

# STARTING SMALL WITH BIG DATA

AN INTRODUCTION TO THE PARSE API, BY VICTOR DADFAR AND EMILY BURNETTE

# WARNING

- The presentation can get a bit technical, so if that makes you uncomfortable, bring a member with programming experience.

# ORIGINS OF STORING DATA ON THE WEB

- Before Parse came around, storing data on one's website was not a simple task
- Your first instinct may be to say: “Why not just store data in javascript variables?”
  - Unfortunately, all temporary data (including javascript variables) is lost when refreshing or reloading the webpage 😞

# WORKAROUNDS TO STORING DATA

- So programmers had to become creative with their solutions. Some notable ones included:

# WORKAROUNDS TO STORING DATA

- Storing javascript variable data in the URL of the webpage i.e. [www.google.com?search=cats](http://www.google.com?search=cats)
  - Works alright for one or two variables, what if you had a dozen to store? What if the data you wanted to store was paragraphs long? And anyone could modify the url thereby changing the data, so security measures were nonexistent.
- Emailing variable data to the programmer' email
  - Great, but now you need to write a script or have the proper backend to actually parse that email, not to mention your inbox will be cluttered with meaningless mail
- Building, Configuring and maintaining your own server
  - Too much work

# ENTER PARSE

- Parse server is a self-hosted that allows you to store data and interact with other services on the web
- It works on a wide variety of devices and platforms:
  - Javascript
  - PHP
  - iOS, OS X, tvOS
  - Android
  - Arduino

# PARSE + JAVASCRIPT

- Today we will be focusing on setting up and accessing your very own Parse Server with Javascript and a simple web page
  - Though once you have your server operational you should be able to access it with any language or platform

# GENERAL WORKFLOW – DATA ACCESS

Website  
uploads data  
to Parse

Parse Server  
determines  
how to store it

Data gets sent  
to remote  
database for  
storage

# GENERAL WORKFLOW – DATA RETRIEVAL

Website sends  
data request to  
Parse

Parse Server  
determines if  
data is  
available

Data is retrieved  
from the remote  
database

Parse ensures its  
the correct data  
and packages it  
up nicely

Data gets sent  
back to the  
website

# HOW IT WILL WORK

- Obtain Parse Server code from Github
- Deploy server code to hosting platform like Heroku
  - This allows us to access the Parse Server remotely, rather than on one local machine
- Create a new database space with a DB platform like mLab (can be done through Heroku)
- Link the Parse Server with the database space we just created
- Profit???

# PREREQUISITES

- Heroku
  - To create an account: <https://dashboard.heroku.com>

# STEP 1: DEPLOY TO HEROKU

- <https://github.com/ParsePlatform/parse-server-example>

## Getting Started With Heroku + mLab Development

-With the Heroku Button

 Deploy to Heroku

## STEP 2 – DEFINE APP VARIABLES

- App Name – optional, but highly recommended
- APP\_ID - must choose a unique identifier (make it the same as your App Name)
- MASTER\_KEY – password for your app if you want to add security in the future
- SERVER\_URL – Change the “yourappname” to the app name you made earlier

## STEP 3 – CREDIT CARD INFORMATION

- Heroku will prompt you to input credit/debit card information before you begin. Don't worry, as you're signed up to use the “sandbox” tier which is completely free. If you don't have a card yourself, see if anyone else on the team does.

## STEP 4 – YOUR APP IS DEPLOYED! NOW TEST IT

- <http://medhacks.org/parseworkshop.html>
- Open the file in a text editor and change both “app\_name”s to the App Name you defined earlier
- Double click on the file to run it,
  - If it worked, you should see the text “Saved heartbeat!”
  - If not, it should say it couldn’t save properly

## FURTHER READINGS

- <http://parseplatform.github.io/docs/js/guide/>
  - Incredibly helpful guide for using Parse with Javascript
- <https://parseplatform.github.io>
  - Guides for all Parse-supported platforms