



Mobile integration API  
v 2.0

Overview.....	3
Requirements .....	3
Environments .....	3
Device Authentication/Registration.....	3
iBeacon Regions .....	4
Geofences.....	5
Beacon Proximity Zones.....	6
Device Profile .....	6
Register/Update User Profile .....	8
App Events.....	9
GPS Location Changes .....	10
iBeacon Location Changes .....	11
Geofence Changes .....	12
Estimote Beacon Proximity Changes.....	12
Unread Signal Count.....	13
Get Signal History.....	14
Set Signal as Read.....	15
Set Signal as Deleted .....	16
Render Signal HTML .....	16

## Overview

The LighthousePE (LHPE) API is a collection of Restful JSON http endpoints that facilitate the integration of the LHPE platform into your mobile app.

## Requirements

You must obtain your LHPE application ID (`api_id`) and Key (`api_key`) in order to communicate with the API. Your application ID and Key will be different for each API environment

All requests are sent in x-www-form-urlencoded format

All requests must be sent via HTTPS (SSL)

## Environments

Staging – <https://staging-api.lighthousepe.com>

Production - <https://api.lighthousepe.com>

## Device Authentication/Registration

When your app has finished starting you will need to register/update the installation with the platform. On iOS, this could be done within the UIApplicationDelegate `didFinishingLaunching` method. On Android, this could be done in your main activity's `OnStart` method or in your main application class' `onCreate` method.

Before you first call this endpoint, you will need to generate or obtain and securely store a unique device identifier. There are several good articles on how to best implement device identification, but we do not have a preferred method of implementation because it can vary so widely by device platform and version, and strategies evolve over time.

A successful request will result in a response with an Authorization header. Store this Authorization key for subsequent requests.

### Request

Path	/auth/device
Method	POST
Headers	None

Body	All fields are required		
	api_id	string	Your LHPE app api_id
	api_key	string	Your LHPE app api_key
	device_identifier	string	Your generated unique device ID
	make	string	Programmatically obtained device make. e.g. "Apple" or "Samsung"
	model	string	Programmatically obtained device model. e.g. "iPhone XR" or "Pixel 2"
	system_name	string	"iOS" or "Android"
	system_version	string	Programmatically obtained OS version e.g. "13.1.2"
	app_version	string	Your app's version number e.g. "1.1.0"

### Response

Headers	Authorization	string	Your installation authorization key
Body	NA		

## iBeacon Regions

If your application uses Apple iBeacon Regions, they should be configured on the LHPE platform, which can be used to manage them and their definitions obtained without having to hardcode them into your app.

### Request

Path	/app/beaconRegions		
Method	GET		
Headers	Authorization	string	Your installation authorization key
Body	NA		

### Response

Headers	NA		
Body	[ { "id": "d21cd31d-3c21-4347-8fde-28c89e1d3815",		

	<pre>         "name": "Sample",         "uuid": "ef800f2c-7c26-43c5-b62d-3cb4368fa501",         "major": 1000,         "minor": 100       }     ] </pre>
--	--

## Geofences

If your application uses circular Geofences, they should be configured on the LHPE platform.

### Request

Path	/app/geofences			
Method	GET			
Headers	<table border="1"> <tr> <td>Authorization</td> <td>string</td> <td>Your installation authorization key</td> </tr> </table>	Authorization	string	Your installation authorization key
Authorization	string	Your installation authorization key		
Body	NA			

### Response

Headers	NA
Body	<pre> [   {     "id": "73a1f0e3-120d-4dd2-a404-328f96e81ac0",     "name": "Oxford MD",     "timeout": 5400,     "h_accuracy_min": 20,     "radius": 26.8902339386184,     "longitude": -76.1659886708754,     "latitude": 38.687593768041   } ] </pre>

## Beacon Proximity Zones

If your application uses Estimote Proximity Beacons, they should be configured on the LHPE platform.

### Request

Path	/app/proximityZones			
Method	GET			
Headers	<table border="1"><tr><td>Authorization</td><td>string</td><td>Your installation authorization key</td></tr></table>	Authorization	string	Your installation authorization key
Authorization	string	Your installation authorization key		
Body	NA			

### Response

Headers	NA
Body	<pre>[   {     "id": "d21cd31d-3c21-4347-8fde-28c89e1d3815",     "name": "Sample Zone",     "proximity_tag": "zone_a",     "proximity_range": 2.5   } ]</pre>

## Device Profile

LHPE provides a means of tracking specific installation attributes and any other attributes you consider important.

### Location Permissions

It's important for LHPE to know the state of the installation's ability to collect location data. Use the operating system's development SDK to obtain the current state of location permissions and inform the platform. We recommend you send this information;

- A) Immediately after the Device Registration request has completed and/or when the application has become active from being in the background.

- B) After you have requested location permissions and the user has granted or denied the request.

**Push Notification Permissions**

LHPE should also be informed of the installation’s ability to receive push notifications. LHPE will not attempt to deliver push notifications if the device registration is not considered approved for push. We recommend you send this information;

- A) Immediately after the Device Registration request has completed and/or when the application has become active from being in the background.
- B) After you have requested location permissions and the user has granted or denied the request.

**Push Notification Token**

LHPE will attempt to send push notifications to your app installations when they automatically trigger the delivery of content/offers or a broadcast is sent manually. Inform LHPE of the push notification token after you have requested permission to send them and the user has approved.

**Temporary Profile**

LHPE has the ability to also track lightweight data about the state of an app installation. When LHPE is properly configured with metadata about the expected field data, it can be used to query your Customer Roster and used as conditions in automated signals. Temporary profile data is generally only used for anonymous installations, or installations where the end user has not registered or authenticated with the app. Once the end user is considered authenticated, you should use the User Profile endpoint. This data should be transmitted as a UTF-8 encoded JSON string. The LHPE customer filter and signal conditions do NOT support nested data structures in this JSON.

**Request**

Path	/app/device										
Method	PUT										
Headers	Authorization	string	Your installation authorization key								
Body	All fields are optional. At least one is required.										
	location_permission	<table border="1"> <tr> <td>null</td> <td>Unknown</td> </tr> <tr> <td>0</td> <td>Denied</td> </tr> <tr> <td>1</td> <td>Always Allowed</td> </tr> <tr> <td>2</td> <td>Only While Open</td> </tr> </table>	null	Unknown	0	Denied	1	Always Allowed	2	Only While Open	
null	Unknown										
0	Denied										
1	Always Allowed										
2	Only While Open										

		3	Only Once
allows_push	boolean	true or false	
push_token	string	Programmatically obtained device push token	
temp_profile	JSON	e.g. {"foo": "bar"}	

## Response

Headers	NA
Body	NA

## Register/Update User Profile

Once the app end user has successfully authenticated, you should provide LHPE with the user's ID and profile. LHPE will associate the user with the device and you will be able to easily locate them in the Customer Roster.

## Foreign ID

After a user has authenticated with the app, you should have access to their unique identifier. This value varies depending on the platform your app uses to register/authenticate them.

## User Profile

LHPE has the ability to also track lightweight data about the attributes of a user such as their name, age, gender, and any other information you feel is relevant to your marketing. When LHPE is properly configured with metadata about the expected field data, it can be used to query your Customer Roster and used as conditions in automated signals.

## Request

Path	/app/user		
Method	PUT		
Headers	Authorization	string	Your installation authorization key
Body	All fields are required.		



	foreign_id	string	Programmatically obtained unique user identifier
	profile	JSON	e.g. {"first_name": "John", "last_name": "Williams"}

## Response

Headers	NA
Body	NA

## App Events

LHPE can also track “events” that occur in your app, such as when the app is launched or terminated. It can also track custom-defined events that you consider important, such as when someone has clicked a certain button or purchased a product or service. Signals can be triggered to run when these events occur.

## Request

Path	/app/device/event		
Method	POST		
Headers	Authorization	string	Your installation authorization key
Body	All fields are optional. At least one is required.		
	event	string	e.g. “LAUNCH”, “SLEEP” or the ID of your custom event
	signal_id	string	If the event is associated with a LHPE Signal, send the ID

## Response

Headers	NA
Body	NA

## GPS Location Changes

A core feature of LHPE is keeping a history of device locations and triggering content delivery based on presence within Geofences. Your app should be collecting GPS location data in both the foreground and background. LHPE has no limitations to how often you send this data, but your method of collecting this data should comply with the guidelines for the App Store and Google Play. Ideally, your app will poll the device location at an interval that provides enough resolution of their activity to satisfy your marketing but is also respectful of their battery.

On iOS, we recommend you consider using the Significant Location Change Service in the background and location polling while the app is in the foreground.

[https://developer.apple.com/documentation/corelocation/getting\\_the\\_user\\_s\\_location/using\\_the\\_significant-change\\_location\\_service](https://developer.apple.com/documentation/corelocation/getting_the_user_s_location/using_the_significant-change_location_service)

On Android, you will have to decide between using a Foreground Service or requesting background location tracking. <https://developer.android.com/training/location/receive-location-updates?authuser=1>

### Location History

We recommend you send location updates to LHPE in batch, no more frequently than once per minute. Your app should keep track of the last date and time location was successfully sent and until a minute has passed, manage a history of location changes that will be sent when that moment has passed. This location history should be sent as an array with a timestamp of when the location was reported by the SDK.

### Request

Path	/app/device/location												
Method	POST												
Headers	Authorization	string	Your installation authorization key										
Body	All fields are optional. At least one is required.												
	history	array	<table border="1"><tr><td>type</td><td>“GPS”</td></tr><tr><td>latitude</td><td>float</td></tr><tr><td>longitude</td><td>float</td></tr><tr><td>altitude</td><td>float</td></tr><tr><td>horizontal_accuracy</td><td>float</td></tr></table>	type	“GPS”	latitude	float	longitude	float	altitude	float	horizontal_accuracy	float
type	“GPS”												
latitude	float												
longitude	float												
altitude	float												
horizontal_accuracy	float												

			vertical_accuracy	float
			created_at	Timestamp in ISO 8601 format

## Response

Headers	NA
Body	NA

## iBeacon Location Changes

LHPE was also designed to support tracking indoor locations via Bluetooth Low Energy Beacons. One form of beacon identification is the Apple iBeacon standard, directly supported by iOS and macOS. Your iBeacon beacon regions can be configured on the LHPE platform and entry and exit events reported through the same endpoint and history array as GPS data.

## Request

Path	/app/device/location			
Method	POST			
Headers	Authorization	string	Your installation authorization key	
Body	All fields are optional. At least one is required.			
	history	array	type	“iBeacon”
			uuid	string
			major	Integer or null
			minor	Integer or null
			event	“ENTER” or “EXIT”
			created_at	Timestamp in ISO 8601 format

## Response

Headers	NA
Body	NA

## Geofence Entry and Exit Events

To detect geofence activity, LHPE typically consumes raw GPS data and uses geospatial queries to determine geofence presence. In other cases, you may want to rely on the iOS or Android platform to detect geofence entries and exits.

### Request

Path	/app/device/location											
Method	POST											
Headers	<table border="1"><tr><td>Authorization</td><td>string</td><td>Your installation authorization key</td></tr></table>	Authorization	string	Your installation authorization key								
Authorization	string	Your installation authorization key										
Body	<p>All fields are optional. At least one is required.</p> <table border="1"><tr><td>history</td><td>array</td><td><table border="1"><tr><td>type</td><td>“Geofence”</td></tr><tr><td>id</td><td>Geofence’s ID, as provided by LHPE</td></tr><tr><td>event</td><td>“ENTER” or “EXIT”</td></tr><tr><td>created_at</td><td>Timestamp in ISO 8601 format</td></tr></table></td></tr></table>	history	array	<table border="1"><tr><td>type</td><td>“Geofence”</td></tr><tr><td>id</td><td>Geofence’s ID, as provided by LHPE</td></tr><tr><td>event</td><td>“ENTER” or “EXIT”</td></tr><tr><td>created_at</td><td>Timestamp in ISO 8601 format</td></tr></table>	type	“Geofence”	id	Geofence’s ID, as provided by LHPE	event	“ENTER” or “EXIT”	created_at	Timestamp in ISO 8601 format
history	array	<table border="1"><tr><td>type</td><td>“Geofence”</td></tr><tr><td>id</td><td>Geofence’s ID, as provided by LHPE</td></tr><tr><td>event</td><td>“ENTER” or “EXIT”</td></tr><tr><td>created_at</td><td>Timestamp in ISO 8601 format</td></tr></table>	type	“Geofence”	id	Geofence’s ID, as provided by LHPE	event	“ENTER” or “EXIT”	created_at	Timestamp in ISO 8601 format		
type	“Geofence”											
id	Geofence’s ID, as provided by LHPE											
event	“ENTER” or “EXIT”											
created_at	Timestamp in ISO 8601 format											

### Response

Headers	NA
Body	NA

## Estimote Beacon Proximity Changes

A popular manufacturer and platform for beacon fleet management is Estimote. LHPE supports Estimote beacon proximity zones by tag. Your beacon proximity activity can be configured on the LHPE platform and entry and exit events reported through the same endpoint and history array as GPS and iBeacon data.

### Request

Path	/app/device/location										
Method	POST										
Headers	Authorization	string	Your installation authorization key								
Body	All fields are optional. At least one is required.										
	history	array	<table border="1"> <tr> <td>type</td> <td>“EstimoteProximity”</td> </tr> <tr> <td>tag</td> <td>Proximity Zone Tag</td> </tr> <tr> <td>event</td> <td>“ENTER” or “EXIT”</td> </tr> <tr> <td>created_at</td> <td>Timestamp in ISO 8601 format</td> </tr> </table>	type	“EstimoteProximity”	tag	Proximity Zone Tag	event	“ENTER” or “EXIT”	created_at	Timestamp in ISO 8601 format
type	“EstimoteProximity”										
tag	Proximity Zone Tag										
event	“ENTER” or “EXIT”										
created_at	Timestamp in ISO 8601 format										

### Response

Headers	NA
Body	NA

## Unread Signal Count

You can quickly obtain a count of the device’s unread signals, rather than pulling the entire signal history and calculating it.

### Request

Path	/app/device/unreadsignalcount		
Method	GET		
Headers	Authorization	string	Your installation authorization key
Body	NA		

### Response

Headers	NA
Body	{

	<pre> "count": 23 } </pre>
--	----------------------------

## Get Signal History

Get an array of signals for the device, sorted by creation date, descending. The response body will be a JSON array of signal data.

### Request

Path	/app/device/signals		
Method	GET		
Headers	Authorization	string	Your installation authorization key
Body	NA		

### Response

Headers	NA
Body	<pre> [   {     "id": "70db7833-3437-4cd7-aaec-35f959e44bfa",     "created_at": "2019-10-08T19:01:51.805Z",     "push_notification_text": "This is the body of the push",     "push_notification_headline": "Hello, World!",     "push_notification_content": "The full body of the signal...",     "push_notification_payload": {       "body": "This is the body of the push",       "type": "lighthouse",       "title": "Hello, World!",       "signalId": "70db7833-3437-4cd7-aaec-35f959e44bfa"     },     "last_view_at": null,     "view_count": 0,     "tags": ["one", "two", "three"],     "liked": null/true/false,     "signalCategory": {       "id": 1,       "name": "General", </pre>

	<pre>         "icon": "fa-star",         "color": "FFD700"       }     }   ] </pre>
--	---

## Set Signal as Read

If your UX provides a means of marking a signal as Read, call this endpoint. Signals are automatically marked as read and the `view_count` is incremented when the rendering URL is called.

### Request

Path	/app/device/signals/{signal_id}		
Method	PUT		
Headers	Authorization	string	Your installation authorization key
Body	NA		

### Response

Headers	NA
Body	<pre> {   "id": "70db7833-3437-4cd7-aaec-35f959e44bfa",   "created_at": "2019-10-08T19:01:51.805Z",   "push_notification_text": "This is the body of the push",   "push_notification_headline": "Hello, World!",   "push_notification_content": "The full body of the signal...",   "push_notification_payload": {     "body": "This is the body of the push",     "type": "lighthouse",     "title": "Hello, World!",     "signalId": "70db7833-3437-4cd7-aaec-35f959e44bfa"   },   "last_view_at": "2019-10-08T19:11:00.007Z",   "view_count": 1,   "tags": ["one", "two", "three"], </pre>

	<pre>"liked": null/true/false }</pre>
--	---------------------------------------

## Set Signal as Deleted

Provide your users with the ability to remove a signal from their history. The signal is only marked as deleted and will no longer appear in results in the Get Signal History endpoint.

### Request

Path	/app/device/signals/{signal_id}		
Method	DELETE		
Headers	Authorization	string	Your installation authorization key
Body	NA		

### Response

Headers	NA		
Body	NA		

## Render Signal HTML

LHPE uses HTML/CSS for richly formatted content. Render the resulting HTML body in a Webview.

### Request

Path	/app/signal/{signal_id}		
Method	GET		
Headers	Authorization	string	Your installation authorization key
Body	NA		



## Response

Headers	NA
Body	The rendered signal HTML