(imageUrl);

            return new PluginResult(PluginResult.Status.OK, 'Upload finished!');
        }
        return null;
    }
}
```

Upload Photo



JS Callback

```
function uploadSuccess (message) {  
    navigator.notification.alert(  
        message, // message  
        alertDismissed, // callback  
        'Success: ', // title  
        'Ok' // buttonName  
    );  
}
```

Demonstration



Camera



Aufruf Phongap

```
function capturePhoto() {  
  
    function onSuccess(imageData) {  
        updateView(imageData);  
    }  
  
    function onFail(message) {  
        // wenn der User kein Bild ausgewählt hat  
    }  
  
    navigator.camera.getPicture(onSuccess, onFail, {  
        sourceType: Camera.PictureSourceType.CAMERA,  
        destinationType: Camera.DestinationType.FILE_URI,  
        quality: 100  
    });  
}
```


Demonstration



Camera



Cordova *Javascript* Library

```
cameraExport.getPicture =  
    function(successCallback, errorCallback, options) {  
        exec(successCallback, errorCallback, "Camera", "takePicture", args);  
    };
```

Camera



Phonegap *Native* Library

```
public CameraLauncher() {  
  
    public boolean execute(String action, JSONArray args,  
        CallbackContext callbackContext) {  
  
        if (action.equals("takePicture")) {  
  
            if (srcType == CAMERA) {  
                this.takePicture(...);  
            }  
        }  
    }  
}
```

Camera



Phonegap *Native* Library

```
public void getImage() {  
  
    Intent intent = new Intent();  
    intent.setType("image/*");  
  
    startActivityForResult(intent);  
    }  
}
```

Javascript Call to Java

1. Javascript Layer: alle Cordova Methoden rufen `Cordova.exec()`
2. Javascript Layer: `Cordova.exec()` ruft die Javascript Methode `prompt()` auf
3. Java Layer: `CordovaChromeClient.class` überschreibt `onJsPrompt()`, wodurch ein Java Event getriggert wird

Javascript Call to Java



2. Aufruf: `prompt()` in *Javascript*

```
prompt(  
    argsJson,    //Aufforderungstext  
    'gap:'+JSON.stringify([service, action, callbackId]) //Feldvorbelegung  
);
```

Source: [promptbasednativeapi.js](#)

Java Chrome Hack



3. Aufruf: `onJsPrompt()` in *Java*

```
public boolean onJsPrompt(WebView view, String url, String message, String  
defaultValue, JsPromptResult result) {
```

```
    if (defaultValue.substring(0, 4).equals("gap:")) {  
        JSONArray array = new JSONArray(defaultValue.substring(4));  
        String service = array.getString(0);  
        String action = array.getString(1);  
        String callbackId = array.getString(2);  
        exec(service, action, callbackId, message);  
    }
```

Source: CordovaChromeClient.class

4. Review

Pros

- Quasi-Native Apps machbar ohne Kenntnisse
 - ➔ Sales-Systeme nutzbar
 - ➔ Phonegap Build Service
- (Begrenzte) Plattform-Unabhängigkeit
 - ➔ Ein Code für mehrere Plattformen

Cons

- Erheblicher Performance Nachteil gegenüber Nativ
➔ Kritisch bei GUI, DB, 3D,..
- Plattform-Unterschiede müssen berücksichtigt werden (auch PG-unabhängige Aspekte)

5. Fazit

Fazit

- Zwitter-Wesen: Wer die Vorteile beider Systeme nutzen will, muss sich mit beiden auseinandersetzen
- Nicht für jede Anwendung geeignet
- Entwicklung nicht aufzuhalten: PG schafft sich ab.

Quellen

- **Phoneygap Download, FAQ, Docs, Blog, Build..**
<http://www.phoneygap.com>
- **Buch: Phoneygap Beginner's Guide**
[Andrew Lunny, Packt Publishing, 2011]
<http://tinyurl.com/dxtsokb>
- **Sebastian Vierecks Entwickler Blog - Eintrag zu PG**
<http://tinyurl.com/c44lekz>
- **Marktanalyse Crossplattform-Tools**
<http://tinyurl.com/84fy73s>
- **Vergleich auf Stackoverflow**
<http://tinyurl.com/yecq9mj>

Danke!