學習

nativescript

#nativescrip
t

1: nativescript

NativescriptWebJShtmliOSAndroid。

- AndroidUI Jankfps
- API
- UI
- TypescriptAngular 2
- Telerik

Examples

Nativescript。

WindowsOSX。

IDE。

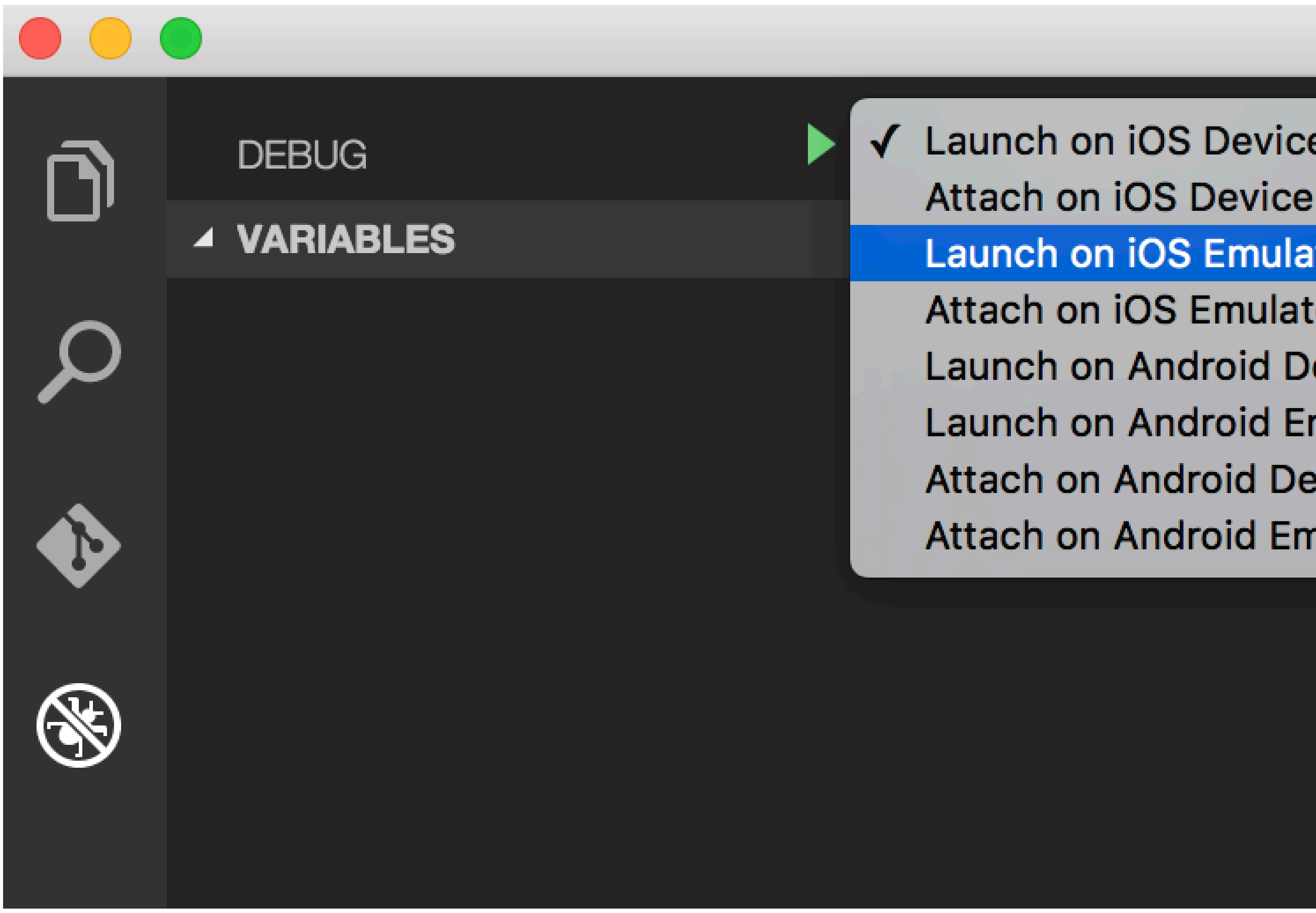
1. Node.js LTS。 Homebrew`brew install node4-lts`。
2. `npm install -g nativescript` 。 EACCES`sudo npm install -g nativescript` 。
3. `ruby -e "$(curl -fsSL https://www.nativescript.org/setup/mac)"` 。 。
4. `tns doctor` 。
5. 。

1. nodeJS LTS
2. `$ npm install -g nativescript`
3. `$ @powershell -NoProfile -ExecutionPolicy Bypass -Command "iex ((new-object net.webclient).DownloadString('https://www.nativescript.org/setup/win'))"` -
4. `$ tns doctor` cmd
5.

Visual StudioNativeScript

Visual Studio CodeMicrosoft。 NativeScript `F1` + `Shift` + `P` `ext install NativeScript` 。

NativeScript。 “”。



Hello World

```
$ mkdir hello-world
$ cd hello-world
$ tns create hello-world --ng
$ tns platform add android #You can only add ios on an OSX machine
```

◦ genymotionAndroid◦

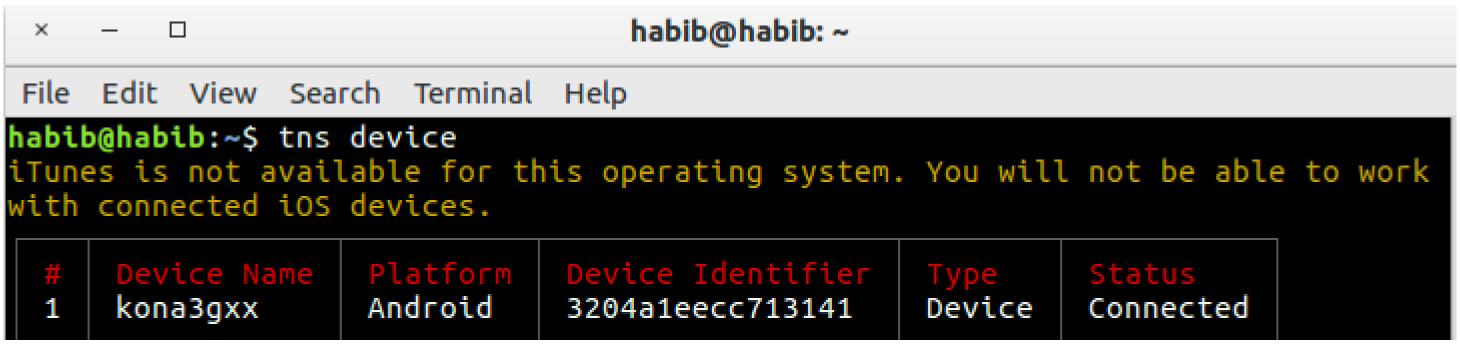
```
$ tns run android
```

android`--emulator`◦

tns 2.5livesync`tns run <platform>`◦ ◦

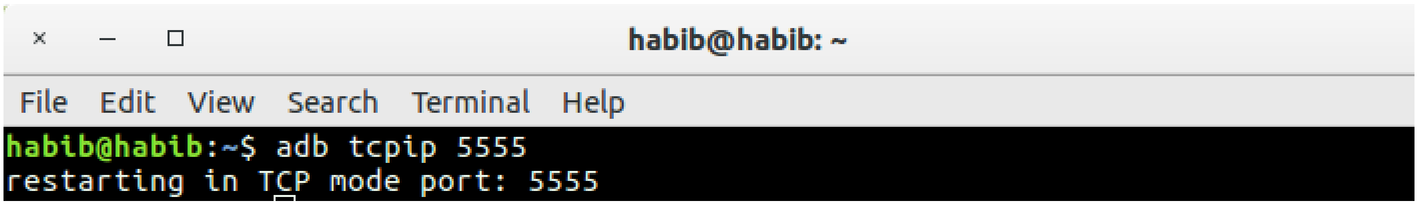
WiFinativescript-androidRoot

1 - USB◦ USB◦ `adb devices` `tns device` ◦



```
habib@habib:~$ tns device
iTunes is not available for this operating system. You will not be able to work
with connected iOS devices.
 │ # │ Device Name │ Platform │ Device Identifier │ Type   │ Status    │
 │ 1 │ kona3gxx    │ Android  │ 3204a1eecc713141  │ Device │ Connected │
```

2-run `adb tcpip 5555`

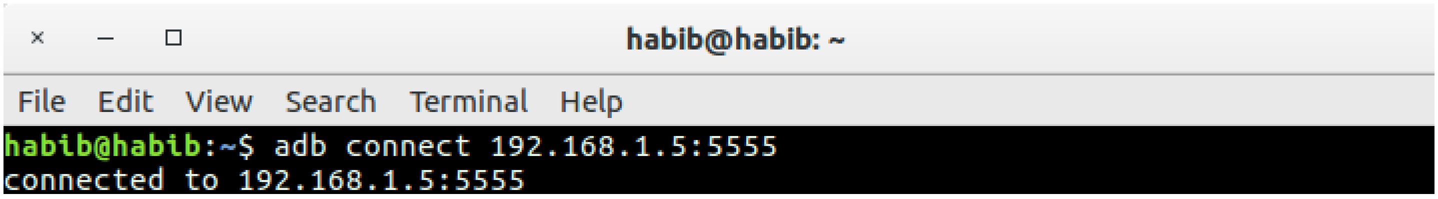


```
habib@habib:~$ adb tcpip 5555
restarting in TCP mode port: 5555
```

3 - USB。

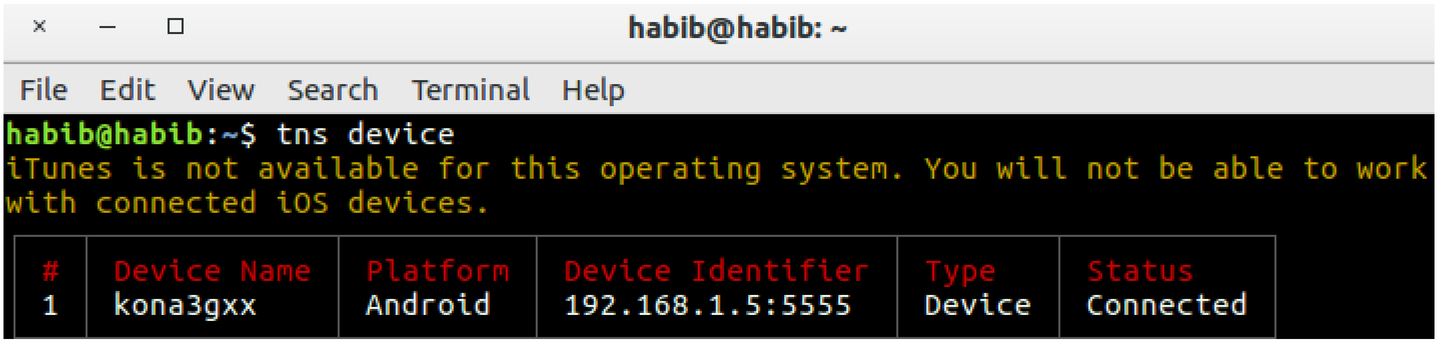
4 - - > - >IP。

5-run `adb connect <IP address of your device>:5555`



```
habib@habib:~$ adb connect 192.168.1.5:5555
connected to 192.168.1.5:5555
```

6 - `adb devices` `tns device` 。



```
habib@habib:~$ tns device
iTunes is not available for this operating system. You will not be able to work
with connected iOS devices.
 │ # │ Device Name │ Platform │ Device Identifier │ Type   │ Status    │
 │ 1 │ kona3gxx    │ Android  │ 192.168.1.5:5555  │ Device │ Connected │
```

7-`tns run android` `tns livesync android`。

1 - WiFi13wifi。 46。

2-Androidwifi。 。

。 Wi-Fi

1-Norman Peitek。 WiFiAndroidRoot。 [] https //futurestud.io/blog/how-to-debug-your-android-app-over-wifi-without-root 。 [201688]。

2 usethe4ce。 Wi-Fi//Android [] http //stackoverflow.com/a/10236938/4146943 。 [201688]。

nativescript https://riptutorial.com/zh-TW/nativescript/topic/921/nativescript

2: Examples

ng2-TNS-AndroidsurfaceView

ng2-nativescriptsurfaceView。 nativescript`surfaceView` `placeholder` 。

```
import {Component} from "@angular/core";
import placeholder = require("ui/placeholder");
let application= require("application");
```

html

`<Placeholder (creatingView)="creatingView($event)"></Placeholder>`

```
public creatingView(args: any) {
  var nativeView = new android.view.SurfaceView(application.android.currentContext);
  args.view = nativeView;
}
```

typescript`android` 。

ng2-nativescript

```
<Placeholder  *ngIf="init" (creatingView)="creatingView($event)"></Placeholder>
```

OnInit

```
import {Component,OnInit} from "@angular/core";
```

OnInit

```
export class AppComponent implements OnInit
```

```
public init: boolean = false;
ngOnInit() {
    this.init = true;
}
```

nativescriptsurfaceView :)

SurfaceView

`getHolder()`

```
  <Placeholder  #surface *ngIf="init" (creatingView)="creatingView($event)"
```

```
(loaded)="onLoaded(surface)"></Placeholder>
```

onLoaded

```
 onLoaded(element){
  let mSurface = element.android;
  let holder =  mSurface.getHolder();
}
```

`android` `element.android` `ngAfterViewInit` `loaded`。

ng2-TNS-AndroidsurfaceView

app.component.ts

```
import {Component,OnInit} from "@angular/core";
import placeholder = require("ui/placeholder");
let application= require("application");

@Component({
    selector: "my-app",
    templateUrl: "app.component.html",
})
export class AppComponent implements OnInit{

  public creatingView(args: any) {
    var nativeView = new android.view.SurfaceView(application.android.currentContext);
    args.view = nativeView;
  }

  onLoaded(element){
    let mSurface = element.android;
    let holder =  mSurface.getHolder();
  }

  public init: boolean = false;
    ngOnInit() {
        this.init = true;
    }
}
```

app.component.html

```
<StackLayout>
   <Placeholder  #surface *ngIf="init" (creatingView)="creatingView($event)"
(loaded)="onLoaded(surface)"></Placeholder>
</StackLayout>
```

https://riptutorial.com/zh-TW/nativescript/topic/5834/

3:

Examples

NativeScript`console`。 `console.log()`

```
console.log("hello world");
```

`consoledump() trace() assert()` 。

```
// Prints the state of a full object.
console.dump({ firstName: "Native", lastName: "Script"});

// Prints the current stack trace
console.trace();

// Asserts a boolean condition, and prints to the console if the assertion fails.
console.assert(1 === 1, "This won’t print as the condition is true");
console.assert(1 === 2, "This will print as the condition is false");
```

JavaScript

NativeScript`timer`。

```
var timer = require("timer")
```

```
var callback = function(){
    console.log("I will be executed once after 500ms");
}
var timeoutId = timer.setTimeout(callback, 500);

// clearing the timeout
timer.clearTimeout(timeoutId);
```

```
var callback = function(){
    console.log("I will be executed every 500 ms")
}
var intervalId = timer.setInterval(callback, 500);

// clearing the interval
timer.clearInterval(intervalId);
```

https://riptutorial.com/zh-TW/nativescript/topic/3133/

4: Nativescript

Examples

StackLayout

stacklayout

/ main.component.ts

```
import {Component, ElementRef, ViewChild} from "@angular/core";
import {Color} from "color";
import {View} from "ui/core/view";

    @Component({
        selector: "main",
        template: `
            <StackLayout #el>
              <Button text="Apply Changes" (tap)="changeBgColor()"></Button>
            </StackLayout>
        `,
        styleUrls: ["pages/main/main-common.css"],
    })

    export class MainComponent {
        @ViewChild("el") el: ElementRef;
        changeBgColor() {
            let el = <View>this.el.nativeElement;
            el.animate({
                backgroundColor: new Color("#222"),
                duration: 300
            });
        }
    }
```

/common.css

```
StackLayout{
    background-color: #333;
}
```

。

/ main.component.ts

```
import {Component, ElementRef, ViewChild} from "@angular/core";
import {View} from "ui/core/view";
import {AnimationCurve} from "ui/enums";

@Component({
    selector: "main",
    template: `
```

```
        <StackLayout>
          <Image #img src="~/assets/images/user-shape.png"></Image>
          <Button text="Apply Changes" (tap)="animateImage()"></Button>
        </StackLayout>
    `,
    styleUrls: ["pages/main/main-common.css"],
})

export class MainComponent {
    @ViewChild("img") img: ElementRef;
    animateImage() {
        let img = <View>this.img.nativeElement;
        img.animate({
            translate: { x: 0, y: 120 },
            duration: 2000,
            curve: AnimationCurve.easeIn
        });
    }
}
```

#snippet

cubicBezier。

1. cubicBezier

```
img.animate({
        translate: { x: 0, y: 120 },
        duration: 2000,
        curve: AnimationCurve.cubicBezier(0.1, 0.2, 0.1, 1)
  });
```

2.

```
img.animate({
    opacity: 0,
    duration: 2000
});
```

```
img.animate({
    translate: { x: 120, y: 0},
    duration: 2000
});
```

```
img.animate({
    scale: { x: 1.5, y: 1.5},
    duration: 2000
});
```

```
img.animate({
    rotate: 270,
    duration: 2000
});
```

Nativescript https://riptutorial.com/zh-TW/nativescript/topic/5970/nativescript

5:

chrome v8ES7。 `"use strict";`。 typescripttypescript。 `self.functionNameglobal.functionName` 。

Examples

2

`/app/services/greeting.service.ts`

```
import { Injectable } from '@angular/core';
import {greetingTypes,request,response}
            from './greeting.interface'

@Injectable()
export class Greeting{

    private worker;
    constructor(){
       this.worker = new Worker('../workers    /greeting.worker');
    }

    sayHello(message:string, answerCallback:Function){
        let requestData:request =
            {'type':greetingTypes.HELLO ,'message':message} ;

        this.worker.postMessage(requestData);
        this.worker.onmessage = (msg)=>{
            let response:response = msg.data;

            if(response.type == greetingTypes.HELLO){
                answerCallback(response.answer)
            }
        }
    }

    sayBye(message:string, answerCallback:Function){
        let requestData:request =     {'type':greetingTypes.BYE ,'message':message};

        this.worker.postMessage(requestData);
        this.worker.onmessage = (msg)=>{
            let response:response = msg.data;

            if(response.type == greetingTypes.BYE)
                answerCallback(response.answer)
        }
    }
}
```

`app/services/greeting.interface.ts`

```
export enum greetingTypes{
    BYE,
    HELLO
}

export interface request{
   type:greetingTypes,
   message:string
}

export interface response{
    type:greetingTypes,
    answer:string
}
```

app/workers/greeting.worker.ts

```
require("globals");
import {greetingTypes,request,response} from
            '../services/greeting.interface';

self.onmessage = (msg)=> {
   let request:request = msg.data;
   let responseData:response;
   if(request.type ==  greetingTypes.HELLO)
        console.log('worker got the message: ' +
                        request.message);
        responseData = {'type':greetingTypes.HELLO,
                            'answer': 'HELLO!'};
        global.postMessage(responseData);

   if(request.type == greetingTypes.BYE )
        console.log('worker got the message: ' +request.message);
        responseData = {'type':greetingTypes.BYE ,
                           'answer':'goodBye!'};
            global.postMessage(responseData);

    };
```

app/app.component.ts

```
import {Component} from "@angular/core";
import {Greeting} from './services/greeting.service';
@Component({
    selector: "my-app",
    templateUrl: "app.component.html",
    providers:[Greeting]
})
export class AppComponent {

    constructor(private greeting:Greeting){}


public tapHello() {

    this.greeting.sayHello('hi',
        (answer)=>{console.log('answer from worker : '+ answer)});
}


public tapBye() {
    this.greeting.sayBye('bye',
        (answer) => {console.log('answer from worker : ' + answer)});
}
```

}

app/app.component.html

```
<StackLayout>
    <Button text="sayBye" (tap)="tapBye()"></Button>
    <Button text="sayHello" (tap) = "tapHello()"></Button>
</StackLayout>
```

https://riptutorial.com/zh-TW/nativescript/topic/7878/

6:

Examples

NativescriptView.OnLayoutChangeListener

```
let playerLayoutChangeListener = new android.view.View.OnLayoutChangeListener( {
      onLayoutChange : function ( v:View, left:number, top:number, right:number,
bottom:number, oldLeft:number, oldTop:number, oldRight:number, oldBottom:number):any {
          if (left != oldLeft || top != oldTop || right != oldRight || bottom != oldBottom) {
              console.log("OnLayoutChangeListener");
              __this.changeSurfaceLayout();
          }
      }
    });
```

surfaceView http://stackoverflow.com/documentation/proposed/changes/79536

```
    surfaceView.addOnLayoutChangeListener(playerLayoutChangeListener);
```

```
    surfaceView.removeOnLayoutChangeListener(playerLayoutChangeListener);
```

https://riptutorial.com/zh-TW/nativescript/topic/5560/

7: RepeaterListView* ngFor for {N} + Angular-2 apps

{N} + Angular-2Repeater * ngRepeatAngular-2。 ListView* ngFor。

Examples

RepeaterNativeScript Core

page.xml

```
<Page xmlns="http://schemas.nativescript.org/tns.xsd" navigatingTo="navigatingTo">
  <Repeater items="{{ myItems }}">
      <Repeater.itemTemplate>
        <Label text="{{ title || 'Downloading...' }}" textWrap="true" />
      </Repeater.itemTemplate>
  </Repeater>
</Page>
```

page.ts

```
import { EventData, Observable } from "data/observable";
import { Page } from "ui/page";

let viewModel = new Observable();
var myItems = [ {title: "Core Concepts"},
                {title: "User Interface"},
                {title: "Plugins"},
                {title: "Cookbook"},
                {title: "Tutorials"} ];

export function navigatingTo(args: EventData) {
    var page = <Page>args.object;

    viewModel.set("myItems", myItems);

    page.bindingContext = viewModel;
}
```

ObservableArrayRepeaterNativeScript Core

page.xml

```
<Page xmlns="http://schemas.nativescript.org/tns.xsd" navigatingTo="navigatingTo">
  <Repeater items="{{ myItems }}">
      <Repeater.itemTemplate>
        <Label text="{{ title || 'Downloading...' }}" textWrap="true" class="title" />
      </Repeater.itemTemplate>
  </Repeater>
</Page>
```

page.ts

```
import { EventData, Observable } from "data/observable";
import { ObservableArray } from "data/observable-array";
import { Page } from "ui/page";

let viewModel = new Observable();
let myItems = new ObservableArray({title: "Core Concepts"}, {title: "User Interface"}, {title:
"Plugins"}, {title: "Cookbook"}, {title: "Tutorials"});

export function navigatingTo(args: EventData) {

    var page = <Page>args.object;
    viewModel.set("myItems", myItems);

    // The Repeater will be updated automatically when new item is pushed.
    myItems.push({title:"Publishing"});

    page.bindingContext = viewModel;
}
```

ListViewObservableArrayNativeScript Core

page.xml

```
<Page xmlns="http://schemas.nativescript.org/tns.xsd" navigatingTo="navigatingTo">
  <ListView items="{{ myItems }}" itemTap="listViewItemTap">
      <ListView.itemTemplate>
        <Label text="{{ title || 'Downloading...' }}" textWrap="true" class="title" />
      </ListView.itemTemplate>
  </ListView>
</Page>
```

page.ts

```
import { EventData, Observable } from "data/observable";
import { ObservableArray } from "data/observable-array";
import { Page } from "ui/page";
import { ItemEventData } from "ui/list-view";

import frameModule = require("ui/frame");

let viewModel = new Observable();
let myItems = new ObservableArray(  {title: "Core Concepts"},
                                    {title: "User Interface"},
                                    {title: "Plugins"},
                                    {title: "Cookbook"},
                                    {title: "Tutorials"}  );

export function navigatingTo(args: EventData) {

    var page = <Page>args.object;
    viewModel.set("myItems", myItems);

    // ListView will be updated automatically when new item is pushed.
    myItems.push({title:"Publishing"});
    page.bindingContext = viewModel;
}

export function listViewItemTap(args:ItemEventData) {
    var itemIndex = args.index;

    // example how to navigate details-page & pass the tapped item context
    // frameModule.topmost().navigate({
    //     moduleName: "./details-page",
    //     context: myItems.getItem(itemIndex);
    // });
}
```

ListViewNativeScript + Angular-2

-listview.component.html

```
<ListView [items]="countries" (itemTap)="onItemTap($event)">
    <template let-country="item" let-i="index">
        <StackLayout orientation="horizontal">
            <Label [text]='(i + 1) +".) "' ></Label>
            <Label [text]='country.name'></Label>
        </StackLayout>
    </template>
</ListView>
```

-listview.component.ts

```
import { Component, ChangeDetectionStrategy, Input }  from "@angular/core";

class Country {
    constructor(public name: string) { }
}

var europianCountries = ["Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czech
Republic",
"Denmark", "Estonia", "Finland", "France","Germany", "Greece", "Hungary", "Ireland", "Italy",
"Latvia", "Lithuania", "Luxembourg", "Malta", "Netherlands","Poland", "Portugal", "Romania",
"Slovakia",
"Slovenia","Spain", "Sweden", "United Kingdom"];

@Component({
    selector: "creating-listview",
    styleUrls:["./creating-listview.component.css"],
    templateUrl: "./creating-listview.component.html",
    changeDetection: ChangeDetectionStrategy.OnPush
})

export class CreatingListViewComponent {
    public countries: Array<Country>;

    constructor() {
        this.countries = [];

        for (var i = 0; i < europianCountries.length; i++) {
            this.countries.push(new Country(europianCountries[i]));
            }
    }

    public onItemTap(args) {
        console.log("Item Tapped at cell index: " + args.index);
    }
}
```

* ngFor Structural DirectivenativeScript + Angular-2

ngfor.component.html

```
<StackLayout>
    <Label *ngFor="let item of items" [text]="item"></Label>
</StackLayout>
```

ngfor.component.ts

```
import { Component } from "@angular/core";

var dataItems = ["data-item 1", "data-item 2", "data-item 3"]

@Component({
    selector: 'ngfor-component',
    styleUrls:["./ngfor.component.css"],
    templateUrl: "./ngfor.component.html",
})

export class NgForComponent {
    public items:Array<string> = [];

    constructor(){
        this.items = dataItems;
    }
}
```

RepeaterJavaScript

page.js

```
var context = {
items: [
        {id: 1, name: "Foo"},
        {id: 2, name: "Bar"},
        {id: 3, name: "Joe"}
    ]
}

exports.loaded = function(args){
    var page = args.object;
    page.bindingContext = context;
}

exports.showEntry = function(args){
    // select the tapped entry without passing an index or anything like that
    var selectedEntry = args.view.bindingContext;
    console.log(selectedEntry.id + " " + selectedEntry.name);
}
```

page.xml

```
<Repeater items="{{ items }}" >

    <Repeater.itemTemplate>
        <Label text="{{ name }}" tap="showEntry" />
    </Repeater.itemTemplate>

</Repeater>
```

RepeaterListView* ngFor for {N} + Angular-2 apps https://riptutorial.com/zh-TW/nativescript/topic/5226/-repeater-listview--ngfor-for--n--plus-angular-2-apps-

8: nativescript

Examples

main.component.ts

```
import {Component} from "@angular/core";

@Component({
    selector: "main",
    template: `
    <StackLayout>
      <TextField hint="some text"></TextField>
      <Button text="Click me" class="btn"></Button>
    </StackLayout>
    `,
    styleUrls: ["pages/main/main-common.css", "pages/main/main.css"]
})
export class MainComponent { }
```

1CSS

app.css - 。

```
StackLayout {
  margin: 10;
  background-color: white;
}
.btn, TextField {
  margin-left: 16;
  margin-right: 16;
}
```

2CSS

platform.android.css - Android。

```
.btn{
    background-color: #191919;
    color: #fff;
}
```

platform.ios.css - ios。

```
.btn{
    background-color: #fff;
    color: #191919;
}
```

app.css

```
@import url("~/platform.css");
```

3CSS

pages / main / main.android.css - android。

```
TextField {
  color: #e1e1e1;
  font-size: 12;
}
```

pages / main / main.ios.css - ios。

```
TextField {
  color: #e3e3e3;
  font-size: 15;
}
```

pages / main / main-common.css - 。

```
TextField {
  padding: 4;
}
```

nativescript https://riptutorial.com/zh-TW/nativescript/topic/3872/nativescript

9:

Examples

/ - android

battryclock ...。

```
let frame = require("ui/frame");
```

```
frame.topmost().android.activity.getWindow().
getDecorView().setSystemUiVisibility(android.view.View.SYSTEM_UI_FLAG_FULLSCREEN);
```

```
frame.topmost().android.activity.getWindow().
getDecorView().setSystemUiVisibility(android.view.View.SYSTEM_UI_FLAG_VISIBLE );
```

statusBarandroid

`APP_Resources/values/styles.xml`

`<item name="android:windowTranslucentStatus">true</item>`

```
<style name="AppThemeBase" parent="Theme.AppCompat.Light.NoActionBar"> </style>
```

。

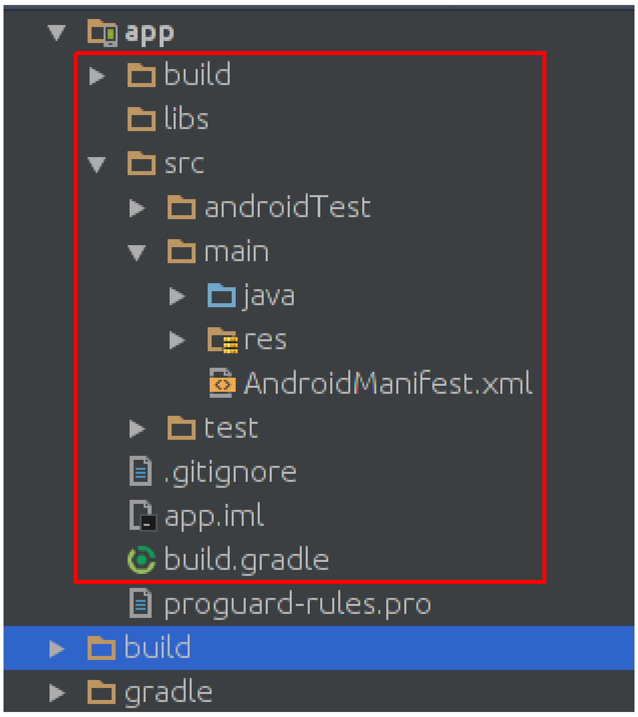
https://riptutorial.com/zh-TW/nativescript/topic/6007/

10: api

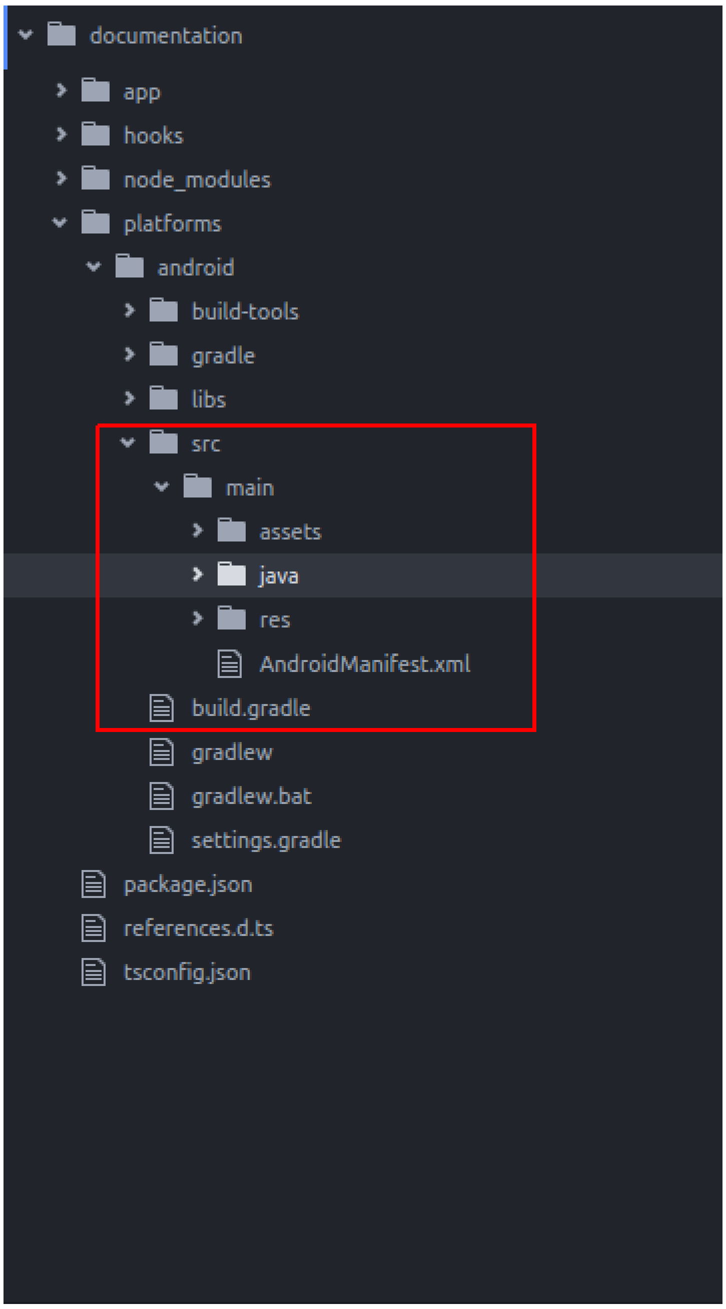
Examples

nativescriptjavajavascript

Android studio

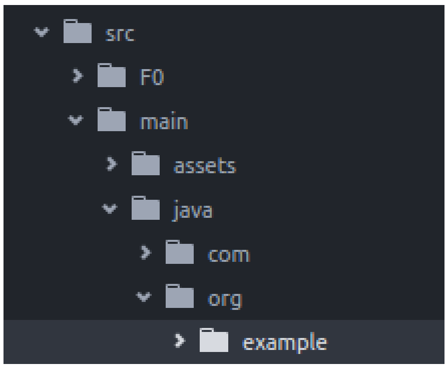


nativescript



◦ android studionativescriptjava。

Toastnativescript。 nativescript`java/org/example`



`exampleMyTrast.java`；

MyToast.java

```
package org.example;

import android.widget.Toast;
import android.content.Context;


public class MyToast{

    public static void showToast(Context context,String text ,String StrDuration ){
      int duration;
      switch (StrDuration){
          case "short":
              duration = Toast.LENGTH_SHORT;
              break;
          case "long":
              duration = Toast.LENGTH_LONG;
              break;
      }
        Toast.makeText(context,text, Toast.LENGTH_SHORT).show();
    }
}
```

;

app.component.ts

```
import {Component} from "@angular/core";
let application = require("application");

declare var org:any;
@Component({
    selector: "my-app",
    templateUrl: "app.component.html",
})
export class AppComponent {
    public counter: number = 16;

    public get message(): string {
        if (this.counter > 0) {
            return this.counter + " taps left";
        } else {
            return "Hoorraaay! \nYou are ready to start building!";
        }
    }

    public onTap() {
        this.counter--;
        org.example.MyToast.showToast(application.android.context,"You pressed the
button","short");
    }
}
```

;

1. showToast`Toast.makeText` `application.android.context`
2. typescript`org` `declare var org:any;`

javascriptnative apis

Toastnativescript。

```
import {Component} from "@angular/core";
let application = require("application");

declare var android:any;

@Component({
    selector: "my-app",
    templateUrl: "app.component.html",
})
export class AppComponent {
    public counter: number = 16;

    public get message(): string {
        if (this.counter > 0) {
            return this.counter + " taps left";
        } else {
            return "Hoorraaay! \nYou are ready to start building!";
        }
    }

    public onTap() {
        this.counter--;
        this.showToast("You pressed the button","short");
    }



    public showToast(text:string ,StrDuration:string ):void{
      let duration:number;
      switch (StrDuration){
          case "short":
              duration = android.widget.Toast.LENGTH_SHORT;
              break;
          case "long":
              duration = android.widget.Toast.LENGTH_LONG;
              break;
      }
        android.widget.Toast.makeText(application.android.context,text,
android.widget.Toast.LENGTH_SHORT).show();
   }
}
```

toast`Toast.makeText` `android.widget.Toast`。 `Toast.makeText`nativescript `application.android.context`

api https://riptutorial.com/zh-TW/nativescript/topic/5188/api

S. No		Contributors
1	nativescript	Adam Diament, Community, George Edwards, HabibKazemi, Hardik Vaghani, Houssem Yahiaoui, Richard Hubley, user6939352
2		HabibKazemi
3		Tim Hallyburton, TJ VanToll
4	Nativescript	Madhav Poudel
5		HabibKazemi
6		HabibKazemi
7	RepeaterListView* ngFor for {N} + Angular-2 apps	Nick Iliev, Tim Hallyburton, William KLEIN
8	nativescript	George Edwards, Madhav Poudel, Nick Iliev
9		HabibKazemi
10	api	HabibKazemi