



eBook Gratuit

APPRENEZ

ionic-framework

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#ionic-

framework

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec ionic-framework.....	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Installation ou configuration.....	2
1. Installez Ionic Framework et Cordova (car les applications Ionic sont basées sur Cordov.....	3
2. Lancer un nouveau projet ionique:.....	3
3. Testez l'application Ionic:.....	4
Ionic Framework Introduction et installation et configuration.....	4
Ionic Framework Hello World App.....	6
Plate-forme ionique (nuage ionique) pour projets ioniques Yo (Yeoman).....	7
Plate - forme ionique :.....	7
Construisez, poussez, déployez et redimensionnez vos applications ioniques de manière simp.....	7
Générateur ionique.....	8
Un générateur pour la structure ionique de Yeoman, l'outil d'échafaudage Web pour les appl.....	8
client-plateforme ionique.....	9
Un client Web qui fournit des interactions avec la plate-forme ionique.....	9
Déploiement ionique.....	10
Poussez les mises à jour en temps réel sur vos applications de production et gérez l'histo.....	10
Analyse ionique.....	11
Afficher le flux en direct des événements ou le nombre brut / unique d'événements / utilis.....	11
Push ionique.....	13
Envoyez des notifications push ciblées et automatisées à vos utilisateurs.....	13
Exemple d'application.....	14
Chapitre 2: Appareil photo et accès à une bibliothèque de photos à partir d'une applicatio.....	16
Remarques.....	16
Exemples.....	16
Camara ouverte et galerie de photos.....	16
Pour Ionic 3 Exemple.....	19

Chapitre 3: Commencer à créer des applications dans Ionic	22
Exemples.....	22
Démarrer une application ionique.....	22
Démarrer une application ionique	22
Débutants de modèles nommés.....	22
Github Repo démarreurs.....	22
Codepen URL starters.....	22
Répertoire local:.....	22
Indicateurs / options de ligne de commande:.....	22
Chapitre 4: Comment utiliser les fonctionnalités d'EcmaScript 6 dans Ionic?	24
Exemples.....	24
En utilisant gulp-babel et gulp-plumber.....	24
Chapitre 5: Composants CSS ioniques	26
Remarques.....	26
Exemples.....	26
Syntaxe de base du système de grille.....	26
Grille de base	26
Grilles Offset	27
Aligner les colonnes	27
Grille réactive	28
Syntaxe de base des éléments de liste.....	28
Utilisation basique des couleurs utilitaires.....	30
Chapitre 6: Connexion ionique avec n'importe quelle base de données	32
Exemples.....	32
Vous ne pouvez pas le faire directement à partir du cadre ionique.....	32
Chapitre 7: Créer un dialogue dans Ionic	33
Paramètres.....	33
Remarques.....	33
Exemples.....	33
Créer un dialogue dans Ionic.....	33
Chapitre 8: Crochets Ionic CLI	34

Remarques.....	34
introduction.....	34
Types de crochet.....	34
Façons de définir des crochets:.....	34
Exemples.....	35
Vérification des erreurs dans vos fichiers Javascript dans before_prepare en utilisant jsh.....	35
Chapitre 9: Défilement infini ionique pour afficher les éléments de charge à la demande (d.....	37
Exemples.....	37
Charger n nombre de données disponibles à la demande.....	37
Chapitre 10: Déployer Ionic en tant que site Web.....	38
Exemples.....	38
Copiez simplement le dossier www sur votre serveur Web.....	38
Chapitre 11: Exécuter une application ionique sur un émulateur ou sur votre téléphone.....	39
Exemples.....	39
Exécuter une application ionique sur un émulateur ou sur votre téléphone.....	39
1. Ajouter une cible de plate-forme.....	39
2. Construisez votre application.....	39
Application de rechargement en direct pendant le développement (version bêta).....	39
Indicateurs de ligne de commande / options pour run et emulate.....	40
3. Émuler votre application.....	40
4. Exécuter votre application.....	41
4.1. Spécifier votre cible.....	41
Chapitre 12: Extensions ioniques AngularJS.....	42
Remarques.....	42
Exemples.....	42
Formulaire de saisie.....	42
Fenêtres modales.....	42
Création de l'objet modal dans la portée.....	42
Contrôler le modal.....	43
Événements modaux.....	43
Chapitre 13: Ionic - Analysez votre application avec jshint et gulp-jshint dans le cadre d.....	45

Remarques.....	45
Qu'est-ce que le linting et comment installer les paquets requis?.....	45
Exemples.....	45
Ajouter une tâche Gulp.....	45
Créer un fichier .jshintrc (facultatif).....	46
Ajouter Makefile.....	46
Chapitre 14: Ionicons.....	48
Remarques.....	48
Exemples.....	48
Utilisation de base.....	48
Utilisation étendue.....	48
Chapitre 15: Nuage ionique pour projets ioniques Yeoman.....	50
Exemples.....	50
Plate-forme ionique (nuage ionique) pour projets ioniques Yo (Yeoman).....	50
Plate - forme ionique :.....	50
Construisez, poussez, déployez et redimensionnez vos applications ioniques de manière simp.....	50
Générateur ionique.....	51
Un générateur pour la structure ionique de Yeoman, l'outil d'échafaudage Web pour les appl.....	51
client-plateforme ionique.....	52
Un client Web qui fournit des interactions avec la plate-forme ionique.....	52
Déploiement ionique.....	53
Poussez les mises à jour en temps réel sur vos applications de production et gérez l'histo.....	53
Analyse ionique.....	54
Afficher le flux en direct des événements ou le nombre brut / unique d'événements / utilis.....	54
Push ionique.....	55
Envoyez des notifications push ciblées et automatisées à vos utilisateurs.....	55
Exemple d'application.....	57
Chapitre 16: Publier votre application ionique.....	58
Exemples.....	58
Version de construction à partir de macOS.....	58
Chapitre 17: Quelle est la différence entre «construction ionique» et «préparation ionique.....	59

Exemples.....	59
construction ionique vs préparation ionique.....	59
Chapitre 18: Services Ionic Backend (ionic.io).....	60
Exemples.....	60
Introduction et configuration.....	60
Chapitre 19: Test de l'application ionique dans un navigateur.....	62
Remarques.....	62
Exemple.....	62
Exemples.....	63
Test dans un navigateur.....	63
LiveReload.....	63
Laboratoire ionique.....	64
Spécifier une adresse IP à utiliser.....	64
Proxies de service.....	64
Indicateurs / options de ligne de commande.....	65
Crédits.....	66

À propos

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

Chapitre 1: Démarrer avec ionic-framework

Remarques

Ionic est un framework pour développer des applications mobiles avec HTML, CSS et JavaScript. Les applications ioniques s'exécutent en tant qu'applications natives et ont une apparence naturelle.

Ionic est basé sur le framework [AngularJS](#) et fournit une solution complète pour concevoir, construire et conditionner des applications mobiles. La conception est réalisée à l'aide d'une collection d'outils de modèles et d'une [bibliothèque d'icônes personnalisée](#). Ionic fournit des [composants CSS / SASS](#) personnalisés ainsi que des [extensions d'interface utilisateur Javascript](#). Les applications ioniques peuvent être créées, émulées et intégrées à leur [interface de ligne de commande \(CLI\)](#).

Les modèles ioniques sont dynamiques et réactifs et s'adaptent à leur environnement pour fournir une apparence naturelle. Cette adaptation comprend la disposition, le style et les icônes. Ionic rend également possible la [personnalisation de plate-forme](#) indépendante. Les applications ioniques utilisant la technologie Web frontale, elles peuvent également être affichées dans un navigateur pour un développement plus rapide.

Les applications ioniques sont construites sur [Apache Cordova](#) par défaut. Ils ont accès à tous les [plug-ins Cordova](#) qui vous permettent d'utiliser des fonctionnalités natives, telles que les notifications push, les caméras, les accéléromètres, etc. Les applications Cordova fonctionnent sur [plusieurs plates - formes](#) et périphériques Cordova peut être remplacé par d' [autres technologies multiplates-formes](#) telles que [trigger.io](#).

Versions

Version	Date de sortie
1.3.1 "le salvador"	2016-05-12
1.3.0 "delhi"	2016-04-21
1.2.0 "zirconium-zeren"	2015-12-09
1.1.0 "xenon-xerus"	2015-08-13
1.0.0 "Licorne d'uranium"	2015-05-12

Exemples

Installation ou configuration

1. Installez Ionic Framework et Cordova (car les applications Ionic sont basées sur Cordova) en utilisant npm (le gestionnaire de paquets de nœuds):

Assurez-vous d'avoir une version à jour de Node.js installée sur votre système. Si vous n'avez pas installé Node.js, vous pouvez l'installer [ici](#) .

De plus, pour les utilisateurs de Mac, la dernière version de Xcode installée sur votre système vous permet de [télécharger des](#) outils de ligne de commande et iOS Simulator.

Ouvrez une fenêtre de terminal (Mac) ou une fenêtre de commande (Windows) et installez Cordova et Ionic:

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

Sur un Mac, vous devrez peut-être utiliser sudo selon la configuration de votre système:

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

Si vous avez déjà installé Cordova et Ionic sur votre ordinateur, assurez-vous de mettre à jour vers la dernière version:

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

ou

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

Suivez les guides de plate-forme [Android](#) et [iOS](#) pour installer les dépendances de plate-forme requises.

Remarque: le développement iOS nécessite Mac OS X. Le simulateur iOS via l'interface Ionic CLI nécessite le package ios-sim npm, qui peut être installé avec la commande:

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

2. Lancer un nouveau projet ionique:

Créez un projet ionique à l'aide d'un des modèles d'applications prédéfinis, ou d'un modèle vierge pour commencer.

```
$ ionic start myApp blank
```

ou

```
$ ionic start myApp tabs
```

ou

```
$ ionic start myApp sidemenu
```

3. Testez l'application Ionic:

Pour tester votre application ionique dans un navigateur de bureau sur les plates-formes iOS et Android:

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

Alors `ionic serve --lab` est idéal pour tester l'interface utilisateur de l'application sur plusieurs plates-formes, cela peut entraîner des problèmes pour la console Javascript ou l'élément Inspection, dans ce cas vous préférez:

```
$ ionic serve
```

Pour tester votre application ionique dans un émulateur:

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

Remplacez `ios` par `Android` pour le test de l'émulateur Android:

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

Pour tester votre application Ionic sur un appareil Android connecté via USB:

```
$ ionic run android
```

Pour tester votre application ionique sur un appareil iOS connecté via USB:

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

Ionic Framework Introduction et installation et configuration

Cadre ionique

Un cadre de développement d'applications mobiles multiplates-formes utilisant les technologies Web Angular JS et Front End.

Site officiel : <http://ionicframework.com/>

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

Installation et configuration

Installation Ionic requise NPM (Node Package Manager) et Cordova.

Vous pouvez télécharger et installer Node JS à partir d' [ici](#), qui est fourni avec NPM.

Pour télécharger Apache Cordova, vous pouvez utiliser NPM en ligne de commande

```
npm install -g cordova
```

Si vous avez déjà NPM et Cordova, vous pouvez installer le framework ionique à partir de la ligne de commande en utilisant la commande suivante.

```
npm install -g ionic
```

Cela va installer et configurer le framework ionique pour l'utiliser depuis la ligne de commande.

Remarque * En fonction de votre environnement système, vous devrez peut-être exécuter avec des privilèges d'administrateur.

Commencer un nouveau projet

Pour démarrer un nouveau projet de structure ionique, vous pouvez utiliser la commande suivante

```
ionic start myproject
```

ou

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

Modèles:

Ionic vous permet de créer un projet en utilisant des modèles intégrés

`tabs` (par défaut): qui créera une application simple avec un onglet.

`sidemenu` : qui créera une application ionique avec menu latéral.

`blank` : qui créera une application ionique vierge.

qui créera un nouveau dossier nommé `myproject` avec tous les fichiers de projet ioniques.

pour tester le projet dans votre navigateur, vous pouvez utiliser la commande suivante

```
ionic serve --lab
```

ou

```
ionic serve
```

Exécuter un émulation Pour exécuter ou tester une application sur un émulateur ou un téléphone, vous devez d'abord ajouter une plate-forme pour laquelle vous pouvez utiliser la commande suivante

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

Noms de plate-forme, vous pouvez directement mentionner Android et iOS pour les plates-formes respectives, vous pouvez mentionner plusieurs noms de plate-forme également séparés par un espace.

Pour exécuter votre application, vous pouvez utiliser

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

Pour de l'aide, vous pouvez utiliser

```
ionic --help
```

ou

```
ionic help
```

Reportez - vous à [ce lien](#) pour une explication détaillée des cli ioniques.

Reportez - vous à [ce lien](#) pour les composants CSS disponibles en version ionique.

Reportez - vous à [ce lien](#) pour une référence API Javascript pour ionique.

Pour un développement plus rapide avec ionique, vous pouvez également essayer [ionic Playground](#) .

Bonne chance avec cadre ionique ...

Ionic Framework Hello World App

Après tout l'installation, pour faire Hello World App

- Pour créer Simple Blank App, exécutez la commande sur le terminal:

```
ionic start HelloWorld blank // create new project
cd HelloWorld // get into HelloWorld directory
```

- ouvrez le projet HelloWorld dans IDE sous-ligne / webstrome:
 - Modifier index.html, dans www / ditectory

```
<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>
```

- Pour exécuter dans le navigateur à partir du terminal:

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

- Pour ajouter une plateforme:

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

- Pour exécuter sur le périphérique:

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

- Pour courir dans livereload:

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

Plate-forme ionique (nuage ionique) pour projets ioniques Yo (Yeoman)



Plate - forme ionique :

Construisez, poussez, déployez et redimensionnez vos applications ioniques de manière simple.

Description du titre:

Ionic Platform est une plate-forme cloud pour la gestion et la mise à l'échelle des applications mobiles multiplates-formes. Les services intégrés vous permettent, à vous et à votre équipe, de

créer, déployer et développer efficacement vos applications.

Objectif du document:

Ionic Platform fonctionne bien avec les projets ioniques standard. Mais les projets qui suivent une structure de répertoire non standard peuvent être confrontés à quelques obstacles. Ce document fournit les étapes pour utiliser Ionic Platform dans les projets ioniques créés à l'aide de Yeoman.

Portée du document:

Ce document décrit les étapes de base pour créer un projet ionique à l'aide de Yeoman et l'intégrer à Ionic Platform à l'aide du client Web Ionic Platform. Ce document couvre les étapes de base pour utiliser Ionic Deploy, Ionic Analytics et Ionic Push.

Public visé:

Le public visé par ce document est constitué par les développeurs d'applications Web / mobile, avec une expertise de niveau débutant et expert, qui connaissent bien les prérequis ci-dessous.

Conditions préalables:

Vous devriez vous familiariser avec les frameworks / outils suivants avant d'essayer ce document.

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

Générateur ionique



Un générateur pour la structure ionique de Yeoman, l'outil d'échafaudage Web pour les applications Web modernes

Node.js est un moteur d'exécution JavaScript basé sur le moteur JavaScript V8 de Chrome. npm est le gestionnaire de paquets pour JavaScript. Téléchargez et installez Node (et npm) à partir de <http://nodejs.org>

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

Yeoman vous aide à lancer de nouveaux projets, en prescrivant les meilleures pratiques et les outils pour vous aider à rester productif.

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

Dans *package.json*, incluez les éléments suivants dans *devDependencies*

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

Dans *bower.json*, incluez les éléments suivants dans les dépendances

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

Dans *Gruntfile.js*, changez le dossier de *scripts* en *"js"* . Changez dans *index.html* si nécessaire.

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

Puis courir

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

client-plateforme ionique



Un client Web qui fournit des interactions avec la plate-forme ionique.

Nous avons besoin de code pour permettre à votre application de communiquer avec la plate-forme ionique. Nous devons ajouter le client Web de la plate-forme ionique pour que l'application ionique puisse s'interfacer avec les plug-ins et la plate-forme Ionic.io.

```
$ ionic io init
```

Dans votre *app.js*, ajoutez la dépendance de module *'ionic.service.core'* . Dans *Gruntfile.js*, ajoutez la tâche *"ionicSettings"* ci-dessous.

```
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']);

```

Ajoutez les *'ionicSettings'* dans *init* et *compressez les tâches* après la *copie* . Dans *index.html*, déplacez la balise ci-dessous après toutes les déclarations de balises.

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

Puis courir

```
$ Grunt serve
```

Déploiement ionique



Poussez les mises à jour en temps réel sur vos applications de production et gérez l'historique des versions.

Ionic Deploy vous permet de mettre à jour votre application à la demande, pour toutes les modifications ne nécessitant pas de modifications binaires, vous permettant de gagner des jours, voire des semaines, de temps d'attente. Suivez la procédure ci-dessous pour configurer Ionic Deploy pour votre application.

Dans *Gruntfile.js*, ajoutez la tâche grunt "*deploy*" comme indiqué ci-dessous.

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

puis courir

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

Code de déploiement ionique:

```
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) {  
  });
```

Déploiement des mises à jour:

Envoyez un nouveau code pour votre application.

Créez un fichier apk et installez votre application. Apportez quelques modifications à votre code et déployez les modifications à l'aide de « *grunt deploy* ». Ensuite, mettez-le à jour depuis votre application.

Vous pouvez également le déployer à partir du tableau de bord *apps.ionic.io*. Vous pouvez déployer l'application sans le paramètre *deploy*. Ensuite, dans le tableau de bord, vous pouvez ajouter les détails des métadonnées et des versions et déployer l'application à partir de là.

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

Analyse ionique



Afficher le flux en direct des événements ou le nombre brut /

unique d'événements / utilisateurs au fil du temps.

Combien d'utilisateurs y a-t-il actuellement sur votre application? Combien d'entre eux utiliseront votre application demain ou la semaine prochaine? Sans information, vous n'avez aucun moyen de savoir si votre application est utilisée de la manière que vous attendez. Suivez la procédure ci-dessous pour configurer Ionic Analytics pour votre application.

Dans votre *app.js*, ajoutez la dépendance de module ' *ionic.service.analytics* ' après le *ionic.service.core* Exécutez la méthode du registre d'analyse dans la fonction d'exécution de notre module.

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

Dans Ionic Analytics, chaque action suivie effectuée par un utilisateur dans votre application est représentée par un objet événement. Un événement est une action unique effectuée à un moment précis. Pour suivre vos propres événements, appelez `$ionicAnalytics.track(eventType, eventData)` chaque fois qu'une action se produit.

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

La directive *ion-track-tap* envoie un événement lorsque son élément hôte est tapé. La directive associée *ion-track-data* joint les données d'événement.

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

Dans le *tableau de bord apps.ionic.io*, vous pouvez afficher les données d'analyse suivantes:

Événements: affichez le nombre brut d'événements au fil du temps ou le nombre d'utilisateurs uniques ayant terminé un événement. Un événement peut être n'importe quoi, depuis un utilisateur chargeant l'application, jusqu'à la confirmation d'un achat.

Entonnoirs: un entonnoir est une séquence d'actions que vous attendez des utilisateurs dans votre application, menant à un objectif défini. L'utilisation judicieuse des entonnoirs vous aidera à améliorer les taux de conversion.

Segments: affiche les événements au fil du temps, regroupés par une propriété spécifiée. Ou bien, calculez le pourcentage d'événements correspondant à une propriété donnée. Les segments vous aident à comprendre votre base d'utilisateurs et à voir comment les propriétés changent avec le temps.

Rétention: suivez la durée pendant laquelle les utilisateurs sont actifs sur votre application avant de cesser de l'utiliser. Ou, indiquez le temps nécessaire aux utilisateurs pour atteindre un objectif défini, comme une vente terminée.

Pulse: Un flux en direct des événements venant de vos utilisateurs.

Push ionique



Envoyez des notifications push ciblées et automatisées à vos utilisateurs.

Ionic Push vous permet de créer des notifications push ciblées via un tableau de bord simple qui sera envoyé automatiquement lorsque les utilisateurs correspondent à des critères spécifiques, et offre une API simple pour envoyer des notifications push depuis vos propres serveurs.

Profils Push Android:

Les notifications push Android utilisent le service *Google Cloud Messaging* (GCM). Ouvrez la [console Google Developers](#) et créez un projet. Copiez le *numéro de votre projet*. Ce sera le **numéro d'expéditeur GCM** ou le **numéro de projet GCM**.

Dans la section *API Manager*, activez l'*API Google Cloud Messaging*. Ensuite, accédez à la section *Informations d'identification* et sélectionnez *Créer des informations d'identification*, puis choisissez *Clé API*, puis *Clé du serveur*. Nommez votre clé API et laissez le champ *Accepter les demandes de ...* vide et cliquez sur *Créer*. Enregistrez votre **clé API**!

Authentification:

Accédez au tableau de bord de votre application sur la [plate - forme ionique](#) et accédez à *Paramètres -> Certificats*. Si vous ne l'avez pas encore fait, créez un nouveau profil de sécurité, puis appuyez sur *edit*. Notez l'**étiquette de profil**.

Maintenant, cliquez sur l'onglet *Android* et recherchez la section marquée *Google Cloud Messaging*, saisissez la *clé API* que vous avez générée sur la console développeur Google, puis cliquez sur *Enregistrer*. Allez dans *Paramètres -> Clés API*. Sous *API Tokens*, créez un nouveau jeton et copiez-le. Ce sera votre **jeton 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
```

Note: phonegap-plugin-push nécessite Android Support Repository version 32+

Dans votre *app.js*, ajoutez la dépendance de module '*ionic.service.push*' après le fichier *ionic.service.core*

Code Push ionique:

Initialisez le service et enregistrez votre appareil dans la fonction d'exécution de votre module.

Vous aurez besoin du jeton de périphérique enregistré par l'utilisateur pour envoyer une notification à l'utilisateur.

```
$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();
```

puis courir

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

Ionic Push vous permet de créer des notifications push ciblées via le tableau de bord. Vous pouvez également envoyer des notifications à partir du serveur dans le format ci-dessous.

```
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"
```

Remarque: les étapes pour configurer Ionic Push pour iOS sont les mêmes, sauf pour la création des profils Push. Pour créer des profils push iOS, reportez-vous à la page <http://docs.ionic.io/v2.0.0-beta/docs/ios-push-profiles>.

Exemple d'application



[Téléchargez l'exemple d'application ici](#) .

Un exemple d'application est joint ici pour référence.

```
IonicApp:
```

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

Remarque: Ceci n'est pas un projet autonome. Le code fourni est uniquement à titre de comparaison avec un projet créé et mis en œuvre en utilisant les procédures indiquées ci-dessus dans ce document, en cas de problèmes ou d'erreurs.

Lire Démarrer avec ionic-framework en ligne: <https://riptutorial.com/fr/ionic-framework/topic/893/demarrer-avec-ionic-framework>

Chapitre 2: Appareil photo et accès à une bibliothèque de photos à partir d'une application ionique

Remarques

renvoie également ce [lien](#)

Exemples

Camara ouverte et galerie de photos

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
}
});

```

Pour Ionic 3 Exemple

Installez les plugins Native Cordova et Ionic:

```

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

```

Votre `app.module.ts` devra injecter la caméra:

```

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 {}

```

L'appareil photo peut être utilisé avec la feuille d'action facilement avec Ionic 3, votre `page.ts` sera

comme ci-dessous:

```
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();

}
}

```

Appelez la fonction `cameragalleryfun` partir de n'importe quel événement comme cliquez sur le bouton, cela retournera la chaîne base64 pour l'image. Plus d'option peut être appliquée. pour une option supplémentaire, voir: <https://github.com/apache/cordova-plugin-camera>

Lire Appareil photo et accès à une bibliothèque de photos à partir d'une application ionique en ligne: <https://riptutorial.com/fr/ionic-framework/topic/7086/appareil-photo-et-acces-a-une-bibliotheque-de-photos-a-partir-d-une-application-ionique>

Chapitre 3: Commencer à créer des applications dans Ionic

Exemples

Démarrer une application ionique

Démarrer une application ionique

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

Les modèles de démarrage peuvent provenir d'un modèle nommé, d'un dépôt Github, d'un Codepen ou d'un répertoire local. Un modèle de démarrage est ce qui devient le répertoire `www` dans le projet Cordova.

Débutants de modèles nommés

- [onglets](#) (par défaut)
- [menu latéral](#)
- [blanc](#)

Github Repo démarreurs

- Toute URL de dépôt Github, ex: <https://github.com/driftyco/ionic-starter-tabs>
- Les modèles nommés sont simplement des alias aux repères ioniques

Codepen URL starters

- Toute URL de Codepen, ex: [<http://codepen.io/ionic/pen/odqCz>][1]
- [Démonstrations ioniques de codepen](#)

Répertoire local:

- Chemin relatif ou absolu vers un répertoire local

Indicateurs / options de ligne de commande:

```
--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
```

Lire Commencer à créer des applications dans Ionic en ligne: <https://riptutorial.com/fr/ionic-framework/topic/3945/commencer-a-creer-des-applications-dans-ionic>

Chapitre 4: Comment utiliser les fonctionnalités d'EcmaScript 6 dans Ionic?

Exemples

En utilisant gulp-babel et gulp-plumber

Ionic utilise Gulp, donc installez gulp-babel et gulp-plumber.

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

Ajoutez babel à `gulpfile.js` comme ceci:

```
//...
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']);
});

//...
```

Modifier `ionic.project` :

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

Maintenant, lorsque vous exécutez `ionic serve`, le code sera transpilé pour vous.

Lire [Comment utiliser les fonctionnalités d'EcmaScript 6 dans Ionic?](https://riptutorial.com/fr/ionic-framework/topic/4847/comment-utiliser-les-fonctionnalites-d-ecma-script-6-dans-ionic) en ligne:

[https://riptutorial.com/fr/ionic-framework/topic/4847/comment-utiliser-les-fonctionnalites-d-](https://riptutorial.com/fr/ionic-framework/topic/4847/comment-utiliser-les-fonctionnalites-d-ecma-script-6-dans-ionic)

Chapitre 5: Composants CSS ioniques

Remarques

Ionic a beaucoup de composants CSS déclarés prêts à vous simplifier la vie tout en codant votre prochaine application mobile hybride. Ces composants varient d'un système de grille de base à la mise en forme de vos formulaires. Ces composants sont à votre disposition si vous choisissez d'installer Ionic avec les feuilles de style CSS prédéfinies.

L'une des fonctions de base d'Ionic CSS vous permet de commencer avec un ensemble de couleurs, mais en règle générale, les couleurs doivent être remplacées. Les couleurs utilitaires sont ajoutées pour vous aider à définir une convention de dénomination. Vous pourriez appeler cela un thème de base de l'application. Pour personnaliser les couleurs, vous pouvez simplement remplacer celles provenant du fichier CSS `ionic.css`. De plus, Ionic étant construit avec Sass, pour plus de puissance et de flexibilité, vous pouvez également modifier et étendre les variables de couleur dans le fichier `_variables.scss`.

Vous pouvez configurer un projet ionique pour utiliser SASS très facilement en exécutant la commande `ionic setup sass` dans votre terminal.

Vous pouvez trouver la documentation officielle sur le CSS ionique ici:

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

Exemples

Syntaxe de base du système de grille

Grille de base

Dans Ionic, vous pouvez déclarer des lignes en définissant la classe `row` sur un élément. Les lignes seront des éléments alignés horizontalement et tout ce qui se trouve dans cet élément appartiendra à la ligne. A l'intérieur de la ligne, vous pouvez déclarer différentes colonnes de largeur. Vous avez le choix entre les déclarations suivantes.

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

Classe	Largeur
.col-67	66.6666%
.col-75	75%
.col-80	80%
.col-90	90%

La valeur maximale des colonnes que peut contenir une ligne est 100. Voici quelques exemples de grille de 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>
```

Grilles Offset

Vous pouvez également définir `col-offset-<value>` sur les colonnes. Dans l'exemple ci-dessous, le tiers de la largeur d'une colonne a un tiers de la largeur, ce qui signifie qu'elle aura une largeur d'un tiers et sera centrée dans la page en raison de son décalage.

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

Aligner les colonnes

Aligner les colonnes verticalement est peut-être en définissant la valeur `col-<align_value>` sur une colonne séparément.

```
<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>
```

Ce qui précède alignera chaque colonne sur elle-même. Si vous souhaitez aligner toutes les colonnes de la ligne, vous pouvez définir la valeur d'alignement sur la ligne elle-même. Dans l'exemple ci-dessous, toutes les colonnes de cette ligne s'aligneront verticalement au centre de la ligne.

```
<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>
```

Grille réactive

Vous pouvez également souhaiter que les colonnes soient réactives à mesure qu'elles s'empileront les unes sur les autres à une certaine largeur de fenêtre. Vous avez trois choix.

Classe	Point d'arrêt (approximativement)
.responsive-sm	Téléphone plus petit que paysage
.responsive-md	Plus petit que la tablette portrait
.responsive-lg	Comprimé plus petit que paysage

Dans cet exemple, ces colonnes s'empileront sous la largeur approximative d'un téléphone de paysage.

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

Vous pouvez également faire vos propres requêtes sur les médias si ces points d'arrêt ne vous conviennent pas et / ou si vous avez besoin de points d'arrêt plus spécifiques.

Syntaxe de base des éléments de liste

Presque chaque application a une sorte de liste. Ionic possède ses propres déclarations CSS d'élément de liste prêtes à l'emploi pour faciliter la création de listes dans votre application. Vous

pouvez soit utiliser des éléments HTML et déclarer une classe pour le ou utiliser la directive `ion-list` pour les créer. L'exemple d'une directive est en bas.

Élément de base de la syntaxe CSS:

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

Liste avec des séparateurs:

```
<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>
```

Liste des éléments avec des icônes:

```
<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>
```

Vous pouvez également définir des icônes des deux côtés des éléments avec la syntaxe suivante:

```
<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>
```

Un élément de liste avec un bouton ou des boutons peut être créé comme ceci:

```
<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>
```

Il est également possible de créer des éléments de liste avec des avatars, des vignettes et des

encarts qui créeront un remplissage autour des éléments de la liste. Ionic gère également la configuration des icônes, etc. dans les éléments de la liste en définissant le remplissage en fonction des éléments de la liste.

Ionic a également ses propres directives pour les cases à cocher, les boutons radio, etc. Voici un exemple de liste de cases à cocher avec Ionic.

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

Utilisation basique des couleurs utilitaires

Preset Ionic CSS aura un thème et des couleurs prédéfinies pour cela. Vous pouvez modifier ou remplacer les valeurs de base dans le fichier `ionic.css` ou dans votre fichier CSS personnalisé. Vous pouvez également les définir avec SASS et, pour utiliser SASS dans Ionic, il vous suffit d'exécuter la commande `ionic setup sass` dans votre terminal.

Utilisation de base des couleurs dans un bouton. La `button-<phrase>` fera de l'arrière-plan du bouton et bordera la couleur du thème défini.

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

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

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

Vos choix de préfixe CSS sont les suivants:

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

Ces classes peuvent être ajoutées également par exemple dans les badges, etc. Voici un exemple de badge:

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

Lire Composants CSS ioniques en ligne: <https://riptutorial.com/fr/ionic-framework/topic/6689/composants-css-ioniques>

Chapitre 6: Connexion ionique avec n'importe quelle base de données

Exemples

Vous ne pouvez pas le faire directement à partir du cadre ionique

La chose est; vous ne pouvez pas vous connecter **directement** à une base de données (MySQL, Postgres, MSSQL, ...). Le mot clé ici est *directement*.

Non, il n'y a pas de solution de contournement, pas de magie impliquée, ce n'est pas la façon dont cela est censé fonctionner. Les travaux ioniques sur Angular et Angular sont des frontaux.

Cependant, la façon de procéder est de créer une API (RESTful) côté serveur.

Cela sera probablement fait avec un langage de serveur (PHP, Go, Python, ...) qui communiquera directement avec votre base de données et l'interrogera.

Après avoir écrit votre API (RESTful), vous pouvez la consommer via vos services dans Angular en utilisant le service `$resource` ou `$http` Angular.

Un exemple d'utilisation de l'API Giphy avec le service `$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);
});
```

Lire Connexion ionique avec n'importe quelle base de données en ligne:

<https://riptutorial.com/fr/ionic-framework/topic/4003/connexion-ionique-avec-n-importe-quelle-base-de-donnees>

Chapitre 7: Créer un dialogue dans Ionic

Paramètres

Paramètres	Détail
<code>title: ''</code> ,	// Chaîne. Le titre de la popup
<code>cssClass: ''</code> ,	// String, le nom de classe CSS personnalisé
<code>subTitle: ''</code> ,	// String (optionnel). Le sous-titre du popup.
<code>template: ''</code> ,	// String (optionnel). Le modèle HTML à placer dans le corps popup.
<code>templateUrl: ''</code> ,	// String (optionnel). L'URL d'un modèle HTML à placer dans le corps popup.
<code>scope: null</code> ,	// Scope (optionnel). Une portée pour créer un lien vers le contenu contextuel.

Remarques

Le service Ionic Popup permet de créer et d'afficher par programme des fenêtres contextuelles nécessitant que l'utilisateur réponde pour continuer.

Exemples

Créer un dialogue dans 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.');
```

Lire Créer un dialogue dans Ionic en ligne: <https://riptutorial.com/fr/ionic-framework/topic/6461/creer-un-dialogue-dans-ionic>

Chapitre 8: Crochets Ionic CLI

Remarques

introduction

Les crochets sont des morceaux de code exécutés par Cordova CLI à certains points de votre application Cordova / Ionic. Les crochets peuvent être utilisés par exemple pour manipuler des fichiers dans notre projet, ajouter automatiquement des plug-ins dans votre application ou, comme dans l'exemple ci-dessus, rechercher des erreurs de code dans vos fichiers.

Note : Il est fortement recommandé d'écrire vos hooks en utilisant Node.js pour qu'ils soient multi-plateforme mais vous pouvez aussi les écrire par exemple en [Javascript](#) .

Types de crochet

Les types de hook suivants sont supportés et l'ordre d'exécution est assez explicite selon le nom.

```
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.
```

Façons de définir des crochets:

Les hooks peuvent être définis dans le `config.xml` du projet en utilisant des éléments `<hook>`, par exemple:

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

En tant que développeur de plug-ins, vous pouvez définir des scripts de hook utilisant les éléments `<hook>` dans un `plugin.xml` comme ceci:

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

`before_plugin_install`, `after_plugin_install`, les `before_plugin_uninstall` `plugin` `before_plugin_uninstall` seront lancés exclusivement pour le plug-in installé / désinstallé.

Remarque : Le placement des hooks dans le répertoire `root/hooks` est considéré comme obsolète en faveur des éléments `hook` dans `config.xml` et `plugin.xml`. Si vous utilisez toutefois cette approche, n'oubliez pas de définir les droits d'exécution sur les fichiers du dossier `root/hooks`.

La documentation pour les crochets Cordova peut être trouvée [ici](#).

Exemples

Vérification des erreurs dans vos fichiers Javascript dans `before_prepare` en utilisant `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);
    }
  });
}

```

Lire Crochets Ionic CLI en ligne: <https://riptutorial.com/fr/ionic-framework/topic/6520/crochets-ionic-cli>

Chapitre 9: Défilement infini ionique pour afficher les éléments de charge à la demande (données déjà disponibles non par requête HTTP)

Exemples

Charger n nombre de données disponibles à la demande

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>
```

Manette :

```
$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')
}
```

Chargez 10 éléments à chaque fois que addMoreItem () appelle.

Lire Défilement infini ionique pour afficher les éléments de charge à la demande (données déjà disponibles non par requête HTTP) en ligne: <https://riptutorial.com/fr/ionic-framework/topic/9490/defilement-infini-ionique-pour-afficher-les-elements-de-charge-a-la-demande--donnees-deja-disponibles-non-par-requete-http->

Chapitre 10: Déployer Ionic en tant que site Web

Exemples

Copiez simplement le dossier www sur votre serveur Web

Ionic 1.2 prend officiellement en charge le déploiement en tant que site Web

Si vous n'utilisez pas de plug-ins Cordova, il n'y a pas de problème (si vous le souhaitez vraiment) à télécharger le contenu du dossier `www` sur votre serveur, et voilà - vous aurez la même application.

Cependant, il est important de noter que Ionic 1 n'a jamais été conçu pour une telle utilisation, et que les utilisateurs de votre "site Web" devront disposer du navigateur le plus récent pour voir le "site Web" correctement (Ionic utilise CSS ou HTML que certains anciens navigateurs ne supportent pas).

Lire Déployer Ionic en tant que site Web en ligne: <https://riptutorial.com/fr/ionic-framework/topic/4001/deployer-ionic-en-tant-que-site-web>

Chapitre 11: Exécuter une application ionique sur un émulateur ou sur votre téléphone

Exemples

Exécuter une application ionique sur un émulateur ou sur votre téléphone

1. Ajouter une cible de plate-forme

iOS:

```
$ ionic platform add ios
```

Android:

```
$ ionic platform add android
```

Les fenêtres:

```
$ ionic platform add windows
```

2. Construisez votre application

iOS:

```
$ ionic build ios
```

Android:

```
$ ionic build android
```

Les fenêtres:

```
$ ionic build windows
```

Application de rechargement en direct pendant le développement (version bêta)

La commande `run` ou `emulate` déploiera l'application sur les périphériques / émulateurs de plate-forme spécifiés. Vous pouvez également exécuter le **rechargement en direct** sur le périphérique

de plate-forme spécifié en ajoutant l'option `--livereload` . La fonctionnalité de rechargement en direct est similaire à `ionic serve` , mais au lieu de développer et de déboguer une application à l'aide d'un navigateur standard, l'application hybride compilée elle-même surveille les modifications apportées à ses fichiers. Cela réduit la nécessité de reconstruire constamment l'application pour les petites modifications. Cependant, toute modification apportée aux plugins nécessitera toujours une reconstruction complète. Pour que le rechargement en direct fonctionne, la machine et le périphérique de développement doivent se trouver sur le même réseau local et le périphérique doit prendre en charge [les sockets Web](#) .

Lorsque le rechargement en direct est activé, les journaux de la console d'une application peuvent également être imprimés sur l'invite de commande du terminal en incluant l'option `--consolelogs` ou `-c` . De plus, les journaux de requêtes du serveur de développement peuvent être imprimés à l'aide des options `--serverlogs` ou `-s` .

Indicateurs de ligne de commande / options pour `run` et `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]
```

Pendant que le serveur est en cours d'exécution pour un rechargement en direct, vous pouvez utiliser les commandes suivantes dans l'interface de ligne de commande:

```
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. Émuler votre application

Déployez l'application Ionic sur un émulateur de plate-forme spécifié. Ceci est simplement un alias pour `run --emulator` .

iOS:

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

Android:

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

Les fenêtres:

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

Pendant l'application émulant dans AVD ou les mobiles, vous pouvez inspecter cette application dans le navigateur chrome. Tapez la commande suivante dans la barre d'adresse du navigateur chrome.

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

4. Exécuter votre application

Déploye l'application ionique sur les périphériques de plate-forme spécifiés. Si un périphérique n'est pas trouvé, il sera ensuite déployé sur un émulateur / simulateur.

iOS:

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

Android:

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

Les fenêtres:

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

4.1. Spécifier votre cible

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

Vous pouvez vérifier le nom cible de votre appareil / émulateur exécutant les `$ adb devices`.

[Lire Exécuter une application ionique sur un émulateur ou sur votre téléphone en ligne:](https://riptutorial.com/fr/ionic-framework/topic/4175/executer-une-application-ionique-sur-un-émulateur-ou-sur-votre-téléphone)
<https://riptutorial.com/fr/ionic-framework/topic/4175/executer-une-application-ionique-sur-un-émulateur-ou-sur-votre-téléphone>

Chapitre 12: Extensions ioniques AngularJS

Remarques

Ionic offre une variété d'extensions Javascript AngularJS à utiliser. Ces extensions peuvent être des entrées de formulaire normales ou des fenêtres modales et rendent le codage de votre application de base beaucoup plus rapide en utilisant ces extensions prêtes.

Exemples

Formulaire de saisie

Les entrées ioniques ne sont pas différentes des entrées HTML normales, mais elles sont conçues différemment et utilisées comme une directive. Vous pouvez également utiliser des entrées HTML normales dans les applications ioniques. Voici quelques exemples de base que Ionic propose prêt à l'emploi.

Case à cocher:

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

Bouton radio:

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

Basculer:

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

Fenêtres modales

Ionic a sa propre extension pour afficher une fenêtre modale. Les modaux peuvent être créés en insérant le modèle directement dans la vue avec une `<script>` ou en utilisant un fichier de modèle distinct. Dans cet exemple, nous supposons que vous avez un fichier HTML nommé `modal-template.html` dans un dossier appelé `templates`. Vous définissez le chemin du modèle dans la fonction d'initialisation modale avec `fromTemplateUrl`.

Création de l'objet modal dans la portée

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>
```

Manette

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

Contrôler le modal

Vous pouvez alors contrôler le modal avec les commandes suivantes.

Modal ouvert

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

Fermer modal

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

Supprimer modal

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

Événements modaux

Vous pouvez écouter des événements modaux avec les fonctions suivantes.

Le modal est caché

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

Le modal est supprimé

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

Lire Extensions ioniques AngularJS en ligne: <https://riptutorial.com/fr/ionic-framework/topic/6446/extensions-ioniques-angularjs>

Chapitre 13: Ionic - Analysez votre application avec jshint et gulp-jshint dans le cadre de votre processus de construction

Remarques

Linting votre application ionique avant de courir a d'énormes avantages. Il analysera le code pour détecter les erreurs potentielles et vous fera gagner énormément de temps.

Qu'est-ce que le linting et comment installer les paquets requis?

"Linting est le processus d'exécution d'un programme qui analysera le code pour détecter les erreurs potentielles." - voir Qu'est-ce que "Linting"?

Votre application ionique est livrée avec un fichier package.json. Accédez à la racine de votre application dans une ligne de commande / terminal et installez les packages suivants:

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

Exemples

Ajouter une tâche Gulp

Dans la racine de votre application ionique, il y a un fichier gulpfile.js. Ouvrez-le dans un éditeur et collez la tâche gulp suivante:

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

Ceci recherche un dossier appelé "js" dans le dossier "www". Si vous avez d'autres dossiers contenant des fichiers JavaScript, ajoutez-les également. Par exemple, ajoutons également un dossier appelé «vues»:

```
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'))
  });
```

Explications:

- 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.

Créer un fichier `.jshintrc` (facultatif)

Créez un fichier nommé `".jshintrc"` dans la racine de votre application, où `package.json` est.

* Remarque sur Windows: créez un fichier nommé `"jshintrc.txt"`. Puis renommez-le en `".jshintrc"`. (remarquez le point à la fin).

Ceci est un fichier de configuration. Il peut par exemple dire à jshint d'ignorer certaines variables et bien d'autres choses. Voici le mien:

```
{
  "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
}
```

Ajouter Makefile

1. Créez un fichier nommé: `"Makefile"` (sans extension) dans la racine de votre application
2. Ouvrez-le dans un éditeur de texte et ajoutez ceci:

```
android:
  gulp lint
```

```
gulp sass
ionic run android --device

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

Cela va compiler votre application et si cela passe, elle compilera sass et créera votre application.

Utilisation: Pour exécuter votre application, au lieu du programme "ionic run android --device", exécutez ces commandes:

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

Lire Ionic - Analysez votre application avec jshint et gulp-jshint dans le cadre de votre processus de construction en ligne: <https://riptutorial.com/fr/ionic-framework/topic/4889/ionic---analysez-votre-application-avec-jshint-et-gulp-jshint-dans-le-cadre-de-votre-processus-de-construction>

Chapitre 14: Ionicons

Remarques

Dans le développement Web moderne, il est courant d'utiliser des polices pour afficher des icônes. Les polices étant des vecteurs, elles sont indépendantes de la résolution et peuvent être facilement colorées via CSS, pour ne citer que quelques avantages par rapport aux images bitmap, etc. Les ioniques ont été créés par la même équipe que Ionic Framework ils sont 100% gratuits et open source. MIT Licensed.

Les ioniques peuvent être utilisés seuls ou avec des composants CSS Ionics quand ils ont certains styles en fonction des éléments parents.

La page d'accueil et la liste des icônes peuvent être trouvées ici: <http://ionicons.com/>

Exemples

Utilisation de base

Les icônes de police sont généralement placées dans une `<i>`. Ionic a des styles CSS par défaut pour les icônes pour une utilisation facile. L'exemple le plus fondamental d'utilisation:

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

Utilisation étendue

Ionic possède des composants CSS dans lesquels vous pouvez utiliser les ioniques par défaut avec un style prédéfini. La classe `range` dans l'élément `<div>` appliquera un style correct à la fois à l'entrée et aux icônes qu'elle contient. Voici un exemple de curseur de plage.

```
<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 autre exemple d'utilisation d'ionicon dans les onglets ioniques créera un menu similaire à un onglet. Les classes `tabs-striped tabs-color-assertive` définissent le style des onglets eux-mêmes. Les icônes sont utilisées avec de simples balises `<i>` et elles obtiennent leur style de position à partir des classes appliquées aux divs parents.

```
<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>
```

Lire Ionicons en ligne: <https://riptutorial.com/fr/ionic-framework/topic/5886/ionicons>

Chapitre 15: Nuage ionique pour projets ioniques Yeoman

Exemples

Plate-forme ionique (nuage ionique) pour projets ioniques Yo (Yeoman)



Plate - forme ionique :

Construisez, poussez, déployez et redimensionnez vos applications ioniques de manière simple.

Description du titre:

Ionic Platform est une plate-forme cloud pour la gestion et la mise à l'échelle des applications mobiles multiplates-formes. Les services intégrés vous permettent, à vous et à votre équipe, de créer, déployer et développer efficacement vos applications.

Objectif du document:

Ionic Platform fonctionne bien avec les projets ioniques standard. Mais les projets qui suivent une structure de répertoire non standard peuvent être confrontés à quelques obstacles. Ce document fournit les étapes pour utiliser Ionic Platform dans les projets ioniques créés à l'aide de Yeoman.

Portée du document:

Ce document décrit les étapes de base pour créer un projet ionique à l'aide de Yeoman et l'intégrer à Ionic Platform à l'aide du client Web Ionic Platform. Ce document couvre les étapes de base pour utiliser Ionic Deploy, Ionic Analytics et Ionic Push.

Public visé:

Le public visé par ce document est constitué par les développeurs d'applications Web / mobile, avec une expertise de niveau débutant et expert, qui connaissent bien les prérequis ci-dessous.

Conditions préalables:

Vous devriez vous familiariser avec les frameworks / outils suivants avant d'essayer ce document.

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

- Plate-forme ionique: <https://ionic.io/platform>
-

Générateur ionique



Un générateur pour la structure ionique de Yeoman, l'outil d'échafaudage Web pour les applications Web modernes

Node.js est un moteur d'exécution JavaScript basé sur le moteur JavaScript V8 de Chrome. npm est le gestionnaire de paquets pour JavaScript. Téléchargez et installez Node (et npm) à partir de <http://nodejs.org>

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

Yeoman vous aide à lancer de nouveaux projets, en prescrivant les meilleures pratiques et les outils pour vous aider à rester productif.

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

Dans *package.json*, incluez les éléments suivants dans devDependencies

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

Dans *bower.json*, incluez les éléments suivants dans les dépendances

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

Dans *Gruntfile.js*, changez le dossier de *scripts* en *"js"* . Changer dans *index.html* si nécessaire.

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

Puis courir

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

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

client-plateforme ionique



Un client Web qui fournit des interactions avec la plate-forme ionique.

Nous avons besoin de code pour permettre à votre application de communiquer avec la plate-forme ionique. Nous devons ajouter le client Web de la plate-forme ionique pour que l'application ionique puisse s'interfacer avec les plug-ins et la plate-forme Ionic.io.

```
$ ionic io init
```

Dans votre *app.js*, ajoutez la dépendance de module *'ionic.service.core'*. Dans *Gruntfile.js*, ajoutez la tâche *"ionicSettings"* ci-dessous.

```
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']);
```

Ajoutez les *'ionicSettings'* dans *init* et *compressez les tâches* après la copie . Dans *index.html*, déplacez la balise ci-dessous après toutes les déclarations de balises.

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

Puis courir

```
$ Grunt serve
```

Déploiement ionique



Poussez les mises à jour en temps réel sur vos applications de production et gérez l'historique des versions.

Ionic Deploy vous permet de mettre à jour votre application à la demande, pour toutes les modifications ne nécessitant pas de modifications binaires, vous permettant de gagner des jours, voire des semaines, de temps d'attente. Suivez la procédure ci-dessous pour configurer Ionic Deploy pour votre application.

Dans *Gruntfile.js*, ajoutez la tâche grunt *"deploy"* comme indiqué ci-dessous.

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

puis courir

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

Code de déploiement ionique:

```
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) {  
});
```

Déploiement des mises à jour:

Envoyez un nouveau code pour votre application.

Créez un fichier apk et installez votre application. Apportez quelques modifications à votre code et déployez les modifications à l'aide de « *grunt deploy* ». Ensuite, mettez-le à jour depuis votre application.

Vous pouvez également le déployer à partir du tableau de bord *apps.ionic.io* . Vous pouvez déployer l'application sans le paramètre *deploy*. Ensuite, dans le tableau de bord, vous pouvez ajouter les détails des métadonnées et des versions et déployer l'application à partir de là.

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

Analyse ionique



Afficher le flux en direct des événements ou le nombre brut / unique d'événements / utilisateurs au fil du temps.

Combien d'utilisateurs y a-t-il actuellement sur votre application? Combien d'entre eux utiliseront votre application demain ou la semaine prochaine? Sans information, vous n'avez aucun moyen de savoir si votre application est utilisée de la manière que vous attendez. Suivez la procédure ci-dessous pour configurer Ionic Analytics pour votre application.

Dans votre *app.js*, ajoutez la dépendance de module ' *ionic.service.analytics* ' après le *ionic.service.core* Exécutez la méthode du registre d'analyse dans la fonction d'exécution de notre module.

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

Dans Ionic Analytics, chaque action suivie effectuée par un utilisateur dans votre application est représentée par un objet événement. Un événement est une action unique effectuée à un moment précis. Pour suivre vos propres événements, appelez `$ionicAnalytics.track(eventType, eventData)` chaque fois qu'une action se produit.

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

La directive *ion-track-tap* envoie un événement lorsque son élément hôte est tapé. La directive

associée *ion-track-data* joint les données d'événement.

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

Dans le *tableau de bord apps.ionic.io*, vous pouvez afficher les données d'analyse suivantes:

Événements: affichez le nombre brut d'événements au fil du temps ou le nombre d'utilisateurs uniques ayant terminé un événement. Un événement peut être n'importe quoi, depuis un utilisateur chargeant l'application, jusqu'à la confirmation d'un achat.

Entonnoirs: un entonnoir est une séquence d'actions que vous attendez des utilisateurs dans votre application, menant à un objectif défini. L'utilisation judicieuse des entonnoirs vous aidera à améliorer les taux de conversion.

Segments: affiche les événements au fil du temps, regroupés par une propriété spécifiée. Ou bien, calculez le pourcentage d'événements correspondant à une propriété donnée. Les segments vous aident à comprendre votre base d'utilisateurs et à voir comment les propriétés changent avec le temps.

Rétention: suivez la durée pendant laquelle les utilisateurs sont actifs sur votre application avant de cesser de l'utiliser. Ou, indiquez le temps nécessaire aux utilisateurs pour atteindre un objectif défini, comme une vente terminée.

Pulse: Un flux en direct des événements venant de vos utilisateurs.

Push ionique



Envoyez des notifications push ciblées et automatisées à vos utilisateurs.

Ionic Push vous permet de créer des notifications push ciblées via un tableau de bord simple qui sera envoyé automatiquement lorsque les utilisateurs correspondent à des critères spécifiques, et offre une API simple pour envoyer des notifications push depuis vos propres serveurs.

Profils Push Android:

Les notifications push Android utilisent le service *Google Cloud Messaging* (GCM). Ouvrez la [console Google Developers](#) et créez un projet. Copiez le *numéro de votre projet*. Ce sera le **numéro d'expéditeur GCM** ou le **numéro de projet GCM**.

Dans la section *API Manager*, activez l'*API Google Cloud Messaging*. Ensuite, accédez à la section *Informations d'identification* et sélectionnez *Créer des informations d'identification*, puis choisissez *Clé API*, puis *Clé du serveur*. Nommez votre clé API et laissez le champ *Accepter les*

demandes de ... vide et cliquez sur *Créer* . Enregistrez votre **clé API** !

Authentification:

Accédez au tableau de bord de votre application sur la [plate - forme ionique](#) et accédez à *Paramètres -> Certificats* . Si vous ne l'avez pas encore fait, créez un nouveau profil de sécurité, puis appuyez sur *edit* . Notez l' **étiquette de profil** .

Maintenant, cliquez sur l'onglet *Android* et recherchez la section marquée *Google Cloud Messaging* , saisissez la **clé API** que vous avez générée sur la console développeur Google, puis cliquez sur *Enregistrer* . Allez dans *Paramètres -> Clés API* . Sous *API Tokens* , créez un nouveau jeton et copiez-le. Ce sera votre **jeton 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
```

Note: phonegap-plugin-push nécessite Android Support Repository version 32+

Dans votre *app.js*, ajoutez la dépendance de module ' *ionic.service.push* ' après le fichier *ionic.service.core*

Code Push ionique:

Initialisez le service et enregistrez votre appareil dans la fonction d'exécution de votre module. Vous aurez besoin du jeton de périphérique enregistré par l'utilisateur pour envoyer une notification à l'utilisateur.

```
$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();
```

puis courir

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

Ionic Push vous permet de créer des notifications push ciblées via le tableau de bord. Vous pouvez également envoyer des notifications à partir du serveur dans le format ci-dessous.

```
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"
```

Remarque: les étapes pour configurer Ionic Push pour iOS sont les mêmes, sauf pour la création des profils Push. Pour créer des profils push iOS, reportez-vous à la page <http://docs.ionic.io/v2.0.0-beta/docs/ios-push-profiles>.

Exemple d'application



Téléchargez l'exemple d'application [ici](#) .

Un exemple d'application est joint ici pour référence.

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

Remarque: Ceci n'est pas un projet autonome. Le code fourni est uniquement à titre de comparaison avec un projet créé et mis en œuvre en utilisant les procédures indiquées ci-dessus dans ce document, en cas de problèmes ou d'erreurs.

Lire Nuage ionique pour projets ioniques Yeoman en ligne: <https://riptutorial.com/fr/ionic-framework/topic/6389/nuage-ionique-pour-projets-ioniques-yeoman>

Chapitre 16: Publier votre application ionique

Exemples

Version de construction à partir de macOS

Construire l'APK

Tout d'abord, nous devons construire l'APK.

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

Générer une clé privée

Ensuite, nous allons créer un fichier de clés pour signer l'APK. `keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000`

- Changez **ma clé de version** avec votre nom de clé.
- Modifiez **alias_name** avec votre alias de clé.

Signer l'APK

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

- Changez **ma clé de version** avec votre nom de clé.
- Modifiez **HelloWorld-release-Unigned** avec votre apk non signé. `ionic-project/platforms/android/build/outputs/apk .`
- Modifiez **alias_name** avec votre alias de clé.

Zip the APK

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

- Vous pouvez trouver `zipalign` dans `/Users/username/Library/Android/sdk/build-tools/XXX/`
- Modifiez **HelloWorld-release-Unigned** avec votre apk non signé. `ionic-project/platforms/android/build/outputs/apk .`
- Changez **HelloWorld.apk** avec votre nom de fichier apk préféré. Cela sera téléchargé sur Google Play.

Lire Publier votre application ionique en ligne: <https://riptutorial.com/fr/ionic-framework/topic/7755/publier-votre-application-ionique>

Chapitre 17: Quelle est la différence entre «construction ionique» et «préparation ionique»?

Exemples

construction ionique vs préparation ionique

De la documentation officielle:

Si vous souhaitez avancer, vous pouvez également ouvrir le fichier de projet pour une plate-forme spécifique en ouvrant le projet XCode ou Android Eclipse requis dans les plates-formes / PLATFORM à la racine de votre projet. Ensuite, vous pouvez créer et tester depuis l'IDE spécifique à la plate-forme. Remarque: si vous allez dans cette voie, je vous recommande de continuer à travailler à l'intérieur du dossier www racine, et lorsque vous aurez apporté des modifications à ce dossier, exécutez la commande: `$ cordova prepare ios` qui mettra à jour le projet spécifique iOS avec le code du dossier `www`. Remarque: cette opération remplace les modifications apportées aux `platforms/ios/www` et à d'autres dossiers spécifiques à la plate-forme.

Donc, pour résumer cette partie - si vous utilisez XCode pour tester et exécuter votre code, après avoir modifié une partie du code, il vous suffit de lancer `ionic prepare` pour mettre à jour le projet iOS que vous continuez à utiliser dans XCode.

`ionic build` commande `ionic build` prépare en fait le fichier final (par exemple sous Android, c'est le fichier .apk) qui pourrait ensuite être copié sur votre appareil et testé en l'exécutant manuellement sur l'appareil (ou en utilisant la commande `ionic emulate` pour le tester sur l'émulateur).

Lire [Quelle est la différence entre «construction ionique» et «préparation ionique»? en ligne: https://riptutorial.com/fr/ionic-framework/topic/2911/quelle-est-la-difference-entre--construction-ionique--et--preparation-ionique--](https://riptutorial.com/fr/ionic-framework/topic/2911/quelle-est-la-difference-entre--construction-ionique--et--preparation-ionique--)

Chapitre 18: Services Ionic Backend (ionic.io)

Exemples

Introduction et configuration

La [plate-forme ionique](#) offre une gamme de services et d'outils de backend mobiles puissants, axés sur les systèmes hybrides, qui facilitent la mise à l'échelle des applications hybrides, belles et performantes.

Pour utiliser Ionic Platform, vous devez avoir [installé Ionic Framework](#) .

1.) Inscription ([inscription](#))

Vous devez entrer: nom, nom d'utilisateur, société, téléphone (facultatif), adresse électronique et mot de passe. Après avoir reçu l'e-mail de confirmation, vous pouvez vous [connecter](#) .

Bienvenue chez Ionic. Créons votre première application! Créez une nouvelle application ci-dessus, puis suivez le guide de [démarrage rapide](#) pour télécharger votre première application.

2.) Configuration

Tout d'abord, [créez votre première application ionique](#) . Maintenant, vous pouvez ajouter le client Web de la plate-forme:

Le client Web vous permet d'interagir avec les services de la plate-forme ionique depuis votre code d'application.

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

Maintenant, vous avez besoin de la plate-forme pour attribuer à votre application un identifiant d'application unique et une clé api. Pour ce faire, utilisez la commande `ionic io init` .

```
$ ionic io init
```

Cela vous invitera automatiquement à vous connecter à votre compte Platform. L'ID d'application et la clé api seront alors stockés dans la [configuration](#) Ionic Platform de votre projet.

Votre application est maintenant connectée à la plate-forme ionique et sera répertoriée dans votre [tableau de bord](#) .

3.) Maintenant, vous pouvez passer à l'installation d'un des services Ionic Platform :

- [Utilisateurs / Auth](#)
- [Déployer](#)

- Pousser
- Paquet
- Analytique

Lire Services Ionic Backend (ionic.io) en ligne: <https://riptutorial.com/fr/ionic-framework/topic/3622/services-ionic-backend--ionic-io->

Chapitre 19: Test de l'application ionique dans un navigateur

Remarques

Les tests de fonctionnalités natives telles que Camera, Vibration et autres, dont beaucoup se trouvent dans la documentation d' [Ionic Native](#) , ne peuvent pas être effectués dans le navigateur. Il s'agit d'une limitation inhérente au fait que Cordova, la plate-forme sur laquelle Ionic dépend pour pouvoir accéder aux API natives Android, iOS et Windows Mobile d'un périphérique, ne peut pas s'exécuter sur le navigateur.

On peut contourner ce problème en se moquant des fonctionnalités du plug-in natif.

Exemple

Voici un exemple sur la façon de simuler le plug-in [Camera](#) :

Allez-y et créez un dossier facultatif dans le dossier racine de votre projet.

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

Ouvrez `camera-mock.ts` et copiez le code suivant:

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

Ensuite, ouvrez `src/app.module.ts` et importez la classe mock "

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

Puis ajoutez-le au tableau des fournisseurs de modules:

```
@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 {}

```

Maintenant, vous pouvez l'utiliser dans n'importe quel composant après l'avoir importé.

Exemples

Test dans un navigateur

Utilisez `ionic serve` à démarrer un serveur de développement local pour le développement et le test des applications. Ceci est utile à la fois pour tester le navigateur de bureau et pour tester dans un navigateur de périphérique connecté au même réseau. En outre, cette commande démarre LiveReload, qui permet de surveiller les modifications du système de fichiers. Dès que vous enregistrez un fichier, le navigateur est automatiquement actualisé. Jetez un coup d'œil aux [documents Sass](#) si vous souhaitez également voir les fichiers Sass du projet `ionic serve`.

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

Pour le navigateur Chrome, vous pouvez inspecter les périphériques (AVD ou Mobiles), tapez la commande suivante dans la barre d'adresse du navigateur Chrome.

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

LiveReload

Par défaut, LiveReload surveille les modifications dans votre répertoire `www/`, à l'exclusion de `www/lib/`. Pour changer cela, vous pouvez spécifier une propriété `watchPatterns` dans le fichier `ionic.project` situé dans la racine de votre projet pour regarder (ou ne pas regarder) les modifications spécifiques.

```

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

```

Pour obtenir une référence sur la syntaxe du modèle glob, consultez les [modèles de globalisation](#) sur le site Web Grunt.

Remarque:

```
$ ionic setup sass
```

ajoutera une `watchPatterns` `watchPatterns` avec les valeurs par défaut de votre fichier `ionic.project`, que vous pourrez ensuite modifier, en plus de la propriété `gulpStartupTasks`, comme décrit dans la [documentation de Sass](#).

Laboratoire ionique

Laboratoire ionique <http://ionicframework.com/img/blog/lab.png>

Ionic Lab est une fonctionnalité de `ionic serve` qui permet de tester facilement votre application dans un cadre téléphonique et avec les plates-formes iOS et Android côte à côte. Pour l'utiliser, lancez simplement

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

Lisez l' [annonce de la version complète](#) pour tous les détails!

Spécifier une adresse IP à utiliser

Supposons que vous souhaitiez spécifier l'adresse à laquelle votre navigateur se connectera, par exemple pour les tests ou les utilisateurs externes. Spécifiez l'adresse avec l'argument `--address`.

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

Proxies de service

La commande `serve` peut ajouter des proxies au serveur http. Ces proxy sont utiles si vous développez dans le navigateur et que vous devez passer des appels vers une API externe. Avec cette fonctionnalité, vous pouvez envoyer une requête proxy à l'API externe via le serveur HTTP ionique en empêchant que l'en `No 'Access-Control-Allow-Origin' header is present on the requested resource` **erreur de** `No 'Access-Control-Allow-Origin' header is present on the requested resource`.

Dans le fichier `ionic.project`, vous pouvez ajouter une propriété avec un tableau de proxies à ajouter. Les proxies sont des objets avec deux propriétés:

- `path`: chaîne qui sera mise en correspondance avec le début de l'URL de demande entrante.
- `proxyUrl`: une chaîne avec l'URL de la destination de la demande par proxy.

```
{  
  "name": "appname",  
  "email": "",  
  "proxies": [  
    {  
      "path": "/api/",  
      "proxyUrl": "http://localhost:3000/"  
    }  
  ]  
}
```

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

En utilisant la configuration ci-dessus, vous pouvez maintenant envoyer des requêtes à votre serveur local à l' `http://localhost:8100/v1` pour qu'il envoie les requêtes de proxy à `https://api.instagram.com/v1`

Par exemple:

```
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;
})

})
```

Voir aussi [cet aperçu](#) pour plus d'aide.

Indicateurs / options de ligne de commande

```
[--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
```

Lire Test de l'application ionique dans un navigateur en ligne: <https://riptutorial.com/fr/ionic-framework/topic/3256/test-de-l-application-ionique-dans-un-navigateur>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec ionic-framework	Akshay Khale , Andrea Macchieraldo , Community , Devid Farinelli , Ian Pinto , leetheguy , Lightbeard , Mazz , Newton Joshua , the_mahasagar
2	Appareil photo et accès à une bibliothèque de photos à partir d'une application ionique	Pritish , sonu
3	Commencer à créer des applications dans Ionic	A-Droid Tech , Ian Pinto
4	Comment utiliser les fonctionnalités d'EcmaScript 6 dans Ionic?	Nikola
5	Composants CSS ioniques	thepio
6	Connexion ionique avec n'importe quelle base de données	Nikola
7	Créer un dialogue dans Ionic	A-Droid Tech
8	Crochets Ionic CLI	thepio
9	Défilement infini ionique pour afficher les éléments de charge à la demande (données déjà disponibles non par requête HTTP)	SANAT
10	Déployer Ionic en tant que site Web	Nikola

11	Exécuter une application ionique sur un émulateur ou sur votre téléphone	A-Droid Tech , Gerard Cuadras , Ian Pinto , Ketan Akbari , olivier , Raymond Ativie
12	Extensions ioniques AngularJS	thepio
13	Ionic - Analysez votre application avec jshint et gulp-jshint dans le cadre de votre processus de construction	Akilan Arasu , Ian Pinto , Sumama Waheed , thepio
14	Ionicons	thepio
15	Nuage ionique pour projets ioniques Yeoman	Newton Joshua
16	Publier votre application ionique	Gerard Cuadras
17	Quelle est la différence entre «construction ionique» et «préparation ionique»?	Nikola
18	Services Ionic Backend (ionic.io)	Tomislav Stankovic
19	Test de l'application ionique dans un navigateur	A-Droid Tech , Ian Pinto , Ketan Akbari , maninak