



Horizon Integration & Messaging API

HTK Technical Document

An overview of the HTK Horizon Integration & Messaging API, which enables third-party developers to write applications which integrate with the Horizon platform.

Issue 2.4 | July 2016

Notice of Confidentiality

© 2016 HTK Limited. All rights reserved.

THIS DOCUMENT IS HTK PROPRIETARY AND CONFIDENTIAL INFORMATION. NEITHER THIS DOCUMENT NOR ITS CONTENTS MAY BE REVEALED OR DISCLOSED TO UNAUTHORIZED PERSONS OR SENT OUTSIDE THE AFOREMENTIONED INSTITUTION WITHOUT PRIOR PERMISSION FROM HTK.

Document History

Date	Description	Version
18 Apr 2013		1.4
		2.1
22 Nov 2015	Added a summary of all available Horizon APIs	2.2
29 Mar 2016	Updated URLs, link to Portal API, extended OAUTH set- up example	2.3
28 Jul 2016	Added new Offer/Rewards methods	2.4

Table of contents

[Document History](#)

[Introduction](#)

[Documentation](#)

[Resources](#)

[Planned Enhancements](#)

[Horizon Client API](#)

[Introduction](#)

[OAuth 2.0 protocol](#)

[Example configuration](#)

[Configuring your project](#)

[Configuring Horizon](#)

[Appendix - API Reference](#)

[/admin/directories/{key}](#)

[/admin/directories](#)

[contacts/{key}](#)

[/contacts/{key}/fields](#)

[/contacts?firstname&lastname&telephone&mobile&fax&sms&pager&email&category&groupid&since&startindex&maxrows](#)

[/groups?name&type](#)

[/interactions/{key}](#)

[/interactions?type&interfacecustomer&interfacehtk&since&to&maxrows](#)

[/sendmessage](#)

[/offers](#)

[URIs](#)

[URI examples](#)

[Security](#)

[Creating and managing your application in Horizon](#)

[Disclaimer](#)

Introduction

The Horizon APIs enable third party developers to write applications which integrate with the Horizon platform. The Horizon APIs provide access to resources by means of a RESTful service¹, meaning resources are accessed via URIs.

Horizon provides two APIs:

- Integration and Messaging API
 - create, maintain, and read Customer data
 - feed in all Customer activity and events to drive Segmentation and Triggered Actions
 - communicate with customers using Email, Push, In-App, SMS, Voice
 - create Offers and Rewards

This document covers the Integration and Messaging API.

- Loyalty Portal API
 - The Loyalty Portal API provides a comprehensive Loyalty capability for a Customer Portal.
 - Create and manage members, with full member life -cycle management
 - Retrieve and redeem personalised, best Offers and Rewards

The Loyalty Portal API is documented here:

<http://www.htk.co.uk/wp-content/uploads/2016/03/HTKHorizon-LoyaltyPortalRESTAPI.pdf>

Documentation

This document provides a general overview of the Integration and Messaging API. Detailed documentation for the current versions are available at:

<https://api.htkhorizon.com/horizon-api/r/documentation.html>

<https://api.htkhorizon.com/horizon-api/r/application.wadl>

¹ For an introduction to RESTful services refer to

<https://www.ibm.com/developerworks/webservices/library/ws-restful/>

Resources

The Horizon Integration and Messaging API provides access to a subset of Horizon resources. The following table lists the resources which can be managed using the API, and the methods which are supported for each.

Resource	Methods	Description
Contact	GET, PUT, POST	Provides access to a contact's basic information, such as first and last names, date of birth, etc. Also these methods allow to insert / update the contacts' media channels and custom fields.
Contact Activity	GET, POST	The Contact Activity resource will list any interactions the contact has had via the Horizon platform. Horizon will track both inbound and outbound activities, split by campaign where appropriate. External applications can log any contact activities such as mobile app activity, ecommerce activity, product purchases, support requests, etc.
Contact Field	GET, PUT	Contact Fields provide the ability for administrators to 'extend' the default Horizon schema through the configuration of additional contact attributes (see also the Directory resource below)
Contact Location	GET, PUT, POST, DELETE	Contact Locations enable recording of a contact's physical location. A Contact Location may be an address ² or point on a map.
Contact Media	GET, PUT, POST	Provides details of a contact's media (channels) such as email, telephone, sms. Note that Contact Media includes a contact's opt-in preferences for each channel.
Contact Note	GET, POST	Contact notes can be used as a 'manual' audit trail, allowing users to record any conversations or other interactions they have had with the contact.
Contact collection	GET	The list of contacts in Horizon. By default the list is limited to the 500 most recently created or updated contacts, but it is also possible to search for specific contacts.

² Postcode details must be included in addresses to enable geographic contact segmentation (cf. [Horizon map groups](#)).

Content	GET, PUT, POST	Content defines the ascii text, html markup, etc. of messages.
ContentCollection	GET	The content collection returns the list of content currently defined.
Directory	GET	A directory can be thought of as an address book or 'rolodesk'. An organisation may have multiple directories (e.g. customers, staff, etc). Each directory may be associated with profiles, allowing the default Horizon schema to be extended with user defined custom fields. Directory details returned by this resource include all profiles and fields (including data types) configured for the specific directory.
Directory collection	GET	The directory collection returns the list of directories currently configured in Horizon for the organisation.
Group collection	GET	Horizon allows contacts to be segmented using different type of group definitions, including list, map and rule groups. The group collection lists the group names and types currently configured in Horizon.
Interactions	GET, POST	Interactions represent individual messages such as SMS, email, pager, and fax messages, or telephone calls. Interaction details include the originator and recipient of the message or call, message contents, and media specific attributes such as DTMF key presses or speech recognition statistics.
Messages	GET	A message is an instance of message content being delivered to a group of contacts at a specified time. Delivery statistics are available for messages which have been completed, or are in progress.
Send message (*see below)	POST	This allows a customer to send an SMS* message to one or multiple recipients with a limit of 100. Note that Contacts that do not exist are created in the Horizon database with an Opt-in (using Organisation's default). The results of the SendMessage request are logged against the Horizon Contact and available to users through the UI.
Offers	POST, PUT, GET	Create and Update Offers.

		Get a set of best Offers for a Contact, taking account of a range of factors including best match to the Contact, location and time of day.
Session	GET	Returns session details, such as the session id.
User	GET	Returns personal information of the currently logged in user, including a full name, email address, etc.

The Horizon API supports both XML and JSON content types. The content type returned by the service is managed using HTTP accept headers as following:

Representation	Requested via
XML	application/xml in the HTTP Accept header ³
JSON	application/json in the HTTP Accept header

³ Note that XML is the default content type. This means XML content will be returned unless otherwise requested in the HTTP Accept header.

Planned Enhancements

In Horizon 3.0.27 the /sendmessages endpoint of the public API provides SMS functionality.

It will be expanded in a future release to also support Email and Voice messages, and SMS templates.

Resource	Methods	Description
email	GET, POST	Allows sending of an email. The response to the POST request will contain a URL including a unique identifier for the email that the client can later use in a GET request to query its status.
SMS	GET, POST	Allows sending of an SMS message. The response to the POST request will contain a URL including a unique identifier for the SMS that the client can later use in a GET request to query its status.
SMS template	POST	Allows sending of an SMS, defining a SMS template id to use in the request to specify the content.
voice	GET, POST	Sends a recorded message or IVR script via an outbound telephone call.

A simplified example (*not yet implemented*) of a JSON email object is shown below:

```
{  
  "from": "sender@htk.co.uk",  
  "to": "a.recipient@example.com",  
  "subject": "Test message",  
  "text": "Hello! This is an example of a JSON email."  
}
```

Horizon Client API

Introduction

The Horizon Client API is a helper library which wraps the Horizon RESTful API to simplify integration with HTK Horizon using the Horizon API. The Horizon Client API is currently available for Java, if required please let us know and we can provide you with the necessary files.

OAuth 2.0 protocol

The Horizon API is secured using the OAuth 2.0 protocol. Application access to Horizon data is controlled by the Horizon Authentication service, meaning Horizon users do not need to divulge their Horizon credentials to third parties and remain in control over both the level and duration of data access grants.

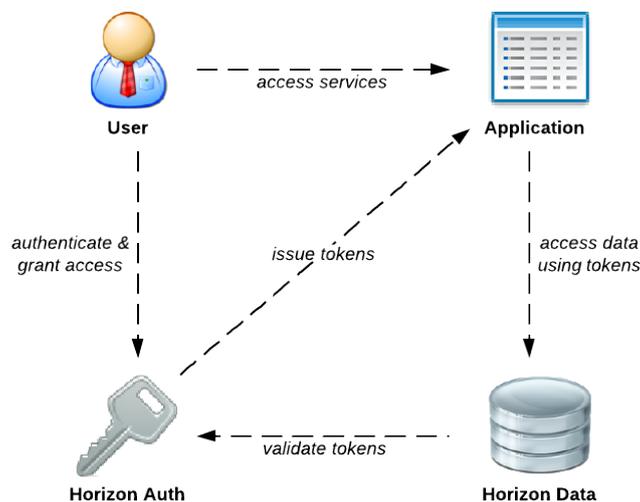


Figure 1. OAuth 2.0 overview

To allow an application to access a user's Horizon data via the Horizon API, it will first need to be registered in Horizon.

Following registration of an application it will be possible to obtain a consumer key and consumer secret token.

These tokens are required to obtain a request token, which will allow a Horizon user to authenticate himself or herself, and subsequently to grant the application access to Horizon data. If application access is authorised by the user, the application will then be granted an access token, which needs to be presented with each request for Horizon data.

Configuration

The configuration process is provided below, together with an example configuration. However HTK Client Services will normally assist you through this process, and support your testing of your authentication implementation.

Example configuration

The following steps show how to create and register an application which uses the Horizon Client API. This example uses the Java version of the Horizon API.

Configuring your project

Add the Horizon dependency and repository to the project object model (pom.xml):

```
<project>
...
<dependencies>
...
<dependency>
  <groupId>com.htkhorizon.api-client</groupId>
  <artifactId>horizon-api-client</artifactId>
  <version>0.0.2</version>
</dependency>
...
<dependencies>
...
<repositories>
...
<repository>
  <id>htkhorizon-nexus</id>
  <url>http://dev.htkhorizon.com/nexus/content/repositories/htkhorizon-maven</url>
</repository>
</repositories>
</project>
```

Configuring Horizon

To allow your application to access Horizon data it first needs to be configured in the Manage Applications

section, Applications in Horizon Settings, using the steps below.

HTK horizon Logged in as pmaterniak Account: PGS change Manage my account | Logout Help Centre Settings

Home Contacts Content Offer Management Campaigns Flood Events Triggered Actions Reports Add-ons Visualisation

Add application

General

Name* Retail CRM Integration

Comments* Integration to sync customer details between CRM and the Horizon platform

Website* http://www.yoursite.faq

Callback URL* http://127.0.0.1:8080/callback

Application icon

Icon APPLICATIONS/HTK/logo/horizon Clear »

Application icon HTK horizon

Application configuration

Integration & Messaging API
For clients using the Integration & Messaging API. Horizon administrator functionality with USER level access.

Loyalty Portal API
For clients using the Loyalty Portal API. Loyalty functionality with CONTACT level access.

Event API
For clients using the Event API. Functionality with CONTACT and USER level access.

User access

Contacts

- Read
- Read and Write
- Delete

Groups

- Read

Messages

- Read
- Read and Write

Directories

- Read

Interactions

- Read
- Read and Write

Broadcasts

- Read

Configuration

- Read

Offers

- Read
- Read and Write

Loyalty & Rewards

- Read
- Read and Write
- Delete

Redemptions

- Read and Write

Save Close »

Figure 2. Example 'Add application' screen.

1. Enter your application details in the General Section, and optionally select an existing icon for your application from the Horizon content repository, or upload a new image.

Field	Description
Name	The name of your application
Comments	A brief description of your application
Website	The site to which users will be directed if they decline permissions to your application. This could be your normal company website, or a specific page describing your application, etc.
Callback URL	The url of your OAuth 2 verification code receiver. Note that the callback URL is case sensitive and must exactly match the redirect url specified in the initial authentication code request.

2. Note that the name, comments and application icon will be presented to the user when they are asked to grant application access to Horizon data:

3. Select the permissions required for the application. Note that users, depending on their role and assigned license(s), will not necessarily have all permissions required by the application (cf. Figure 3). It is therefore important that applications are able to deal with “missing permissions” appropriately.

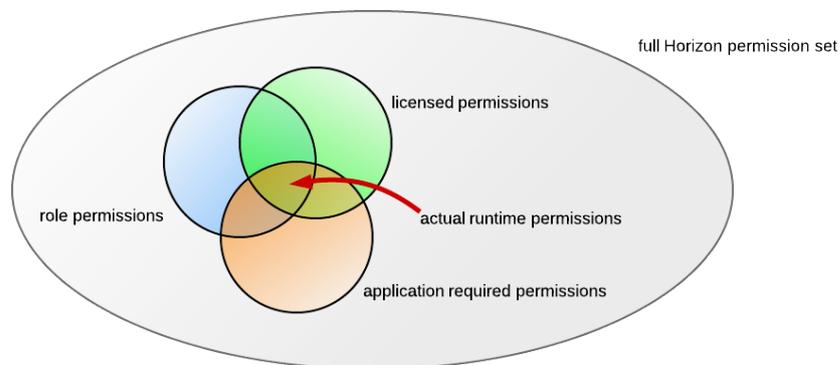


Figure 4. Application runtime permissions model

4. Save the application details.

5. In the list of applications, click 'View' to see details of your application:

Applications

[Add new application](#)

Name	Type	Description	Website	Last activity	Actions
Retail CRM Integration	Integration & Messaging	Integration to synchronise customer details between CRM and the Horizon platform	http://www.yoursite.faq		  

Page 1 of 1

Figure 5. Click to view application details.

View application

Authentication

Client id e8YZRf4AUIQM0laNgxclmZexErQ=

Client secret vOLbXK0Cj53CaUB8DSxu28FVrOSojPSasehIAheqR0E=

Authorization URL https://api.experiencehorizon.com/oauth/authorise

Access token URL https://api.experiencehorizon.com/oauth/access_token

Callback URL http://127.0.0.1:8080/callback

General

Application configuration

User access

Contacts Read
 Read and Write
 Delete

Groups Read

Messages Read
 Read and Write

Directories Read

Interactions Read
 Read and Write

Broadcasts Read

Configuration Read

Offers Read
 Read and Write

Loyalty & Rewards Read
 Read and Write
 Delete

Redemptions Read and Write

[Cancel](#)

Figure 6. Application details

6. Make a note of the 'Client id' and 'Client secret' values, as well as the 'Authorization' and 'Access token' URLs.

Field	Value
Client id	yw9zFTOuMgLRf7m5jG3uNXbNtss=
Client secret	q5LzM/bGlzik45X+XmWIduFdgEf8OB7NEqVnCdVmaAk=
Authorization URL	https://api.htkhorizon.com/horizon-api/r/oauth/authorize
Access token URL	https://api.htkhorizon.com/horizon-api/r/oauth/token

The 'Client id' and 'Client secret' values will be needed later to pass as the client_id and client_secret fields in the OAuth token request.

7. Store the consumer credentials in a class. Note that these credentials are specific to your application and should be kept both secure and confidential, meaning it is not appropriate to store them as clear text in a Java Properties or other kind of text file.

```
package com.theretailcompany.crmintegration;

public class ConsumerCredentials {
    /** Value of the "API Consumer key" */
    public static final String CONSUMER_KEY = "yW9....";

    /** Value of the "API Consumer secret" */
    public static final String CONSUMER_SECRET = "q5L....";
}
```

8. You will need your credentials when instantiating a Horizon OAuth Basic Client:

```
public class ExampleApiClient {
    private static BasicClient client;

    public static void main(String[] args) {
        try {
            VerificationCodeReceiver receiver = new LocalServerReceiver();
```

```
        client = new BasicClient(
            ConsumerCredentials.CONSUMER_KEY,
            ConsumerCredentials.CONSUMER_SECRET);
    ...
} catch (Exception e) {
    System.err.println(e.getMessage());
}
}
}
```

9. Send a post request to the `/oauth/authorise` endpoint including the client ID.
10. This will return a login form. Login as an HTK User.
12. This will return the provider screen. A page asking what permissions to grant the application behalf of the User. Submit this form.
13. An auth token will be returned.
14. Post this auth token to the `/oauth/token` endpoint including the client ID and client secret.
15. This will return an access token to use on future Q/R webservice requests.

As covered above, HTK Client Services will normally assist you with this process.

Appendix - API Reference

This section gives a snapshot of the API for information. For current detailed documentation please see:

<https://api.htkhorizon.com/horizon-api/q/documentation.html> <https://api.htkhorizon.com/horizon-api/q/application.wadl>

/admin/directories/{key}

Parameters

parameter	value	description
key	long	unique identifier for a directory

Methods

GET

Retrieves details for the directory with the specified identifier.

Available responses

200	OK	The response includes all available details for the directory identified by <i>key</i> , including the number of contacts in the directory, as well as any profiles (including field definitions) associated with the directory.
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The directory with the given identifier does not exist

/admin/directories

Methods

GET

Retrieves collection of directories configured for the organisation.

Available responses

200	OK	The response contains details of all directories.
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The directory with the given identifier does not exist

contacts/{key}

Parameters

parameter	value	description
key	long	unique identifier for a contact

Methods

GET

Retrieves details for the contact with the specified identifier.

Available responses

200	OK	The response includes all available details for the contact identified by <i>key</i> , including any media, locations, profile fields, etc, but does not include any contact activity.
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The contact with the given identifier does not exist

PUT

Updates details for the contact with the specified identifier.

Available responses

200	OK	The contact has been updated
400	Bad Request	<i>key</i> could not be parsed

404	Not Found	The contact with the given identifier does not exist
-----	-----------	--

DELETE

Deletes the contact with the specified identifier.

Available responses

200	OK	The contact has been deleted
400	Bad Request	key could not be parsed
404	Not Found	The contact with the given identifier does not exist

/contacts/{key}/fields

Parameters

parameter	value	description
key	long	unique identifier for a contact

Methods

GET

Retrieves profile field data for the contact with the specified identifier.

Available responses

200	OK	
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The contact with the given identifier does not exist

PUT

Updates details for the contact with the specified identifier.

Available responses

200	OK	The contact has been updated
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The contact with the given identifier does not exist

DELETE

Deletes the contact with the specified identifier.

Available responses

200	OK	The contact has been deleted
400	Bad Request	key could not be parsed
404	Not Found	The contact with the given identifier does not exist

**/contacts?firstname&lastname&telephone&mobile&fax&sms&pager&email
&category&groupid&since&startindex&maxrows**

Methods

GET

Retrieves a collection of contacts, optionally matching one or more search criteria as specified in the query string.

This method will return the 500 most recently created or updated contacts if no query parameters are provided. Use *maxrows* to specify the maximum number of contacts which should be returned.

request query parameters

parameter	value	description
firstname	string	Return contacts with a first name which matches <i>firstname</i>
lastname	string	Return contacts with a last name which matches <i>lastname</i>
telephone	string	Return contacts who have a telephone number which matches <i>telephone</i>
mobile	string	Return contacts who have a mobile number which matches <i>mobile</i>
fax	string	Return contacts who have a fax number which matches <i>fax</i>
sms	string	Return contacts who have a sms number which matches <i>sms</i>
pager	string	Return contacts who have a pager number which matches <i>pager</i>
email	string	Return contacts who have an email address which matches <i>email</i>
category	string	Return contacts who have opted in to <i>category</i> type messages
groupid	long	Return contacts who are members of the group with the specified <i>groupid</i>
since	datetime	Return only contacts which were added or updated in Horizon since the given date and time
startindex	long	Used to control the starting point of the collection returned

maxrows **integer** Limit the number of contacts returned by the query to *maxrows*.

Note: The API will never return more than 2,500 contacts in a single request

Available responses

200	OK	One or more contacts matching the criteria were returned
204	No Content	Returned if no contacts were found which matched the criteria

POST

Adds a new contact.

Available responses

200	OK	A new contact has been created. The contact details returned will include the unique contact identifier assigned to this contact
400	Bad Request	It was not possible to create the contact because the data supplied did not meet one or more validation rules (e.g. no value was supplied for a required field)

/groups?name&type

Methods

GET

Retrieves a collection of groups, optionally matching one or more search criteria as specified in the query string.

request query parameters

parameter	value	description
name	string	Return groups with a name which matches <i>name</i>
type	string	Return groups of the given <i>type</i> . The type should be one of <ul style="list-style-type: none">• list• rule• map• combined• variant

Available responses

200	OK	One or more groups matching the criteria were returned
204	No Content	Returned if no groups were found which matched the criteria
400	Bad Request	Invalid type

/interactions/{key}

Parameters

parameter	value	description
key	guid	unique identifier for an interaction

Methods

GET

Retrieves details of the interaction with the specified identifier.

Available responses

200	OK	
400	Bad Request	<i>key</i> could not be parsed
404	Not Found	The interaction with the given identifier does not exist

/interactions?type&interfacecustomer&interfacehtk&since&to&maxrows

Methods

GET

Retrieves a collection of interactions, optionally matching one or more search criteria as specified in the query string.

This method will return the 500 most recent interactions if no query parameters are provided. Use *maxrows* to specify the maximum number of interactions which should be returned.

The interfaces (i.e. *interfacecustomer* and *interfacehtk*) detail the media values (such as telephone number or email address) of the parties involved in the interaction.

request query parameters

parameter	value	description
type	string	Return interactions of the given <i>type</i> . Type should be one of <ul style="list-style-type: none">• email• fax• pager• sms• voice
interfacecustomer	string	The customer media interface
interfacehtk	string	The HTK media interface
since	datetime	Return only interactions which were created since the given date and time
to	datetime	Return only interactions which were created before the given date and time
maxrows	integer	Limit the number of interactions returned by the query to <i>maxrows</i> . Note: The API will never return more than 2,500 interactions in a single request

Available responses

200	OK	One or more interactions matching the criteria were returned
204	No Content	Returned if no interactions were found which matched the criteria
400	Bad Request	Invalid <i>type</i> , <i>from</i> , or <i>to</i>

POST

Initiates a new interaction.

Available responses

200	OK	A new interaction has been initiated. The details returned will include the unique identifier for this interaction
400	Bad Request	It was not possible to initiate the interaction because the data supplied did not meet one or more validation rules (e.g. no value was supplied for a required field)

/sendmessage

POST

Sends an SMS message to the defined recipients

Available responses

201	OK	The send message request has been accepted and an attempt will be made to send the message to the defined phone numbers.
400	Bad Request	It was not possible to request the message send because the data supplied did not meet one or more validation rules. E.g. too many recipients defined or that the from field was not a valid interface.
429	Too Many Requests	This response code is returned when the daily limit for send message requests has been hit for the requesting user.
460	Contact not opted in	This will be returned if a contact with that media value is not opted in.

/offers

POST

Adds a new Offer or Reward. The unique offer identifier assigned to this offer is returned.

request query parameters

parameter	value	description
offerName	string	Offer Name
availableForRecommendation	boolean	Controls if Offer is available for presenting to Contacts
offerType	string	Available values: PERSONALIZED_OFFER; TIERED_REWARD
externalOfferId	string	An id for the Offer derived from an external system, optional
merchant	string	Merchant or Supplier name, optional
startDate	datetime	Start date and time of the availability of the Offer, mandatory if Offer is available for Recommendation
endDate	datetime	End date and time of the availability of the Offer, optional
stockLevel	integer	Starting stock level
offerTags	string	Set of Recommendation Tags that are associated with this Offer
content - shortDescription	string	Short display text.
content - longDescription	string	Long display text.
imageURLs	string	Up to three URLs to images for display.
questions	string	Up to three questions to ask after the user accepts the offer

Available responses

201	OK	The create Offer request has been accepted and the record created.
400	Bad Request	It was not possible to request the message send because the data supplied did not meet one or more validation rules.

PUT

Updates an Offer or Reward. All the Offer values are replaced by the values included in the request. The parameters are the same as for the POST method, with the addition of offer_id to identify the offer to be updated.

request query parameters

parameter	value	description
offer_id	integer	Unique identifier for the offer

Available responses

201	OK	The update Offer request has been accepted and the record updated.
400	Bad Request	It was not possible to request the message send because the data supplied did not meet one or more validation rules.

URIs

The base URI for the Horizon API is

<https://api.htkhorizon.com/horizon-api/r>

Use the path fragments as documented in the resource definitions to create a fully qualified path to a resource.

URI examples

URI	Method	Description
https://api.htkhorizon.com/horizon-api/r/contacts	GET	Return the 500 most recently created or updated contacts
https://api.htkhorizon.com/horizon-api/r/contacts	POST	Create a new contact using the details provided
https://api.htkhorizon.com/horizon-api/r/contacts/12345/locations	PUT	Updates the locations for the contact with 12345 as its unique identifier

Security

The Horizon API is secured using OAuth 2.0.

Creating and managing your application in Horizon

Applications need to be configured in the Horizon UI⁴ to enable access to the Horizon API (see also the section on Security).

Disclaimer

HTK reserves the right to make changes to the Horizon API at any time. Any changes will be notified to customers and made available immediately. Previous versions of the API will remain available for at least six months after notice of a change is given, allowing time for Horizon API customers to make any necessary application modifications.

⁴ The application configuration section in Horizon is only available to administrators with a Horizon API License.