

Cristian Muñoz Álvarez

NOTIFICACIONES PUSH CON PARSE

¿Qué son las notificaciones
Push?

¿Qué son las notificaciones Push?

La tecnología push es un tipo de comunicación en la que es el servidor el que inicia la petición al cliente (el móvil, por ejemplo) cuando tiene una información o notificación nueva, permitiendo un importante ahorro de recursos y tiempo respecto a la tecnología convencional pull, en la que es el cliente el que envía una petición al servidor.

Ventajas

La dos ventajas fundamentales de la mensajería push con respecto a las basadas en tecnología pull son:

- La notificación es instantánea, recibimos los mensajes en el acto según llegan al servidor.
- Las notificaciones push no requieren que la aplicación esté ejecutándose en el dispositivo y realizando llamadas constantes al servidor, sino que permite que ésta esté en segundo plano y se active al recibir un mensaje. Esto implica un importante ahorro de batería y recursos del teléfono

Parse

¿Qué es Parse?

- Parse es un servicio web que permite enviar notificaciones Push (entre otro montón de servicios y utilidades pero, ese es el que nos importa para este tutorial) no solo a dispositivos Android sino también IOS, Windows Phone, etc.
- Provee una interfaz fácil de administrar y llena de información útil, como los dispositivos que tienes suscritos, los mensajes que se enviaran o han enviado, programación de mensajes, etc.
- Parse tiene un plan gratuito de hasta un millón de notificaciones al mes y otro de pago por si tus necesidades exceden esa cuota.

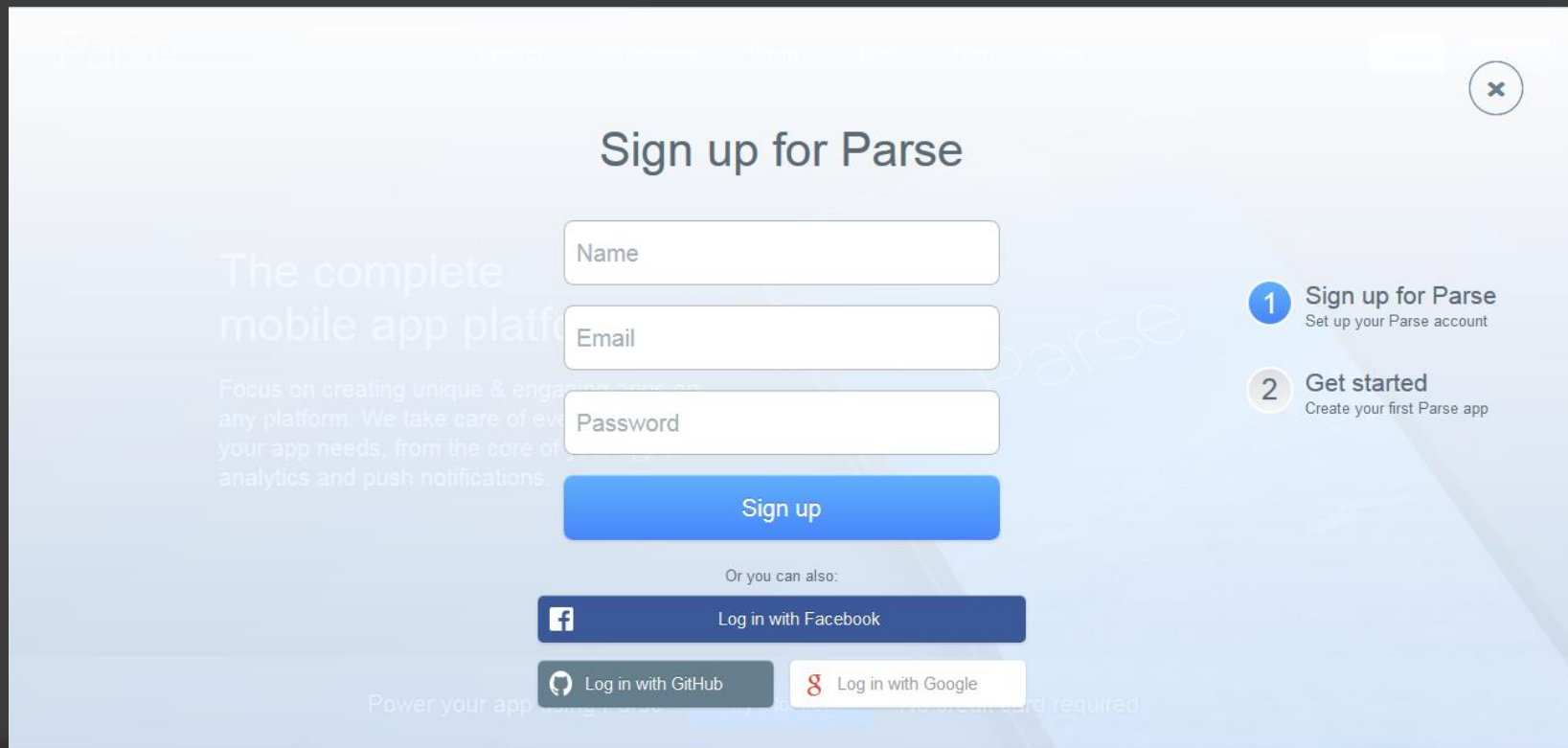
¿Por qué Parse?

¿Porque usar esto y no el Google Cloud Messaging que es la plataforma oficial?
Por una simple razón: es muchísimo más fácil de configurar y ponerlo a funcionar es cuestión de minutos.

Registrarse en Parse

Lo primero de todo para empezar a usar este servicio es registrarse en parse:

<https://parse.com/>



The screenshot shows the Parse website's sign-up page. At the top, the Parse logo is on the left, and navigation links for 'create', 'learn more', 'pricing', 'faq', and 'contact us' are on the right. A close button (X) is in the top right corner. The main heading is 'Sign up for Parse'. Below it, there are three input fields: 'Name', 'Email', and 'Password'. A blue 'Sign up' button is positioned below the password field. To the left of the input fields, there is a paragraph of text: 'The complete mobile app platform. Focus on creating unique & engaging apps on any platform. We take care of everything your app needs, from the core of your app to the analytics and push notifications.' To the right of the input fields, there is a numbered list: '1 Sign up for Parse Set up your Parse account' and '2 Get started Create your first Parse app'. Below the 'Sign up' button, there is a link 'Or you can also:'. Below this link, there are three buttons: 'Log in with Facebook' (with a Facebook icon), 'Log in with GitHub' (with a GitHub icon), and 'Log in with Google' (with a Google icon). At the bottom, there is a footer with the text 'Power your app with Parse' and a link to 'Sign up for Parse'.

Parse

create learn more pricing faq contact us

Sign up for Parse

The complete mobile app platform. Focus on creating unique & engaging apps on any platform. We take care of everything your app needs, from the core of your app to the analytics and push notifications.

Name

Email

Password

Sign up

Or you can also:

Log in with Facebook

Log in with GitHub

Log in with Google

1 Sign up for Parse
Set up your Parse account

2 Get started
Create your first Parse app

Power your app with Parse

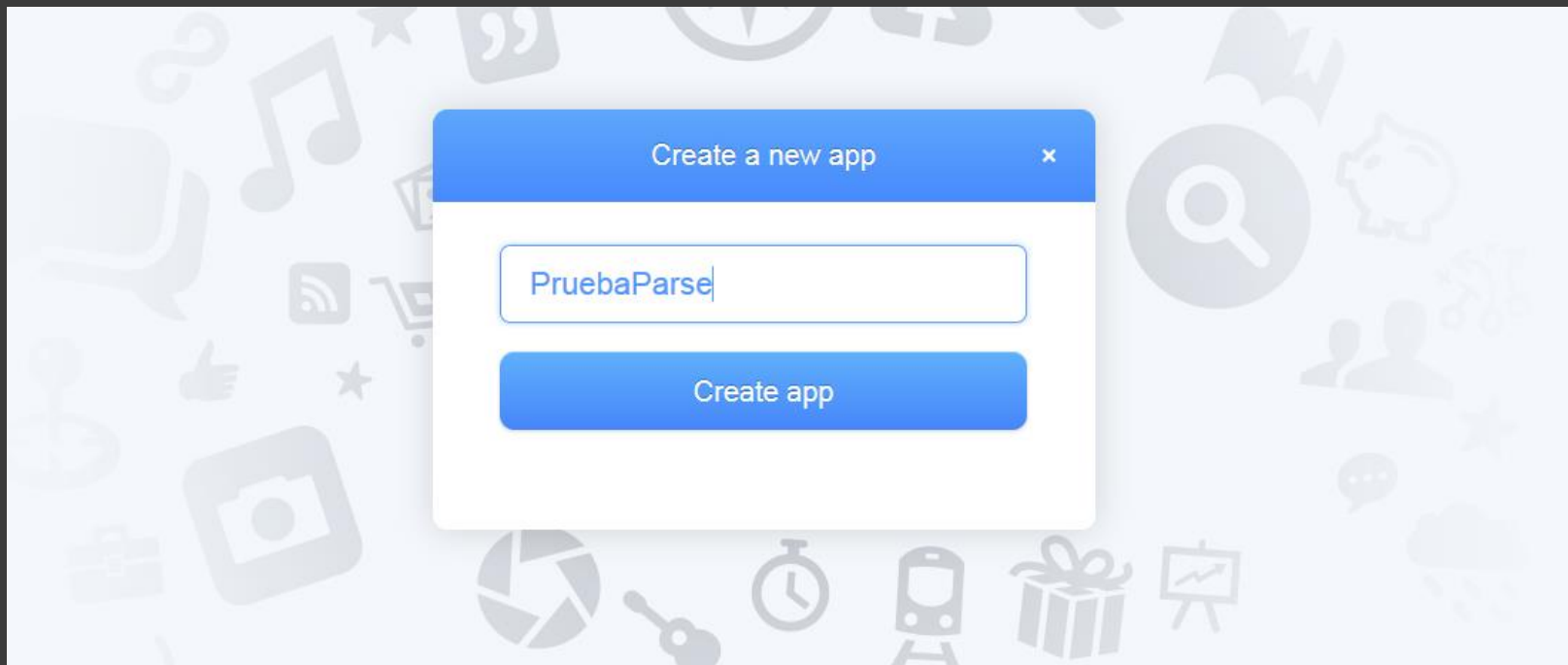
Sign up for Parse

Crear una Aplicación

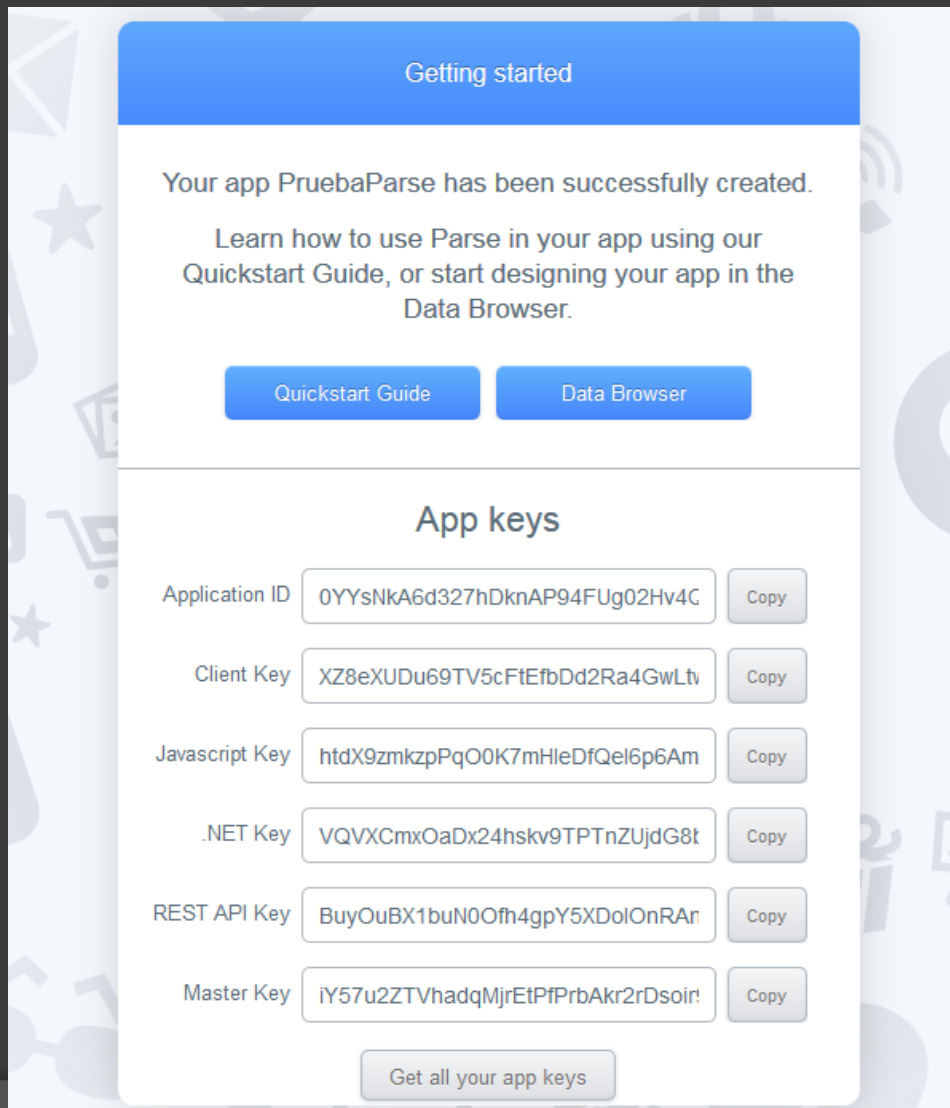
Para interactuar con la plataforma y poder enviar notificaciones, es necesario crear una aplicación dentro de Parse, una app, eso lo haces dando clic en la pestaña *Create a New App*.

Nombre de la Aplicación

Aparecerá un cuadro de dialogo en el indicaremos el nombre de la aplicación.



Claves de la Aplicación



Estas claves son las que usaremos para conectar nuestra aplicación Android con la aplicación que hemos creado en el servidor Parse.

Cliente Android

Crear un nuevo proyecto Android

Dejamos apartado Parse por el momento y creamos un nuevo proyecto de Android en Eclipse.

Librerías Parse

Necesitamos añadir al proyecto las librerías que proporciona Parse. Se pueden descargar desde la siguiente dirección:

<https://parse.com/downloads/android/Parse/latest>

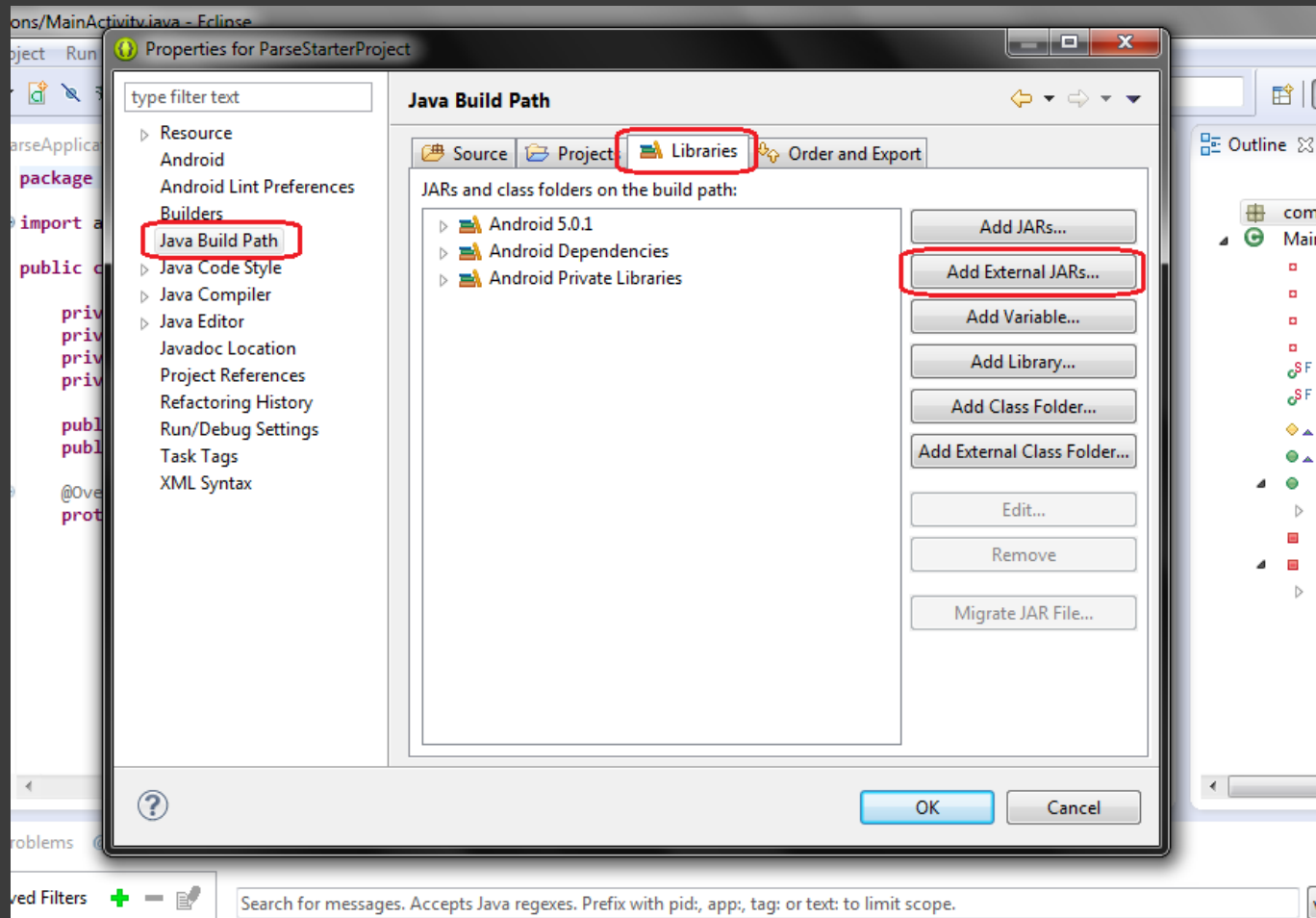
Añadir librerías

Para añadir las librerías tenemos que abrir las propiedades de nuestro proyecto (Botón derecho → Properties).

Seleccionamos *Java Build Path* en la lista de la izquierda y nos situamos en la pestaña *Libraries*.

Hacemos click en *Add External JARs...* y seleccionamos las librerías de Parse que habíamos descargado.

Agregar librerías



Clase Application

Para poder conectar nuestra aplicación con Parse, tenemos que definir una clase que herede de Application e implementar el método onCreate desde el que llamaremos *Parse.initialize* .

En los parámetros de este método introduciremos el *Application ID* y el *Client Key* de nuestra aplicación de Parse.

Clase Application

```
package net.androidsensei.pushsensei;

import android.app.Application;
import com.parse.Parse;

public class PushSenseiApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        Parse.initialize(this, "[AQUI VA TU Application ID]",
            "[AQUI VA TU Client Key]");
        ParseInstallation.getCurrentInstallation()
            .saveInBackground();
    }
}
```

AndroidManifest

Tenemos que actualizar el *AndroidManifest.xml* e indicar el nombre de nuestra *Application*:

`<application android:name= "paquete.nombreApplication" >`

```
<application
    android:name="net.androidsensei.pushsensei.PushSenseiApplication"
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

Permisos

Para que la aplicación funcione correctamente, será necesario configurarle una serie de permisos en el AndroidManifest:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

<!--
    IMPORTANT: Change "com.parse.starter.permission.C2D_MESSAGE" in the lines below
    to match your app's package name + ".permission.C2D_MESSAGE".
-->
<permission android:protectionLevel="signature"
    android:name="com.parse.starter.permission.C2D_MESSAGE" />
<uses-permission android:name="com.parse.starter.permission.C2D_MESSAGE" />
```

BroadcastReceiver

Ahora solo queda agregar unos cuantos BroadcastReceiver que vienen incluidos en las librerías de Parse, para que se encarguen de mostrar los mensajes en la barra de título, hacer vibrar el teléfono, en si procesar todas las notificaciones entrantes enviadas desde Parse.

Añadimos al *AndroidManifest.xml* las siguientes líneas:

BroadcastReceiver

```
<service android:name="com.parse.PushService" />
<receiver android:name="com.parse.ParseBroadcastReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.intent.action.USER_PRESENT" />
    </intent-filter>
</receiver>
<receiver android:name="com.parse.ParsePushBroadcastReceiver"
    android:exported="false">
    <intent-filter>
        <action android:name="com.parse.push.intent.RECEIVE" />
        <action android:name="com.parse.push.intent.DELETE" />
        <action android:name="com.parse.push.intent.OPEN" />
    </intent-filter>
</receiver>
<receiver android:name="com.parse.GcmBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND">
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <action android:name="com.google.android.c2dm.intent.REGISTRATION" />

    <!--
        IMPORTANT: Change "com.parse.starter" to match your app's package name.
    -->
    <category android:name="com.parse.starter" />
    </intent-filter>
</receiver>
```

Enviar Notificaciones desde Parse

Enviando Notificaciones

Regresamos al Dashboard de la App en Parse que creamos al iniciar el tutorial y buscamos una pestaña que diga Push Notifications y presionamos el boton +Send a push.

Tienes opción de elegir el tipo de cliente al que enviaras la notificación, cuantos dispositivos están enlazados, si quieres enviar las notificaciones a todos o a un segmento (Parse te permite que tus clientes se suscriban en distintos canales para recibir información segmentada), el mensaje que quieres mandar y el momento en que quieres mandarlo.

Elegir Destinatarios

P prueba ▾

Core Analytics **Push** Settings Docs


Cristian Muñiz

[← Back](#)


Send a push notification

Choose Your Recipients

Send to everyone, or use segments to choose the right users.


Everyone

or


Segment


This will be sent to **4 devices**

A/B Testing


Experiment with different messages or send times to discover the optimal campaign variables.

[Docs](#) [Billing](#) [Downloads](#) [Help](#) [Status](#) [Blog](#) [Parse.com](#)


Elegir Destinatarios


Everyone

or


Segment

CHOOSE PLATFORMS


Android

INSTALLATION CONDITIONS

channels

contains one of


Goles x Se

Add condition

Elegir la hora de entrega


Choose a Delivery Time

We can send the campaign immediately, or wait until just the right moment.



Now


or



Later

Deliver on

2015-02-20



4

:

09

AM

Time Zone

GMT ☒ Local to User

Expires

No ☒ Yes

Escribir mensaje

Write Your Message

The best campaigns use short and direct messaging.

Your message here

Message Type

Plain Text ☒ JSON

Increment App Badge

No ☒ Yes

Profundizando en Parse

Installations

Cada aplicación instalada en un dispositivo tiene asociado un objeto *Installation*.

En el objeto Installation es donde se guardan todos los datos necesarios para manejar las notificaciones.

Se utiliza la API *ParseInstallation* para salvar y recuperar los datos.

Installations

Para acceder al actual objeto *Installation* de la aplicación se utiliza el método:

```
ParseInstallation.getCurrentInstallation();
```

La primera vez que se guarda un *ParseInstallation*, Parse agrega la *Installation* y habilita el dispositivo para recibir notificaciones.

```
ParseInstallation.getCurrentInstallation()  
    .saveInBackground();
```

Installations

ParseInstallation dispone de una serie de campos para gestionar el dispositivo:

- ⦿ **installationId**: identificador único para el dispositivo utilizado por Parse (*sólo lectura*) .
- ⦿ **appName**: nombre de la aplicación cliente a la que pertenece la instalación (*sólo lectura*) .
- ⦿ **timeZone**: La zona horaria actual en la que se encuentra el dispositivo de destino. Este valor se sincroniza cada vez que un objeto *Installation* se guarda desde el dispositivo (*sólo lectura*) .
- ⦿ **channels**: conjunto de los canales a los que un dispositivo está suscrito actualmente.

Entre otros más. Además podemos configurar nuestro propios campos.

Channels

La forma más sencilla de empezar a enviar notificaciones está utilizando canales. Estos permiten utilizar un modelo de editor-suscriptor para enviar notificaciones. El servicio comienza suscribiéndose a uno o más canales, y después, las notificaciones se pueden enviar a los suscriptores. Los canales se almacenan en el campo channels de *Installation*.

Channels

Para suscribir el dispositivo a un canal utilizamos:

```
ParsePush.subscribeInBackground("nombreCanal");
```

Y para darlo de baja:

```
ParsePush.unsubscribeInBackground("nombreCanal");
```

Podemos obtener la lista de los canales a los que está suscrito mediante:

```
List<String> subscribedChannels =  
ParseInstallation.getCurrentInstallation().getList("channels");
```

Channels

Podemos mandar notificaciones a los suscriptores de los canales que queramos desde la aplicación cliente de la siguiente manera:

```
ParsePush push = new ParsePush();  
push.setChannel("nombreCanal");  
push.setMessage("mensaje");  
push.sendInBackground();
```

Lo habitual es que sea el servidor el que genera los mensajes.

Campos personalizados

Podemos introducir nuevos campos definidos por nosotros para *Installation* que podamos requerir para gestionar con mayor precisión los notificaciones de nuestra aplicación. Se utiliza:

```
ParseInstallation installation =  
ParseInstallation.getCurrentInstallation();
```

```
//installation.put(String clave, Object value);  
//value != null
```

```
installation.put("gameResults",true);  
installation.put("injuryReports",true);
```

```
installation.saveInBackground();
```

Bibliografía

Bibliografía

Notificaciones Push

- <https://androidsensei.net/notificaciones-push-en-android-sin-sudor-ni-dolor-ni-lagrimas-con-parse/>
- <http://es.slideshare.net/ykro/cultura-android-push-notifications-con-parsecom>
- <http://netosolis.com/push-notifications-como-wathsapp-android-con-parse-javascript/>

Bibliografía

Tutoriales y Documentación de Parse

- ◉ <https://parse.com/>
- ◉ <https://parse.com/docs>
- ◉ <https://parse.com/tutorials/android-push-notifications>
- ◉ https://parse.com/docs/push_guide#top/Android
- ◉ <https://parse.com/docs/android/api/>

Bibliografía

AMPLIACIÓN:

Notificaciones Push con Google Cloud Messaging

- <http://www.sgoliver.net/blog/notificaciones-push-android-google-cloud-messaging-gcm-introduccion/>
- <http://www.sgoliver.net/blog/notificaciones-push-android-google-cloud-messaging-gcm-implementacion-servidor/>
- <http://www.sgoliver.net/blog/notificaciones-push-android-google-cloud-messaging-gcm-implementacion-cliente/>