Heroku Parse Install

Instructions For Installing Parse To Heroku

Setting Your App Up On Heroku

- Create a Heroku account. You will need a credit card even though this is a free tier account.
- Once you have created an account and can log in then go to this link: here
- Scroll down until you see a purpose button that says "Deploy To Heroku". CLICK that button!
- Leave everything alone except the **appname**, **appid** and **masterkey**. Fill these in and copy paste somewhere convenient. (Tip: Use your bundle identifier for the appid)
- Add your appname to the url
- e.g. http://mycoolapp.herokuapp.com/parse
  - Here my app's name is `mycoolapp` so this gets added to the url before .herokuapp.com
- Click deploy! Wait until the app shows that it was

successfully deployed.

Installing Parse SDK & Testing

- You can install the Parse SDK using either Cocoapods, Carthage, or manually. I will just show the Cocoapods install here. Be sure you update the pod spec file using: `pod install --repo-update`

1. create a blank single view application in xCode.

2. cd into the root directory of the project from terminal.

3. type `pod init`

4. open the Podfile. Uncomment `use_frameworks!` for Swift, and uncomment `platform :ios, '10.0'`

5)paste `pod 'Parse'` between `do` and `end` inside the main target statement

6. save the Podfile and type `pod install` in terminal and let it run.

7. close your project and open it from the `.xcworkspace` file instead of the normal `.xcodeproj` file.

8. for Swift you need a bridging header. To create one select the main target's group and right click and select `New file...` . Select `Cocoatouch` and just make it an `Objective-C` file with any super class and name it whatever you want. When you click create you will be prompted to create a `bridging header` . Accept the

prompt. This will create a .h file with the name `YourProjectName-Bridging-Header.h` . Delete the *other* .h and .m files now.

9. add your import into the bridging header `#import <Parse/Parse.h>`

10. follow the quickstart instructions for creating an object [here]()

```
let configuration = ParseClientConfiguration {
      $0.applicationId = "afdafsdflj888888"
      $0.clientKey = "myMasterKey"
      $0.server = "http://myapp8.herokuapp.com/parse"
    }
    Parse.initialize(with: configuration)
    let testObject = PFObject(className: "TestObject")
    testObject["foo"] = "bar"
    testObject.saveInBackground { (success: Bool, error: Error?) in
      if let error = error {
        print(#line, error)
      }
      guard success = true else {
        print(#line, "object not saved, WTF!")
        return
      }

      print("Object has been saved!")
    }
```

11. Add the key `App Transport Security Settings` to the info.plist file (unless you accepted the default url). It should be a dictionary.

12. Add the value `Allow Arbitrary Loads` , which is a Boolean, and make it `YES`

13. Run the app.

14. Check the mongo db by clicking on mLab MongoDB :: Mongodb link in the heroku dashboard

15. The link will probably fail. If it does, just delete the last part of the path starting with `heroku\_...` and you should see your data

16. You can install the Parse dashboard locally/remotely to help you manage your model inside a browser. I will quickly take you through installing this locally.

Installing Parse Dashboard Locally

15. The instructions for Parse dashboard installation are here.

16. First make sure you have Node.js version >= 4.3 installed. Go to terminal and type `Node -v` to get the version if any. If it doesn't meet the requirement then install the latest version by going here. Download and install it.

17. With node installed, run `npm install -g parse-dashboard` .

18. To view your app you have to enter the following into terminal, and fill out the places for `yourAppId`, `yourMasterKey`, `https://example.com/parse` with your url and `optionalName` in the following script `parse-dashboard --appId yourAppId --masterKey yourMasterKey --serverURL "https://example.com/parse" --appName optionalName`. Paste the command into an editor and change those values then copy/paste into terminal and run.

19. Now open your browser to `http://localhost:4040`