



AWESOME NATIVE APPS WITH NATIVESCRIPT!

Rowdy Rabouw

- Freelance web and app developer
- Lead developer Nationale-Nederlanden Pension APP
- Progress Developer Expert for Nativescript
- Organizer NativeScript Developer Day Europe
- Curator nativescript.nl

NativeScript

NATIVE

NativeScript

- Open source framework for building truly native mobile apps with JavaScript
- together with markup (XML/HTML) and CSS
- and native code if you want and dare
- Cross Platform : one codebase for iOS and Android



Not like
PhoneGap/Cordova
with Ionic

Not like PhoneGap/Cordova with Ionic

- No WebView
- No DOM to manipulate
- No HTML elements styled like native components
- Real native components



Not like
Xamarin

Not like Xamarin

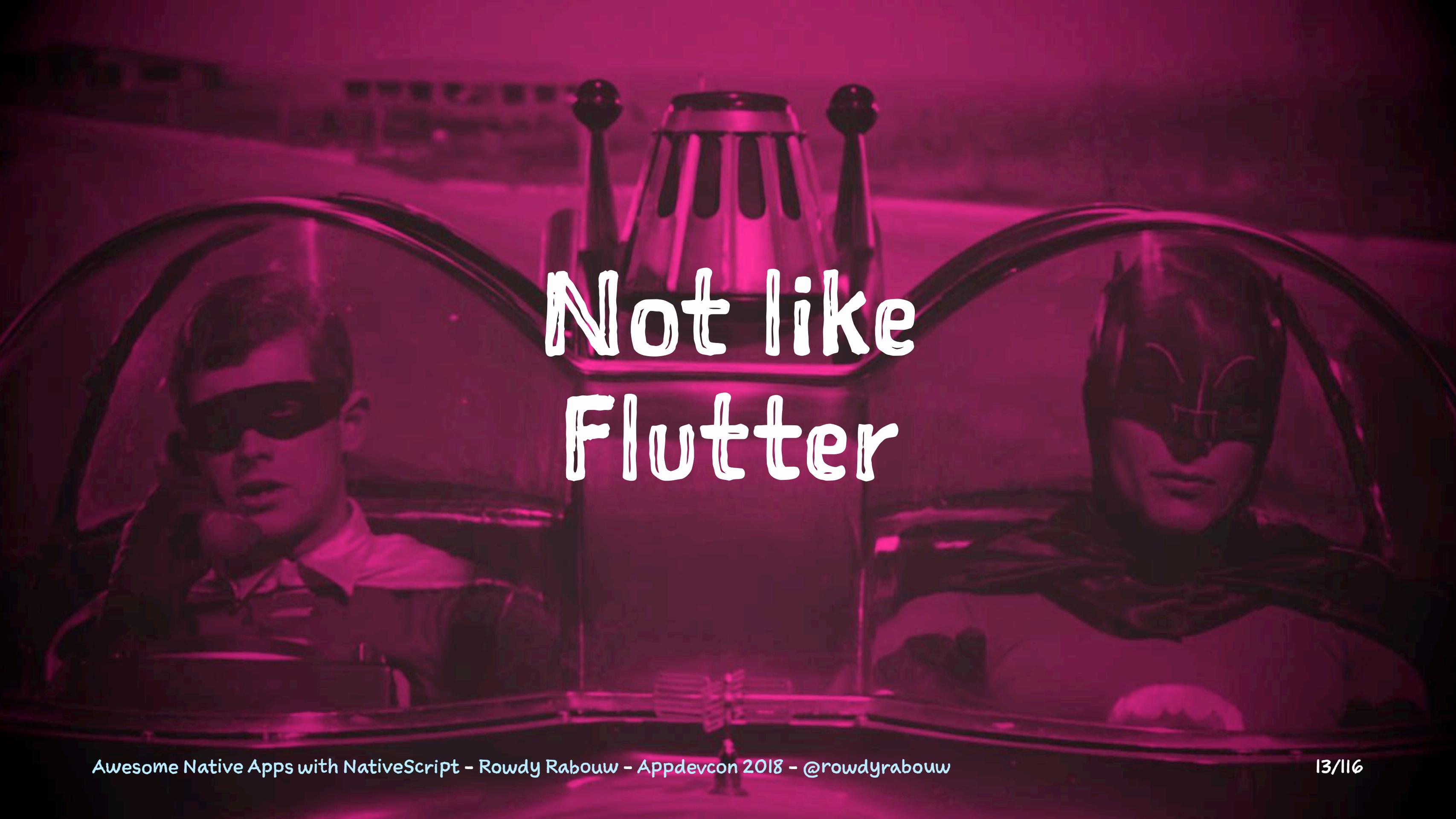
- No Cross Compiling
- 100% access to native APIs without writing bindings
- No .NET or C#



Not like React Native

Not like React Native

- Not bound to a specific framework
- No need writing ObjectiveC, Swift or Java to access native APIs
- {N} JavaScript has 100% access to native APIs
- Maturity: 0.54 vs 3.4 (4.0 in April)
- No Hot Module Reloading yet (expected in 4.2 in July)

A photograph of a man in a white shirt and dark trousers playing a traditional Indian tabla drum set. He is wearing a white cloth over his head. In the background, a woman in a pink top and patterned pants is seated, looking towards the camera. The setting appears to be a dimly lit room or stage.

Not like
Flutter

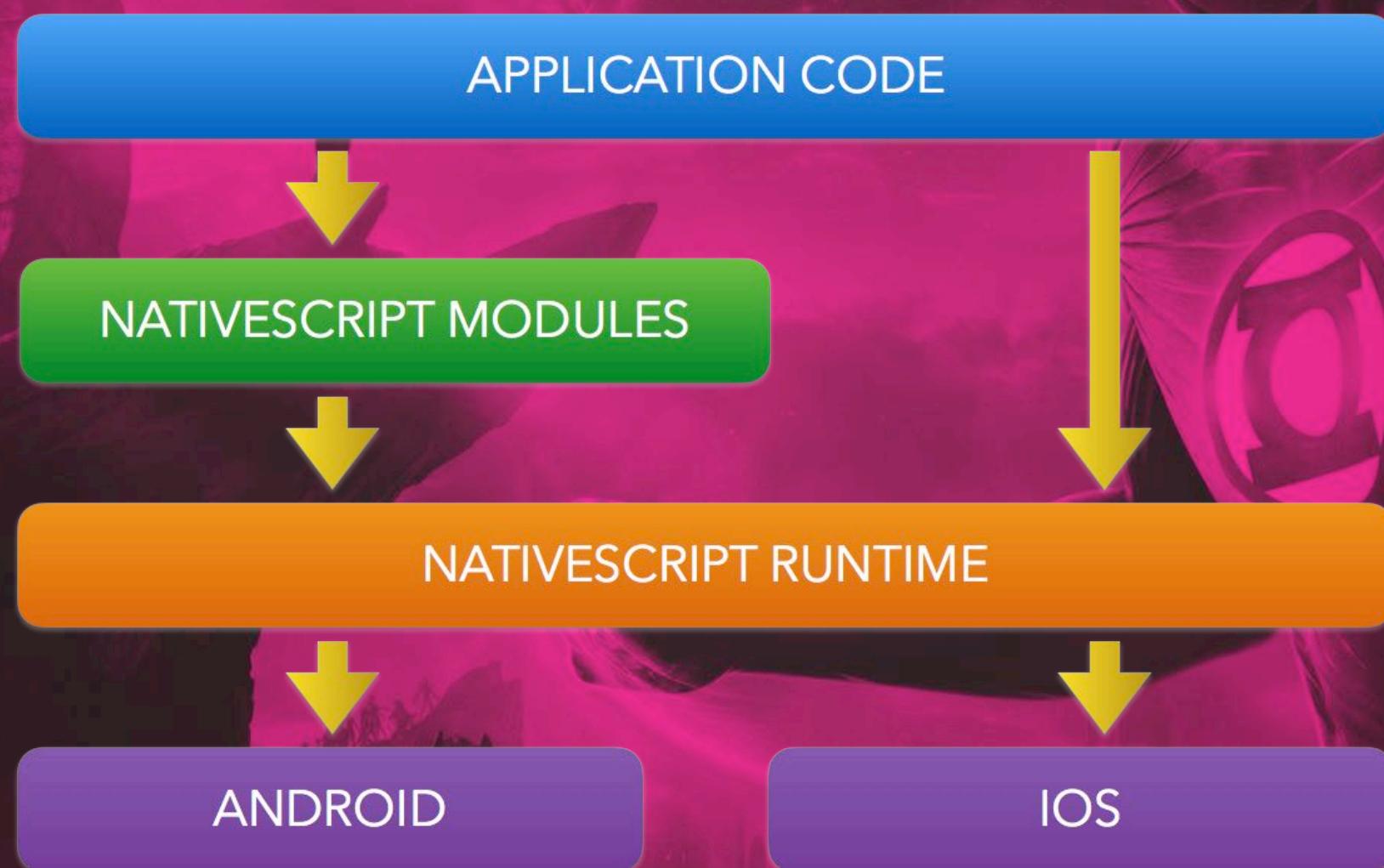
Not like Flutter

- Not bound to a specific language (Dart)
- No Cross Compiling
- Maturity: 0.2.1 vs 3.4 (4.0 in April)
- No Hot Module Reloading yet (expected in 4.2 in July)



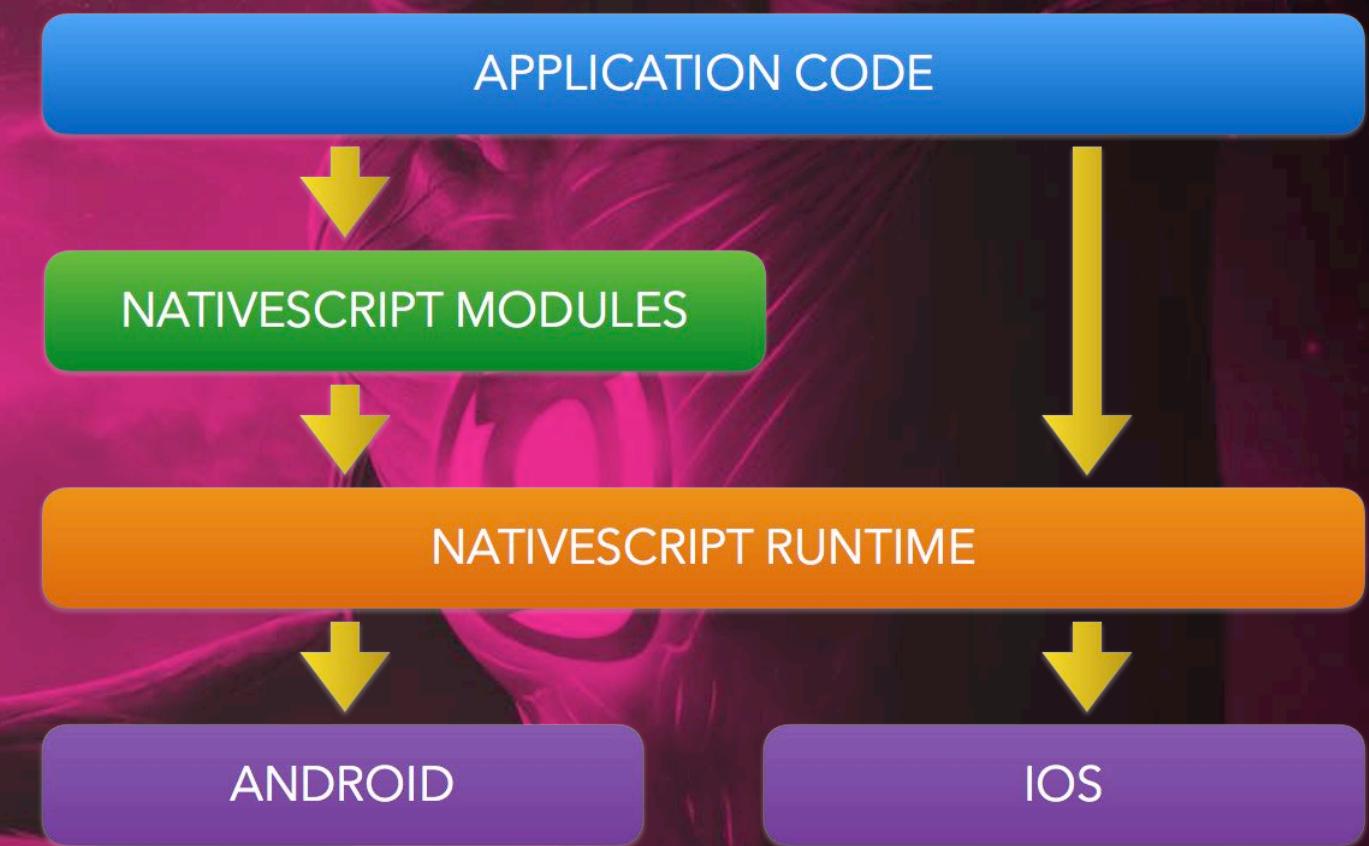
How does NativeScript work?

How does NativeScript work?



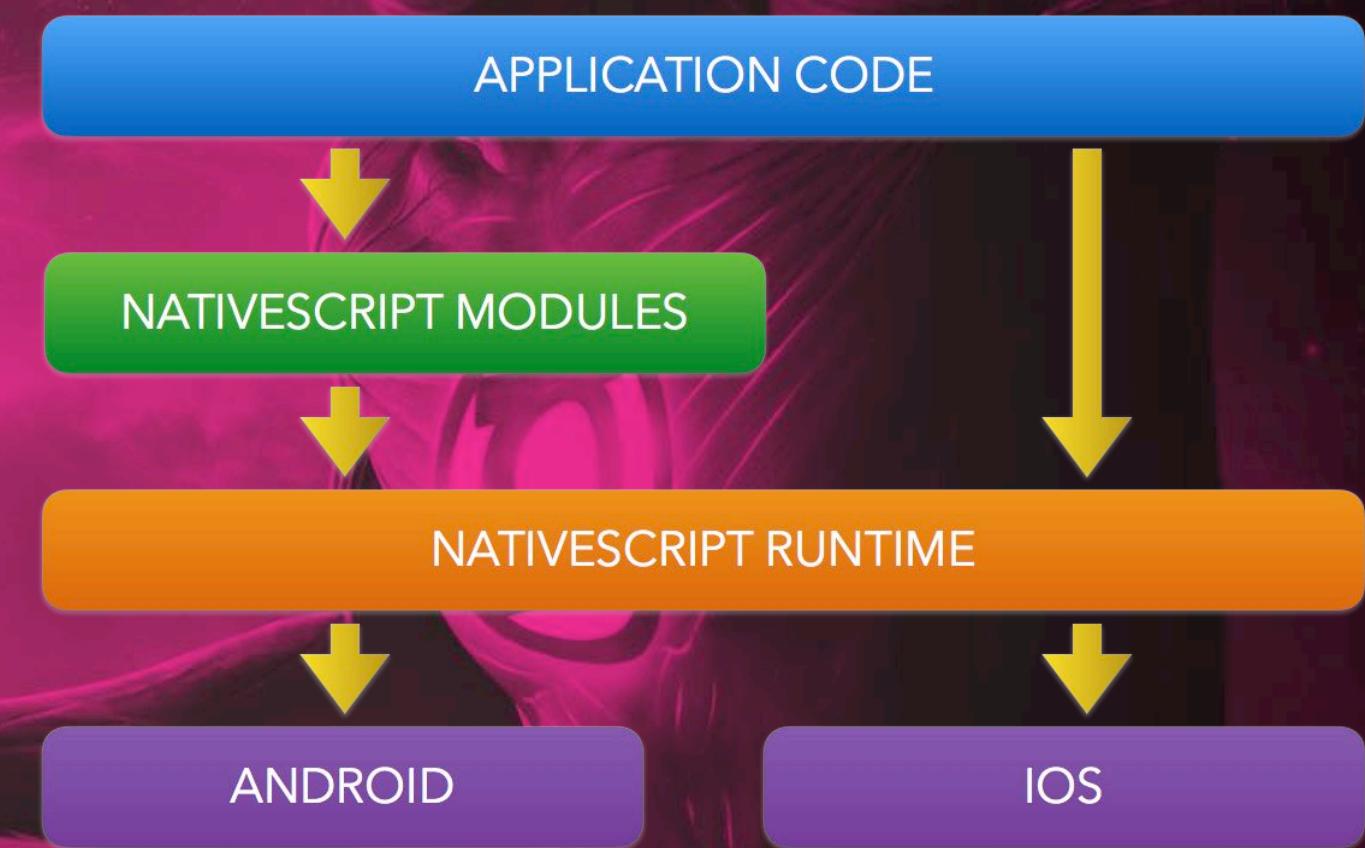
Application Code

- JavaScript code
- Page layouts in markup
- CSS for styling



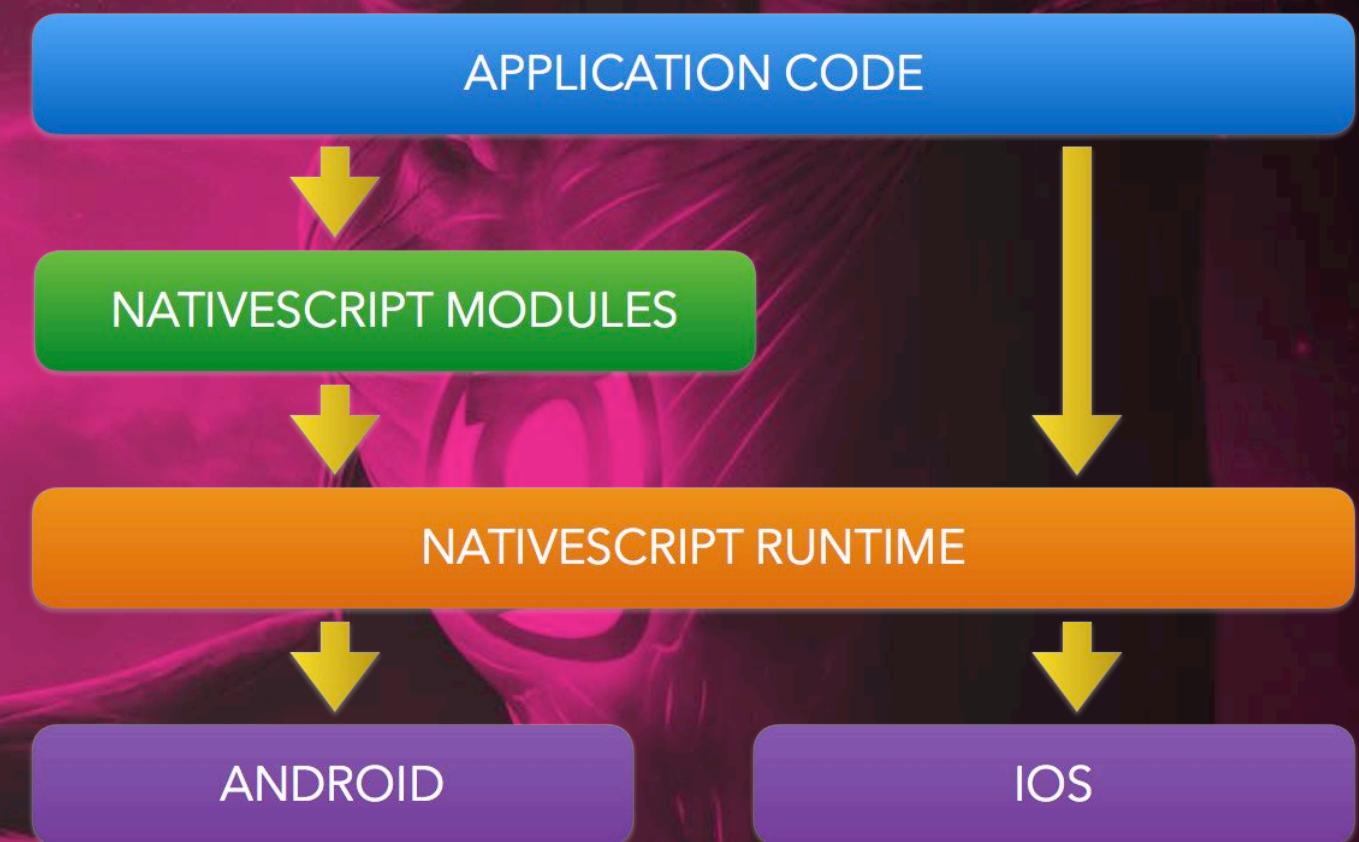
NativeScript Modules

- NativeScript Core
- UI components
- File system access
- Platform information
- NativeScript Plugins
 - Camera
 - Bluetooth
 - Fingerprint Authentication



NativeScript Runtime

- Runs on JavaScript Virtual Machines
 - V8 (Android)
 - JavaScriptCore (iOS)
- Executes C++ code to invoke native code
- Uses reflection to generate metadata
 - Examines, introspects, and modifies its own structure and behavior at runtime
- No separate binding layers between {N} and each mobile platform API
- New features are available immediately



A group of Justice League characters standing together in a dark, smoky environment. From left to right: Cyborg, Superman, Starro, Wonder Woman, Batman, and Aquaman. They are all in their iconic costumes, looking towards the camera.

JavaScript

JavaScript

JS

JavaScript

JS TS

JavaScript

JS TS V

JavaScript

JS

TS

Vue

Angular



Angular

Angular

- Opinionated JavaScript framework
- Typescript
- Full-featured routing
- Dependency injection
- Data binding

The NativeScript logo, consisting of the letters 'N' and 'S' stacked vertically, is displayed on a large red 3D cube. The cube is positioned in front of a person's face, which is partially visible and looking directly at the camera. The background is a dark, textured surface.

NativeScript



Performance Optimization

Performance Optimization

- Webpack
 - traverses your source tree via module imports
 - only include modules that are used
- Ahead-of-Time compilation
 - pre-compiles application components and templates
 - Angular compiler not included in bundle

Performance Optimization

- Uglify
 - code minification
 - reduces names of local variables
- Lazy-Load modules
 - not all modules are needed at startup
 - pre-load in background



Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS)

```
button,  
.btn,  
button[btn-type="primary"] {  
    height: 40;  
    font-size: 16;  
    color: rgb(197, 46, 54);  
    background-color: #f6c600;  
    text-transform: uppercase;  
    opacity: 0.5;  
}
```

Cascading Style Sheets (CSS)

- subset of CSS is supported
- device-independent pixels
- <https://docs.nativescript.org/ui/styling>

NativeScript Theme Core

Ready to use color schemes for iOS and Android



Syntactically Awesome Style Sheets

CSS preprocessor (nested rules, functions, ...)

Sass

A woman with long blonde hair, wearing a red dress, stands in a field with mountains in the background. She is looking directly at the camera. The image has a slightly faded, warm-toned effect.

Native elements with markup

Native elements with markup

```
<ActionBar title="Native elements"></ActionBar>

<StackLayout>
    <Button text="Button" (tap)="onButtonTap()"></Button>
    <Switch checked="false"></Switch>
    <SegmentedBar [items]="segmentedBarItemItems"></SegmentedBar>
    <Progress value="0" maxValue="100"></Progress>
    <Slider value="0" minValue="0" maxValue="100"></Slider>
    <DatePicker year="2018" month="1" day="1"
                minDate="1970-01-01" maxDate="2100-12-31"></DatePicker>
</StackLayout>
```

Native elements with markup

```
<Button text="Button" (tap)="onButtonTap()"></Button>
```

```
onButtonTap() {  
    let options = {  
        title: "Superheroes",  
        message: "Choose your favorite",  
        cancelButtonTitle: "Cancel",  
        actions: ["Wonder Woman", "Wolverine", "Black Widow", "Superman"]  
    };  
  
    action(options).then(result) => {  
        alert(result);  
    });  
}
```



Native elements with markup

```
<SegmentedBar [items]="segmentedBarItems"></SegmentedBar>
```

```
import { SegmentedBar, SegmentedBarItem } from "ui/segmented-bar";
```

```
private getSegmentedBarItems = () => {
    let segmentedBarItem1 = new SegmentedBarItem();
    segmentedBarItem1.title = "Item 1";
    let segmentedBarItem2 = new SegmentedBarItem();
    segmentedBarItem2.title = "Item 2";
    let segmentedBarItem3 = new SegmentedBarItem();
    segmentedBarItem3.title = "Item 3";
    return [segmentedBarItem1, segmentedBarItem2, segmentedBarItem3];
}
```

```
segmentedBarItems: Array<SegmentedBarItem> = this.getSegmentedBarItems();
```

Native elements

BUTTON



ITEM 1

ITEM 2

ITEM 3



31

Dec

2017

01

Jan

2018

02

Feb

2019

Native elements

Button



Item 1

Item 2

Item 3



29

October

2015

30

November

2016

31

December

2017

1

January

2018

2

February

2019

3

March

2020

4

April

2021

A full-body photograph of a person wearing a dark spacesuit with a reflective visor, floating in the void of space. The person's arms are slightly outstretched. In the background, a large, hazy blue and white planet dominates the center, with numerous small, glowing stars scattered across the dark expanse.

TextField

Text Field

```
<TextField></TextField>
```

```
<TextField text=""></TextField>
```

Text Field: hint

```
<TextField></TextField>
```

```
<TextField text=""></TextField>
```

```
<TextField hint="Enter your name"></TextField>
```

Text Field: autocapitalization

```
<!-- Capitalize all characters automatically -->
```

```
<TextField autocapitalizationType="allCharacters"></TextField>
```

```
<!-- Capitalize the first letter of each sentence automatically; default -->
```

```
<TextField autocapitalizationType="sentences"></TextField>
```

```
<!-- Capitalize the first letter of each word automatically -->
```

```
<TextField autocapitalizationType="words"></TextField>
```

Text Field: autocapitalization

```
<!-- Capitalize all characters automatically -->
```

```
<TextField autocapitalizationType="allCharacters"></TextField>
```

```
<!-- Capitalize the first letter of each sentence automatically; default -->
```

```
<TextField autocapitalizationType="sentences"></TextField>
```

```
<!-- Capitalize the first letter of each word automatically -->
```

```
<TextField autocapitalizationType="words"></TextField>
```

```
<!-- Do not capitalize any text automatically -->
```

```
<TextField autocapitalizationType="none"></TextField>
```

Text Field: autocorrect

```
<!-- Enables autocorrection; default -->
```

```
<TextField autocorrect="true"></TextField>
```

```
<!-- Disables autocorrection -->
```

```
<TextField autocorrect="false"></TextField>
```

Text Field: keyboardType

<!-- Sets the soft keyboard type -->

```
<TextField keyboardType="number"></TextField>
<TextField keyboardType="datetime"></TextField>
<TextField keyboardType="phone"></TextField>

<TextField keyboardType="email"></TextField>

<TextField keyboardType="url"></TextField>
```

keyboardType

datetime

number

phone

email

url

default

keyboardType

datetime

number

phone

email

url

default

Text Field: more attributes

```
<!-- Sets text-alignment style property -->  
<TextField textAlignment=""></TextField>
```

```
<!-- Sets the visibility of the view -->  
<TextField visibility=""></TextField>
```

```
<!-- Sets the desired width of the view -->  
<TextField width=""></TextField>
```

```
<!-- Limits input to a certain number of characters -->  
<TextField maxLength=""></TextField>
```

Layouts

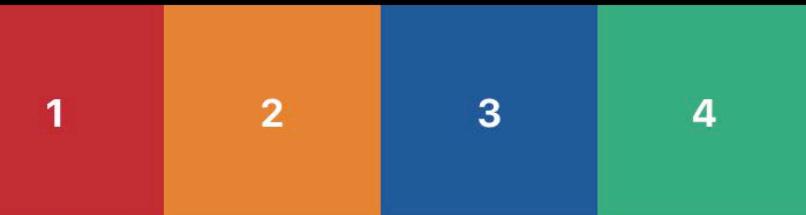
StackLayout

```
<StackLayout orientation="horizontal">
    <Label text="1"></Label>
    <Label text="2"></Label>
    <Label text="3"></Label>
    <Label text="4"></Label>
</StackLayout>
```

StackLayout

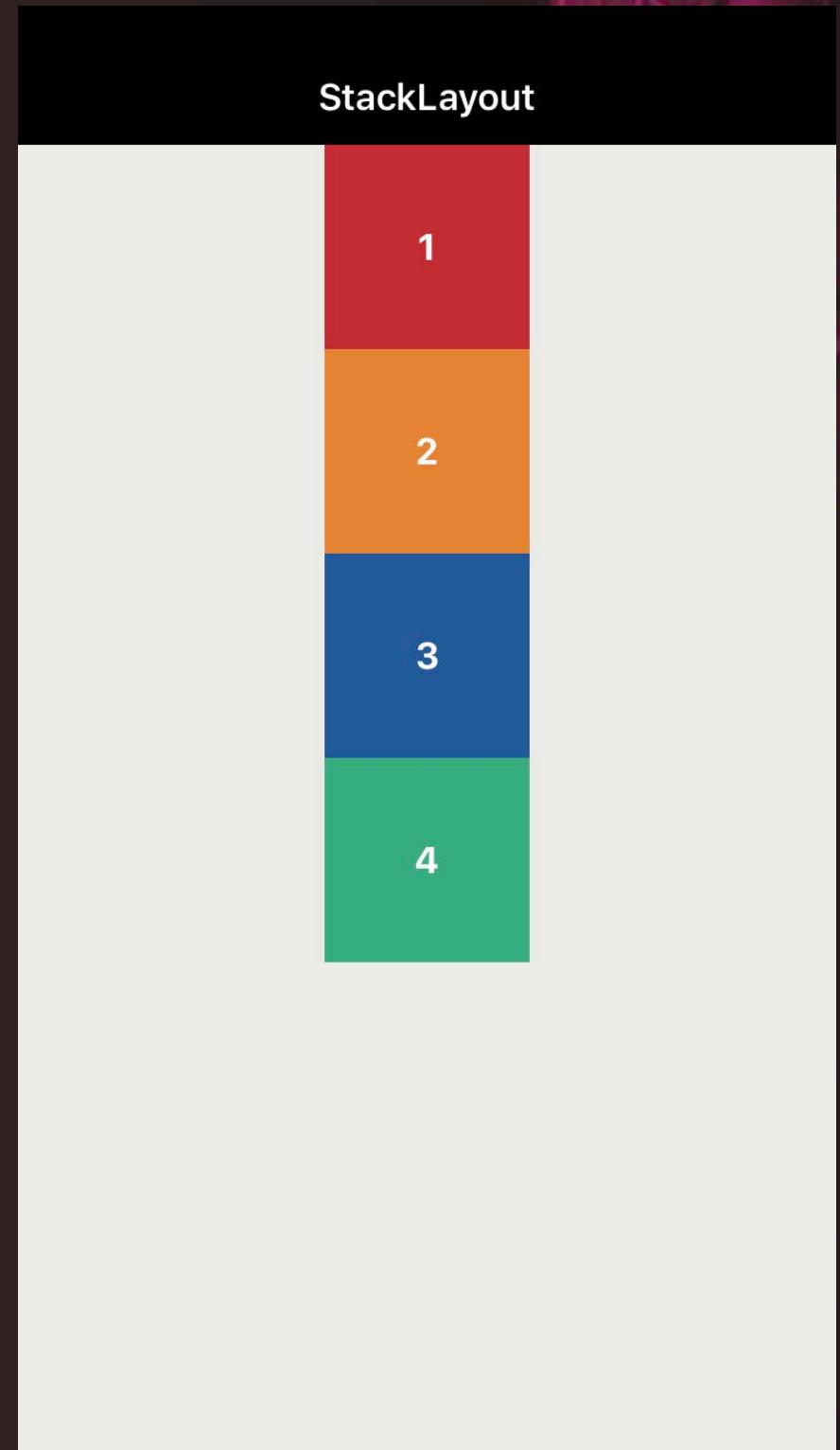
```
<StackLayout orientation="horizontal">
    <Label text="1"></Label>
    <Label text="2"></Label>
    <Label text="3"></Label>
    <Label text="4"></Label>
</StackLayout>
```

StackLayout orientation="horizontal"



StackLayout

```
<StackLayout>
    <Label text="1"></Label>
    <Label text="2"></Label>
    <Label text="3"></Label>
    <Label text="4"></Label>
</StackLayout>
```

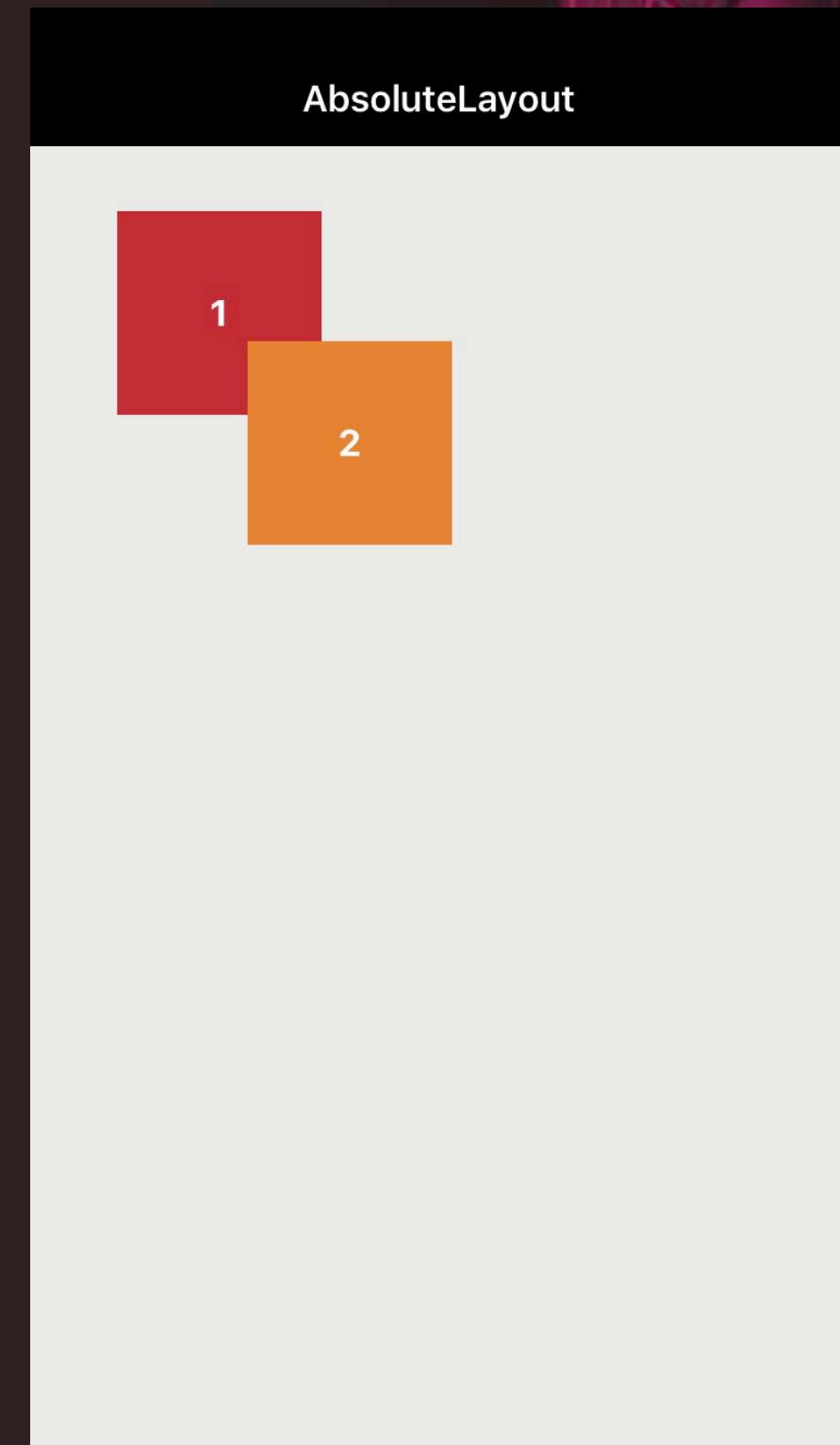


AbsoluteLayout

```
<AbsoluteLayout>
    <Label text="1" left="40" top="30"></Label>
    <Label text="2" left="100" top="90"></Label>
</AbsoluteLayout>
```

AbsoluteLayout

```
<AbsoluteLayout>
  <Label text="1" left="40" top="30"></Label>
  <Label text="2" left="100" top="90"></Label>
</AbsoluteLayout>
```

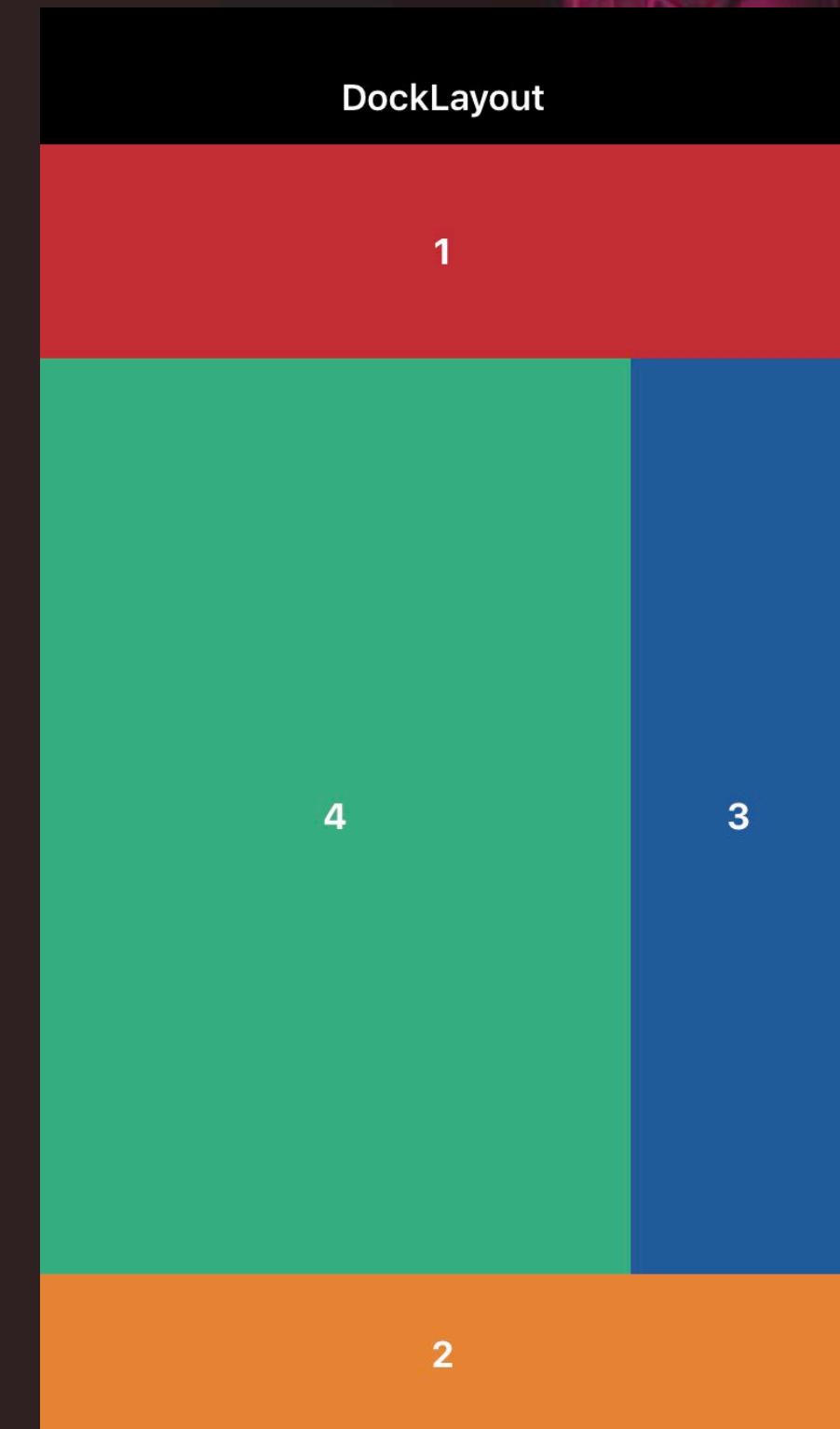


DockLayout

```
<DockLayout height="100%" stretchLastChild="true">
    <Label text="1" dock="top" height="100"></Label>
    <Label text="2" dock="bottom" height="75"></Label>
    <Label text="3" dock="right" width="100"></Label>
    <Label text="4" dock="left"></Label>
</DockLayout>
```

DockLayout

```
<DockLayout height="100%" stretchLastChild="true">
    <Label text="1" dock="top" height="100"></Label>
    <Label text="2" dock="bottom" height="75"></Label>
    <Label text="3" dock="right" width="100"></Label>
    <Label text="4" dock="left"></Label>
</DockLayout>
```

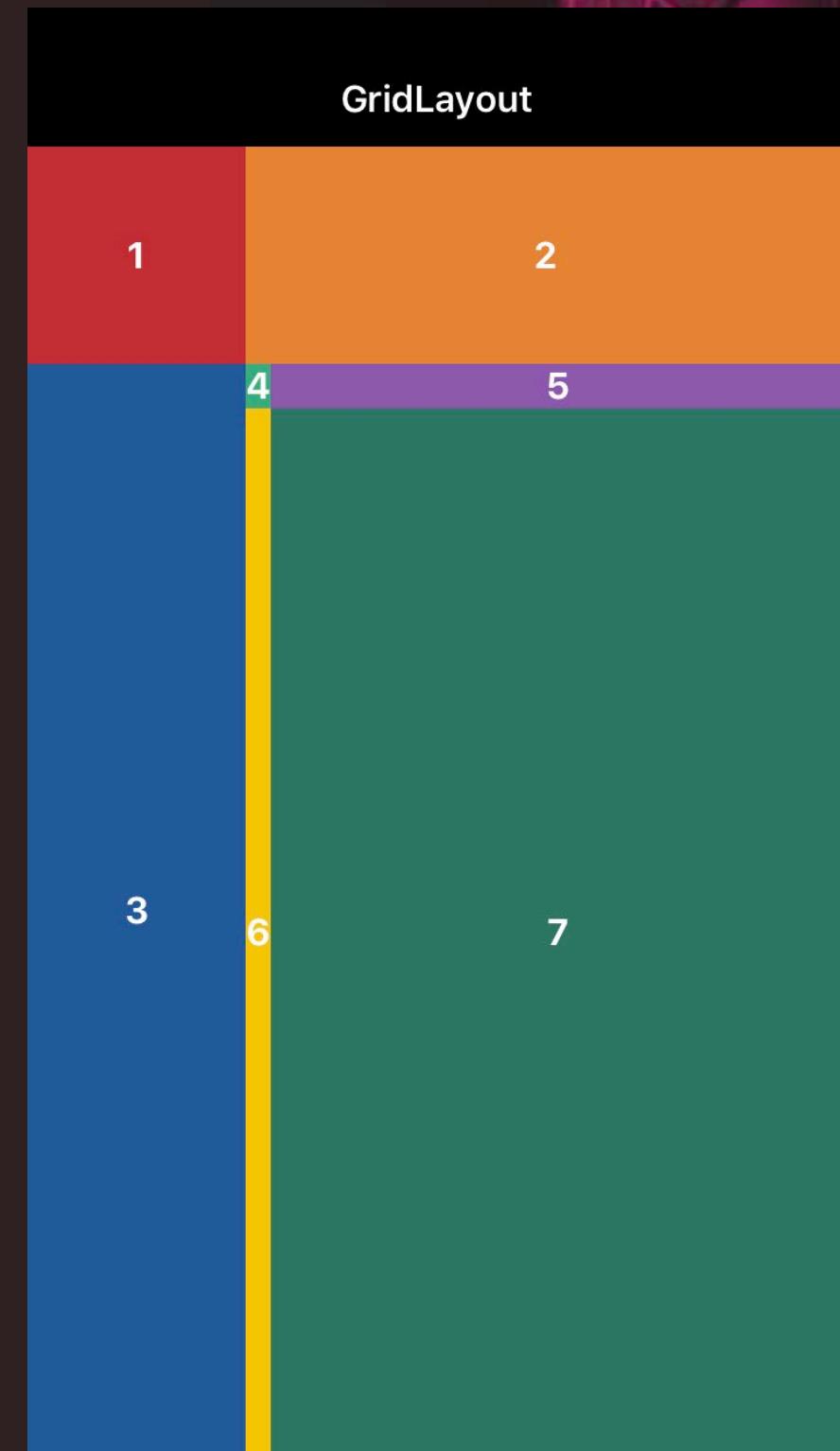


GridLayout

```
<GridLayout  
    columns="100, auto, *" rows="100, auto, *"  
    >  
    <Label text="1" row="0" col="0"></Label>  
    <Label text="2" row="0" col="1" colSpan="2"></Label>  
    <Label text="3" row="1" col="0" rowspan="2"></Label>  
    <Label text="4" row="1" col="1"></Label>  
    <Label text="5" row="1" col="2"></Label>  
    <Label text="6" row="2" col="1"></Label>  
    <Label text="7" row="2" col="2"></Label>  
</GridLayout>
```

GridLayout

```
<GridLayout  
    columns="100, auto, *" rows="100, auto, *"  
    <Label text="1" row="0" col="0"></Label>  
    <Label text="2" row="0" col="1" colSpan="2"></Label>  
    <Label text="3" row="1" col="0" rowSpan="2"></Label>  
    <Label text="4" row="1" col="1"></Label>  
    <Label text="5" row="1" col="2"></Label>  
    <Label text="6" row="2" col="1"></Label>  
    <Label text="7" row="2" col="2"></Label>  
</GridLayout>
```



WrapLayout

```
<WrapLayout orientation="vertical">
    <Label text="1"></Label>
    <Label text="2"></Label>
    <Label text="3"></Label>
    <Label text="4"></Label>
</WrapLayout>
```

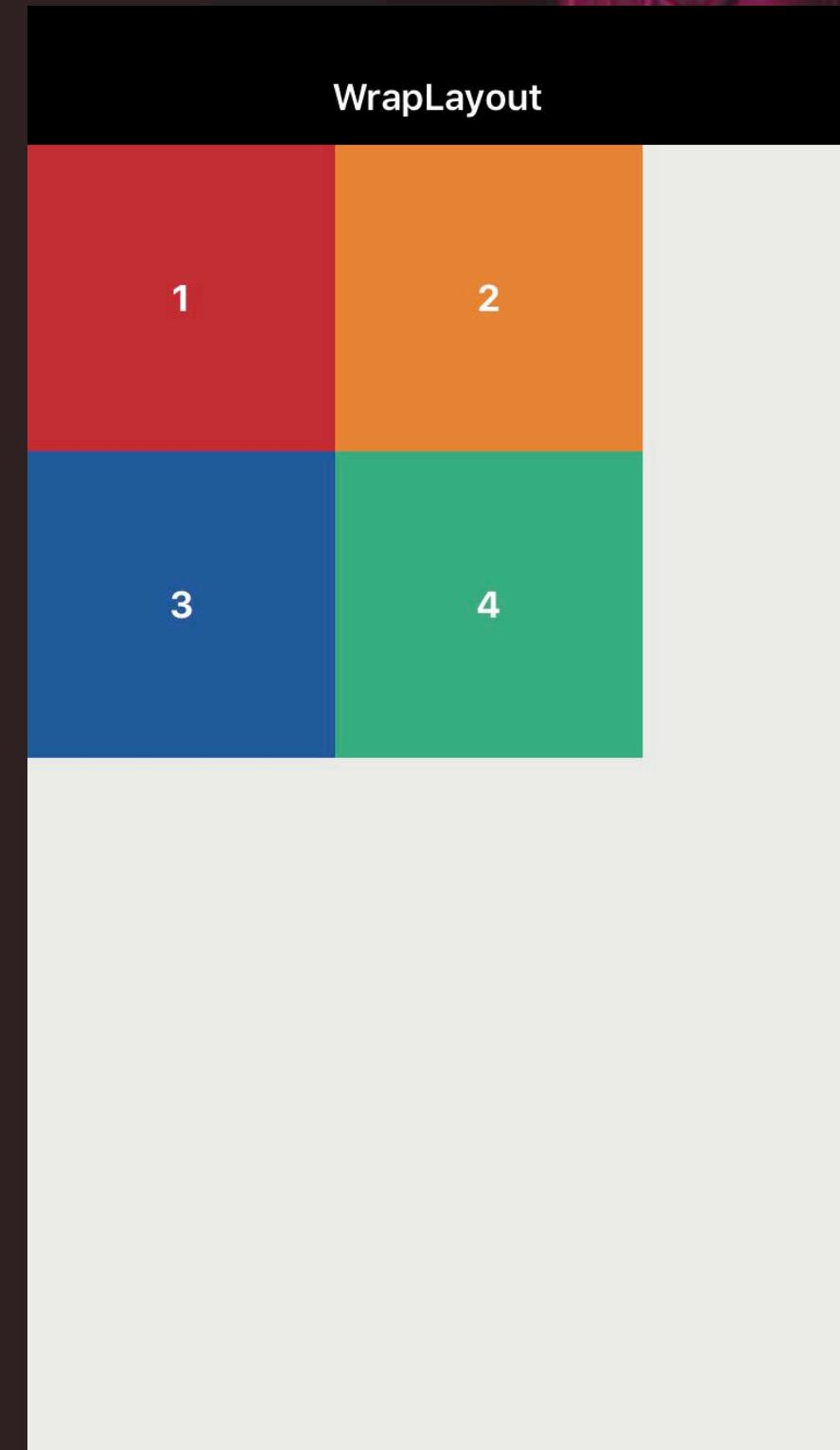
WrapLayout

```
<WrapLayout orientation="vertical">  
  <Label text="1"></Label>  
  <Label text="2"></Label>  
  <Label text="3"></Label>  
  <Label text="4"></Label>  
</WrapLayout>
```



WrapLayout

```
<WrapLayout>
    <Label text="1"></Label>
    <Label text="2"></Label>
    <Label text="3"></Label>
    <Label text="4"></Label>
</WrapLayout>
```



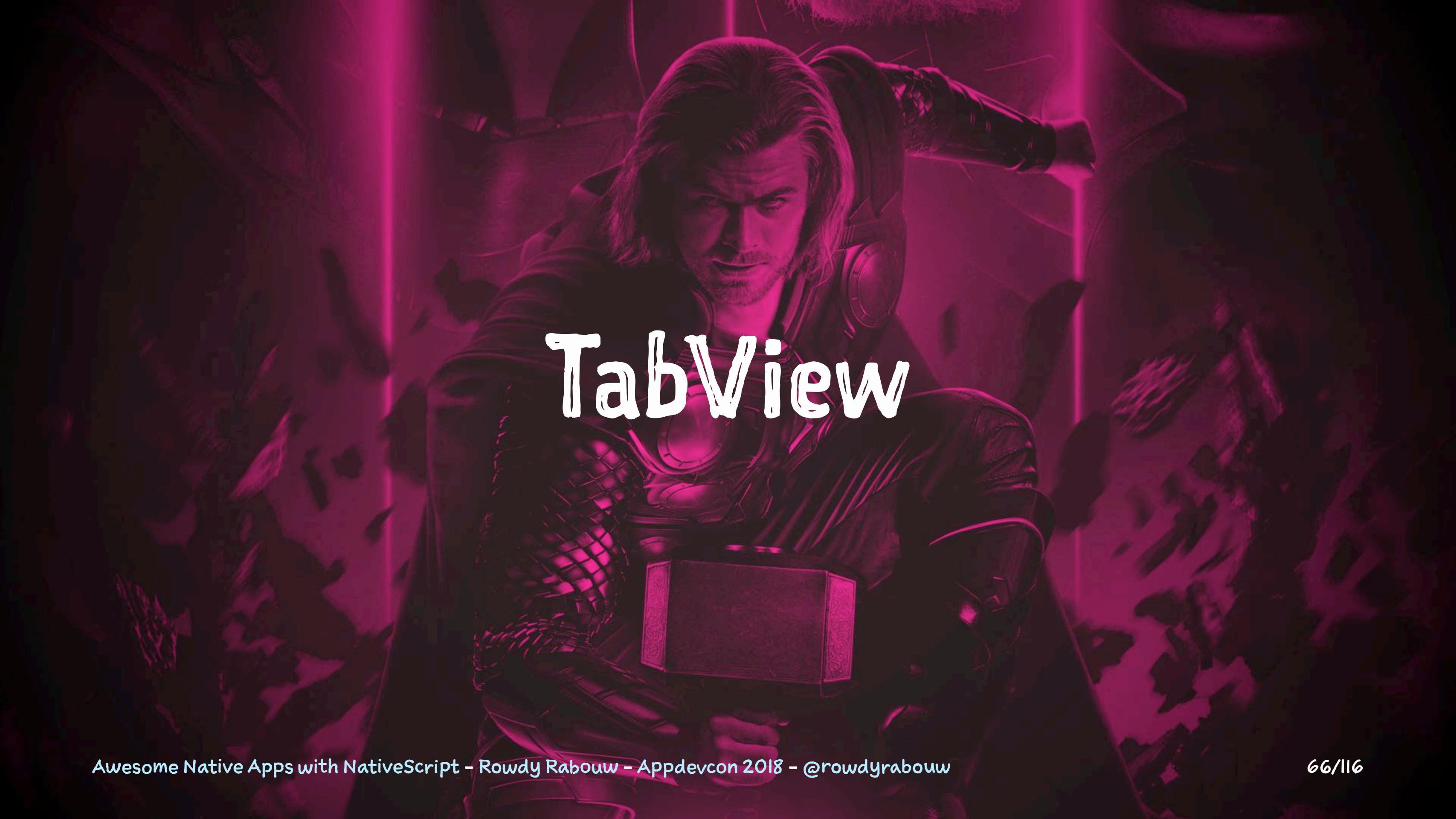
FlexboxLayout

```
<FlexboxLayout flexDirection="row-reverse">
  <Label text="1" flexGrow="1"></Label>
  <Label text="2" flexGrow="2"></Label>
  <Label text="3" flexGrow="3"></Label>
  <Label text="4" flexGrow="4"></Label>
  <Label text="5" flexGrow="5"></Label>
  <Label text="6" flexGrow="6"></Label>
</FlexboxLayout>
```

FlexboxLayout

```
<FlexboxLayout flexDirection="row-reverse">
  <Label text="1" flexGrow="1"></Label>
  <Label text="2" flexGrow="2"></Label>
  <Label text="3" flexGrow="3"></Label>
  <Label text="4" flexGrow="4"></Label>
  <Label text="5" flexGrow="5"></Label>
  <Label text="6" flexGrow="6"></Label>
</FlexboxLayout>
```



A dramatic, low-key lighting photograph of Thor from the Marvel Cinematic Universe. He is shown from the waist up, wearing his iconic hammer-shaped胸甲 (thorax plate) and gauntlets. His long blonde hair flows down his back. He has a serious, intense expression, looking directly at the viewer. The background is dark and textured, suggesting a metallic or rocky environment.

TabView

TabView

```
<TabView>
  <StackLayout *tabItem="{title: 'Tab 1'}">
    <!-- Tab 1 content -->
  </StackLayout>
  <StackLayout *tabItem="{title: 'Tab 2'}">
    <!-- Tab 2 content -->
  </StackLayout>
  <StackLayout *tabItem="{title: 'Tab 3'}">
    <!-- Tab 3 content -->
  </StackLayout>
</TabView>
```

TabView

```
<TabView height="100%">
  <StackLayout *tabItem="{title: 'Rocket Raccoon'}" class="full rocket">
    </StackLayout>
  <StackLayout *tabItem="{title: 'Harley Quinn'}" class="full harley">
    </StackLayout>
  <StackLayout *tabItem="{title: 'Hulk'}" class="full hulk">
    </StackLayout>
</TabView>
```

TabView

```
.full {  
    background-size: cover;  
    background-position: center;  
    background-repeat: no-repeat;  
}  
  
.rocket {  
    background-image: url("~/images/rocket-raccoon.jpg");  
}  
  
.harley {  
    background-image: url("~/images/harley-quinn.jpg");  
}  
  
.hulk {  
    background-image: url("~/images/hulk.jpg");  
}
```

TabView



ROCKET
RACCOON

HARLEY QUINN

HULK

TabView



Rocket Raccoon

Harley Quinn

Hulk

A group of diverse people, including men and women of various ethnicities, are gathered around a white tablet device. They are all looking intently at the screen, which displays a NativeScript application. The background is a dark, modern interior space.

Native code

slider.component.html

```
<ActionBar title="Slider"></ActionBar>

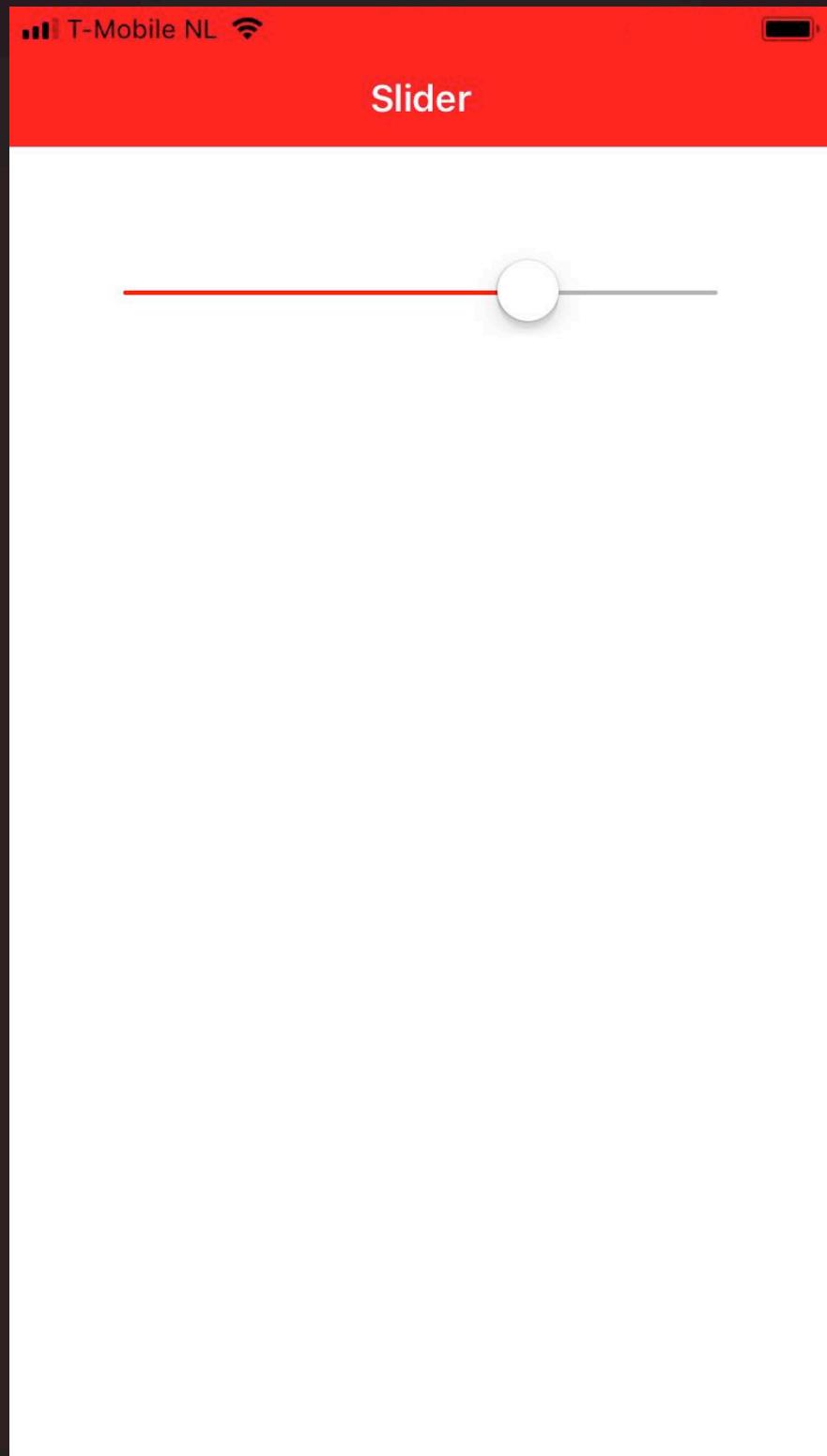
<StackLayout>
    <Slider value="70"></Slider>
</StackLayout>
```

slider.component.ts

```
import { Component } from "@angular/core";  
  
@Component({  
    selector: "app-slider",  
    moduleId: module.id,  
    templateUrl: "slider.component.html",  
    styleUrls: ["slider.component.css"]  
})  
export class SliderComponent {  
    constructor() {}  
}
```

slider.component.css

```
ActionBar {  
    color: white;  
    background-color: red;  
}  
  
StackLayout {  
    padding: 50;  
}  
  
Slider {  
    background-color: red;  
}
```



slider.component.html

```
<ActionBar title="Slider"></ActionBar>

<StackLayout>
    <Slider slider-icon value="70"></Slider>
</StackLayout>
```

- attribute directive
- changes the appearance or behavior of an element

slider.directive.ts

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "platform";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      let uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

slider.directive.ts

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "platform";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      let uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

slider.directive.ts

```
import { Directive, ElementRef } from "@angular/core";
import { isIOS } from "platform";

@Directive({
  selector: "[slider-icon]"
})
export class SliderIconDirective {
  constructor(private el: ElementRef) {
    if (isIOS) {
      let uiSlider = this.el.nativeElement.ios;
      uiSlider.setThumbImageForState(
        UIImage.imageNamed("image.png"), UIControlState.Normal);
    }
  }
}
```

slider.module.ts

```
import { NgModule, NO_ERRORS_SCHEMA } from "@angular/core";
import { NativeScriptModule } from "nativescript-angular/nativescript.module";

import { SliderIconDirective } from "./slider.directive";

import { SliderRoutingModule } from "./slider-routing.module";
import { SliderComponent } from "./slider.component";

@NgModule({
    imports: [NativeScriptModule, SliderRoutingModule],
    declarations: [SliderComponent, SliderIconDirective],
    schemas: [NO_ERRORS_SCHEMA]
})
export class SliderModule {}
```



Demo

A group of four people (three men and one woman) are standing behind a large, muscular man in a tattered suit. The man in the suit has a textured, almost stone-like or metallic appearance. He is flexing his right arm, showing a large bicep. The background is dark and smoky.

Node Package Manager

Node Package Manager

- commonly known as npm
- ready to use JavaScript modules
- almost 500.000 packages of free, reusable code

Android Arsenal

→ libraries for Android (Java / Kotlin)

Cocoapods

→ libraries for iOS (Objective-C / Swift)

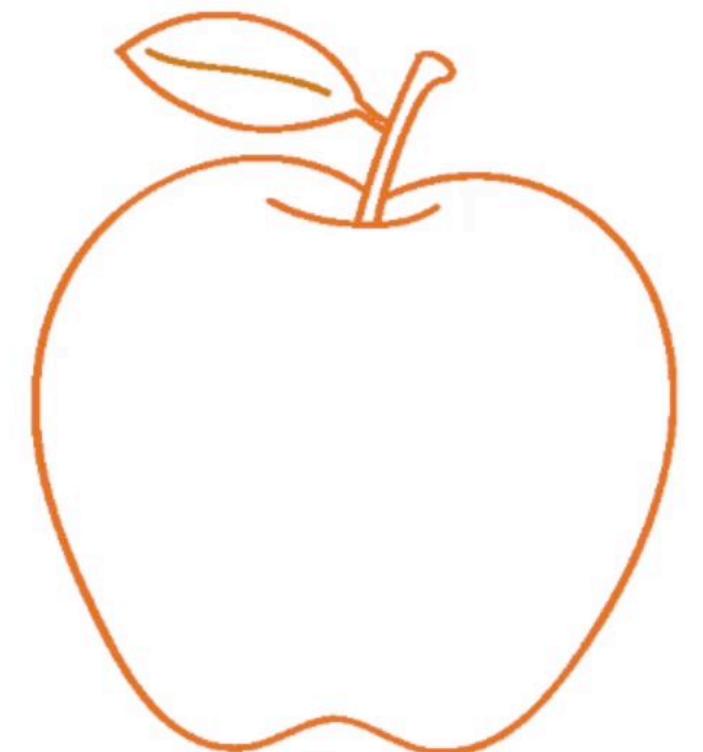
A medium shot of a man with curly hair, wearing a dark t-shirt, sitting at a wooden table. He is looking down at a laptop screen. In front of him is a small potted plant. The background is a brick wall.

Multilingual with ngx-translate

Multilingual with ngx-translate

- internationalization library for Angular 2+
- define translations in different languages
- switch between them easily
- no hardcoded text/labels, all in one place
- start directly, even with one language

Mijn Inkomen Later



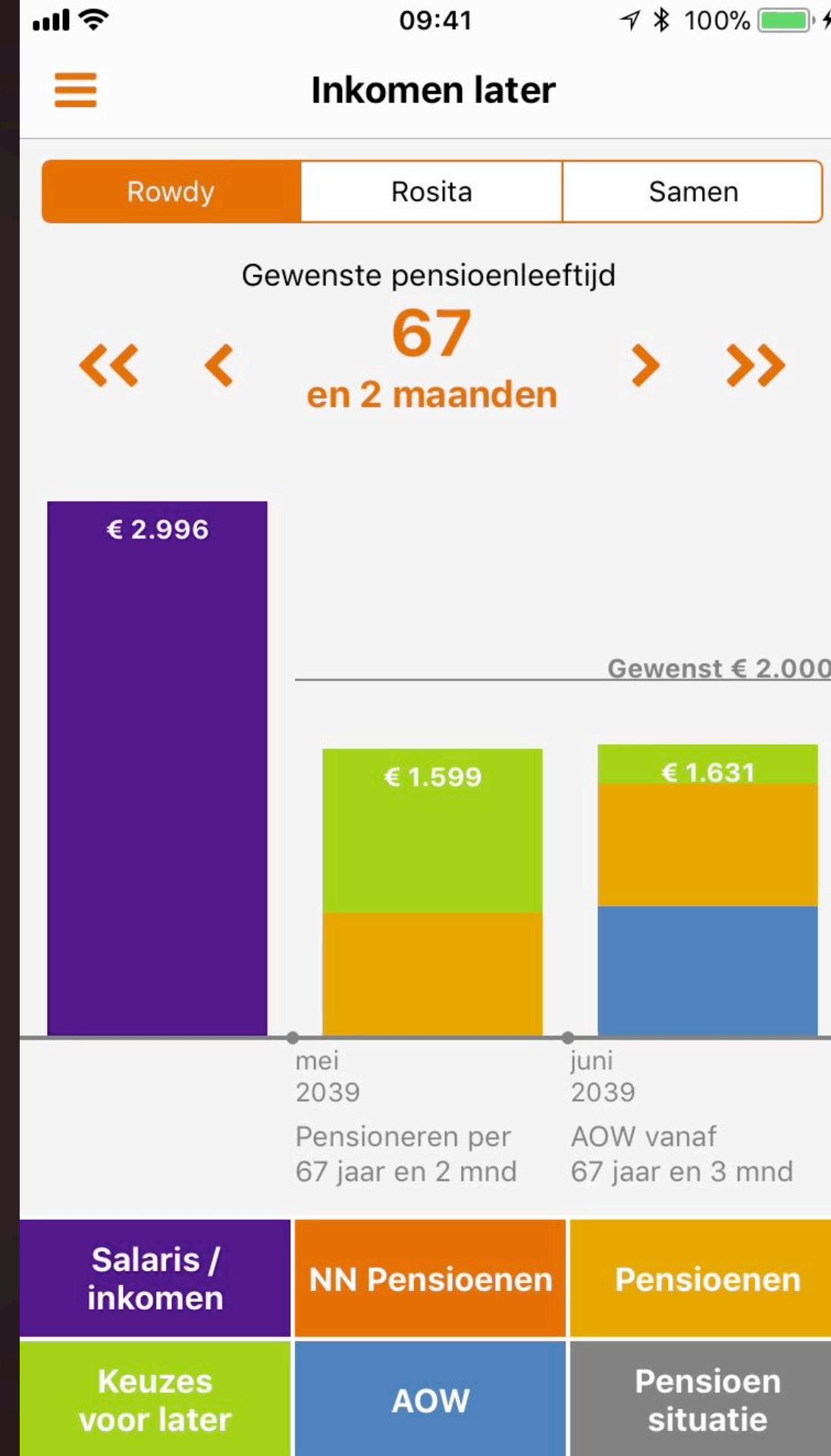
POWERED BY



nationale
nederlanden

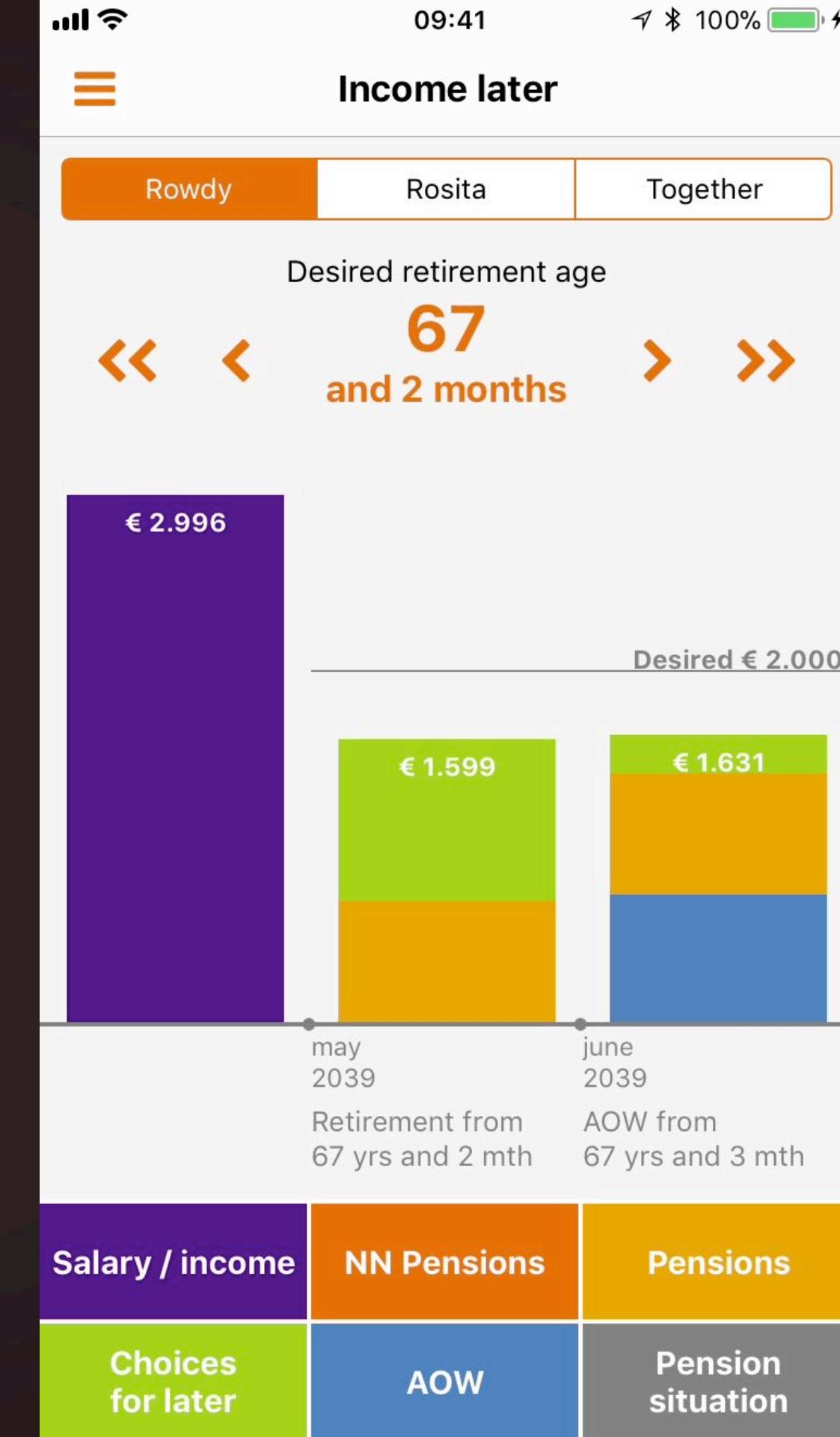
nl.json

```
{  
  "GRAPH": {  
    "PAGETITLE": "Inkomen later",  
    "HDRTXT": "Gewenste pensioenleeftijd",  
    "BTN_01": "Salaris / inkomen",  
    "BTN_02": "NN Pensioenen",  
    "BTN_03": "Pensioenen",  
    "BTN_04": "Keuzes voor later",  
    "BTN_05": "AOW",  
    "BTN_06": "Pensioen situatie"  
  }  
}
```



en.json

```
{  
  "GRAPH": {  
    "PAGETITLE": "Income later",  
    "HDRTXT": "Desired retirement age",  
    "BTN_01": "Salary / income",  
    "BTN_02": "NN Pensions",  
    "BTN_03": "Pensions",  
    "BTN_04": "Choices for later",  
    "BTN_05": "AOW",  
    "BTN_06": "Pension situation"  
  }  
}
```



Multilingual with ngx-translate

The screenshot shows a software interface for managing multilingual translations. On the left, a sidebar titled "Translation IDs" lists categories like "GRAPH", "BTN_01" through "BTN_06", "HDRTXT", and "PAGETITLE". The main area displays a specific entry for "GRAPH.PAGETITLE". It shows two language entries: "en-US" with the value "Income later" and "nl-NL" with the value "Inkommen later". Both entries have a checked "Approved" checkbox to their right.

BabelEdit by Andreas Löw - @CodeAndWeb
<https://www.codeandweb.com/babeledit>

Multilingual with ngx-translate

```
<Label [text]="'GRAPH.HDRTXT' | translate"></Label>
```

[] = one way data binding in Angular

| = display-value transformations



Inkomen later

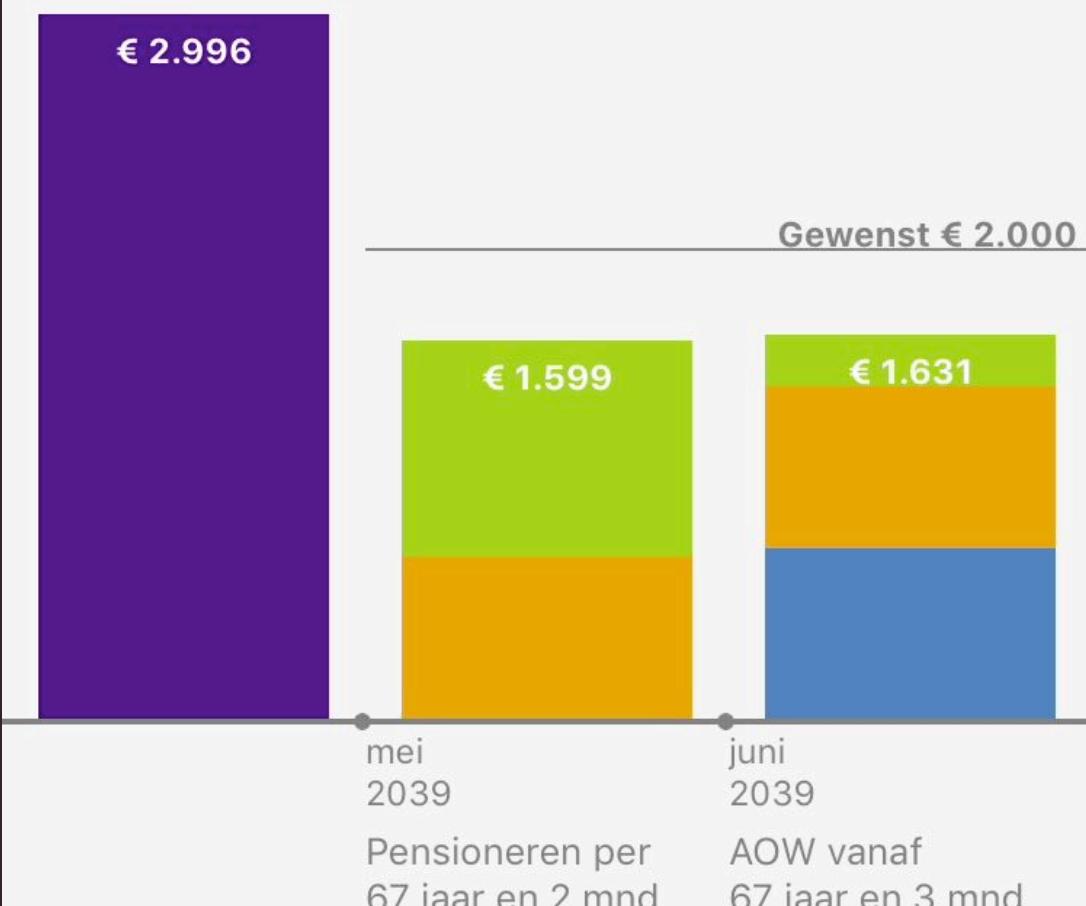
Rowdy

Rosita

Samen

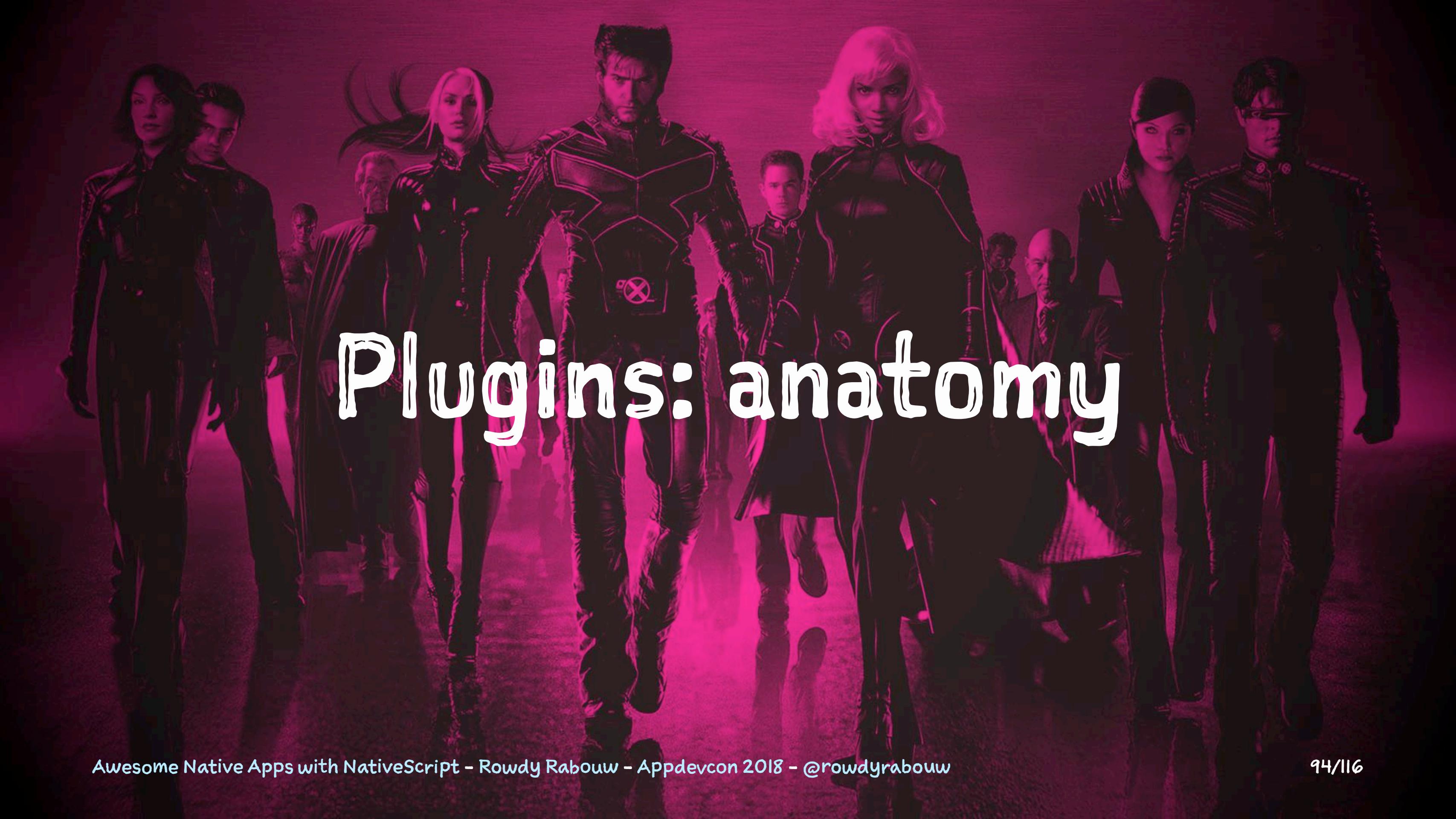
Gewenste pensioenleeftijd

67
en 2 maanden



A group of X-Men characters standing in a dark, dramatic setting. From left to right: Jean Grey, Cyclops, Emma Frost, Wolverine, Iceman, Magneto, Storm, Bishop, Professor X, and Cyclops again. The background is dark with some glowing elements.

Plugins

A group of Marvel X-Men characters standing together in a dark, dramatic setting. From left to right, the characters include: Anna (Wanda Maximoff), Cyclops (Scott Summers), Jean Grey (Phoenix), Wolverine (Logan), Beast (Kurt Wagner), Iceman (Bobby Drake), Emma Frost (White Queen), Gambit (Remy LeBeau), Bishop (Peter Rasputin), Magneto (Erik Lehnsherr), and Colossus (Raven).

Plugins: anatomy

version-number.ios.ts

```
export class VersionNumber {
    get() {
        let version = NSBundle mainBundle objectForInfoDictionaryKey("CFBundleShortVersionString");
        return version;
    }
}
```

version-number.android.ts

```
import * as application from "application";

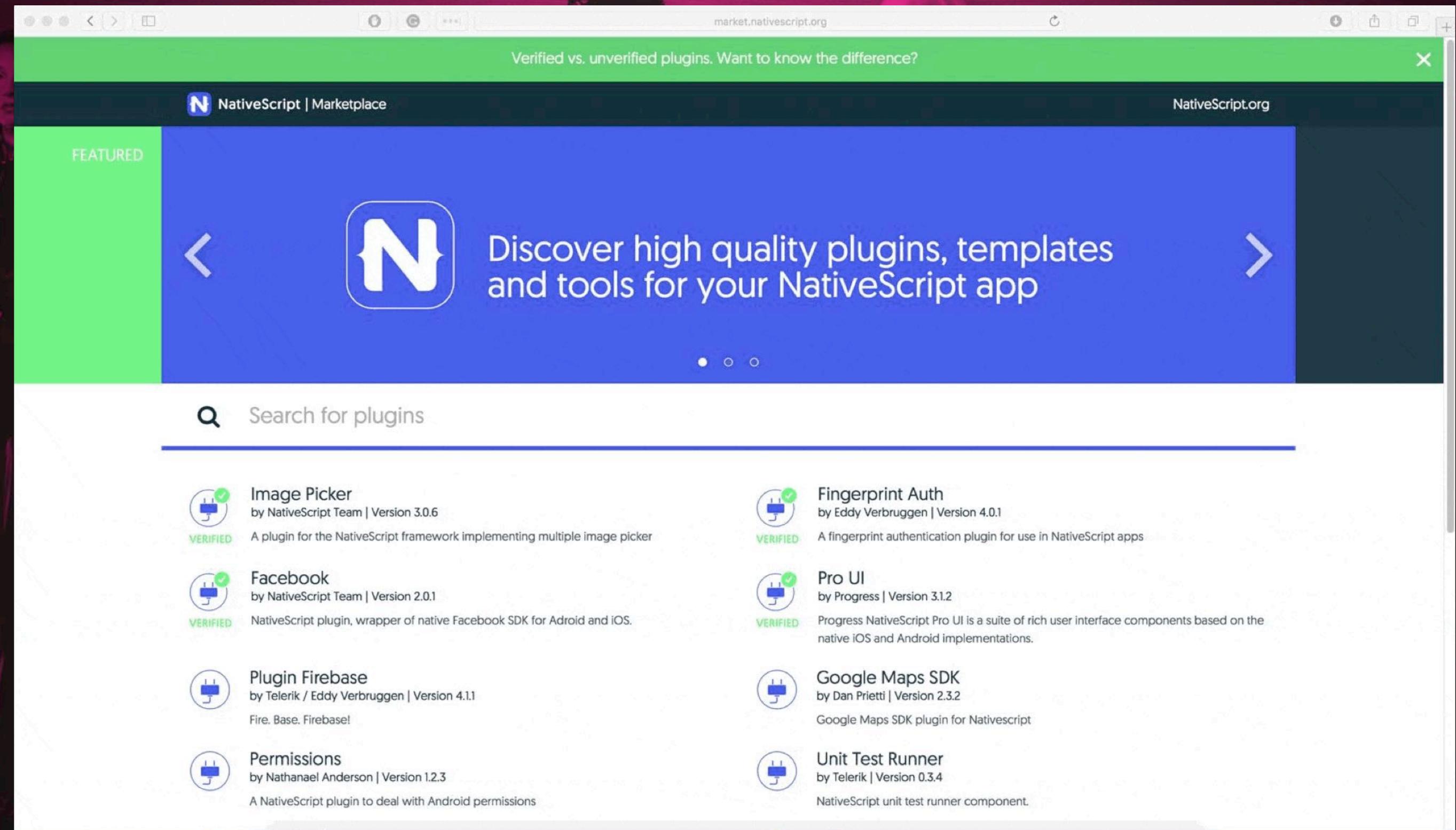
export class VersionNumber {
    get() {
        let PackageManager = android.content.pm.PackageManager;
        let pkg = application.android.context
            .getPackageManager()
            .getPackageInfo(application.android.context.getPackageName(), PackageManager.GET_META_DATA);
        return pkg.versionName;
    }
}
```

version-number.d.ts

```
export declare class VersionNumber {  
    get(): any;  
}
```

```
tns plugin add nativescript-version-number
```

```
import { VersionNumber } from "nativescript-version-number";  
  
let version = new VersionNumber().get();
```



The screenshot shows the NativeScript Marketplace homepage. At the top, there's a green banner with the text "Verified vs. unverified plugins. Want to know the difference?" and a close button. Below it is a dark header with the NativeScript logo and "NativeScript | Marketplace" on the left, and "NativeScript.org" on the right. A large blue banner in the center features the NativeScript logo and the text "Discover high quality plugins, templates and tools for your NativeScript app". To the left of the banner is a green "FEATURED" box with a left arrow icon. Below the banner is a search bar with the placeholder "Search for plugins". The main content area displays a grid of eight plugin cards:

- Image Picker** by NativeScript Team | Version 3.0.6 (VERIFIED)
A plugin for the NativeScript framework implementing multiple image picker
- Facebook** by NativeScript Team | Version 2.0.1 (VERIFIED)
NativeScript plugin, wrapper of native Facebook SDK for Adroid and iOS.
- Plugin Firebase** by Telerik / Eddy Verbruggen | Version 4.1.1 (VERIFIED)
Fire. Base. Firebase!
- Permissions** by Nathanael Anderson | Version 1.2.3 (VERIFIED)
A NativeScript plugin to deal with Android permissions
- Fingerprint Auth** by Eddy Verbruggen | Version 4.0.1 (VERIFIED)
A fingerprint authentication plugin for use in NativeScript apps
- Pro UI** by Progress | Version 3.1.2 (VERIFIED)
Progress NativeScript Pro UI is a suite of rich user interface components based on the native iOS and Android implementations.
- Google Maps SDK** by Dan Prietti | Version 2.3.2 (VERIFIED)
Google Maps SDK plugin for Nativescript
- Unit Test Runner** by Telerik | Version 0.3.4 (VERIFIED)
NativeScript unit test runner component.



nativescript-texttospeech
nativescript-audio

nativescript-texttospeech

```
tns plugin add nativescript-texttospeech
```

```
import { TNSTextToSpeech, SpeakOptions } from "nativescript-texttospeech";

let textToSpeech = new TNSTextToSpeech();
let speakOptions: SpeakOptions = {
    text: "Hello world!",
    locale: "en-GB"
};

textToSpeech.speak(speakOptions);
```

nativescript-audio

```
tns plugin add nativescript-audio
```

```
import { TNSPlayer } from "nativescript-audio";

let player = new TNSPlayer();

player.initFromFile({
  audioFile: "~/audio/song.mp3"
});

player.play();
```

A group of X-Men characters standing in a dark, dramatic setting. From left to right: Jean Grey (Phoenix), Cyclops, Emma Frost (White Queen), Wolverine, Beast, Iceman, Storm, Bishop, Magneto, and Cyclops again. They are all dressed in their iconic costumes, with a red and black color palette.

Win a T-shirt!



Turn up your NativeScript training

Are you ready to start learning NativeScript? Get the FREE getting started guides for NativeScript Core and NativeScript with Angular.

Do you already know the basics and want to go deeper? Our NativeScript Pro courses will take you to a finished and polished product that you can launch in the app stores. Courses can be purchased on-demand, or save money with our bundles! Only 100 of each bundle are made available for the special pre-launch price.

Grab them while we have them.



NativeScript Core



NativeScript with Angular

Courses for the NativeScript Core minded folks. MVVM, TypeScript, and no other UI frameworks.



NativeScript Plugins:
Creating Custom View
Components



NEW

Nathan Walker

LEARN MORE



NativeScript Core Getting
Started Guide

NEW

Alex Ziskind

FREE COURSE

FREE COURSE



NativeScript Core Pro



Alex Ziskind

PRESALE

LEARN MORE

A dark, atmospheric poster for the X-Men movies, featuring the main cast of mutants standing in a row. From left to right, the characters include: Jean Grey (Phoenix), Cyclops, Emma Frost (White Queen), Wolverine, Beast, Iceman, Storm, Magneto, and Cyclops again. They are all dressed in their iconic superhero costumes, set against a background of a city at night with lights reflecting on water.

nativescript-speech-recognition
nativescript-videooplayer
nativescript-texttospeech

nativescript-speech-recognition

```
tns plugin add nativescript-speech-recognition
```

```
import { SpeechRecognition } from "nativescript-speech-recognition";

private speechRecognition = new SpeechRecognition();

checkAvailability() {
    this.speechRecognition.available().then(
        (available: boolean) => console.log(available ? "YES!" : "NO"),
        (err: string) => console.log(err)
    );
}
```

nativescript-speech-recognition

```
this.speechRecognition.requestPermission().then((granted: boolean) => {  
    console.log("Granted? " + granted);  
});
```

nativescript-speech-recognition

```
this.speechRecognition.startListening({
    locale: "en-US",
    returnPartialResults: true,
    onResult: (transcription: SpeechRecognitionTranscription) => {
        console.log(`User said: ${transcription.text}`);
    }
});
```

nativescript-speech-recognition

```
this.speechRecognition.stopListening().then(() => {  
    console.log(`stopped listening`);  
    // do something with the recognized text  
    this.processInput();  
});
```

nativescript-videoplayer

```
tns plugin add nativescript-videoplayer
```

```
<VideoPlayer src="video.mp4"></VideoPlayer>
```

A group of Marvel X-Men characters standing in a dark, dramatic setting. In the foreground, from left to right, are Jean Grey (Phoenix), Cyclops, Emma Frost (White Queen), Wolverine, Iceman, and Storm. Behind them are Magneto, Bishop, and Professor X. The scene is set against a dark background with some glowing elements.

Demo

A collage of characters from the movie Guardians of the Galaxy. In the foreground, Rocket Raccoon (a raccoon in a suit) holds a large gun. Behind him is Star-Lord (Peter Quill) in his signature leather jacket. To the right is Gamora (Neve Campbell), holding a sword. In the background, Drax the Destroyer (Dave Bautista) is visible, along with other characters like Nebula and Groot. The scene is set against a dark, star-filled background.

2xr.nl/AppDevCon2018

A collage of characters from the Guardians of the Galaxy movie. Star-Lord (Peter Quill) is in the upper left, Gamora (Neidhe) is in the center, Drax (Drax the Destroyer) is on the right, Rocket Raccoon is in the lower left, and Groot is in the lower right. They are set against a dark background with stars and a nebula.

Thank you!

Links and Resources

Links

NativeScript styling: <http://2xr.nl/css>

Native elements playground: <http://2xr.nl/Native>

keyboardType playground: <http://2xr.nl/keyboardType>

TabView playground: <http://2xr.nl/TabView>

Plugin demo app: <http://2xr.nl/AppDevCon2018code>

Resources

NativeScript.nl - <https://nativescript.nl>

NativeScript Book - <https://nativescript.org/get-the-nativescript-book>

NativeScript Documentation - <https://docs.nativescript.org>

NativeScript Marketplace - <https://market.nativescript.org>

NativeScript Playground - <https://play.nativescript.org>

NativeScript Sidekick - <https://www.nativescript.org/nativescript-sidekick>