



EBook Gratuito

APPENDIMENTO ionic-framework

Free unaffiliated eBook created from
Stack Overflow contributors.

#ionic-

framework

Sommario

Di.....	1
Capitolo 1: Iniziare con il framework ionic	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione o configurazione.....	2
1. Installa Ionic Framework e Cordova (poiché le app Ionic sono basate su Cordova) utilizz.....	2
2. Avvia un nuovo progetto Ionic:.....	3
3. Testare l'app Ionic:.....	4
Introduzione e installazione di Ionic Framework e installazione.....	4
Ionic Framework Hello World App.....	6
Ionic Platform (Ionic Cloud) per Yo (Yeoman) Ionic Projects.....	7
Piattaforma ionic :.....	7
Costruisci, spinge, distribuisce e ridimensiona le tue app Ionic, in modo semplice.....	7
Generatore di quadri ionic.....	8
Un generatore per Ionic Framework di Yeoman, lo strumento di Scaffolding del Web per le mo.....	8
ionico-piattaforma web-client.....	9
Un client web che fornisce interazioni con la piattaforma Ionic.....	9
Distribuzione ionic.....	10
Invia aggiornamenti in tempo reale alle app di produzione e gestisci la cronologia delle v.....	10
Analytics ionic.....	11
Visualizza il feed dal vivo degli eventi o il numero grezzo / univoco di eventi / utenti n.....	11
Ionic Push.....	12
Invia notifiche push mirate e automatizzate ai tuoi utenti.....	12
App di esempio.....	14
Capitolo 2: Accesso alla fotocamera e alla libreria fotografica del dispositivo dall'appli	16
Osservazioni.....	16
Examples.....	16
Apri camara e galleria fotografica.....	16
Per l'esempio di Ionic 3.....	19

Capitolo 3: Collegamento di Ionic con qualsiasi database	22
Examples	22
Non puoi farlo direttamente dalla struttura ionica	22
Capitolo 4: Come utilizzare le funzioni di EcmaScript 6 in Ionic?	23
Examples	23
Usando gulp-babel e gulp-idraulico	23
Capitolo 5: Componenti CSS ionici	24
Osservazioni	24
Examples	24
Sintassi del sistema di griglia di base	24
Griglia di base	24
Griglie sfalsate	25
Allinea le colonne	25
Griglia reattiva	26
Sintassi delle voci di base dell'elenco	26
Utilizzo di base dei colori di utilità	28
Capitolo 6: Crea finestra di dialogo in Ionic	30
Parametri	30
Osservazioni	30
Examples	30
Crea finestra di dialogo in Ionic	30
Capitolo 7: Distribuisci Ionic come sito web	31
Examples	31
Copia semplicemente la cartella www sul tuo server web	31
Capitolo 8: Esegui l'app Ionic on Emulator o sul tuo telefono	32
Examples	32
Esegui l'app Ionic on Emulator o sul tuo telefono	32
1. Aggiungi un target di piattaforma	32
2. Crea la tua app	32
Live Reload App During Development (beta)	32
Flag / opzioni della riga di comando per l' run e l' emulate	33

3. Emulazione della tua app.....	33
4. Esecuzione della tua app.....	34
4.1. Specifica il tuo obiettivo.....	34
Capitolo 9: Estensioni Ionic AngularJS.....	35
Osservazioni.....	35
Examples.....	35
Input di forma.....	35
Finestre modali.....	35
Creazione dell'oggetto modale nell'ambito.....	35
Controlla il modale.....	36
Eventi modali.....	36
Capitolo 10: Ganci Ionico CLI.....	38
Osservazioni.....	38
introduzione.....	38
Tipi di gancio.....	38
Modi per definire i ganci:.....	38
Examples.....	39
Verifica degli errori nei file Javascript in before_prepare utilizzando jshint.....	39
Capitolo 11: Inizia a creare app su Ionic.....	41
Examples.....	41
Avvio di un'applicazione ionica.....	41
Avvio di un'applicazione ionica.....	41
Starter modello con nome.....	41
Github Repo starters.....	41
Codepen URL starters.....	41
Avviatori di directory locali:.....	41
Flag / opzioni della riga di comando:.....	41
Capitolo 12: Ionic Cloud per Yeoman Ionic Projects.....	43
Examples.....	43
Ionic Platform (Ionic Cloud) per Yo (Yeoman) Ionic Projects.....	43
Piattaforma ionica :.....	43

Costruisci, spinge, distribuisce e ridimensiona le tue app Ionic, in modo semplice.....	43
Generatore di quadri ionici.....	44
Un generatore per Ionic Framework di Yeoman, lo strumento di Scaffolding del Web per le mo.....	44
ionico-piattaforma web-client.....	45
Un client web che fornisce interazioni con la piattaforma Ionic.....	45
Distribuzione ionica.....	46
Invia aggiornamenti in tempo reale alle app di produzione e gestisci la cronologia delle v.....	46
Analytics ionico.....	47
Visualizza il feed dal vivo degli eventi o il numero grezzo / univoco di eventi / utenti n.....	47
Ionic Push.....	48
Invia notifiche push mirate e automatizzate ai tuoi utenti.....	48
App di esempio.....	50
Capitolo 13: Ionic: analizza la tua app con jshint e gulp-jshint come parte del processo d.....	51
Osservazioni.....	51
Che cosa è linting e come installare i pacchetti richiesti?.....	51
Examples.....	51
Aggiungi un'attività gulp.....	51
Creare il file .jshintrc (facoltativo).....	52
Aggiungi Makefile.....	52
Capitolo 14: Ionicons.....	54
Osservazioni.....	54
Examples.....	54
Utilizzo di base.....	54
Utilizzo prolungato.....	54
Capitolo 15: Pubblicare la tua app Ionic.....	56
Examples.....	56
Costruire la versione di rilascio da macOS.....	56
Capitolo 16: Qual è la differenza tra "costruzione ionica" e "preparazione ionica"?.....	57
Examples.....	57
build ionico vs preparazione ionica.....	57
Capitolo 17: Scorrimento infinito ionico per mostrare gli elementi di carico su richiesta.....	58

Examples.....	58
Carica n numero di dati disponibili su richiesta.....	58
Capitolo 18: Servizi backend ionici (ionic.io).....	59
Examples.....	59
Introduzione e configurazione.....	59
Capitolo 19: Testare l'app Ionic in un browser.....	61
Osservazioni.....	61
Esempio.....	61
Examples.....	62
Test in un browser.....	62
LiveReload.....	62
Ionic Lab.....	63
Specifica di un indirizzo IP da utilizzare.....	63
Proxy di servizio.....	63
Flag / opzioni della riga di comando.....	64
Titoli di coda.....	65

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ionic-framework](#)

It is an unofficial and free ionic-framework ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ionic-framework.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con il framework ionic

Osservazioni

Ionic è un framework per lo sviluppo di applicazioni mobili con HTML, CSS e JavaScript. Le applicazioni ioniche funzionano come applicazioni native e hanno un "aspetto grafico" nativo.

Ionic è basato sul framework [AngularJS](#) e fornisce una soluzione completa per progettare, costruire e pacchettizzare applicazioni mobili. Il design è realizzato con una raccolta di strumenti modello e una [libreria di icone personalizzate](#) . Ionic fornisce [componenti CSS / SASS](#) personalizzati e le [estensioni dell'interfaccia utente JavaScript](#) . Le app ioniche possono essere costruite, emulate e impacchettate con la loro [interfaccia a riga di comando \(CLI\)](#) .

I modelli ionic sono dinamici e reattivi e si adattano al loro ambiente per fornire un "look and feel" nativo. Questo adattamento include layout, stile e icone. Ionic rende anche disponibile la [personalizzazione](#) indipendente della [piattaforma](#) . Poiché le app Ionic utilizzano la tecnologia Web front-end, possono essere visualizzate in un browser per uno sviluppo più rapido.

Le app Ionic sono costruite su [Apache Cordova](#) per impostazione predefinita. Hanno accesso a tutti i [plugin Cordova](#) che consentono di utilizzare funzionalità native, quali notifiche push, fotocamera, accelerometro, ecc. Le app Cordova funzionano su [più piattaforme](#) e dispositivi (telefoni, tablet, ecc.) Con uno sforzo minimo. Cordova può essere sostituito con [altre tecnologie multiplatforma](#) come [trigger.io](#) .

Versioni

Versione	Data di rilascio
1.3.1 "el salvador"	2016/05/12
1.3.0 "delhi"	2016/04/21
1.2.0 "zirconio-zeren"	2015/12/09
1.1.0 "xenon-xerus"	2015/08/13
1.0.0 "uranio-unicorno"	2015/05/12

Examples

Installazione o configurazione

1. Installa Ionic Framework e Cordova (poiché le app Ionic

sono basate su Cordova) utilizzando npm (Node Package Manager):

Assicurati di avere una versione aggiornata di Node.js installata sul tuo sistema. Se non hai installato Node.js, puoi installarlo da [qui](#) .

Inoltre, per gli utenti Mac, l'ultima versione di Xcode installata sul tuo sistema ti porta gli strumenti da riga di comando e il simulatore iOS, [scaricalo qui](#) .

Aprire una finestra di terminale (Mac) o una finestra di comando (Windows) e installare Cordova e Ionic:

```
$ npm install -g cordova ionic
```

Su un Mac, potresti dover utilizzare sudo in base alla configurazione del tuo sistema:

```
$ sudo npm install -g cordova ionic
```

Se hai già installato Cordova e Ionic sul tuo computer, assicurati di aggiornare alla versione più recente:

```
$ npm update -g cordova ionic
```

o

```
$ sudo npm update -g cordova ionic
```

Seguire le guide della piattaforma [Android](#) e [iOS](#) per installare le dipendenze della piattaforma richieste.

Nota: lo sviluppo iOS richiede Mac OS X. Il simulatore iOS tramite la CLI Ionic richiede il pacchetto ios-sim npm, che può essere installato con il comando:

```
$ sudo npm -g install ios-sim
```

2. Avvia un nuovo progetto Ionic:

Crea un progetto Ionic utilizzando uno dei modelli di app già pronti o uno vuoto per iniziare.

```
$ ionic start myApp blank
```

o

```
$ ionic start myApp tabs
```

```
$ ionic start myApp sidemenu
```

3. Testare l'app Ionic:

Per testare la tua app Ionic in un browser desktop su entrambe le piattaforme iOS e Android:

```
$ ionic serve --lab
```

Considerando che `ionic serve --lab` è ottimo per testare l'interfaccia utente dell'applicazione su più piattaforme, potrebbe portare ad alcuni problemi per la console JavaScript o l'ispezione degli elementi, in tal caso cosa si può preferire:

```
$ ionic serve
```

Per testare la tua app Ionic in un emulatore:

```
$ cd myApp
$ ionic platform add ios android
$ ionic build ios
$ ionic emulate ios
```

Sostituisci `ios` con `Android` per il test dell'emulatore Android:

```
$ ionic build android
$ ionic emulate android
```

Per testare la tua app Ionic su un dispositivo Android connesso tramite USB:

```
$ ionic run android
```

Per testare la tua app Ionic su un dispositivo iOS collegato tramite USB:

```
$ ionic run ios --device
```

Introduzione e installazione di Ionic Framework e installazione

Quadro ionic

Un framework di sviluppo di applicazioni mobili multiplatforma che utilizza le tecnologie Web Angular JS e Front End.

Sito Web ufficiale : <http://ionicframework.com/>

Documentazione : <http://ionicframework.com/docs/>

Installazione e configurazione

Installazione Ionic richiede NPM (Node Package Manager) e Cordova.

È possibile scaricare e installare Node JS da [qui](#), fornito con NPM out of the Box.

Per scaricare Apache Cordova puoi utilizzare NPM dalla riga di comando

```
npm install -g cordova
```

Se hai già NPM e Cordova, puoi installare il framework ionic dalla riga di comando usando il seguente comando.

```
npm install -g ionic
```

Questo installerà e installerà il framework ionic affinché tu possa usarlo dalla riga di comando.

Nota * In base all'ambiente di sistema, potrebbe essere necessario eseguire i privilegi di amministratore.

Iniziare un nuovo progetto

Per avviare un nuovo progetto Ionic Framework è possibile utilizzare il seguente comando

```
ionic start myproject
```

O

```
ionic start myproject [template_name]
```

Modelli:

Ionic ti consente di creare progetti utilizzando alcuni modelli incorporati

`tabs` (impostazione predefinita): che creerà una semplice app con visualizzazione a schede.

`sidemenu` : che creerà app ionic con menu laterale.

`blank` : che creerà un'app ionica vuota.

che creerà una nuova cartella denominata `myproject` con tutti i file di progetto ionic.

per testare il progetto nel tuo browser puoi usare il seguente comando

```
ionic serve --lab
```

O

```
ionic serve
```

Esegui un emulatore Per eseguire o testare l'app su emulatore o telefono, dovrai prima aggiungere una piattaforma per poter usare il seguente comando

```
ionic platform [Platform Name]
ionic build [Platform Name]
ionic emulate [platform name]
```

Nomi di piattaforma che puoi menzionare direttamente Android e iOS per le rispettive piattaforme, puoi menzionare più nomi di piattaforme separati da spazio.

Per eseguire la tua app puoi utilizzare

```
ionic run [platform name]
```

Per aiuto puoi usare

```
ionic --help
```

o

```
ionic help
```

Fare riferimento a [questo link](#) per una spiegazione dettagliata del cli ionic.

Fare riferimento a [questo link](#) per i componenti CSS disponibili in ionic.

Fare riferimento a [questo collegamento](#) per il riferimento API Javascript per ionic.

Per uno sviluppo più veloce con ionic puoi provare anche [Playground ionic](#) .

Buona fortuna con la struttura ionic ...

Ionic Framework Hello World App

After All the setup, Per rendere Hello World App

- Per creare una semplice app vuota, esegui il comando sotto sul terminale:

```
ionic start HelloWorld blank // create new project

cd HelloWorld // get into HelloWorld directory
```

- apri il Progetto HelloWorld in IDE sublime / webstrom:
 - Modifica index.html, in www / ditactory

```
<body ng-app="starter">
  <ion-pane>
    <ion-header-bar class="bar-stable">
      <h1 class="title">Ionic Hello World App</h1>
    </ion-header-bar>
```

```
<ion-content>
  <div class="center">Hello World..!</div>
</ion-content>
</ion-pane>
</body>
```

- Per eseguire nel browser dal terminale:

```
ionic serve // run the app in browser
```

- Per aggiungere una piattaforma:

```
ionic platform add android // add android platform
ionic platform add ios // add ios platform
```

- Per eseguire sul dispositivo:

```
adb devices // to check devices is connected
ionic run android // to run on android devices
ionic run ios // to run on ios devices
```

- Per correre nel fegato:

```
ionic run android -c -s -l // to check app in live reload with console.
```

Ionic Platform (Ionic Cloud) per Yo (Yeoman) Ionic Projects



Piattaforma ionica :

Costruisci, spinge, distribuisci e ridimensiona le tue app Ionic, in modo semplice.

Descrizione del titolo:

La piattaforma Ionic è una piattaforma cloud per la gestione e il ridimensionamento di app mobili multiplatforma. I servizi integrati consentono a te e al tuo team di creare, distribuire e far crescere le app in modo efficiente.

Obiettivo del documento:

Ionic Platform funziona bene con i progetti Ionic standard. Ma i progetti che seguono una qualsiasi struttura di directory non standard possono incontrare alcuni ostacoli. Questo documento fornisce i passaggi per utilizzare Ionic Platform nei progetti Ionic creati usando Yeoman.

Ambito del documento:

Questo documento illustra i passaggi fondamentali per la creazione di un progetto Ionic utilizzando Yeoman e integrandolo con la piattaforma Ionic utilizzando il client Web della piattaforma Ionic. Questo documento illustra i passaggi fondamentali per utilizzare Ionic Deploy, Ionic Analytics e Ionic Push.

Destinatari:

Il pubblico previsto per questo documento è gli sviluppatori di app Web / Mobile, con esperienza sia per principianti che per esperti, che hanno familiarità con i prerequisiti seguenti.

Prerequisiti:

Dovresti avere familiarità con i seguenti framework / strumenti prima di provare questo documento.

- AngularJs: <https://docs.angularjs.org/guide>
- IonicFramework: <http://ionicframework.com/docs/guide>
- Yeoman: <http://yeoman.io/codelab/index.html>
- Generatore ionic: <https://github.com/diegonetto/generator-ionic>
- Piattaforma ionic: <https://ionic.io/platform>

Generatore di quadri ionic



Un generatore per Ionic Framework di Yeoman, lo strumento di Scaffolding del Web per le moderne applicazioni web

Node.js è un runtime JavaScript creato sul motore JavaScript V8 di Chrome. npm è il gestore di pacchetti per JavaScript. Scarica e installa Node (e npm) da <http://nodejs.org>

```
$ npm install npm -g
$ npm install -g yo
```

Yeoman ti aiuta a dare il via a nuovi progetti, a prescrivere best practice e strumenti per aiutarti a rimanere produttivo.

```
$ yo ionic [app-name]
```

In *package.json* include quanto segue in devDependencies

```
"grunt-string-replace": "^1.2.1"
```

In *bower.json* includere quanto segue nelle dipendenze

```
"ionic-platform-web-client": "^0.7.1"
```

In *Gruntfile.js* cambia la cartella degli *script* in *'js'* . Cambia anche in *index.html* se necessario.

```
grunt.initConfig({  yeoman: {.....  
  scripts: 'js',  
  ..... } })
```

Quindi corri

```
$ bower install && npm install  
$ grunt  
$ grunt serve  
  
$ cordova platform add android  
$ grunt build:android --debug
```

ionico-piattaforma web-client



Un client web che fornisce interazioni con la piattaforma Ionic.

Abbiamo bisogno di un codice per consentire alla tua app di parlare con la piattaforma Ionic. Abbiamo bisogno di aggiungere il client web della piattaforma Ionic per l'app Ionic per l'interfaccia con i plugin e la piattaforma Ionic.io.

```
$ ionic io init
```

Nel tuo *app.js* aggiungi la dipendenza del modulo *'ionic.service.core'* . In *Gruntfile.js* aggiungere il task grunt *'ionicSettings'* come indicato di seguito.

```
grunt.initConfig({  
  ionicSettings: JSON.stringify(grunt.file.readJSON('./.io-config.json')),  
  
  ionicIoBundlePath: 'www/bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js',  
  
  'string-replace': {  
    ionicSettings: {  
      files: {  
        '<%= ionicIoBundlePath %>': '<%= ionicIoBundlePath %>',  
      },  
      options: {  
        replacements: [  
          {  
            pattern:
```

```

        "IONIC_SETTINGS_STRING_START";"IONIC_SETTINGS_STRING_END"',
        replacement:
        "IONIC_SETTINGS_STRING_START";var settings =<%= ionicSettings %>; return { get:
function(setting) { if (settings[setting]) { return settings[setting]; } return null; }
};"IONIC_SETTINGS_STRING_END";'
    }
  ]
}
},
copy: {
  ionicPlatform:{
    expand: true,
    cwd: 'app/bower_components/ionic-platform-web-client/dist/',
    src: ['**'],
    dest: 'www/bower_components/ionic-platform-web-client/dist'
  }
}
});

grunt.registerTask('ionicSettings', ['copy:ionicPlatform','string-replace:ionicSettings']);

```

Aggiungi *"ionicSettings"* in *init* e *comprimi* le attività dopo la *copia* . In *index.html* sposta il tag sottostante dopo tutte le dichiarazioni dei tag.

```
<script src="bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js"></script>
```

Quindi corri

```
$ Grunt serve
```

Distribuzione ionica



Invia aggiornamenti in tempo reale alle app di produzione e gestisci la cronologia delle versioni.

Ionic Deploy ti consente di aggiornare la tua app su richiesta, per eventuali modifiche che non richiedono modifiche binarie, risparmiando giorni o addirittura settimane di attesa. Segui la procedura seguente per configurare Ionic Deploy per la tua app.

In *Gruntfile.js* aggiungi l'attività grunt *"deploy"* come indicato di seguito.

```

grunt.registerTask('deploy', function () {
  return grunt.task.run(['init', 'ionic:upload' + this.args.join()]);
});

```

quindi corri

```
$ ionic plugin add ionic-plugin-deploy
```

Codice di distribuzione ionico:

```
var deploy = new Ionic.Deploy();

// Check Ionic Deploy for new code
deploy.check().then(function(hasUpdate) {
}, function(err) {
});

// Update app code with new release from Ionic Deploy
deploy.update().then(function(result) {
}, function(error) {
}, function(progress) {
});
```

Distribuzione degli aggiornamenti:

Invia nuovo codice per la tua app.

Crea un apk e installa la tua app. Apporta poche modifiche al tuo codice e distribuisce le modifiche usando ' *grunt deploy* '. Quindi aggiornalo dalla tua app.

Puoi anche distribuirlo dalla dashboard *apps.ionic.io* . È possibile distribuire l'app senza il parametro *deploy* . Quindi, nella dashboard puoi aggiungere i metadati e i dettagli della versione e distribuire l'app da lì.

```
$ grunt build:android --debug

$ grunt deploy --note "release notes"
$ grunt deploy --note "release notes" --deploy=production
```

Analytics ionico



Visualizza il feed dal vivo degli eventi o il numero grezzo / univoco di eventi / utenti nel tempo.

Quanti utenti ci sono nella tua app adesso? Quanti di questi useranno la tua app domani o la prossima settimana? Senza informazioni, non hai modo di dire se la tua app viene utilizzata nei modi che ti aspetti. Segui la procedura seguente per configurare Ionic Analytics per la tua app.

Nel tuo *app.js* aggiungi la dipendenza del modulo ' *ionic.service.analytics* ' dopo *ionic.service.core*

Esegui il metodo di registrazione analitica nella funzione di esecuzione del nostro modulo.

```
$ionicAnalytics.register();
```

In Ionic Analytics, ogni azione tracciata che un utente effettua nella tua app è rappresentata da un oggetto evento. Un evento è una singola azione eseguita in un momento specifico. Per tenere traccia dei tuoi eventi, chiama `$ionicAnalytics.track(eventType, eventData)` ogni volta che si verifica un'azione.

```
$ionicAnalytics.track('User Login', {  
  user: $scope.user  
});
```

La direttiva `ion-track-tap` invia un evento quando viene toccato il suo elemento host. La direttiva `ion-track-data` associa i dati dell'evento.

```
<button ion-track-tap="eventType" ion-track-data="expression"></button>
```

Nel dashboard *dell'app.ionic.io* puoi visualizzare i seguenti dati di analisi,

Eventi: visualizza il numero grezzo di eventi nel tempo o il numero di utenti unici che hanno completato un evento. Un evento può essere qualsiasi cosa, da un utente che carica l'app, a confermare un acquisto.

Canalizzazioni: una canalizzazione è una sequenza di azioni che ti aspetti che gli utenti eseguano nella tua app, raggiungendo un obiettivo definito. L'uso attento delle canalizzazioni ti aiuterà a migliorare i tassi di conversione.

Segmenti: visualizza gli eventi nel tempo, raggruppati in base a una proprietà specificata. Oppure, calcola la percentuale di eventi che corrispondono a una determinata proprietà. I segmenti ti aiutano a capire la tua base di utenti e vedere come cambiano le proprietà nel tempo.

Conservazione: tieni traccia di quanto tempo gli utenti sono attivi sulla tua app prima che smettano di usarla. In alternativa, identifica il tempo necessario agli utenti per raggiungere un obiettivo definito, ad esempio una vendita completata.

Pulse: un feed dal vivo degli eventi che arrivano dai tuoi utenti.

Ionic Push



Invia notifiche push mirate e automatizzate ai tuoi utenti.

Ionic Push consente di creare notifiche push mirate tramite un semplice dashboard che verrà

inviato automaticamente quando gli utenti soddisfano criteri specifici e offre una semplice API per inviare notifiche push dai propri server.

Profili push Android:

Le notifiche push di Android utilizzano il servizio *Google Cloud Messaging* (GCM). Apri la [Google Developers Console](#) e crea un progetto. Copia il *numero del tuo progetto* . Questo sarà l' *ID mittente GCM* o il **Numero progetto GCM** .

Nella sezione *Gestione API* , attiva l' *API di Google Cloud Messaging* . Quindi accedere alla sezione *Credenziali* e selezionare *Crea credenziali*, quindi selezionare *Chiave API*, quindi *Chiave server*. Assegna un nome alla chiave API e lascia vuoto il campo *Accetta richieste da ...* e fai clic su *Crea* . Salva la tua **chiave API** !

Autenticazione:

Vai alla dashboard della tua app su [Ionic Platform](#) e vai su *Impostazioni -> Certificati* . Se non lo hai già fatto, crea un nuovo profilo di sicurezza, quindi premi *Modifica* . Annotare il **tag del profilo** .

Ora, fai clic sulla scheda *Android* e trova la sezione contrassegnata con *Google Cloud Messaging* , inserisci la *chiave API* generata nella Google Developer Console, quindi fai *clic* su *Salva* . Vai a *Impostazioni -> Chiavi API* . Sotto *Token API* , crea un nuovo token e copialo. Questo sarà il tuo **token API** .

```
$ ionic plugin add phonegap-plugin-push --variable SENDER_ID="GCM_PROJECT_NUMBER"
$ ionic config set gcm_key <your-gcm-project-number>
$ ionic config set dev_push false
$ ionic io init
```

Nota: phonegap-plugin-push richiede il repository del supporto Android versione 32+

Nel tuo *app.js* aggiungi la dipendenza del modulo ' *ionic.service.push* ' dopo *ionic.service.core*

Codice Ionico Push:

Inizializza il servizio e registra il tuo dispositivo nella funzione di esecuzione del modulo. Avrai bisogno del token del dispositivo che è registrato dall'utente per l'invio di notifiche all'utente.

```
$ionicPush.init({
  debug: true,
  onNotification: function (notification) {
    console.log('token:', notification.payload);
  },
  onRegister: function (token) {
    console.log('Device Token:', token);
    $ionicPush.saveToken(token); // persist the token in the Ionic Platform
  }
});

$ionicPush.register();
```

quindi corri

```
$ grunt build:android --debug
```

Ionic Push ti consente di creare notifiche push mirate attraverso la dashboard. Puoi anche inviare notifiche dal server nel formato seguente.

```
curl -X POST -H "Authorization: Bearer API_TOKEN" -H "Content-Type: application/json" -d '{
  "tokens": ["DEVICE_TOKEN"],
  "profile": "PROFILE_TAG",
  "notification": {
    "message": "Hello World!"
  }
  "android": {
    "title": "Hi User",
    "message": "An update is available for your App",
    "payload": {
      "update": true
    }
  }
}' "https://api.ionic.io/push/notifications"
```

Nota: i passaggi per configurare Ionic Push per iOS sono gli stessi tranne per la creazione dei profili push. Per creare profili push iOS, consulta <http://docs.ionic.io/v2.0.0-beta/docs/ios-push-profiles>

App di esempio



Scarica l'app di esempio qui .

Un'app campione è allegata qui per riferimento.

```
IonicApp:
|
|  bower.json
|  Gruntfile.js
|  package.json
|
└─ app
    |  index.html
    |
    └─ js
        |  app.js
        |  controllers.js
        |
    └─ templates
        |  home.html
        |  menu.html
```

Nota: questo non è un progetto autonomo. Il codice fornito è solo per il confronto con un progetto creato e implementato utilizzando le procedure sopra riportate in questo documento, in caso di problemi o errori.

Leggi Iniziare con il framework ionico online: <https://riptutorial.com/it/ionic-framework/topic/893/iniziare-con-il-framework-ionico>

Capitolo 2: Accesso alla fotocamera e alla libreria fotografica del dispositivo dall'applicazione ionica

Osservazioni

rimanda anche a questo [link](#)

Examples

Apri camera e galleria fotografica

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="Content-Security-Policy" content="default-src *; script-src 'self' 'unsafe-inline' 'unsafe-eval' *; style-src 'self' 'unsafe-inline' *"/>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">
    <title></title>

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">

    <!-- IF using Sass (run gulp sass first), then uncomment below and remove the CSS includes above
    <link href="css/ionic.app.css" rel="stylesheet">
    -->

    <!-- ionic/angularjs js -->
    <script src="lib/ionic/js/ionic.bundle.js"></script>

    <!-- cordova script (this will be a 404 during development) -->
    <script src="js/ng-cordova.min.js"></script>
    <script src="cordova.js"></script>

    <!-- your app's js -->
    <script src="js/app.js"></script>
    <script src="js/controllers.js"></script>
    <script src="js/services.js"></script>
  </head>
  <body ng-app="starter">
    <ion-content class="has-header padding" ng-controller="ImageController">
      <button class="button button-full button-energized" ng-click="addMedia()">
        Add image
      </button>
      <button class="button button-full button-positive" ng-click="sendEmail()">
        Send mail
    </ion-content>
  </body>
</html>
```

```

</button>
<br><br>
<ion-scroll direction="y" style="height:200px; min-height: 200px; overflow: scroll; white-
space: nowrap;">
  
</ion-scroll>

</ion-content>
</body>
</html>

```

controller.js

```

angular.module('starter')

.controller('ImageController', function($scope, $cordovaDevice, $cordovaFile, $ionicPlatform,
$cordovaEmailComposer, $ionicActionSheet, ImageService, FileService) {

  $ionicPlatform.ready(function() {
    $scope.images = FileService.images();
    $scope.$apply();
  });

  $scope.addMedia = function() {
    $scope.hideSheet = $ionicActionSheet.show({
      buttons: [
        { text: 'Take photo' },
        { text: 'Photo from library' },
        { text: '<b ng-disabled="user.length<1">Delete</b>',
          type: 'input type="file"'
        },
      ],
      titleText: 'Add images',
      cancelText: 'Cancel',
      buttonClicked: function(index) {
        $scope.addImage(index);
      }
    });
  }

  $scope.addImage = function(type) {
    $scope.hideSheet();
    ImageService.handleMediaDialog(type).then(function() {
      $scope.$apply();
    });
  }

  $scope.sendEmail = function() {
    if ($scope.images != null && $scope.images.length > 0) {
      var mailImages = [];
      var savedImages = $scope.images;
      for (var i = 0; i < savedImages.length; i++) {
        mailImages.push('base64:attachment'+i+'.jpg/' + savedImages[i]);
      }
      $scope.openMailComposer(mailImages);
    }
  }

  $scope.openMailComposer = function(attachments) {
    var bodyText = '<html><h2>My Images</h2></html>';

```

```

var email = {
  to: '',
  attachments: attachments,
  subject: 'Devdactic Images',
  body: bodyText,
  isHtml: true
};

$cordovaEmailComposer.open(email, function(){
  console.log('email view dismissed');

}, this);
});

```

service.js

```

angular.module('starter')

.factory('FileService', function() {
  var images;
  var IMAGE_STORAGE_KEY = 'dav-images';

  function getImages() {
    var img = window.localStorage.getItem(IMAGE_STORAGE_KEY);
    if (img) {
      images = JSON.parse(img);
    } else {
      images = [];
    }
    return images;
  };

  function addImage(img) {
    images.push(img);
    window.localStorage.setItem(IMAGE_STORAGE_KEY, JSON.stringify(images));
  };

  return {
    storeImage: addImage,
    images: getImages
  }
});

.factory('ImageService', function($cordovaCamera, FileService, $q, $cordovaFile) {

  function optionsForType(type) {
    var source;
    switch (type) {
      case 0:
        source = Camera.PictureSourceType.CAMERA;
        break;
      case 1:
        source = Camera.PictureSourceType.PHOTOLIBRARY;
        break;
    }
    return {
      quality: 90,
      destinationType: Camera.DestinationType.DATA_URL,
      sourceType: source,

```

```

    allowEdit: false,
    encodingType: Camera.EncodingType.JPEG,
    popoverOptions: CameraPopoverOptions,
    saveToPhotoAlbum: false,
    correctOrientation:true
  };
}

function saveMedia(type) {
  return $q(function(resolve, reject) {
    var options = optionsForType(type);

    $cordovaCamera.getPicture(options).then(function(imageBase64) {
      FileService.storeImage(imageBase64);
    });
  })
}
return {
  handleMediaDialog: saveMedia
}
});

```

Per l'esempio di Ionic 3

Installa i plugin Cordova e Ionic Native:

```

$ ionic cordova plugin add cordova-plugin-camera
$ npm install --save @ionic-native/camera

```

La tua `app.module.ts` dovrà iniettare una telecamera:

```

import { Camera } from '@ionic-native/camera';
.....
@NgModule({
  declarations: [
    MyApp
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    .....
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp
  ],
  providers: [
    StatusBar,
    SplashScreen,
    Camera,
    {provide: ErrorHandler, useClass: IonicErrorHandler},
    .....
  ]
})
export class AppModule {}

```

La fotocamera può essere utilizzata facilmente con il foglio di azione con Ionic 3, la tua pagina.ts

sarà come di seguito:

```
import { Camera, CameraOptions } from '@ionic-native/camera';
.....

export class YourPage {

  private base64:any;

  constructor(private camera: Camera,private actionsheetCtrl: ActionSheetController) { }

  cameragalleryfun(){

    let actionSheet = this.actionsheetCtrl.create({
      title: 'Camera-Gallery',
      cssClass: 'action-sheets-basic-page',
      buttons: [
        {
          text: 'Camera',
          icon: 'camera',
          handler: () => {
            //console.log('Camera');
            const options: CameraOptions = {
              quality: 60,
              destinationType: this.camera.DestinationType.DATA_URL,
              encodingType: this.camera.EncodingType.JPEG,
              mediaType: this.camera.MediaType.PICTURE,
              sourceType : this.camera.PictureSourceType.CAMERA,
              targetWidth: 500,
              saveToPhotoAlbum: false,
              correctOrientation:true
            }

            this.camera.getPicture(options).then((imageData) => {
              this.base64 = 'data:image/jpeg;base64,' + imageData;
            }, (err) => {
              // Handle error
            });
          }
        },
        {
          text: 'Gallery',
          icon: 'images',
          handler: () => {
            //console.log('Gallery');
            const options: CameraOptions = {
              quality: 60,
              destinationType: this.camera.DestinationType.DATA_URL,
              encodingType: this.camera.EncodingType.JPEG,
              mediaType: this.camera.MediaType.PICTURE,
              sourceType : this.camera.PictureSourceType.PHOTOLIBRARY,
              targetWidth: 500,
              saveToPhotoAlbum: false,
              correctOrientation:true
            }

            this.camera.getPicture(options).then((imageData) => {
              this.base64 = 'data:image/jpeg;base64,' + imageData;
            }, (err) => {
              // Handle error
            });
          }
        }
      ]
    });
  }
}
```

```

        });
    }
},
{
  text: 'Cancel',
  role: 'cancel', // will always sort to be on the bottom
  icon: 'close',
  handler: () => {
    //console.log('Cancel clicked');
  }
}
]
});
actionSheet.present();

}
}

```

Chiama la funzione `cameragalleryfun` da qualsiasi evento come fare clic sul pulsante, questo restituirà la stringa base64 per l'immagine. Più opzioni possono essere applicate. per l'opzione extra vedi: <https://github.com/apache/cordova-plugin-camera>

Leggi [Accesso alla fotocamera e alla libreria fotografica del dispositivo dall'applicazione ionica](https://riptutorial.com/it/ionic-framework/topic/7086/accesso-alla-fotocamera-e-alla-libreria-fotografica-del-dispositivo-dall-applicazione-ionica) online: <https://riptutorial.com/it/ionic-framework/topic/7086/accesso-alla-fotocamera-e-alla-libreria-fotografica-del-dispositivo-dall-applicazione-ionica>

Capitolo 3: Collegamento di Ionic con qualsiasi database

Examples

Non puoi farlo direttamente dalla struttura ionica

La cosa è; non è possibile collegare Ionic a nessun database (MySQL, Postgres, MSSQL, ...) **direttamente** . La parola chiave qui è *direttamente* .

No, non c'è soluzione, nessuna magia coinvolta, non è solo il modo in cui dovrebbe funzionare. Le opere ioniche in cima a Angolare e Angolare è una struttura di frontend.

Tuttavia, il modo in cui dovresti farlo è che in pratica crei un'API (RESTful) sul tuo server.

Molto probabilmente questo sarà realizzato con un linguaggio server (PHP, Go, Python, ...) che parlerà direttamente al tuo database e lo interrogherà.

Dopo aver scritto la tua API (RESTful) puoi consumarla tramite i tuoi servizi in Angular utilizzando il servizio `$resource` o `$http` Angular.

Un esempio di consumo dell'API Giphy con il servizio `$http` Angular:

```
var search = 'cats';
var link = 'http://api.giphy.com/v1/gifs/search?api_key=dc6zaTOxFJmzC&q=' + search;

$http.get(link).then(function(result) {
  console.log(result);
});
```

Leggi [Collegamento di Ionic con qualsiasi database online](https://riptutorial.com/it/ionic-framework/topic/4003/collegamento-di-ionic-con-qualsiasi-database): <https://riptutorial.com/it/ionic-framework/topic/4003/collegamento-di-ionic-con-qualsiasi-database>

Capitolo 4: Come utilizzare le funzioni di EcmaScript 6 in Ionic?

Examples

Usando gulp-babel e gulp-idraulico

Ionic usa Gulp, quindi installa gulp-babel e gulp-idraulico.

```
npm install --save-dev gulp-babel gulp-plumber
```

Aggiungi babel a `gulpfile.js` modo:

```
//...
var babel = require("gulp-babel");
var plumber = require("gulp-plumber");

var paths = {
  es6: ['./src/es6/*.js'],
  sass: ['./scss/**/*.scss']
};

gulp.task('default', ['babel', 'sass']);

gulp.task("babel", function () {
  return gulp.src(paths.es6)
    .pipe(plumber())
    .pipe(babel())
    .pipe(gulp.dest("www/js"));
});

//...

gulp.task('watch', function() {
  gulp.watch(paths.es6, ['babel']);
  gulp.watch(paths.sass, ['sass']);
});

//...
```

Modifica `ionic.project` :

```
"gulpStartupTasks": [
  "babel",
  "sass",
  "watch"
],
```

Ora quando esegui il `ionic serve`, il codice verrà trascritto per te.

Leggi [Come utilizzare le funzioni di EcmaScript 6 in Ionic? online](https://riptutorial.com/it/ionic-framework/topic/4847/come-utilizzare-le-funzioni-di-ecmascript-6-in-ionic): [https://riptutorial.com/it/ionic-framework/topic/4847/come-utilizzare-le-funzioni-di-ecmascript-6-in-ionic-](https://riptutorial.com/it/ionic-framework/topic/4847/come-utilizzare-le-funzioni-di-ecmascript-6-in-ionic)

Capitolo 5: Componenti CSS ionici

Osservazioni

Ionic ha un sacco di componenti CSS dichiarati pronti per semplificarti la vita mentre codifichi la tua prossima applicazione mobile ibrida. Questi componenti variano da un sistema di griglia di base allo stile delle forme. Questi componenti sono in uso se si sceglie di installare Ionic con i fogli di stile CSS preimpostati.

Una delle funzioni di base che Ionic CSS ti porta in mano è che viene fornito con un set di colori per iniziare, ma come regola generale i colori devono essere sovrascritti. I colori dell'utilità vengono aggiunti per aiutare a impostare una convenzione di denominazione. Potresti chiamarlo un tema di base dell'applicazione. Per personalizzare i colori puoi semplicemente ignorare quelli provenienti dal file CSS `ionic.css`. Inoltre, poiché Ionic è stato creato usando Sass, per maggiore potenza e flessibilità è anche possibile modificare ed estendere le variabili di colore all'interno del file `_variables.scss`.

È possibile impostare un progetto Ionic per utilizzare SASS molto facilmente eseguendo il comando `ionic setup sass` nel terminale.

Puoi trovare la documentazione ufficiale su Ionic CSS qui:

<http://ionicframework.com/docs/components/>

Examples

Sintassi del sistema di griglia di base

Griglia di base

In Ionic è possibile dichiarare le righe impostando la classe di `row` su un elemento. Le righe saranno elementi allineati orizzontalmente e qualsiasi cosa all'interno di questo elemento apparterrà alla riga. All'interno della riga è possibile dichiarare colonne di larghezza diverse. Avete una scelta delle seguenti dichiarazioni.

Classe	Larghezza
<code>.col-10</code>	10%
<code>.col-20</code>	20%
<code>.col-25</code>	25%
<code>.col-33</code>	33,3333%
<code>.col-50</code>	50%

Classe	Larghezza
.col-67	66,6666%
.col-75	75%
.col-80	80%
.col-90	90%

Le colonne del valore massimo possono avere all'interno di una riga 100. Ecco alcuni esempi della griglia di base.

```
<div class="row">
  <div class="col col-50">.col.col-50</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>

<div class="row">
  <div class="col col-75">.col.col-75</div>
  <div class="col">.col</div>
</div>

<div class="row">
  <div class="col">.col</div>
  <div class="col col-75">.col.col-75</div>
</div>

<div class="row">
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

Griglie sfalsate

Puoi anche impostare `col-offset-<value>` sulle colonne. Nell'esempio seguente la colonna di un terzo di una larghezza ha un terzo dell'offset di larghezza, il che significa che sarà largo un terzo della larghezza e centrato nella pagina a causa del suo offset.

```
<div class="row">
  <div class="col col-33 col-offset-33">.col</div>
  <div class="col">.col</div>
</div>
```

Allinea le colonne

Allineare le colonne verticalmente è possibilmente impostando il `col-<align_value>` su una colonna separatamente come questo.

```
<div class="row">
  <div class="col col-top">.col</div>
  <div class="col col-center">.col</div>
  <div class="col col-bottom">.col</div>
  <div class="col">1<br>2<br>3<br>4</div>
</div>
```

Quanto sopra allinea tutte le colonne su se stesse. Se si desidera allineare tutte le colonne all'interno della riga, è possibile impostare il valore di allineamento sulla riga stessa. Nell'esempio seguente tutte le colonne all'interno di questa riga si allineeranno verticalmente nel centro della riga.

```
<div class="row row-center">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">1<br>2<br>3<br>4</div>
</div>
```

Griglia reattiva

Potresti anche volere che le colonne siano reattive e che si impilino l'una sull'altra ad una certa larghezza di vista. Hai tre scelte.

Classe	Breakpoint (circa)
.responsive-sm	Più piccolo di un telefono panoramico
.responsive-md	Più piccolo del tablet ritratto
.responsive-lg	Più piccolo del tablet del paesaggio

In questo esempio, queste colonne si impilano sotto la larghezza di circa un telefono orizzontale.

```
<div class="row responsive-sm">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

Puoi anche formulare le tue domande sui media se questi punti di interruzione non sono adatti a te e / o se hai bisogno di punti di interruzione più specifici.

Sintassi delle voci di base dell'elenco

Quasi ogni applicazione ha una sorta di elenco. Ionic ha le proprie dichiarazioni CSS predefinite per l'elenco delle voci predefinite per semplificare la creazione di elenchi all'interno dell'applicazione. Puoi usare gli elementi HTML e dichiarare una classe per o usare la direttiva

ion-list per renderli. L'esempio di una direttiva è in fondo.

Sintassi CSS dell'elenco di base:

```
<ul class="list">
  <li class="item"></li>
</ul>
```

Elenco con divisori:

```
<div class="list">
  <a class="item" href="#">
    List item
  </a>
  <div class="item item-divider">
    Divider that looks a bit different from items
  </div>
  <a class="item" href="#">
    Another list item
  </a>
</div>
```

Elenca elementi con icone:

```
<div class="list">
  <a class="item item-icon-left" href="#">
    <i class="icon ion-trash-b"></i>
    List item with a trashcan icon on the left
  </a>
</div>
```

Puoi anche impostare le icone su entrambi i lati degli elementi con la seguente sintassi:

```
<div class="list">
  <a class="item item-icon-left item-icon-right" href="#">
    <i class="icon ion-trash-b"></i>
    List item with a trashcan icon on the left and a briefcase icon on the right
    <i class="icon ion-briefcase"></i>
  </a>
</div>
```

Una voce di elenco con pulsante o pulsanti può essere creata in questo modo:

```
<div class="list">
  <div class="item item-button-right">
    Item with a button on the right that has a clock icon in it
    <button class="button button-positive">
      <i class="icon ion-clock"></i>
    </button>
  </div>
</div>
```

È anche possibile creare voci di elenco con avatar, miniature e inserti che creeranno il riempimento attorno agli elementi dell'elenco. Ionic gestisce anche le icone di impostazione in voci

di elenco impostando il padding in base agli elementi dell'elenco.

Anche Ionic ha le sue direttive per le caselle di controllo, i pulsanti di opzione ecc. Ecco un esempio di un elenco di caselle di controllo con Ionic.

```
<ion-list>
  <ion-checkbox ng-model="choice1">Choice 1</ion-checkbox>
  <ion-checkbox ng-model="choice2">Choice 2</ion-checkbox>
</ion-list>
```

Utilizzo di base dei colori di utilità

Preset Ionic CSS avrà un tema e i colori preimpostati. È possibile modificare o sovrascrivere i valori di base in `ionic.css` o nel proprio file CSS personalizzato. Puoi anche definirli con SASS e usare SASS in Ionic hai solo bisogno di eseguire il comando `ionic setup sass` nel tuo terminale.

Uso di base dei colori in un pulsante. La classe `button-<phrase>` renderà il pulsante background e confina con il colore del tema impostato.

```
<button class="button button-positive">
  button-positive
</button>

<button class="button button-calm">
  button-calm
</button>

<button class="button button-balanced">
  button-balanced
</button>
```

Le tue scelte di prefisso CSS sono le seguenti:

- `<element>-light`
- `<element>-stable`
- `<element>-positive`
- `<element>-calm`
- `<element>-balanced`
- `<element>-energized`
- `<element>-assertive`
- `<element>-royal`
- `<element>-dark`

Queste classi possono essere aggiunte anche per esempio in badge ecc. Ecco un esempio di badge:

```
<span class="badge badge-positive">Positive badge</span>
```

Leggi Componenti CSS ionic online: <https://riptutorial.com/it/ionic-framework/topic/6689/componenti-css-ionic>

Capitolo 6: Crea finestra di dialogo in Ionic

Parametri

parametri	Dettaglio
title: '',	// String. Il titolo del popup.
cssClass: '',	// String, il nome della classe CSS personalizzata
subTitle: '',	// String (opzionale). Il sottotitolo del popup.
template: '',	// String (opzionale). Il modello html da inserire nel corpo del popup.
templateUrl: '' ,	// String (opzionale). L'URL di un modello html da inserire nel corpo del popup.
scope: null,	// Scope (opzionale). Un ambito per collegare al contenuto popup.

Osservazioni

Il servizio Ionic Popup consente di creare e mostrare a livello di programmazione finestre popup che richiedono all'utente di rispondere per continuare.

Examples

Crea finestra di dialogo in Ionic

```
// An alert dialog
$scope.showAlert = function() {
  var alertPopup = $ionicPopup.alert({
    title: 'Don\'t eat that!',
    template: 'It might taste good'
  });

  alertPopup.then(function(res) {
    console.log('Hello your first example.');
```

Leggi Crea finestra di dialogo in Ionic online: <https://riptutorial.com/it/ionic-framework/topic/6461/crea-finestra-di-dialogo-in-ionic>

Capitolo 7: Distribuisci Ionic come sito web

Examples

Copia semplicemente la cartella `www` sul tuo server web

Ionic 1.2 [supporta](#) ufficialmente [la distribuzione come sito web](#)

Se non utilizzi alcun plug-in Cordova, non c'è alcun problema (se lo desideri davvero) per caricare i contenuti della cartella `www` sul tuo server e voila - avrai la stessa app.

Tuttavia, è importante notare che Ionic 1 non è concepito per tale uso e che gli utenti del tuo "sito web" dovranno avere il browser più recente per visualizzare correttamente il "sito web" (non suddiviso per alcune funzionalità che Ionic sta usando sia in CSS che in HTML che alcuni browser meno recenti non supportano).

Leggi [Distribuisci Ionic come sito web online](#): <https://riptutorial.com/it/ionic-framework/topic/4001/distribuisci-ionic-come-sito-web>

Capitolo 8: Esegui l'app Ionic on Emulator o sul tuo telefono

Examples

Esegui l'app Ionic on Emulator o sul tuo telefono

1. Aggiungi un target di piattaforma

iOS:

```
$ ionic platform add ios
```

Android:

```
$ ionic platform add android
```

Finestre:

```
$ ionic platform add windows
```

2. Crea la tua app

iOS:

```
$ ionic build ios
```

Android:

```
$ ionic build android
```

Finestre:

```
$ ionic build windows
```

Live Reload App During Development (beta)

La `run` o `emulate` comando distribuire l'applicazione per i dispositivi di piattaforma / emulatori specificati. Puoi anche eseguire il **live ricaricamento** sul dispositivo della piattaforma specificato aggiungendo l'opzione `--livereload`. La funzionalità live reload è simile a `ionic serve`, ma invece

di sviluppare e eseguire il debug di un'app utilizzando un browser standard, l'app ibrida compilata stessa sta osservando eventuali modifiche ai suoi file e ricaricando l'app quando necessario. Ciò riduce il requisito di ricreare costantemente l'app per piccole modifiche. Tuttavia, qualsiasi modifica ai plug-in richiederà comunque una ricostruzione completa. Affinché la ricarica live funzioni, la macchina e il dispositivo dev devono trovarsi sulla stessa rete locale e il dispositivo deve supportare [socket Web](#) .

Con il live reload abilitato, i registri della console di un'app possono anche essere stampati sul terminale / prompt dei comandi includendo l'opzione `--consolelogs o -c` . Inoltre, i log delle richieste del server di sviluppo possono essere stampati usando `--serverlogs o -s options`.

Flag / opzioni della riga di comando per l' `run` e l' `emulate`

```
[--livereload|-l] ..... Live Reload app dev files from the device (beta)
[--consolelogs|-c] ..... Print app console logs to Ionic CLI (live reload req.)
[--serverlogs|-s] ..... Print dev server logs to Ionic CLI (live reload req.)
[--port|-p] ..... Dev server HTTP port (8100 default, live reload req.)
[--livereload-port|-i] .. Live Reload port (35729 default, live reload req.)
[--debug|--release]
```

Mentre il server è in esecuzione per la ricarica live, è possibile utilizzare i seguenti comandi all'interno della CLI:

```
restart or r to restart the client app from the root
goto or g and a url to have the app navigate to the given url
consolelogs or c to enable/disable console log output
serverlogs or s to enable/disable server log output
quit or q to shutdown the server and exit
```

3. Emulazione della tua app

Distribuisce l'app Ionic su un emulatore di piattaforma specificato. Questo è semplicemente un alias per `run --emulator` .

iOS:

```
$ ionic emulate ios [options]
```

Android:

```
$ ionic emulate android [options]
```

Finestre:

```
$ ionic emulate windows [options]
```

Durante l'emulazione di app in AVD o cellulari, puoi ispezionare l'app in Chrome Browser. Digitare il seguente comando nella barra degli indirizzi del browser Chrome.

4. Esecuzione della tua app

Distribuisce l'app Ionic su dispositivi con piattaforma specificata. Se un dispositivo non viene trovato verrà quindi distribuito su un emulatore / simulatore.

iOS:

```
$ ionic run ios [options]
```

Android:

```
$ ionic run android [options]
```

Finestre:

```
$ ionic run windows [options]
```

4.1. Specifica il tuo obiettivo

```
$ ionic run [ios/android/windows] --target="[target-name]"
```

Puoi controllare il nome di destinazione del tuo dispositivo / emulatore con `$ adb devices`.

Leggi [Esegui l'app Ionic on Emulator o sul tuo telefono online: https://riptutorial.com/it/ionic-framework/topic/4175/esegui-l-app-ionic-on-emulator-o-sul-tuo-telefono](https://riptutorial.com/it/ionic-framework/topic/4175/esegui-l-app-ionic-on-emulator-o-sul-tuo-telefono)

Capitolo 9: Estensioni Ionic AngularJS

Osservazioni

Ionic offre una varietà di estensioni Javascript AngularJS da utilizzare. Queste estensioni possono essere qualsiasi cosa, dai normali input di form alle finestre modali e rendono la codifica della tua app di base molto più veloce usando queste estensioni pronte.

Examples

Input di forma

Gli input ionici non sono diversi dai normali input HTML, ma hanno uno stile diverso e sono usati come una direttiva. Puoi anche utilizzare i normali input HTML nelle app Ionic. Ecco alcuni esempi di base che Ionic offre pronto all'uso.

casella di controllo:

```
<ion-checkbox ng-model="checkbox">Label</ion-checkbox>
```

Pulsante di opzione:

```
<ion-radio ng-model="radio" ng-value="'radiovalue'">Value</ion-radio>
```

Toggle:

```
<ion-toggle ng-model="toggle" toggle-class="toggle-calm">Toggle</ion-toggle>
```

Finestre modali

Ionic ha la propria estensione per la visualizzazione di una finestra modale. I modali possono essere creati inserendo il modello direttamente nella vista con un tag `<script>` o utilizzando un file modello separato. In questo esempio stiamo assumendo che tu abbia un file html chiamato `modal-template.html` in una cartella chiamata `templates`. Si imposta il percorso del modello nella funzione di inizializzazione modale con `fromTemplateUrl`.

Creazione dell'oggetto modale nell'ambito

HTML

```
<ion-modal-view>
  <ion-header-bar>
    <h1>Modal title</h1>
  </ion-header-bar>
```

```
<ion-content>
  <p>Modal content</p>
</ion-content>
</ion-modal-view>
```

controllore

```
$ionicModal.fromTemplateUrl('/templates/modal-template.html', {
  scope: $scope,
  animation: 'slide-in-up'
}).then(function(modal) {
  $scope.modal = modal;
});
```

Controlla il modale

È quindi possibile controllare il modale con i seguenti comandi.

Apri modale

```
$scope.modal.show();
```

Chiudi modale

```
$scope.modal.hide();
```

Rimuovi modale

```
$scope.modal.remove();
```

Eventi modali

È possibile ascoltare gli eventi modali con le seguenti funzioni.

Il modale è nascosto

```
$scope.$on('modal.hidden', function() {
  // Do something when modal is hidden
});
```

Il modale è stato rimosso

```
$scope.$on('modal.removed', function() {
  // Do something when modal is removed
});
```

Leggi Estensioni Ionic AngularJS online: <https://riptutorial.com/it/ionic->

Capitolo 10: Ganci Ionico CLI

Osservazioni

introduzione

I ganci sono pezzi di codice che Cordova CLI esegue in determinati punti della build dell'applicazione Cordova / Ionic. Gli hook possono essere utilizzati ad esempio per manipolare i file nel nostro progetto, aggiungere automaticamente plug-in alla tua applicazione o, come nell'esempio sopra, verificare la presenza di errori di codice nei tuoi file.

Nota : si consiglia vivamente di scrivere i propri hook usando Node.js in modo che siano multiplatforma ma è possibile scriverli anche per esempio in [Javascript](#) .

Tipi di gancio

Sono supportati i seguenti tipi di hook e l'ordine di esecuzione è abbastanza autoesplicativo in base al nome.

```
after_build
after_compile
after_docs
after_emulate
after_platform_add
after_platform_rm
after_platform_ls
after_plugin_add
after_plugin_ls
after_plugin_rm
after_plugin_search
after_prepare
after_run
after_serve
before_build
before_compile
before_docs
before_emulate
before_platform_add
before_platform_rm
before_platform_ls
before_plugin_add
before_plugin_ls
before_plugin_rm
before_plugin_search
before_prepare
before_run
before_serve
pre_package/ <-- Applicable to Windows 8 and Windows Phone only. This hook is deprecated.
```

Modi per definire i ganci:

Gli hook possono essere definiti nel `config.xml` del progetto usando gli elementi `<hook>` , ad esempio:

```
<hook type="after_build" src="scripts/appAfterBuild.js" />
```

Come sviluppatore di plugin puoi definire gli hook script usando gli elementi `<hook>` in un file `plugin.xml` come questo:

```
<hook type="after_build" src="scripts/afterBuild.js" />
```

`before_plugin_install` , `after_plugin_install` , i plugin `before_plugin_uninstall` verranno attivati esclusivamente per il plug-in installato / disinstallato.

Nota : il posizionamento degli hook nella directory `root/hooks` è considerato deprecato a favore degli elementi hook in `config.xml` e `plugin.xml` . Se tuttavia si utilizza questo approccio, ricordare di impostare i diritti di esecuzione sui file nella cartella `root/hooks` .

La documentazione per i ganci Cordova è disponibile [qui](#) .

Examples

Verifica degli errori nei file Javascript in `before_prepare` utilizzando `jshint`

```
#!/usr/bin/env node

var fs = require('fs');
var path = require('path');
var jshint = require('jshint').JSHINT;
var async = require('async');

var foldersToProcess = [
  'js'
];

foldersToProcess.forEach(function(folder) {
  processFiles("www/" + folder);
});

function processFiles(dir, callback) {
  var errorCount = 0;
  fs.readdir(dir, function(err, list) {
    if (err) {
      console.log('processFiles err: ' + err);
      return;
    }
    async.eachSeries(list, function(file, innercallback) {
      file = dir + '/' + file;
      fs.stat(file, function(err, stat) {
        if (!stat.isDirectory()) {
```

```

        if (path.extname(file) === ".js") {
            lintFile(file, function(hasError) {
                if (hasError) {
                    errorCount++;
                }
                innercallback();
            });
        } else {
            innercallback();
        }
    } else {
        processFiles(file);
    }
});
}, function(error) {
    if (errorCount > 0) {
        process.exit(1);
    }
});
});
}

function lintFile(file, callback) {
    console.log("Linting " + file);
    fs.readFile(file, function(err, data) {
        if (err) {
            console.log('Error: ' + err);
            return;
        }
        if (jshint(data.toString())) {
            console.log('File ' + file + ' has no errors.');
```

```

            console.log('-----');
```

```

            callback(false);
        } else {
            console.error('Errors in file ' + file);
            var out = jshint.data(),
                errors = out.errors;
            for (var j = 0; j < errors.length; j++) {
                console.error(errors[j].line + ':' + errors[j].character + ' -> ' + errors[j].reason +
                    ' -> ' + errors[j].evidence);
            }
            console.error('-----');
```

```

            setTimeout(function() {
                callback(true);
            }, 10);
        }
    });
}

```

Leggi Ganci Ionico CLI online: <https://riptutorial.com/it/ionic-framework/topic/6520/ganci-ionico-cli>

Capitolo 11: Inizia a creare app su Ionic

Examples

Avvio di un'applicazione ionica

Avvio di un'applicazione ionica

```
$ ionic start myapp [template]
```

I modelli di avvio possono provenire da un modello con nome, un repository Github, un Codepen o una directory locale. Un modello iniziale è ciò che diventa la directory `www` all'interno del progetto Cordova.

Starter modello con nome

- [schede](#) (predefinito)
- [sidemenu](#)
- [vuoto](#)

Github Repo starters

- Qualsiasi URL di repository Github, ad esempio: <https://github.com/driftyco/ionic-starter-tabs>
- I modelli nominati sono semplicemente alias dei repository di avvio ionici

Codepen URL starters

- Qualsiasi URL Codepen, ad esempio: [<http://codepen.io/ionic/pen/odqCz>][1]
- [Demo di Codepen ionici](#)

Avviatori di directory locali:

- Percorso relativo o assoluto di una directory locale

Flag / opzioni della riga di comando:

```
--appname, -a ..... Human readable name for the app (Use quotes around the name)
--id, -i ..... Package name set in the <widget id> config ex: com.mycompany.myapp
```

```
--no-cordova, -w .... Do not create an app targeted for Cordova
```

Leggi Inizia a creare app su Ionic online: <https://riptutorial.com/it/ionic-framework/topic/3945/inizia-a-creare-app-su-ionic>

Capitolo 12: Ionic Cloud per Yeoman Ionic Projects

Examples

Ionic Platform (Ionic Cloud) per Yo (Yeoman) Ionic Projects



Piattaforma ionica :

Costruisci, spinge, distribuisci e ridimensiona le tue app Ionic, in modo semplice.

Descrizione del titolo:

La piattaforma Ionic è una piattaforma cloud per la gestione e il ridimensionamento di app mobili multiplatforma. I servizi integrati consentono a te e al tuo team di creare, distribuire e far crescere le app in modo efficiente.

Obiettivo del documento:

Ionic Platform funziona bene con i progetti Ionic standard. Ma i progetti che seguono una qualsiasi struttura di directory non standard possono incontrare alcuni ostacoli. Questo documento fornisce i passaggi per utilizzare Ionic Platform nei progetti Ionic creati usando Yeoman.

Ambito del documento:

Questo documento illustra i passaggi fondamentali per la creazione di un progetto Ionic utilizzando Yeoman e integrandolo con la piattaforma Ionic utilizzando il client Web della piattaforma Ionic. Questo documento illustra i passaggi fondamentali per utilizzare Ionic Deploy, Ionic Analytics e Ionic Push.

Destinatari:

Il pubblico previsto per questo documento è gli sviluppatori di app Web / Mobile, con esperienza sia per principianti che per esperti, che hanno familiarità con i prerequisiti seguenti.

Prerequisiti:

Dovresti avere familiarità con i seguenti framework / strumenti prima di provare questo documento.

- AngularJs: <https://docs.angularjs.org/guide>
- IonicFramework: <http://ionicframework.com/docs/guide>

- Yeoman: <http://yeoman.io/codelab/index.html>
- Generatore ionico: <https://github.com/diegonetto/generator-ionic>
- Piattaforma ionica: <https://ionic.io/platform>

Generatore di quadri ionici



Un generatore per Ionic Framework di Yeoman, lo strumento di Scaffolding del Web per le moderne applicazioni web

Node.js è un runtime JavaScript creato sul motore JavaScript V8 di Chrome. npm è il gestore di pacchetti per JavaScript. Scarica e installa Node (e npm) da <http://nodejs.org>

```
$ npm install npm -g
$ npm install -g yo
```

Yeoman ti aiuta a dare il via a nuovi progetti, a prescrivere best practice e strumenti per aiutarti a rimanere produttivo.

```
$ yo ionic [app-name]
```

In *package.json* include quanto segue in devDependencies

```
"grunt-string-replace": "^1.2.1"
```

In *bower.json* includere quanto segue nelle dipendenze

```
"ionic-platform-web-client": "^0.7.1"
```

In *Gruntfile.js* cambia la cartella degli *script* in *'js'* . Cambia anche in *index.html* se necessario.

```
grunt.initConfig({  yeoman: {.....
  scripts: 'js',
  ..... } })
```

Quindi corri

```
$ bower install && npm install
$ grunt
$ grunt serve

$ cordova platform add android
$ grunt build:android --debug
```

ionico-piattaforma web-client



Un client web che fornisce interazioni con la piattaforma Ionic.

Abbiamo bisogno di un codice per consentire alla tua app di parlare con la piattaforma Ionic. Abbiamo bisogno di aggiungere il client web della piattaforma Ionic per l'app Ionic per l'interfaccia con i plugin e la piattaforma Ionic.io.

```
$ ionic io init
```

Nel tuo *app.js* aggiungi la dipendenza del modulo *'ionic.service.core'*. In *Gruntfile.js* aggiungere il task grunt *'ionicSettings'* come indicato di seguito.

```
grunt.initConfig({
  ionicSettings: JSON.stringify(grunt.file.readJSON('./.io-config.json')),

  ionicIoBundlePath: 'www/bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js',

  'string-replace': {
    ionicSettings: {
      files: {
        '<%= ionicIoBundlePath %>': '<%= ionicIoBundlePath %>',
      },
      options: {
        replacements: [
          {
            pattern:
              '"IONIC_SETTINGS_STRING_START";"IONIC_SETTINGS_STRING_END"',
            replacement:
              '"IONIC_SETTINGS_STRING_START";var settings =<%= ionicSettings %>; return { get:
function(setting) { if (settings[setting]) { return settings[setting]; } return null; }
};"IONIC_SETTINGS_STRING_END";'
          }
        ]
      }
    }
  },

  copy: {
    ionicPlatform: {
      expand: true,
      cwd: 'app/bower_components/ionic-platform-web-client/dist/',
      src: ['**'],
      dest: 'www/bower_components/ionic-platform-web-client/dist'
    }
  }
});

grunt.registerTask('ionicSettings', ['copy:ionicPlatform','string-replace:ionicSettings']);
```

Aggiungi *"ionicSettings"* in *init* e *comprimi* le attività dopo la *copia* . In *index.html* sposta il tag sottostante dopo tutte le dichiarazioni dei tag.

```
<script src="bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js"></script>
```

Quindi corri

```
$ Grunt serve
```

Distribuzione ionica



Invia aggiornamenti in tempo reale alle app di produzione e gestisci la cronologia delle versioni.

Ionic Deploy ti consente di aggiornare la tua app su richiesta, per eventuali modifiche che non richiedono modifiche binarie, risparmiando giorni o addirittura settimane di attesa. Segui la procedura seguente per configurare Ionic Deploy per la tua app.

In *Gruntfile.js* aggiungi l'attività grunt *"deploy"* come indicato di seguito.

```
grunt.registerTask('deploy', function () {  
  return grunt.task.run(['init', 'ionic:upload' + this.args.join()]);  
});
```

quindi corri

```
$ ionic plugin add ionic-plugin-deploy
```

Codice di distribuzione ionico:

```
var deploy = new Ionic.Deploy();  
  
// Check Ionic Deploy for new code  
deploy.check().then(function(hasUpdate) {  
  }, function(err) {  
  });  
  
// Update app code with new release from Ionic Deploy  
deploy.update().then(function(result) {  
  }, function(error) {  
  }, function(progress) {  
  });
```

Distribuzione degli aggiornamenti:

Invia nuovo codice per la tua app.

Crea un apk e installa la tua app. Apporta poche modifiche al tuo codice e distribuisce le modifiche usando ' *grunt deploy* '. Quindi aggiornalo dalla tua app.

Puoi anche distribuirlo dalla dashboard *apps.ionic.io* . È possibile distribuire l'app senza il parametro *deploy* . Quindi, nella dashboard puoi aggiungere i metadati e i dettagli della versione e distribuire l'app da lì.

```
$ grunt build:android --debug  
  
$ grunt deploy --note "release notes"  
$ grunt deploy --note "release notes" --deploy=production
```

Analytics ionic



Visualizza il feed dal vivo degli eventi o il numero grezzo / univoco di eventi / utenti nel tempo.

Quanti utenti ci sono nella tua app adesso? Quanti di questi useranno la tua app domani o la prossima settimana? Senza informazioni, non hai modo di dire se la tua app viene utilizzata nei modi che ti aspetti. Segui la procedura seguente per configurare Ionic Analytics per la tua app.

Nel tuo *app.js* aggiungi la dipendenza del modulo ' *ionic.service.analytics* ' dopo *ionic.service.core* . Esegui il metodo di registrazione analitica nella funzione di esecuzione del nostro modulo.

```
$ionicAnalytics.register();
```

In Ionic Analytics, ogni azione tracciata che un utente effettua nella tua app è rappresentata da un oggetto evento. Un evento è una singola azione eseguita in un momento specifico. Per tenere traccia dei tuoi eventi, chiama `$ionicAnalytics.track(eventType, eventData)` ogni volta che si verifica un'azione.

```
$ionicAnalytics.track('User Login', {  
  user: $scope.user  
});
```

La direttiva *ion-track-tap* invia un evento quando viene toccato il suo elemento host. La direttiva *ion-track-data* associata allega i dati dell'evento.

```
<button ion-track-tap="eventType" ion-track-data="expression"></button>
```

Nel dashboard *dell'apps.ionic.io* puoi visualizzare i seguenti dati di analisi,

Eventi: visualizza il numero grezzo di eventi nel tempo o il numero di utenti unici che hanno completato un evento. Un evento può essere qualsiasi cosa, da un utente che carica l'app, a confermare un acquisto.

Canalizzazioni: una canalizzazione è una sequenza di azioni che ti aspetti che gli utenti eseguano nella tua app, raggiungendo un obiettivo definito. L'uso attento delle canalizzazioni ti aiuterà a migliorare i tassi di conversione.

Segmenti: visualizza gli eventi nel tempo, raggruppati in base a una proprietà specificata. Oppure, calcola la percentuale di eventi che corrispondono a una determinata proprietà. I segmenti ti aiutano a capire la tua base di utenti e vedere come cambiano le proprietà nel tempo.

Conservazione: tieni traccia di quanto tempo gli utenti sono attivi sulla tua app prima che smettano di usarla. In alternativa, identifica il tempo necessario agli utenti per raggiungere un obiettivo definito, ad esempio una vendita completata.

Pulse: un feed dal vivo degli eventi che arrivano dai tuoi utenti.

Ionic Push



Invia notifiche push mirate e automatizzate ai tuoi utenti.

Ionic Push consente di creare notifiche push mirate tramite un semplice dashboard che verrà inviato automaticamente quando gli utenti soddisfano criteri specifici e offre una semplice API per inviare notifiche push dai propri server.

Profili push Android:

Le notifiche push di Android utilizzano il servizio *Google Cloud Messaging* (GCM). Apri la [Google Developers Console](#) e crea un progetto. Copia il *numero del tuo progetto* . Questo sarà l' *ID mittente GCM* o il **Numero progetto GCM** .

Nella sezione *Gestione API* , attiva l' *API di Google Cloud Messaging* . Quindi accedere alla sezione *Credenziali* e selezionare *Crea credenziali*, quindi selezionare *Chiave API*, quindi *Chiave server*. Assegna un nome alla chiave API e lascia vuoto il campo *Accetta richieste da ...* e fai clic su *Crea* . Salva la tua **chiave API** !

Autenticazione:

Vai alla dashboard della tua app su [Ionic Platform](#) e vai su *Impostazioni -> Certificati* . Se non lo hai già fatto, crea un nuovo profilo di sicurezza, quindi premi *Modifica* . Annotare il **tag** del **profilo** .

Ora, fai clic sulla scheda *Android* e trova la sezione contrassegnata con *Google Cloud Messaging* , inserisci la *chiave API* generata nella Google Developer Console, quindi fai clic su *Salva* . Vai a

Impostazioni -> Chiavi API . Sotto Token API , crea un nuovo token e copialo. Questo sarà il tuo token API .

```
$ ionic plugin add phonegap-plugin-push --variable SENDER_ID="GCM_PROJECT_NUMBER"
$ ionic config set gcm_key <your-gcm-project-number>
$ ionic config set dev_push false
$ ionic io init
```

Nota: phonegap-plugin-push richiede il repository del supporto Android versione 32+

Nel tuo *app.js* aggiungi la dipendenza del modulo ' *ionic.service.push* ' dopo *ionic.service.core*

Codice Ionico Push:

Inizializza il servizio e registra il tuo dispositivo nella funzione di esecuzione del modulo. Avrai bisogno del token del dispositivo che è registrato dall'utente per l'invio di notifiche all'utente.

```
$ionicPush.init({
  debug: true,
  onNotification: function (notification) {
    console.log('token:', notification.payload);
  },
  onRegister: function (token) {
    console.log('Device Token:', token);
    $ionicPush.saveToken(token); // persist the token in the Ionic Platform
  }
});

$ionicPush.register();
```

quindi corri

```
$ grunt build:android --debug
```

Ionic Push ti consente di creare notifiche push mirate attraverso la dashboard. Puoi anche inviare notifiche dal server nel formato seguente.

```
curl -X POST -H "Authorization: Bearer API_TOKEN" -H "Content-Type: application/json" -d '{
  "tokens": ["DEVICE_TOKEN"],
  "profile": "PROFILE_TAG",
  "notification": {
    "message": "Hello World!"
  },
  "android": {
    "title": "Hi User",
    "message": "An update is available for your App",
    "payload": {
      "update": true
    }
  }
}' "https://api.ionic.io/push/notifications"
```

Nota: i passaggi per configurare Ionic Push per iOS sono gli stessi tranne per la creazione dei profili push. Per creare profili push iOS, consulta <http://docs.ionic.io/v2.0.0-beta/docs/ios-push->

App di esempio



Scarica l'app di esempio qui .

Un'app campione è allegata qui per riferimento.

```
IonicApp:
|
|  bower.json
|  Gruntfile.js
|  package.json
|
└─ app
    |  index.html
    |
    └─ js
        |  app.js
        |  controllers.js
        |
    └─ templates
        |  home.html
        |  menu.html
```

Nota: questo non è un progetto autonomo. Il codice fornito è solo per il confronto con un progetto creato e implementato utilizzando le procedure sopra riportate in questo documento, in caso di problemi o errori.

Leggi Ionic Cloud per Yeoman Ionic Projects online: <https://riptutorial.com/it/ionic-framework/topic/6389/ionic-cloud-per-yeoman-ionic-projects>

Capitolo 13: Ionic: analizza la tua app con jshint e gulp-jshint come parte del processo di creazione

Osservazioni

Lasciare la tua app ionica prima di correre ha enormi vantaggi. Analizzerà il codice per potenziali errori e ti farà risparmiare moltissimo tempo.

Che cosa è linting e come installare i pacchetti richiesti?

"Linting è il processo di esecuzione di un programma che analizzerà il codice per potenziali errori."
- vedi cos'è "Linting"?

La tua app ionica viene fornita con un file package.json. Vai alla root della tua app in una riga di comando / terminale e installa i seguenti pacchetti:

```
npm install jshint --save-dev
npm install jshint-stylish --save-dev
npm install gulp-jshint --save-dev
```

Examples

Aggiungi un'attività gulp

Nella radice della tua app ionica, c'è un file gulpfile.js. Aprilo in un editor e incolla il seguente compito gulp:

```
gulp.task('lint', function() {
  return gulp.src(['./www/js/**/*.js'])
    .pipe(jshint('.jshintrc'))
    .pipe(jshint.reporter('jshint-stylish'))
    .pipe(jshint.reporter('fail'))
});
```

Questo cerca una cartella chiamata 'js' nella cartella 'www'. Se hai altre cartelle contenenti file JavaScript, aggiungi anche quelle. Ad esempio, consente anche di aggiungere una cartella chiamata 'viste':

```
gulp.task('lint', function() {
  return gulp.src(['./www/js/**/*.js', './www/views/**/*.js'])
    .pipe(jshint('.jshintrc'))
    .pipe(jshint.reporter('jshint-stylish'))
    .pipe(jshint.reporter('fail'))
});
```

spiegazioni:

- 1) `/**/*.js` - This syntax means to look at all the js files in the subfolders too
- 2) `.jshintrc` - This is a configuration file that we will create in the next example.

Creare il file `.jshintrc` (facoltativo)

Crea un file chiamato `.jshintrc` nella radice della tua app, dove è `package.json`.

* Nota su Windows: crea un file chiamato `"jshintrc.txt"`. Quindi rinominalo in `".jshintrc"`. (notare il punto alla fine).

Questo è un file di configurazione. Ad esempio, può dire a jshint di ignorare determinate variabili e molte altre cose. Ecco il mio:

```
{
  "predef": [
    "window",
    "console",
    "cordova",
    "device",
    "alert",
    "document",
    "debug",
    "setServiceVars",
    "StatusBar",
    "config"
  ],
  "globals": {
    "angular"      : false,
    "myApp"        : false,
    "myControllers": false,
    "myDirectives" : false,
    "localStorage" : false,
    "navigator"    : false,
    "emit"         : false,
    "atob"         : false,
    "moment"       : false,
    "btoa"         : false
  },
  "node"          : true
}
```

Aggiungi Makefile

1. Crea un file chiamato `"Makefile"` (senza estensione) nella radice della tua app
2. Aprilo in un editor di testo e aggiungilo:

```
android:
  gulp lint
  gulp sass
  ionic run android --device

ios:
```

```
gulp lint
gulp sass
ionic build ios
```

Questo lascerà la tua app e se questo passa, si compilerà sass e costruirai app.

Utilizzo: per eseguire la tua app, anziché il normale "ionic run android --device", esegui questi comandi:

```
Android: make android
iOS      : make ios
```

Leggi Ionic: analizza la tua app con jshint e gulp-jshint come parte del processo di creazione online: <https://riptutorial.com/it/ionic-framework/topic/4889/ionic--analizza-la-tua-app-con-jshint-e-gulp-jshint-come-parte-del-processo-di-creazione>

Capitolo 14: Ionicons

Osservazioni

Nel moderno sviluppo del web è pratica comune usare i caratteri per visualizzare le icone. Poiché i font sono vettori, sono indipendenti dalla risoluzione e possono essere facilmente colorati tramite CSS, solo per citare alcuni vantaggi rispetto alle immagini bitmap. Ionicons è stato creato dallo stesso team creato da Ionic Framework e può essere utilizzato in qualsiasi progetto dal sono gratuiti al 100% e open source. MIT con licenza.

Ionicons possono essere utilizzati da soli o con componenti Ionics CSS quando hanno determinati stili in base agli elementi padre.

La homepage e l'elenco delle icone sono disponibili qui: <http://ionicons.com/>

Examples

Utilizzo di base

Le icone dei caratteri sono generalmente collocate all'interno di un tag `<i>`. Ionic ha stili CSS predefiniti per le icone per un facile utilizzo. L'esempio più semplice di utilizzo:

```
<i class="icon ion-home"></i>
```

Utilizzo prolungato

Ionic ha alcune componenti CSS in cui è possibile utilizzare Ionicons come impostazione predefinita con uno stile predefinito. La classe di `range` nell'elemento `<div>` applicherà lo stile corretto sia per l'input che per le icone al suo interno. Ecco un esempio di un cursore di intervallo.

```
<div class="item range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>
```

Un altro esempio di utilizzo di Ionicon nelle schede ioniche che creerà un menu simile a una scheda. Le `tabs-striped tabs-color-assertive` definiscono lo stile delle schede stesse. Le icone vengono utilizzate con semplici tag `<i>` e ottengono il loro stile posizionale dalle classi applicate alle div parent.

```
<div class="tabs-striped tabs-color-assertive">
  <div class="tabs">
    <a class="tab-item" href="#">
      <i class="icon ion-home"></i>
      Home
    </a>
```

```
<a class="tab-item" href="#">
  <i class="icon ion-gear-b"></i>
  Settings
</a>
</div>
```

Leggi Ionicons online: <https://riptutorial.com/it/ionic-framework/topic/5886/ionicons>

Capitolo 15: Pubblicare la tua app Ionic

Examples

Costruire la versione di rilascio da macOS

Costruisci l'APK

Prima di tutto abbiamo bisogno di costruire l'APK.

```
ionic build --release android
```

Genera chiave privata

Quindi creeremo un keystore per firmare l'APK. `keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000`

- Cambia **my-release-key** con il tuo nome chiave.
- Cambia **alias_name** con il tuo alias chiave.

Firma l'APK

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore HelloWorld-release-unsigned.apk alias_name
```

- Cambia **my-release-key** con il tuo nome chiave.
- Cambia **HelloWorld-release-unsigned** con il tuo apk non firmato. `ionic-project/platforms/android/build/outputs/apk .`
- Cambia **alias_name** con il tuo alias chiave.

Chiudi l'APK

```
zipalign -v 4 HelloWorld-release-unsigned.apk HelloWorld.apk
```

- Puoi trovare zip in `/Users/username/Library/Android/sdk/build-tools/XXX/`
- Cambia **HelloWorld-release-unsigned** con il tuo apk non firmato. `ionic-project/platforms/android/build/outputs/apk .`
- Cambia **HelloWorld.apk** con il tuo nome di file apk preferito. Questo sarà caricato su Google Play.

Leggi [Pubblicare la tua app Ionic online](https://riptutorial.com/it/ionic-framework/topic/7755/pubblicare-la-tua-app-ionic): <https://riptutorial.com/it/ionic-framework/topic/7755/pubblicare-la-tua-app-ionic>

Capitolo 16: Qual è la differenza tra "costruzione ionica" e "preparazione ionica"?

Examples

build ionico vs preparazione ionica

Dalla documentazione ufficiale:

Se vuoi essere avanzato, puoi anche aprire il file di progetto per una piattaforma specifica aprendo il progetto XCode o Android Eclipse richiesto nelle piattaforme / PLATFORM all'interno della radice del tuo progetto. Quindi, è possibile creare e testare dall'interno dell'IDE specifico della piattaforma. Nota: se segui questo percorso, ti consiglio di continuare a lavorare all'interno della cartella radice di `www` e, quando hai apportato modifiche a questa cartella, esegui il comando: `$ cordova prepare ios` che aggiornerà il progetto specifico iOS con il codice dal cartella `www`. Nota: questo sovrascriverà qualsiasi modifica apportata alle `platforms/ios/www` e altre cartelle specifiche della piattaforma.

Quindi, per riassumere questa parte - se usi XCode per testare ed eseguire il tuo codice, dopo aver modificato parte del codice devi solo eseguire `ionic prepare` l'aggiornamento del progetto iOS che poi continuerai a utilizzare in XCode.

`ionic build` comando `ionic build` realtà prepara il file finale (ad esempio in Android è il file `.apk`) che poi può essere copiato sul tuo dispositivo e test eseguendolo manualmente sul dispositivo (o usando il comando `ionic emulate` per testarlo sull'emulatore).

Leggi Qual è la differenza tra "costruzione ionica" e "preparazione ionica"? online:
<https://riptutorial.com/it/ionic-framework/topic/2911/qual-e-la-differenza-tra--costruzione-ionica--e--preparazione-ionica-->

Capitolo 17: Scorrimento infinito ionico per mostrare gli elementi di carico su richiesta (dati già disponibili non per richiesta Http)

Examples

Carica n numero di dati disponibili su richiesta

HTML:

```
<li class="item" ng-repeat="schedule in Schedules | filter:scheduleSearch |
limitTo:numberOfItemsToDisplay">
  Display some data
</li>
<ion-infinite-scroll on-infinite="addMoreItem()" ng-if="Schedules.length >
numberOfItemsToDisplay"></ion-infinite-scroll>
```

Controller:

```
$scope.numberOfItemsToDisplay = 10; // Use it with limit to in ng-repeat
$scope.addMoreItem = function(done) {
  if ($scope.Schedules.length > $scope.numberOfItemsToDisplay)
    $scope.numberOfItemsToDisplay += 10; // load number of more items
  $scope.$broadcast('scroll.infiniteScrollComplete')
}
```

Carica 10 voci ogni volta che si chiama addMoreItem ().

Leggi Scorrimento infinito ionico per mostrare gli elementi di carico su richiesta (dati già disponibili non per richiesta Http) online: <https://riptutorial.com/it/ionic-framework/topic/9490/scorrimento-infinito-ionico-per-mostrare-gli-elementi-di-carico-su-richiasta--dati-gia-disponibili-non-per-richiasta-http->

Capitolo 18: Servizi backend ionic (ionic.io)

Examples

Introduzione e configurazione

La [piattaforma Ionic](#) offre una gamma di potenti servizi e strumenti di backend mobile basati sull'ibrido per rendere più semplice scalare applicazioni ibride performanti e belle ad un ritmo rapido.

Per poter utilizzare Ionic Platform è necessario avere [installato Ionic Framework](#) .

1.) Registrazione ([iscrizione](#))

È necessario inserire: nome, nome utente, azienda, telefono (opzionale), e-mail e password. Dopo aver ricevuto l'e-mail di conferma, puoi effettuare il [log-in](#) .

Benvenuto in Ionic. Creiamo la tua prima app! Crea una nuova app in alto, quindi segui la guida di [avvio rapido](#) per caricare la tua prima app.

2.) Installazione

Per prima cosa [crea la tua app Ionic](#) . Ora puoi aggiungere il client web della piattaforma:

Il client web ti offre un modo per interagire con i servizi della piattaforma Ionic dal tuo codice app.

```
$ ionic add ionic-platform-web-client
```

Ora, è necessario che la piattaforma assegni all'app un ID app univoco e una chiave API. Per farlo usa il comando `ionic io init` .

```
$ ionic io init
```

Questo ti chiederà automaticamente di accedere al tuo account piattaforma. L'ID app e la chiave API verranno quindi archiviati nella [configurazione della](#) piattaforma Ionic del progetto.

La tua app è ora collegata alla piattaforma Ionic e sarà elencata nella tua [Dashboard](#) .

3.) Ora puoi passare all'installazione di uno dei servizi di Ionic Platform :

- [Utenti / Autori](#)
- [schierare](#)
- [Spingere](#)
- [Pacchetto](#)
- [analitica](#)

Leggi Servizi backend ionic (ionic.io) online: <https://riptutorial.com/it/ionic-framework/topic/3622/servizi-backend-ionic--ionic-io->

Capitolo 19: Testare l'app Ionic in un browser

Osservazioni

Il test delle funzionalità del dispositivo nativo come fotocamera, vibrazione e altri, molti dei quali si trovano nella documentazione di [Ionic Native](#), non può essere fatto nel browser. Questo è un limite intrinseco del fatto che Cordova, la piattaforma su cui Ionic dipende da essere in grado di accedere alle API native Android, iOS e Windows Mobile di un dispositivo, non può essere eseguita sul browser.

Uno può aggirare questo problema prendendo in giro la funzionalità del plugin nativo.

Esempio

Ecco un esempio su come prendere in giro il plugin [Camera](#) :

Vai avanti e crea una cartella opzionale nella cartella principale del progetto.

```
cd src
mkdir mocks
cd mocks
touch camera-mock.ts
```

Apri `camera-mock.ts` e copia incolla il seguente codice:

```
export class CameraMock {
  getPicture(params) {
    return new Promise((resolve, reject) => {
      resolve("BASE_64_IMAGE_DATA");
    });
  }
}
```

Quindi apri `src/app.module.ts` e importa la classe di simulazione "

```
import { CameraMock } from "../mocks/camera-mock";
```

Quindi aggiungilo all'array dei provider di moduli:

```
@NgModule({
  declarations: [
    MyApp,
    HomePage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp)
  ],
  bootstrap: [IonicApp],
```

```

entryComponents: [
  MyApp,
  HomePage
],
providers: [
  StatusBar,
  SplashScreen,
  CameraMock,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
})
export class AppModule {}

```

Ora puoi usarlo in qualsiasi componente dopo averlo importato.

Examples

Test in un browser

Utilizzare `ionic serve` per avviare un server di sviluppo locale per lo sviluppo e il test delle app. Questo è utile sia per i test del browser desktop, sia per testare all'interno di un browser del dispositivo che è connesso alla stessa rete. Inoltre, questo comando avvia LiveReload che viene utilizzato per monitorare le modifiche nel file system. Non appena si salva un file, il browser viene aggiornato automaticamente. Dai un'occhiata ai [documenti di Sass](#) se vuoi anche avere un `ionic serve` guardare i file Sass del progetto.

```
$ ionic serve [options]
```

Per browser Chrome puoi controllare i dispositivi (AVD o cellulari), digitare il comando seguente nella barra degli indirizzi del browser Chrome.

```
chrome://inspect/#devices
```

LiveReload

Per impostazione predefinita, LiveReload controllerà le modifiche nella tua `www/` directory, escluso `www/lib/`. Per cambiare questo, puoi specificare una proprietà `watchPatterns` nel file `ionic.project` che si trova nella root del progetto per guardare (o non guardare) le modifiche specifiche.

```

{
  "name": "myApp",
  "app_id": "",
  "watchPatterns": [
    "www/js/*",
    "!www/css/**/*"
  ]
}

```

Per un riferimento sulla sintassi del pattern glob, controlla i [pattern globbing](#) sul sito web di Grunt.

Nota:

```
$ ionic setup sass
```

aggiungerà `watchPatterns` correttamente con i valori predefiniti al file `ionic.project` che è possibile modificare, oltre alla proprietà `gulpStartupTasks` come descritto nella [documentazione di Sass](#) .

Ionic Lab

Ionic Lab <http://ionicframework.com/img/blog/lab.png>

Ionic Lab è una funzione al servizio di `ionic serve` che semplifica il test della tua app in una cornice del telefono e con piattaforme iOS e Android affiancate. Per usarlo, basta correre

```
$ ionic serve --lab
```

Leggi l' [annuncio di rilascio completo](#) per tutti i dettagli!

Specifica di un indirizzo IP da utilizzare

Supponi di voler specificare a quale indirizzo si conetterà il tuo browser, ad esempio per testare o utenti esterni. Specificare l'indirizzo con l'argomento `--address` .

```
$ ionic serve --address 68.54.96.105
```

Proxy di servizio

Il comando `serve` può aggiungere alcuni proxy al server http. Questi proxy sono utili se stai sviluppando nel browser e devi effettuare chiamate a un'API esterna. Con questa funzione è possibile inoltrare la richiesta al API esterno tramite il server http ionico, impedendo che l' No `'Access-Control-Allow-Origin' header is present on the requested resource` sull'errore di No `'Access-Control-Allow-Origin' header is present on the requested resource` .

Nel file `ionic.project` è possibile aggiungere una proprietà con una serie di proxy che si desidera aggiungere. I proxy sono oggetti con due proprietà:

- `path` : stringa che verrà confrontata con l'inizio dell'URL della richiesta in entrata.
- `proxyUrl` : una stringa con l'url di dove deve andare la richiesta proxy.

```
{
  "name": "appname",
  "email": "",
  "app_id": "",
  "proxies": [
    {
      "path": "/v1",
      "proxyUrl": "https://api.instagram.com/v1"
    }
  ]
}
```

```
}
```

Utilizzando la configurazione di cui sopra, ora puoi effettuare richieste al tuo server locale su `http://localhost:8100/v1` per fare in modo che le richieste proxy vengano inviate a `https://api.instagram.com/v1`

Per esempio:

```
angular.module('starter.controllers', [])
.constant('InstagramApiUrl', '')
// .constant('InstagramApiUrl', 'https://api.instagram.com')
// In production, make this the real URL

.controller('FeedCtrl', function($scope, $http, InstagramApiUrl) {

  $scope.feed = null;

  $http.get(InstagramApiUrl +
'/v1/media/search?client_id=1&lat=48&lng=2.294351').then(function(data) {
  console.log('data ', data)
  $scope.feed = data;
})

})
```

Vedi anche [questo succo](#) per ulteriore aiuto.

Flag / opzioni della riga di comando

```
[--consolelogs|-c] ..... Print app console logs to Ionic CLI
[--serverlogs|-s] ..... Print dev server logs to Ionic CLI
[--port|-p] ..... Dev server HTTP port (8100 default)
[--livereload-port|-i] .. Live Reload port (35729 default)
[--nobrowser|-b] ..... Disable launching a browser
[--nolivereload|-r] ..... Do not start live reload
[--noproxy|-x] ..... Do not add proxies
```

Leggi Testare l'app Ionic in un browser online: <https://riptutorial.com/it/ionic-framework/topic/3256/testare-l-app-ionic-in-un-browser>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con il framework ionic	Akshay Khale , Andrea Macchieraldo , Community , Devid Farinelli , Ian Pinto , leetheguy , Lightbeard , Mazz , Newton Joshua , the_mahasagar
2	Accesso alla fotocamera e alla libreria fotografica del dispositivo dall'applicazione ionic	Pritish , sonu
3	Collegamento di Ionic con qualsiasi database	Nikola
4	Come utilizzare le funzioni di EcmaScript 6 in Ionic?	Nikola
5	Componenti CSS ionic	thepio
6	Crea finestra di dialogo in Ionic	A-Droid Tech
7	Distribuisci Ionic come sito web	Nikola
8	Esegui l'app Ionic on Emulator o sul tuo telefono	A-Droid Tech , Gerard Cuadras , Ian Pinto , Ketan Akbari , olivier , Raymond Ativie
9	Estensioni Ionic AngularJS	thepio
10	Ganci Ionico CLI	thepio
11	Inizia a creare app su Ionic	A-Droid Tech , Ian Pinto
12	Ionic Cloud per	Newton Joshua

	Yeoman Ionic Projects	
13	Ionic: analizza la tua app con jshint e gulp-jshint come parte del processo di creazione	Akilan Arasu , Ian Pinto , Sumama Waheed , thepio
14	Ionicons	thepio
15	Publicare la tua app Ionic	Gerard Cuadras
16	Qual è la differenza tra "costruzione ionica" e "preparazione ionica"?	Nikola
17	Scorrimento infinito ionico per mostrare gli elementi di carico su richiesta (dati già disponibili non per richiesta Http)	SANAT
18	Servizi backend ionici (ionic.io)	Tomislav Stankovic
19	Testare l'app Ionic in un browser	A-Droid Tech , Ian Pinto , Ketan Akbari , maninak