

Chapitre 1. Cordova

1.1. Installation poste de développement

Sur votre mac, au niveau de votre utilisateur vous devez avoir un fichier (~/bash_profile, ~/.zshrc, ~/.profile, or ~/.bashrc) qui vous permettra de configurer vos variables d'environnement et votre path.

1.1.1. Node.js

- Installez le module global node cordova

```
npm install -g cordova
```

- Vérifiez que cordova est bien installé `cordova help`

```
cordova help
Synopsis

    cordova command [options]

Global Commands
  create ..... Create a project
  help ..... Get help for a command
  telemetry ..... Turn telemetry collection on or off
  config ..... Set, get, delete, edit, and list
global cordova options

Project Commands
  info ..... Generate project information
  requirements ..... Checks and print out all the
requirements
                                for platforms specified

  platform ..... Manage project platforms
  plugin ..... Manage project plugins

building
  prepare ..... Copy files into platform(s) for
  compile ..... Build platform(s)
  clean ..... Cleanup project from build artifacts

  run ..... Run project
              (including prepare && compile)
  serve ..... Run project with a local webserver
              (including prepare)
```

Learn more about `command` options using '`cordova help <command>`'

Aliases

```
build -> cordova prepare && cordova compile
emulate -> cordova run --emulator
```

Options

```
-v, --version ..... prints out this utility's version
-d, --verbose ..... debug mode produces verbose log
output for all activity,
--no-update-notifier ..... disables check for CLI updates
--nohooks ..... suppress executing hooks
(taking RegExp hook patterns as
parameters)
```

Examples

```
cordova create myApp org.apache.cordova.myApp myApp
cordova plugin add cordova-plugin-camera
cordova platform add android
cordova plugin add cordova-plugin-camera --nosave
cordova platform add android --nosave
cordova requirements android
cordova build android --verbose
cordova run android
cordova build android --release -- --keystore="..\android.keystore"
--storePassword=android --alias=mykey
cordova config ls
cordova platform add ios --nofetch
cordova plugin add cordova-plugin-camera --nofetch
```

- Créez un premier projet cordovaFormation
- Ajoutez les plateformes browser, android et ios
- lancez l'application avec la plateforme browser

```
cordova run browser
```

1.1.2. Android

- Java doit être installé sur la machine et la variable d'environnement `$JAVA_HOME` déclarée dans le fichier (`~/.bash_profile`, `~/.zshrc`, `~/.profile`, or `~/.bashrc`).

```
// sur ma machine
export JAVA_HOME=$(/usr/libexec/java_home)
```

- Android sdk doit être installé sur la machine (via Android Studio) et la variable d'environnement `$ANDROID_HOME` déclarée

```
// sur ma machine
// Pensez à changer le user ;)
export ANDROID_HOME=/Users/nartawak/Library/Android/sdk
```

- Les tools et les platforms-tools doivent être déclarés dans le path

```
// sur ma machine
// Pensez à changer le user ;)
export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools
```

- Fermez et réouvrez votre terminal (ou source ~/.profile)
- Vous devriez avoir le contexte android disponible en ligne de commande. Pour vérifier **adb devices**
- Dans le projet cordovaFormation, exécutez la commande suivante pour vérifier que le contexte android est bien en place

```
cordova requirements android
```

1.1.3. iOS

- Vérifiez que xcode est bien installé

```
xcode-select -p
```

- Installez les Xcode Command line tools

```
xcode-select --install
```

- Installez le module global node ios-deploy

```
npm install -g ios-deploy
```

- Cocoapods doit être présent sur la machine. C'est un exécutable ruby. Ruby doit donc être installé sur la machine, par chance c'est installé par défaut sur un mac.



La bonne pratique serait normalement d'installer un ruby version manager (rbenv ou rvm), principalement parce que le ruby du mac nécessite des droits root pour installer des dépendances.

Pour la formation, nous allons, pour des questions de simplicité, le faire quand même :(

```
ruby -v // Vérifier que ruby est quand même bien là !!  
sudo gem install cocoapods  
pod setup // Met les pods à jour
```

- Dans le projet cordovaFormation, exécutez la commande suivante pour vérifier que le contexte ios est bien en place

```
cordova requirements ios
```

1.2. Run & Debug

Afin de connaître les différentes options pour lancer une application

```
cordova help run
```

1.2.1. Android

La plateforme Android doit être ajoutée préalablement au projet.

Run

- Branchez votre device Android sur la machine
- Exécutez la commande Cordova

```
[source, shell]  
----  
    cordova run android --device  
----
```

Debug

Chrome doit être installé sur le poste de développement. Vous devez avoir activé les outils de développements sur le device.

- Lancez l'application comme précédemment
- Ouvrez Chrome
- Dans la barre d'adresse, saisissez l'URL suivante: `chrome://inspect/#devices`
- Cliquez sur `inspect` pour votre application

Vous pouvez aussi, ouvrir l'application Android Native (./platforms/android) dans Android Studio. Cela permet de pouvoir déboguer natif en même temps que la partie web.

1.2.2. iOS

La plateforme iOS doit être ajoutée préalablement au projet.

Dans la version actuelle de Cordova, une issue ne permet pas de lancer l'application sur le simulateur. Exécuter ces commandes dans votre projet

```
cd platforms/ios/cordova/node_modules/  
npm install ios-sim@latest
```

Run

Vous pouvez visualiser la liste des simulateurs disponibles

```
cordova run ios --list
```

- Exécutez la commande Cordova

```
cordova run ios --target=iPhone-7-Plus
```

Debug

- Lancez l'application comme précédemment
- Ouvrez Safari
- Ouvrez les préférences de Safari. Avancées ⇒ cocher Afficher le module Développement dans la barre des menus.
- Safari, Menu Développement ⇒ Simulator ⇒ nom de l'application

Vous pouvez aussi, ouvrir l'application iOS Native (./platforms/ios) dans Xcode. Cela permet de pouvoir déboguer natif en même temps que la partie web.

Chapitre 3. Ionic

3.1. Installation poste de développement



La configuration du poste de développement pour Cordova doit être réalisée préalablement

3.1.1. Installation basique

- Installez le module node global Ionic

```
npm install -g ionic
```

- Créez une application Ionic sans template

```
ionic start firstProject blank
```

```
# Répondre No à l'installation du pack Ionic Pro
```

- Allez dans le dossier de l'application

```
cd firstProject
```

- Vérifiez que Ionic est bien installé

```
(_) _ _ _ ( ) _ _ _  
| | / - \ | ' - \ | / _ _ |  
| | ( ) | | | | ( _  
|_ \ _ _ / |_ | _ | \ _ _ | CLI PRO 3.18.0
```

```
$ ionic <command> [<args>] [--help] [--verbose] [--quiet] [--no-interactive]
[--confirm] [options]
```

```

config <subcommand> ..... Manage CLI and project config values (subcommands:
get, set)
docs ..... Open the Ionic documentation website
info ..... Print system/environment info
login ..... Login with your Ionic ID
signup ..... Create an Ionic account
ssh <subcommand> ..... Commands for configuring SSH keys (subcommands:
add, delete, generate, list, setup, use)
start ..... Create a new project
telemetry ..... (deprecated) Opt in and out of telemetry

```

```

    build ..... Build web assets and prepare your app for any
platform targets
    cordova <subcommand> ..... Cordova functionality (subcommands: build, compile,
emulate, platform, plugin, prepare,
                                requirements, resources, run)
    doctor <subcommand> ..... Commands for checking the health of your Ionic
project (subcommands: check, ignore, list)
    generate ..... Generate pipes, components, pages, directives,
providers, and tabs (ionic-angular >=
                    3.0.0) (alias: g)
    git <subcommand> ..... Commands relating to git (subcommands: remote)
    integrations <subcommand> Add or disable various integrations in your app
(subcommands: disable, enable)
    link ..... Connect your local app to Ionic
    monitoring <subcommand> .. Commands relating to Ionic Pro error monitoring
(subcommands: syncmaps)
    serve ..... Start a local dev server for app dev/testing

```

- 12

ionic serve

- Regardez les options de `ionic serve`

```
ionic-pro-sample git:(master) ionic help serve
```

ionic serve - Start a local dev server for app dev/testing

Easily spin up a development server which launches in your browser. It watches for changes in your source files and automatically reloads with the updated build.

By default, ionic serve boots up a development server on all network interfaces and prints the external address(es) on which your app is being served. It also broadcasts your app to the Ionic DevApp on your network. To disable the DevApp and bind to localhost, use **--local**.

Try the **--lab** option to see multiple platforms at once.

Usage:

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

Options:

--consolelogs, -c Print app console logs to Ionic CLI
--no-open Do not open a browser window
--local Disable external network usage
--lab, -l Test your apps on multiple platform types in the browser
--platform, -t Start serve with a specific platform (android, ios)

Advanced Options:

--address Use specific address for the dev server (default: 0.0.0.0)
--port, -p Use specific port for HTTP (default: 8100)
--livereload-port, -r Use specific port for live-reload (default: 35729)
--dev-logger-port Use specific port for dev server communication (default: 53703)
--no-devapp Do not publish DevApp service
--no-proxy Do not add proxies
--browser, -w Specifies the browser to use (safari, firefox, google chrome)
--browseroption, -o Specifies a path to open to (/#/tab/dash)

Examples:

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

- Testez notamment cette option

```
ionic serve --lab -c
```

- Démarrez l'application sur votre device Android

```
ionic cordova run android --livereload
```

- Démarrez l'application sur le simulateur iOS

```
ionic cordova run ios --livereload
```

- Que vous développiez sur les devices ou dans le navigateur, dès que vous allez effectuer un changement dans votre code, l'interface sera mise à jour lorsque vous sauvegarderez vos modifications dans l'application Angular. Si vous modifiez, le code natif, par exemple avec l'ajout d'un plugin, vous devrez alors relancer les commandes précédentes afin de mettre l'application native à jour sur votre device.
- Le client Ionic permet aussi de générer vos composants comme le client Angular-cli
- Ajouter un nouveau composant

```
ionic generate component myComponent
```

- Ajouter le à l'application (dans un template HTML)
- Dans le navigateur et dans vos devices, les modifications ont été rechargées sans action utilisateur

3.2. Template

- Regardez l'aide de ionic start

```
ionic help start
```

- Listez les templates disponibles

```
ionic start --list
```

- Créez une ou plusieurs applications en utilisant les template de votre choix, n'y passez pas 1H non plus ;)

3.4. TP: Trailer application

Nous allons créer une application de gestion de films. Celle-ci permettra d'afficher une liste de trailers et une page de détail en mode anonyme. Il sera possible de s'y connecter pour un utilisateur.

Des fonctionnalités supplémentaires seront alors disponibles:

- Ajoutez un film dans sa liste de favoris
- Une page de préférence utilisateur
- logout

3.4.1. Architecture de l'application

- Créez l'application ionic-trailers sans template
- Mettez en place un menu à gauche ([doc](#))
- Mettez en place toute la navigation de l'application. ([doc](#)).

Toutes les pages auront un titre dans le header. Voici la liste des pages à mettre en place : * Page Trailers: c'est la root page. Titre de la page : Trailers * Page Trailer: Titre de la page : Trailer * Page Login: Titre de la page : Login * Page Favoris: Titre de la page: Favoris * Page Préférences: Titre de la page: Préférences

- Listez toutes les pages dans le menu, au click sur une item de la liste naviguer vers la page correspondante

3.4.2. Page Trailers

- Récupérez les trailers via un appel HTTP sur l'[API](#)
- Triez la liste par ordre alphabétique sur le titre
- Affichez dans la page Trailers la liste récupérée avec une miniature de l'image, le titre et le(s) genre(s)
- Au touch sur un trailer de la liste naviguer vers la page de détail en envoyant en paramètre les informations nécessaires à l'affichage de la page de détail.

3.4.3. Page Trailer

- Affichez les informations complètes d'un trailer
- Dans l'ordre, le titre, l'image (toute la largeur de l'écran), le(s) genre(s), les acteurs
- Une flèche de retour doit être présent dans le header pour revenir sur la page trailers

3.4.4. Page Login

- Le formulaire est composé de 3 champs de saisie, email, checkbox se souvenir de moi. Les règles de gestions sont les suivantes :

- les champs sont tous requis
- le champ email doit être un email valide
- le champ password doit avoir au moins 6 caractères
- des messages d'erreurs doivent être affichés en dessous de chaque champ si ceux-ci ne sont pas valides
- Le bouton de soumission du formulaire est actif quand tous les champs sont valides
- Au click sur le bouton de soumission, vérifiez que le mot de passe est égal à 1234. L'utilisateur est alors considéré comme authentifié. Garder le user en mémoire.
- Si l'utilisateur a coché se souvenir de moi, ne rien faire pour l'instant, nous reviendrons dessus plus tard

3.4.5. Contexte User

Maintenant que nous avons un user, nous pouvons conditionner un peu l'application.

- Dans le menu, rendre la page Préférences accessible que si le user est authentifié et la page Login accessible si il ne l'est pas
- Dans le menu si l'utilisateur click sur la page favoris et qu'il n'est pas authentifié, lui afficher une modale lui proposant de le faire avec deux choix.
 - non, fermer le menu
 - oui, naviguer vers la page de login
- Dans le menu, si le user est authentifié, ajoutez un nouvel item logout, qui purge de la mémoire notre user

3.4.6. Favoris

Toutes ces fonctionnalités ne sont accessibles que si le user est authentifié

- Dans la page de détail d'un trailer mettez un bouton avec Ionic-icons pour ajouter un trailer aux favoris du user
- Dans la page Favoris, affichez une liste contenant le titre et la miniatures de l'image des trailers favoris
- Si le user n'a pas de favoris, affichez lui un message l'incitant à en ajouter
- Bonus si vous avez le temps, mettez une icone différente sur le trailer quand il déjà fait parti de la liste de favoris. Au click, il faut alors le supprimer de cette liste ;)

3.4.7. Storage

- Sur la page de login, si le user à cocher se souvenir de moi, au click sur la validation du formuaire, persistez en base son email.
- A l'initialisation de cette page, regardez en base si le mail est présent, si c'est le cas, pré-remplissez le champ email et cocher la checkbox
- Persistez les favoris du user à chaque ajout

3.4.8. Offline

Nous souhaitons que l'application puisse fonctionner offline.

- Ajoutez le plugin Ionic Native [Network](#)
- Lors de la récupération des trailers via l'appel réseau, persistez en base les trailers
- A l'initialisation de l'application, si il n'y a pas de réseaux récupérez les trailers depuis la base pour l'affichage de la page
- Faites apparaitre un Toast à chaque changement du réseau (pas de connection ou connection oK)

3.4.9. Secure Storage

Nous souhaitons sécuriser les informations sensibles de notre user. Son mail est une information sensible ;)

- Ajoutez le plugin Ionic Native [Secure Storage](#)
- Utilisez ce plugin pour persister le mail du user
- Dans la page Préférence, ajoutez un toggle qui sera actif si le email est persisté. Lorsqu'il passe inactif, supprimer les données persistées.

3.4.10. Profil User

- Dans la page préférence de l'utilisateur, ajoutez un bouton afin de prendre une photo de profil
- Si une photo de profil est définie pour le user, affichez la dans le header de la page préférence