Parse: A Perfect Cloud-based Backend Services for Mobile Application Developers

Pravin Pandey1 1 Kamlesh Prajapati2, Dashrath Mane3
1 M.Tech. Student 2,3 Assistant Professor
1,2 Department of MCA
1,2,3 VESIT, Mumbai

Abstract— Parse is the most popular MBaaS to provide a backend for different platforms such as mobile and web applications. Parse Development makes creation of mobile apps easier because it provides some of the features such as CustomObjects, Pushnotifications, Users, Files, Geolocations, Social integration etc. If we want our data to be accessible outside of single device then we can use parse as backend service.

Key words: Mbass, IaaS, PaaS, BaaS, NoSQL, object

I. INTRODUCTION

“Parse’s vision is to let developers build any mobile app without dealing with servers.” Parse provides cloud-based backend services for mobile application developers. The San Francisco-based startup delivers a full stack of mobile services so that developers can focus on developing applications and leave the infrastructure to Parse. Launched in 2012, this fast-growing start-up provides server management for over 180,000 Android, iOS, and Windows mobile applications, which run on more than 200 million mobile devices.

The Parse mobile SDK provides cloud-based APIs and services for iOS, Android, and Windows® applications. Parse SDK also provides a JavaScript and REST APIs. Using the Parse API, you can cloud-enable your mobile applications very quickly and with minimal effort.

A mobile application that is integrated with the Parse API can easily store data objects and files on the Parse cloud, send and listen to push notifications, manage users, handle geo-location data, and use social media platforms such as Twitter and Facebook. For mobile applications that need to scale, the Parse SDK offers all the elasticity of a cloud platform.

II. BACKGROUND

With the increased use of mobile devices, it is common for applications to offer features such as backup storage, data sync, data sharing etc. Building stand-alone applications that only function and save their data on the device they are installed, are at times not feasible. A backend is usually needed where data can be saved and manipulated for the app to provide the service it was intended for.

Building this requires time, a different skill set and other resources (e.g. servers, push notification services, etc). Fortunately, there are several platforms that provide ready-made customizable backends that you can integrate with your apps. These are known as ‘Backend as a Service’, or BaaS in short.

A. Backend As A Service (BaaS)

1) What is BaaS?

BaaS is an approach for providing web and mobile app developers with a way to connect their applications to backend cloud storage and processing while also providing common features such as user management, push notifications, social networking integration, and other features that mobile users demand from their apps these days.

This new breed of BaaS services are provided via custom software development kits (SDK) and application programming interfaces (APIs). BaaS is a relatively recent development in cloud computing, with most BaaS startups dating from 2011 or later. The global BaaS market is estimated to grow from $216.5 million in 2012 to $7.7 billion in 2017 from a report published by MarketsandMarkets.

B. How Does BaaS Differ From IaaS and PaaS?

BaaS has evolved out of frustration around deployment of IaaS platforms like Amazon Web Services, just to fire up a single new mobile application—accompanied with the fact that the traditional PaaS offerings have not stayed on top of what is needed for mobile developers. BaaS is about abstracting away the complexities of launching and managing your own infrastructure, then bridging a stack of meaningful resources targeting exactly what developers need to build the next generation of mobile apps.

1) What Are The Benefits of BaaS?

BaaS is all about making developers lives easier. BaaS is born out of a shortage of mobile developer talent and an overwhelming demand for high quality mobile apps, not just on iOS, but across Android, Windows and Blackberry devices.

2) BaaS Delivers:

- Efficiency Gains - Reducing overhead in all aspects of mobile app development, increasing efficiency at all stages of development
- Faster Times to - Reducing the obstacles to take a mobile app from idea to production and overhead with operations once in production
- App Delivery With Fewer Resources - BaaS supports development with fewer developers and supporting data and IT resources
- Optimize for Mobile and Tablets - BaaS providers have put a lot of time and resources into optimization of data and network for mobile apps, and reduce fragmentation problems across multiple platforms and devices.
- Secure and Scalable Infrastructure - BaaS provides a bundled infrastructure that deals with scalability, security, performance and other operational headaches, leaving developers to do what they do best
- Stack of Common API resources - BaaS brings common and essential 3rd party API resources into a single stack, preventing developers from having to go gather them separately.
III. MOBILE BACKEND AS A SERVICE(MBaaS)

A. What is MBaaS?
Mobile Backend As A Service(MBaaS) is also known as Backend As A service. MBaaS is basically cloud computing category and which should be easy for developer setup backend operations. Some essential backend operations are cloud based data storage, user management, push notification and file management. All these backend operations consider as service and provides full abstract layer on all these features. MBaaS is not a new idea. It is gradually grow up from Infrastructure as a Service (IaaS), Platform as a Service (PaaS). Some other backend service like analytics, locations, messaging and crash reporting also consider as a service in MBaaS.

B. Why MBaaS?
Web and mobile app require some similar set of operations on the backend like push notification, data management and user management. When a developer begins to develop an app should need to redevelop each of these operation one by one. Developer can avoid all these repeat operation by MBaaS. Then also app development team need to
- Reduce the development cost.
- Reduce the development time.
- Reduce maintenance cost.
- Reduce Time To Market(TTM).
- Improve quality.
- Handle security.
- Handle app growth.

Acquired by Facebook back in 2013, Parse offers three main products. Parse Core is the basic MBaaS offering, letting you store data securely, connect with social networks and the like. Parse Push is, as the name suggests, a push notification platform, while Parse Analytics lets you measure acquisition, retention, engagement, push notification campaigns and everything else you’d expect out of an analytics platform. In terms of pricing, Parse Core is offered for free to try out at 30 requests per second and you can tailor up from here. When you get to 80 req/s that price jumps to $500 and 210 req/s hits $1800 per month (you’re about halfway up the scale at this point).

IV. PARSE - CLOUD CODE FOR MOBILE APPLICATIONS
Parse's vision is to let developers build any mobile app without dealing with servers. For complex apps, sometimes you just need a bit of logic that isn't running on a mobile device. Cloud Code makes this possible.

Cloud Code is easy to use because it's built on the same JavaScript SDK that powers thousands of apps. The only difference is that this code runs in the Parse Cloud rather than running on a mobile device. When you update your Cloud Code, it becomes available to all mobile environments instantly. You don't have to wait for a new release of your application. This lets you change app behavior on the fly and add new features faster.

Even if you're only familiar with mobile development, we hope you'll find Cloud Code straightforward and easy to use.

A. Setting Up Cloud Code
Before setting up cloud code you need to install Parse’s command line tool for the computer you use for development. And once you have installed parse The next step is to create a directory to store the code that you will run in the cloud. The command parses new sets up this directory, and will prompt you to pick which app you are creating Cloud Code for:

$ parse new MyCloudCode
Email: ninja@gmail.com
Password:
1:MyApp
Select an App: 1
$ cd MyCloudCode

Use the email address and password for your Parse account to log in. This will create a directory called MyCloudCode in the current directory. Several files are automatically created for you:

```
- config/
  - global.json
- cloud/
  - main.js
- public/
  - index.html
```

The config directory contains a JSON configuration file that you shouldn’t normally need to deal with, the cloud directory stores your Cloud Code, and the public directory stores any static content that you want to host on Parse. In the cloud directory, you'll typically just be editing main.js, which stores all of your Cloud Code functions. For now, just check that these files were created successfully. If you're using source control, you can check all of these files in.

B. Simple Function
Let's see how to run the simplest possible function in the cloud. If you take a look at cloud/main.js, you'll see an example function that just returns a string:

```
Parse.Cloud.define("hello", function(request, response) {
  response.success("Hello world!");
});
```

To deploy the code from your machine to the Parse Cloud, run:

$ parse deploy
To run this function once it's deployed, run:

```
Parse.Cloud.callFunctionInBackground("hello", new
HashMap<String, Object>(), new
FunctionCallback<String>() {
  void done(String result, ParseException e) {
    if (e == null) {
      // result is "Hello world!"
    }
  }
});
```

You should see this response:

```
"result": "Hello world!"
```

Congratulations! You have successfully deployed and run Cloud Code.
V. CONCLUSION

We highly recommend Parse because it makes our app development a breeze. The primary advantages of Parse are that you'll be able to work with application-level concepts like user accounts and push notifications rather than technology-focused concepts like databases and socket I/O. Parse helps us to share native models across server-side code and multiple clients. It's mainly for front-end mobile (iOS, Android etc) developers who have little backend skills or resources but need some simple backend for their mobile apps.

Basically our project was working fine while we were building the system. We had quite experienced developers working on the platform, the code was well organized, and performances were taken into consideration since we already produced lots of apps which expect heavy load. But once we started to do stress testing before delivery, lots of problems occurred, and they were really hard to be found in the docs.

And the problems which we came across while developing app using Parse are:
- Limited API requests per minute for entire app.
- Limited number of COUNT operations.
- Log system remembers only 100 last logs.
- Push notifications are done well, but sometimes there is a delay, up to an hour which can be serious problem.

Parse would be a great platform if all these limits are removed and their performance is improved.

REFERENCES