horizontal line

Welcome to **Code Push**

This plugin allows you to easily add a dynamic update experience to your Cordova app(s).

Installation in Ionic App:

1. Install the Cordova and Ionic Native plugins:

|  |
| --- |
| **$ ionic cordova plugin add cordova-plugin-code-push  $ npm install --save @ionic-native/code-push** |

1. Add this plugin to your app's module

Installation of Code Push CLI:

|  |
| --- |
| 1. **sudo npm install -g code-push-cli** 2. **code-push login** |

# 

* code-push login command will open the browser and you will have to create an account with the code-push cloud services (you can use your Github, Google, Hotmail one). After successful sign-up, you must copy the token provided by the browser in the terminal.

Add Project in Code-Push Account:

|  |
| --- |
| **code-push app add project\_name-ios ios cordova**  **code-push app add project\_name-android android cordova** |

* It's better to add two different apps in your code-push account for each platform so you can handle them separately.
* After these commands you will get deployment channels along with deployment keys. Initially “Production” and “Staging” deployment channels are created.

**To see again the deployment keys:**

|  |
| --- |
| **code-push deployment ls project\_name-android -k**  **code-push deployment ls project\_name-ios -k** |

**Now set up these keys in your config.xml:**

<platform name="android">  
 <preference name="CodePushDeploymentKey" value="YOUR-ANDROID-DEPLOYMENT-KEY" />  
</platform>

<platform name="ios">  
 <preference name="CodePushDeploymentKey" value="YOUR-IOS-DEPLOYMENT-KEY" />  
</platform>

Release the updates:

Each time you modify anything in your code, you release an update like this:

**For iOS:**

|  |
| --- |
| 1. **ionic cordova prepare ios** 2. **code-push release codepush-ionic-test-ios ./platforms/ios/www/ 0.0.1 --description "Your awesome change description" -d "Staging"** |

**For Android:**

|  |
| --- |
| 1. **ionic cordova prepare android** 2. **code-push release codepush-ionic-test-android ./platforms/android/assets/www/ 0.0.1 --description "Your awesome change description" -d "Staging"** |

(the -d flag can be "Production" or "Staging", so you can manage and test your releases against staging APPs)

**Note:**

The update will only be released for those devices matching the 0.0.1 version of your app.

Create New Deployment Channel:

* Initially “Production” and “Staging” deployment channels are created.
* To view list of existing deployment channels use:

|  |
| --- |
| **code-push deployment ls <appName> -k**  (-k :Specifies whether to display the deployment keys) |

* Create new deployment channel use:

|  |
| --- |
| **code-push deployment add <appName> [deploymentName]** |

# **Ex. code-push deployment add project-name dummychannel**

Adds deployment "**dummychannel**" to app "**project-name**"

Deployment App On New Channel:

* After creation of new deployment channel, you will get deployment key for that channel. To use new deployment channel goto config.xml of the app and replace older deployment key with new deployment key(new deployment Channel).
* Else pass deployment key to sync() as a parameter like:

|  |
| --- |
| **sync(syncOptions?: SyncOptions, downloadProgress?: SuccessCallback<DownloadProgress>): Observable<SyncStatus>;** |

Where SyncOptions can have:

|  |
| --- |
| **ignoreFailedUpdates?: boolean;**  **updateDialog?: boolean | UpdateDialogOptions;**  **deploymentKey?: string;** |

* Actually, the app first try to search deployment key in sync() call, if it's not available it tries to find it in config.xml
* To deploy app on new channel use:

|  |
| --- |
| **code-push release-cordova project\_name android -d "dummy" (android)**  **code-push release-cordova project\_name ios -d "dummy"(ios)** |

**Tips:**

The simplest (and recommended) way to do this is to use the release-cordova comand in the CodePush CLI, which will handle preparing and releasing your update to the CodePush server.

|  |
| --- |
| **code-push release-cordova <appName> <platform>** |

# **Note:**

Make sure that project\_name in release command is valid ,otherwise above command will give error like:

[Error] App "project\_name" does not exist.

***Solution:***

To solve this error , check app name(project\_name) by entering following command:

|  |
| --- |
| **code-push app ls** |

This will you list of projects added into your code-push account. Find name of desired project and insert in command accordingly.

There are multiple flags in release command and those are as follows:

|  |
| --- |
| **code-push patch <appName> <deploymentName> [--label <releaseLabel>] [--mandatory <isMandatory>] [--description <description>] [--rollout <rolloutPercentage>] [--disabled <isDisabled>] [--targetBinaryVersion <targetBinaryVersion>]** |

To get detailed information about each of these flags refer link given below:

<https://microsoft.github.io/code-push/docs/cli.html#rollout-parameter>

**Tips:**

Run ionic serve on another terminal on a background. This will help us to get the latest build of the app.

Functions:

* getCurrentPackage()

### getPendingPackage()

### checkForUpdate(deploymentKey)

### notifyApplicationReady()

### restartApplication()

### sync(syncOptions, downloadProgress)

To get detailed informations for each of these functions refer link given below:

<https://ionicframework.com/docs/native/code-push/#installation>

Generate build of the app:

To build app’s executable file these are ionic commands:

|  |
| --- |
| **ionic cordova build ios(.ios file for ios)**  **ionic cordova build android(.apk file for android)** |

Benefits of Code-Push over Ionic Pro:

1. Code-push provides total applications deployment from e2e through command line.
2. Adjustable packages and pricing according to requirements from Microsoft App Center.