Push notifications to devices

Push iOS and Android notifications

With Apstrata you can easily push notifications to any iOS or Android devices, asynchronously or synchronously just by invoking the PushNotifications API. You also can create reusable device groups, called channels, to which you can push simultaneously, or you can dynamically build a list of devices further to the execution of a no-SQL query on your content or user directory for example. To configure your connection to a push notification middle tier such as APNS or FCM, you invoke the AddCertificate API or, even simpler, resort to the Apstrata Workbench.

Back to the map   **Next station**: send email

Example: push a promotion code to all players who have been registered to your game for more than 6 months

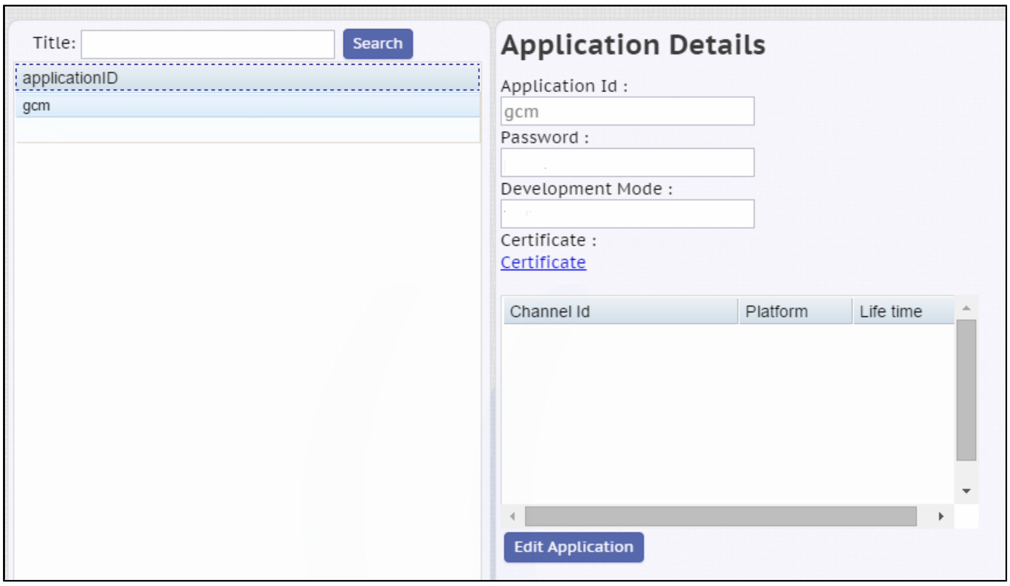
In order to generate more traction to your cool mobile app game, let us imagine that you decide to regularly push promotions through notifications to some of your players, in the current case, those who have been using registered to your game for more that 6 months. Let us show you how easy it is to achieve this with Apstrata but first, let us configure Apstrata by uploading your provider's credentials to your Apstrata application.

Configure Apstrata to push

Sign in to the Apstrata Workbench, click on "Manage app > Pus notifications" then click on a store ("DefaultStore" by default). In the panel that opens, click on "New" at the bottom right corner of the panel in order to create a new configuration (called "application"). In the form that is displayed, enter an application id.

- APNS
  - If you intend to push through APNS, fill the "password" field with the password of your APNS certificate
  - From the "certificate" field, upload your certificate
  - If your certificate is for development only, set the value of the "development" field to "true", otherwise set it to "false"
- FCM
  - Leave the password field empty
  - Click on the "certificate" field to upload text file that contains your API Key
  - Leave the "development" field empty

Click on "Save" and you are set.



Application Id : gcm

Start pushing

Although you can directly invoke the PushNotification API from your client side, it makes more sense in our example to execute the push from the application back-end, since we notably need to prepare the list of recipients (players registered for more than 6 months). therefore, the example is implemented only using an Apstrata server-side script.

```
<script>
        <scriptACL>
                <execute>nobody</execute>
                <read>nobody</read>
                <write>nobody</write>
        </scriptACL>
        <code>
                <![CDATA[

                        // Prepare the parameters to pass to the ListUsers API in order to search for
users
                        // who registered more than n months ago
                        var queryUsersParams = {
                            // Search for users who have a creation time <= the given date and that
                            "apsdb.query": "apsdb.creationTime<date> <= \"2014-07-01\" and pushToken is
not null",
                            // only return the login and pushToken attributes of the user's profile. We
assume that we added the "pushToken" field to the users profiles
                            "apsdb.attributes": "login, pushToken"
                        }

                        // Execute the request by invoking the ListUsers API
                        var response = apsdb.callApi("ListUsers", queryUsersParams, null);
                        if (response.metadata.status == "failure") {
                            return response.metadata;
                        }

                        var users = response.result.users;
                        if (users.length == 0) {
                            return {
                                "result": "No matching users"
                            }
                        }

                        // Create an array of device tokens (FCM registration ids) from the data
returned by the query
                        var tokens = [];
                        for (var i = 0; i < users.length; i++) {
                            tokens.push(users[i].pushToken[0]); // We assume that users only have one
device token (FCM registration id)
                        }

                        // Prepare the parameters to pass to the push notification API
                    var notificationStr = "{\"Content\":\"Get 50% using this promotion code for your
next purchase: AZX34W \"}";
                        var pushParams = {
                            "apsdb.store": "DefaultStore",
                            "adhoc.tokens": tokens,
                "adhoc.lifeTime": "3600", // the lifetime in seconds of a notification before it is
discarded if not pushed
                "adhoc.applicationID": "fcm", // the identifier of the push application in order to use
the correct credentials
                "adhoc.platform": "Android",
                "fireAndForget": "false", // asynchronous pus, do not wait for a response
                "notification": notificationStr
                        };

                        // Call the PushNotification API
                        var pushNotificationsResponse = apsdb.callApi("PushNotification", pushParams,
null);
            if (pushNotificationsResponse.metadata.status == "failure") {
                return pushNotificationsResponse.metadata;
            }

            return pushNotificationsResponse;
                ]]>
        </code>
</script>
```

Try it!

Download the following apk and deploy it to your Android device. Once done, run the application, which will display your device registration id and open an email client. Enter your email address and send, then copy/paste the registration id that you received into the below field.

Registration id

Push

Dig deeper

- Push Notifications APIs

Related tutorials

Push notifications to devices	**Send emails**