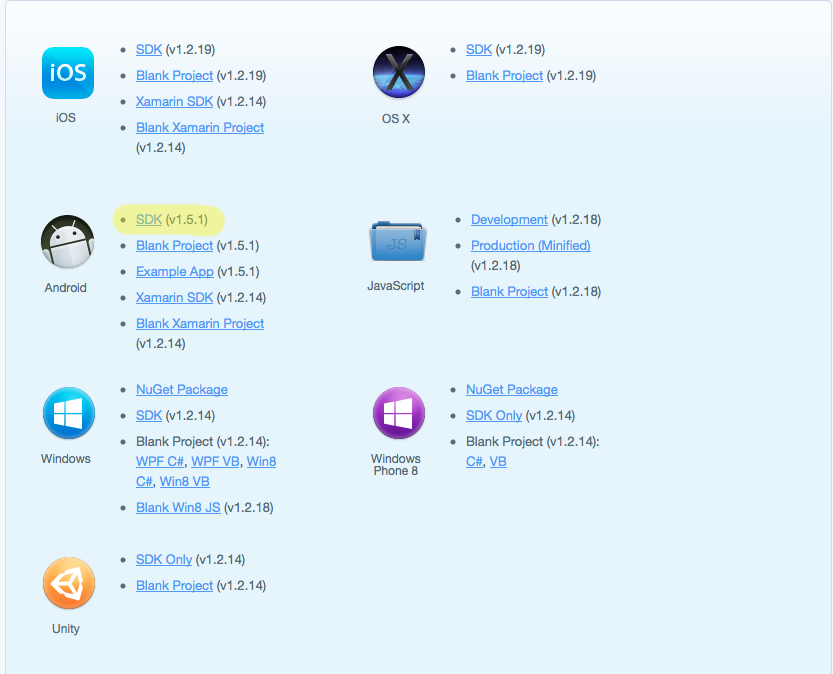
Group 2:

How to use Parse.com in Java

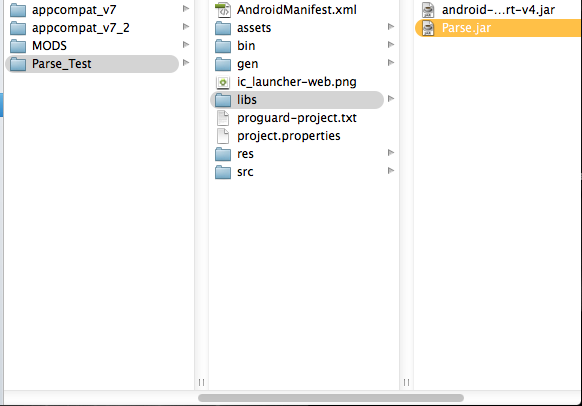
This guide teaches you how to store and retrieve data onto parse.com. This does require your phone to be connected to the internet. I am using a mac, so my picture might be a little different than most, but the basic instructions are the same.

Getting Parse to work:

Step 1: Download the Parse jar file. This can be done here: <https://parse.com/docs/downloads> . Download the Android SDK version.

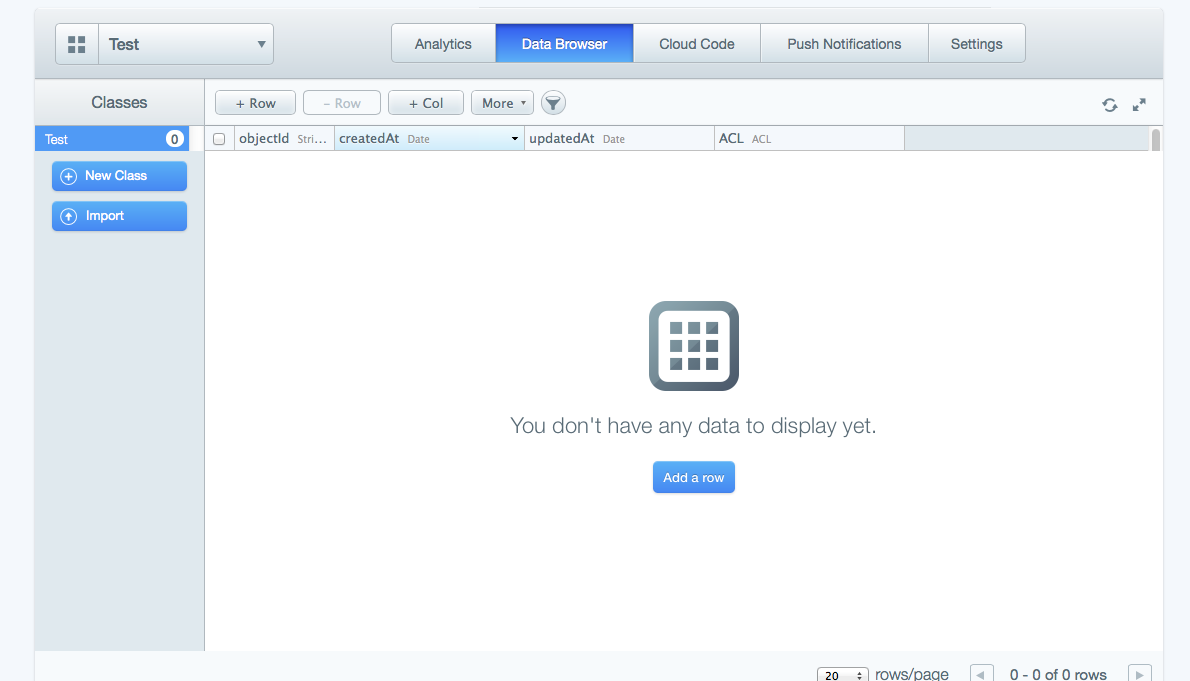


Step 2: Open up the folder you just downlaoded ( unzip it if you need to) and take only the file that ends in .jar. Move that file intot he libs folder of your Project.

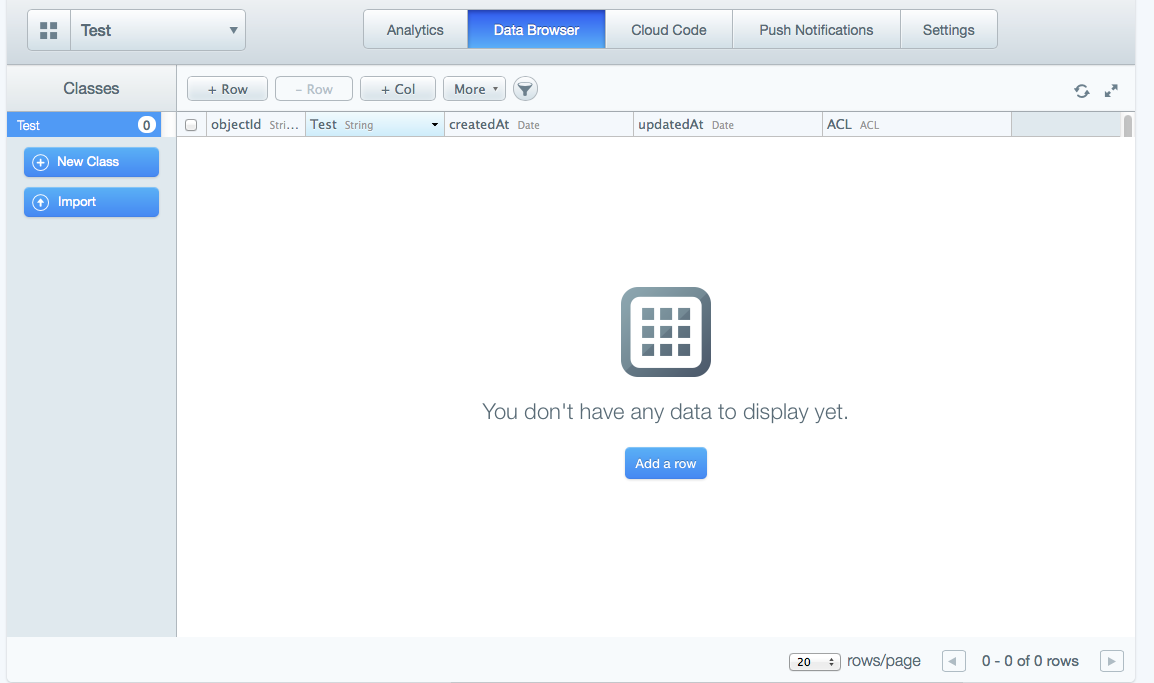


Part 2: Setting up your data on Parse.

Step 1: Assuming you already have an account on Parse.com create a new Project. Once that is done (you can exit out of the getting started menu. We will go back to that soon) Go to the menu on top of the page and select Data Browser, and create a new Parse Class. You can call it whatever you want. At the end it should look like this.



Step 2: First you have to know what data you are storing on this website. Once you have done that, add a column for each type of data you are storing. For this example I’m going to store a string called test.



Step 3: Finally go to settings, and then application keys. We are going to need these soon.

Part 3: The code

Step 1: First in each class that you are using parse code you will need 3 imports.

**import** com.parse.Parse;

**import** com.parse.ParseAnalytics;

**import** com.parse.ParseObject;

Step 2: In the class you want upload data to parse with, you have to initialize parse with your application and client keys. This can be done like so:

Parse.*initialize*(**this**, "Application\_key\_goes\_here", "Client\_key\_goes\_here");

Step 3: Now in order to upload data to parse, you first must create a parse object. It has one parameter, the name of the data class you created on the website. Once that is done, you just use the put() method. The put method takes two perimeters. A string that is the name of the column that you created on parse, and the variable you are uploading. You can use the put method as many times as you want, with as many columns that you want. Finally when you are done use the saveInBackground() method. When you run the program, the data should be saved onto parse.

ParseObject data = **new** ParseObject("test");

data.put("test", var);

data.saveInBackground();

Step 4: This step is used in order to retrieve the data. The next part is more confusing, so I’m going to post the code first and then explain parts of it.

//(a)

Parse.*initialize*(**this**, " Application\_key\_goes\_here ", " Client\_key\_goes\_here ");

//(b)

ParseQuery<ParseObject> query = ParseQuery.*getQuery*("test");

//(c)

query.whereEqualTo("culomn\_name", value\_you\_want);

//(d)

query.findInBackground(**new** FindCallback<ParseObject>() {

@Override

//(e)

**public** **void** done(List<ParseObject> list, ParseException e) { //(f)}

A: initialize parse like we did when we wanted to upload to parse

B: This creates a list made up from your data. The parameters for the getQuery method are the name of your data class that you are getting the data from.

C: this step is optional, but is very useful in some cases. What this does is that it only retrieves the data that has one variable equal to a certain value. For example if you are storing data of age groups and exhibits they liked, and you want to retrieve all the data of only entries that are age 6, you would use this method to do that.

D: just copy this line like it is.

E: This creates a method within the onCreate() method called done. This method will be called automatically.

F: within this method you have to do anything that you wanted to do with the data. This is the only way I figured out on how to use the data, as you cannot use the list outside the method.