NativeScript and OpenEdge

Miguel Saez, Julian Steiner
Senior Solution Engineers
2017-11-16
930% YoY npm download growth
(Aug ’16 to Aug ’17)
4200+
NativeScript 3
Developer Experience

Performance
Chrome Dev Tools

Elements
NativeScript for Angular Mobile Development

by Nathan Walker & Nathanael Anderson

Available on Packt
The NativeScript Book
by The Brosteins

Available today. For FREE.

nativescript.org/book
What is NativeScript?

An open source runtime for building and running native iOS, Android and Windows Phone apps with a single, JavaScript code base.
Architecture Choices

- JavaScript (JS)
- TypeScript
- Angular
Not sure...

if this is PhoneGap
No WebView/DOM

Direct access to native APIs in JS

No cross compilation
NativeScript Android example

```javascript
var time = new android.text.format.Time();
time.set(1, 0, 2015);
console.log(time.format("%D"));
```

Output: "01/01/15"
100% access to Android and iOS APIs

```javascript
var alert = new UIAlertView();
alert.message = "Hello world!";
alert.addButtonWithTitle("OK");
alert.show();
```
var alert = new UIAlertView();
alert.message = "Hello world!";
alert.addButtonWithTitle( "OK" );
alert.show();
How does this work?
Platform Approaches

- Google V8 JavaScript Engine
- Webkit JavaScript Core
var time = new android.text.format.Time();
time.set( 1, 0, 2015 );
console.log( time.format( "%D" ) );

Runs on V8 JavaScript VM

var alert = new UIAlertView();
alert.message = "Hello world!";
alert.addButtonWithTitle( "OK" );
alert.show();

Runs on JavaScriptCore VM
Calling Android API

V8 JavaScript Engine

```javascript
var file = new java.io.File(path);
```
Calling Android API

V8 JavaScript Engine

var file = new java.io.File(path);

Type Conversion Service

Native Android

java.lang.String
Calling Android API

V8 JavaScript Engine

var file = new java.io.File(path);

Type Conversion Service

Native Android

java.lang.String

Metadata

java.io.File()
But I don't want to
Write iOS and Android code
NativeScript modules for all the things
NativeScript modules

Code → Modules → Platform API
Example: NativeScript file module

```javascript
import { File } from "file-system";
new File();

new java.io.File( path );

NSFileManager.defaultManager();
fileManager.createFileAtPathContentsAttributes(path);
```
NativeScript Modules—There are a lot
Web UI != Mobile UI

<StackLayout>
  <Label id="Label1" text="This is Label!" />
  <Button text="This is Button!" tap="buttonTap" />
</StackLayout>
Native Layouts

Absolute

Dock

Grid

Stack

Wrap
StackLayout

This layout arranges its children horizontally or vertically. The direction is set with the orientation property.

```xml
<Page>
  <StackLayout orientation="horizontal">
    <Label text="This is Label 1" />
    <Label text="This is Label 2" />
  </StackLayout>
</Page>
```
The `GridLayout` defines a rectangular layout area that consists of columns and rows.

```xml
<GridLayout columns="80, *, auto" rows="auto, *" >
  <Button col="0" />
  <Button col="1" />
  <Button col="2" />
  <Button row="1" colSpan="3" />
</GridLayout>
```
margin-right: -100px !important;
CSS

Label {
    color: red;
    font-size: 20;
    width: 200;
    margin: 20;
}
Supported Properties

This is the list of the properties that can be set in CSS or through the style property of each View:

<table>
<thead>
<tr>
<th>CSS Property</th>
<th>JavaScript Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>color</td>
<td>Sets a solid-color value to the matched view’s foreground.</td>
</tr>
<tr>
<td>background-color</td>
<td>backgroundColor</td>
<td>Sets a solid-color value to the matched view’s background.</td>
</tr>
<tr>
<td>font-size</td>
<td>fontSize</td>
<td>Sets the font size of the matched view (only supports device-independent units).</td>
</tr>
</tbody>
</table>
But what about OpenEdge? (experimental – you have been warned!)
@Component({
    selector: "checkbox",
    templateUrl: "checkbox.html"
});

<switch>

<input type="checkbox">
One syntax for all

**Property Binding:**  
[property]

**Event Binding:**  
(event)

**Intercepting input:**  
#idhandler

**Conditions:**  
*ngIf="expression"

**Loops:**  
*ngFor="expression"

**Styling:**  
[class.style1]="expression"
Angular 2 meets NativeScript
Vanilla NativeScript HTTP example

```javascript
var http = require( "http" );
http.getJSON( "https://api.myservice.com" )
 .then(function( result ) {
    // result is JSON Object
 });
```
NativeScript Vanilla

Destination

Implementation

Abstraction Layer

Native iOS call

http / XHR

<NS Module>

Native Android call

iOS App

Android App
Angular HTTP example

```typescript
@Injectable()
export class FooService {
    constructor(http: Http) {
        let foo = http.get('http://someurl/api/foo.json')
            .map(response => response.json());
    }
}
```
NativeScript Angular

Layer

Abstraction

HTTP
<Angular Component>

XMLHttpRequest
</Browser>

Implementation

Native
iOS call

Native
Android call

Ajax call

Destination

iOS App

Android App

Web App