If you are creating a client-side library, application, or open source project that will be redistributed and installed by end-users, you may want to require each of your users to register their own application rather than including your own API token with the software. See our Knowledge Base (https://support.pushover.net/i37) for more information.

Users, Groups, and Devices

Once you have an API token, you'll need the user key and optional device name for each user to which you are pushing notifications. If a device name is not specified for a user, or the specified device name is no longer enabled/valid, notifications will be sent to all active devices for that user to avoid losing messages. Messages may be addressed to multiple specific devices by joining them with a comma (such as `device` = `iphone,nexus5`).

Instead of a user key, a group key (/api/groups) may be supplied. Group keys look identical to user keys and from your application's perspective, you do not need to distinguish between them. When sending notifications to a group key, all active users listed in the group will have the notification delivered to them and the response will look the same.

Alternatively, a message may be sent to multiple users in one request by specifying a comma-separated list (with no spaces) of user keys as the `user` parameter. These requests are currently limited to 50 users in a single request.

When sending to delivery groups not belonging to a Pushover for Teams organization, or specifying multiple users in a single request, the `device` parameter will be ignored. Group users will have their specific device honored according to how they are entered in the group.

When sending to a single Team-owned group, the `device` name *is* honored, and will restrict sending the message to just the team member devices matching the name in the group. If no devices match, the message will not be broadcast to any users and the API will return a failure status.

As with application API tokens, user keys should be considered private and not disclosed to 3rd parties. Users should be able to update their identifiers and/or device names with your application or service.

Example User Identifier: `uQiRzpo4DXghDmr9QzzfQu27cmVRsG`

Example Group Identifier: `gznej3rKEVAvPUxu9vvNnqpmZpokzF`

Example User Device Name: `droid2`

User and group identifiers are 30 characters long, case-sensitive, and may contain the character set `[A-Za-z0-9]`. Device names are optional, may be up to 25 characters long, and will contain the character set `[A-Za-z0-9_-]`.

As an optional step, your application may validate user or group identifiers after they have been submitted to you. This will ensure that a user has copied his or her identifier properly, that the account is valid, and that there is at least one active device on the account.

Pushing Messages

Messages must contain a `message` parameter that contains the message body and an optional `title` parameter. If the title is not specified, the application's name will be shown by default. HTTP and HTTPS URLs included in messages will be automatically parsed by the device clients and shown as clickable links. To include a clickable link outside of your message body, see the supplemental URL documentation.

In this example, we will use the application token, user key, and device name above to push a message about a completed process. Using an HTTPS library available in your application's language, construct a `POST` request (not a `GET` request which is often the default) to the following URL:

```
https://api.pushover.net/1/messages.json
```

The `.json` suffix requests that the response be in JSON format. `https://api.pushover.net/1/messages.xml` may be used instead to receive an XML response. Note that this does not affect how you send your parameters to our server, which is controlled by the `Content-Type` header you send with your input. We do not support receiving XML-encoded parameters, only the standard percent-encoding (https://en.wikipedia.org/wiki/Percent-encoding) that most HTTP libraries default to, and JSON with a `Content-Type` of `application/json`.

HTTPS is required for all API calls, and for security purposes, your application should enable your HTTP library's TLS/SSL verification. The POST method is required be used for the API call to push messages.

Include the `token`, `user`, `device` (optional), `title` (optional), and `message` parameters in the body of the request as standard key-value pairs. Continuing with our example, these parameters would be:

`token` = `azGDORePK8gMaC0QOYAMyEEuzJnyUi`

`user` = `uQiRzpo4DXghDmr9QzzfQu27cmVRsG`

`device` = `droid4`

`title` = `Backup finished - SQL1`

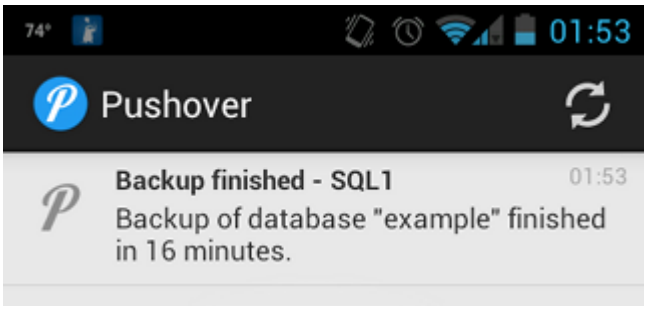
`message` = `Backup of database "example" finished in 16 minutes.`

Those parameters would look like this when POSTed as a URL-encoded (also known as percent-encoded) request:

```
POST /1/messages.json HTTP/1.1
Host: api.pushover.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 180

token=azGDORePK8gMaC0QOYAMyEEuzJnyUi&user=uQiRzpo4DXghDmr9QzzfQu27cmVRsG&device=droid4&title=Backup+finishe
```

That message would appear like this in the Pushover client on an Android device:



HTML/Message Styling

As of version 2.3 of our device clients, messages can be formatted with HTML tags. As of version 3.4, messages can be formatted with a monospace font.

To enable HTML formatting, include an `html` parameter set to `1`. The normal message content in your `message` parameter will then be displayed as HTML.

To enable monospace messages, include a `monospace` parameter set to `1`. `monospace` may not be used if `html` is used, and vice versa.

Due to limitations with notifications on mobile platforms, HTML tags and monospace formatting are stripped out when displaying your message as a notification (leaving just the plain text of your message). Once the device client is opened and your message has been downloaded from our server, it will be displayed with appropriate HTML or