Parse

The following is an overview of how to use the Parse API for the “On the Map”.

The Parse API

Parse, not unlike other API’s, treats data as objects. These objects are very similar to objects from object-oriented programming languages. For “On the Map”, you will be working with StudentLocation objects that we and other students have created in Parse.

Using the Parse API for “On the Map”

Before you can use Parse, you will need to know our Parse Application ID and REST API Key for “On the Map”:

- Parse Application ID = QrX47CA9cyuGewLdsL7o5Eb8iug6Em8ye0dnAbIr
- REST API Key = QuWThTdiRmTux3YaDseUSEpUKo7aBYM737yKd4gY

You will be required to specify the Parse Application ID and REST API Key for all requests to Parse so that you can access StudentLocation data for “On the Map”. Since everyone will be using the same Parse Application ID and REST API Key, the StudentLocation data could be overwritten and changed at anytime (although not likely). In normal cases this would be very bad, but since this is a learning application don’t sweat it!

StudentLocation Objects

In Parse, StudentLocation objects are represented as follows:

Key	Description	Parse Type	Example Value
objectId	an auto-generated id/key generated by Parse which uniquely identifies a StudentLocation	String	8ZExGR5uX8
uniqueKey	an extra (optional) key used to uniquely identify a	String	1234

	StudentLocation; **you should populate this value using your Udacity account (user) id**		
firstName	the first name of the student which matches their Udacity profile first name	String	John
lastName	the last name of the student which matches their Udacity profile last name	String	Doe
mapString	the location string used for geocoding the student location	String	Mountain View, CA
mediaURL	the URL provided by the student	String	https://udacity.com
latitude	the latitude of the student location (ranges from -90 to 90)	Number*	37.386052
longitude	the longitude of the student location (ranges from -180 to 180)	Number*	-122.083851
createdAt	the date when the student location was created	Date	Feb 25, 2015, 01:10
updatedAt	the date when the student location was last updated	Date	Mar 09, 2015, 23:34
ACL	the Parse access and control list (ACL), i.e. permissions, for this StudentLocation entry	ACL	Public Read and Write

* These may be casted into floating-point values in Swift

When you are parsing StudentLocation(s), the Parse Types can be casted into equivalent Swift types. **YOU DO NOT HAVE TO WORRY ABOUT PARSING DATE OR ACL TYPES.**

GETting StudentLocations

Method: https://api.parse.com/1/classes/StudentLocation
Method Type: GET
Optional Parameters:

- limit - (Number) specifies the maximum number of StudentLocation objects to return in the JSON response
  - ex: https://api.parse.com/1/classes/StudentLocation?limit=100
- skip - (Number) use this parameter with limit to paginate through results
  - ex: https://api.parse.com/1/classes/StudentLocation?limit=200&skip=400
- order - (String) a comma-separate list of key names that specify the sorted order of the results

○ Prefixing a key name with a negative sign reverses the order (default order is descending)
○ ex: https://api.parse.com/1/classes/StudentLocation?order=-updatedAt

Example Request:

```
let request = NSMutableURLRequest(URL: NSURL(string:
"https://api.parse.com/1/classes/StudentLocation")!)
request.addValue("QrX47CA9cyuGewLdsL7o5Eb8iug6Em8ye0dnAbIr",
forHTTPHeaderField: "X-Parse-Application-Id")
request.addValue("QuWThTdiRmTux3YaDseUSEpUKo7aBYM737yKd4gY",
forHTTPHeaderField: "X-Parse-REST-API-Key")

let session = NSURLSession.sharedSession()
let task = session.dataTaskWithRequest(request) { data, response, error
in
   if error != nil { // Handle error...
       return
   }
   println(NSString(data: data, encoding: NSUTF8StringEncoding))
}
task.resume()
```

Example JSON Response: get-student-locations.json

POSTing a StudentLocation

Method: https://api.parse.com/1/classes/StudentLocation
Method Type: POST
Example Request:

```
let request = NSMutableURLRequest(URL: NSURL(string:
"https://api.parse.com/1/classes/StudentLocation")!)
request.HTTPMethod = "POST"
request.addValue("QrX47CA9cyuGewLdsL7o5Eb8iug6Em8ye0dnAbIr",
forHTTPHeaderField: "X-Parse-Application-Id")
request.addValue("QuWThTdiRmTux3YaDseUSEpUKo7aBYM737yKd4gY",
forHTTPHeaderField: "X-Parse-REST-API-Key")
request.addValue("application/json", forHTTPHeaderField: "Content-
Type")
request.HTTPBody = "{\"uniqueKey\": \"1234\", \"firstName\": \"John\",
\"lastName\": \"Doe\",\"mapString\": \"Mountain View, CA\",
\"mediaURL\": \"https://udacity.com\",\"latitude\": 37.386052,
\"longitude\": -122.083851}".dataUsingEncoding(NSUTF8StringEncoding)

let session = NSURLSession.sharedSession()
let task = session.dataTaskWithRequest(request) { data, response, error
in
      if error != nil { // Handle error…
            return
      }
      println(NSString(data: data, encoding: NSUTF8StringEncoding))
}

task.resume()
```

Example JSON Response: post-student-location.json

Exceeding Expectations

The following sections are not required to complete “On the Map”. However, if you would like to exceed expectations, then you will find these useful.

Querying for a StudentLocation

Method: https://api.parse.com/1/classes/StudentLocation
Method Type: GET
Required Parameters:

- where - (Parse Query) a SQL-like query allowing you to check if an object value matches some target value
    - ex: https://api.parse.com/1/classes/StudentLocation?where=%7B%22uniqueKey%22%3A%221234%22%7D
    - the above URL is the escaped form of… https://api.parse.com/1/classes/StudentLocation?where={"uniqueKey":"1234"}
    - you can read more about these types of queries in Parse’s REST API documentation

Example Request:

```
let urlString = "https://api.parse.com/1/classes/StudentLocation?
where=%7B%22uniqueKey%22%3A%221234%22%7D"
let url = NSURL(string: urlString)

let request = NSMutableURLRequest(URL: url!)
request.addValue("QrX47CA9cyuGewLdsL7o5Eb8iug6Em8ye0dnAbIr",
forHTTPHeaderField: "X-Parse-Application-Id")
request.addValue("QuWThTdiRmTux3YaDseUSEpUKo7aBYM737yKd4gY",
forHTTPHeaderField: "X-Parse-REST-API-Key")

let session = NSURLSession.sharedSession()
let task = session.dataTaskWithRequest(request) { data, response, error
in
if error != nil { /* Handle error */ return }
      println(NSString(data: data, encoding: NSUTF8StringEncoding))
}

task.resume()
```

Example JSON Response: query-student-location.json

PUTing (Updating) a StudentLocation

Method: https://api.parse.com/1/classes/StudentLocation/<objectId>
Method Type: PUT
Required Parameters:

- objectId - (String) the object ID of the StudentLocation to update; specify the object ID right after StudentLocation in URL as seen below
    - ex: https://api.parse.com/1/classes/StudentLocation/8ZExGR5uX8

Example Request:

```
let urlString =
"https://api.parse.com/1/classes/StudentLocation/8ZExGR5uX8"
let url = NSURL(string: urlString)

let request = NSMutableURLRequest(URL: url!)
request.HTTPMethod = "PUT"
request.addValue("QrX47CA9cyuGewLdsL7o5Eb8iug6Em8ye0dnAbIr",
forHTTPHeaderField: "X-Parse-Application-Id")
request.addValue("QuWThTdiRmTux3YaDseUSEpUKo7aBYM737yKd4gY",
forHTTPHeaderField: "X-Parse-REST-API-Key")
request.addValue("application/json", forHTTPHeaderField: "Content-
Type")
request.HTTPBody = "{\"uniqueKey\": \"1234\", \"firstName\": \"John\",
\"lastName\": \"Doe\",\"mapString\": \"Cupertino, CA\", \"mediaURL\":
\"https://udacity.com\",\"latitude\": 37.322998, \"longitude\":
-122.032182}".dataUsingEncoding(NSUTF8StringEncoding)

let session = NSURLSession.sharedSession()
let task = session.dataTaskWithRequest(request) { data, response, error
in
      if error != nil { // Handle error…
            return
      }
      println(NSString(data: data, encoding: NSUTF8StringEncoding))
}

task.resume()
```

Example JSON Response: put-student-location.json