

API Portal - Developer Guide

Draft : 20210421 (ver 1.1)

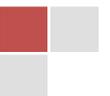


TABLE OF CONTENTS

1. Welcome!.....	1
2. Apply developer account.....	2
3. Create Developer Application.....	6
4. Subscribe API for the Application	9
5. Generate access token for the application	13
6. Learn and test APIs at API Console	16
7. CN region and Global region differences.....	23
8. API Integration to your own application	24
9. Certification	26

1. WELCOME!

Welcome to Infrasy Cloud Partner API Portal!

You can find that this is an easy way to extend the functionality of your system via the partner API Portal.

From there, we provide the RESTful APIs for the developer integration.

Please follow the steps and start your API development.

- Apply Developer Account for Infrasy Cloud API Portal**
- Create Developer Application
- Subscribe APIs for the Application (to get sandbox access key)
- Learn and test APIs at API Console (with sandbox)
- API Integration to your own application
- Certification (to get production access key)**

*** Application Form and approval process by Infrasy API Portal Admin Team is required*

After Verification in Certification, you will receive the keys to access Infrasy Cloud production environment and your application will be put into our feature market place for Infrasy Cloud customer subscription.

At that moment, your application can work with us to provide the world class service to our customers.

Let's start now!

2. APPLY DEVELOPER ACCOUNT

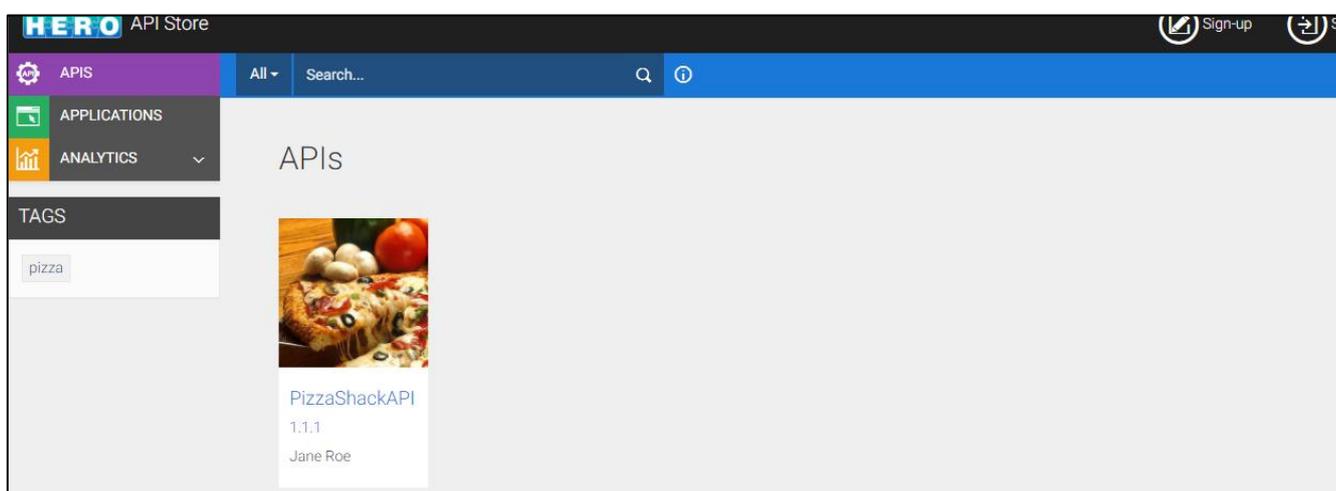
Infrasys Cloud API Portal is for the partner development only. Therefore, the developer should receive the invitation before submitting your account application.

For the API list that should be subscribed by your application, you must inform Infrasys Business Unit prior to raising your application-API subscription. Please make sure you have submitted a registration form provided by Infrasys Business Unit.

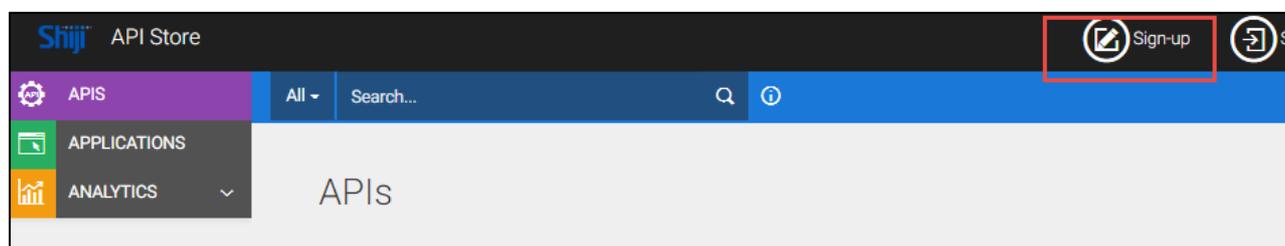
Step 1: Go to Infrasys Cloud Developer Portal

CN region: <https://api.hero-cloud.cn:9443/store>

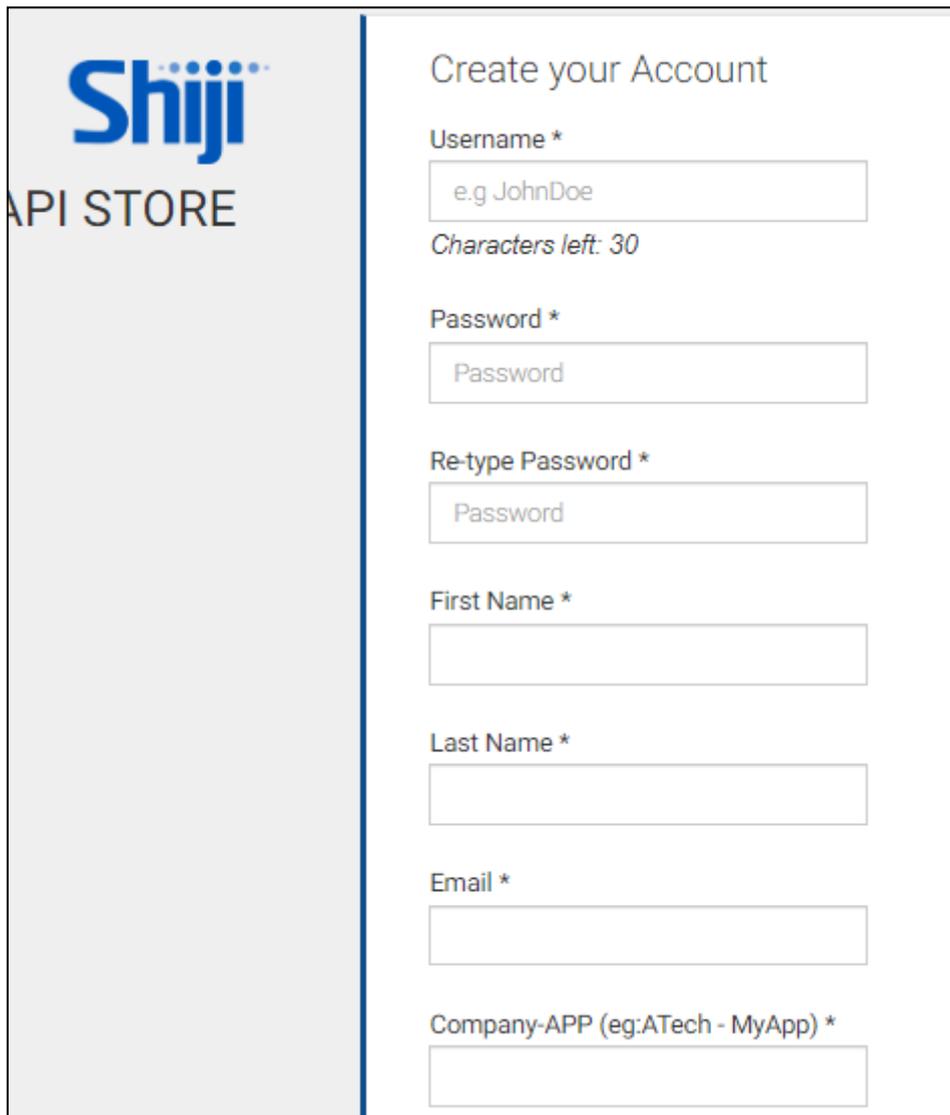
Global region: <https://api.hero-cloud.com:9443/store>



Step 2: Click "Sign-up" button at the top-right corner



Step 3: Input Developer Information.



Shiji
API STORE

Create your Account

Username *

Characters left: 30

Password *

Re-type Password *

First Name *

Last Name *

Email *

Company-APP (eg:ATech - MyApp) *

Step 4: Please enter your company and application name in “Company-APP” field. This will be the application name shown in market place for user selection.

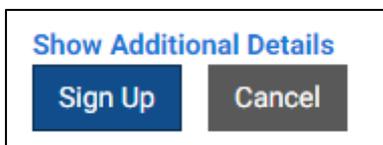
e.g.

Company name : Future Technology

Application : Butterfly App

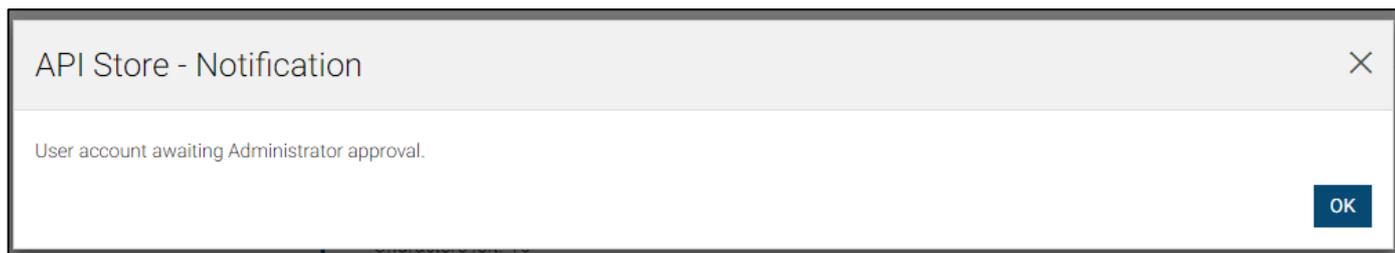
Company-APP: “Future Technology – Butterfly App”

If Company-APP Field is hidden, please press “Show Additional Details” to show the field.

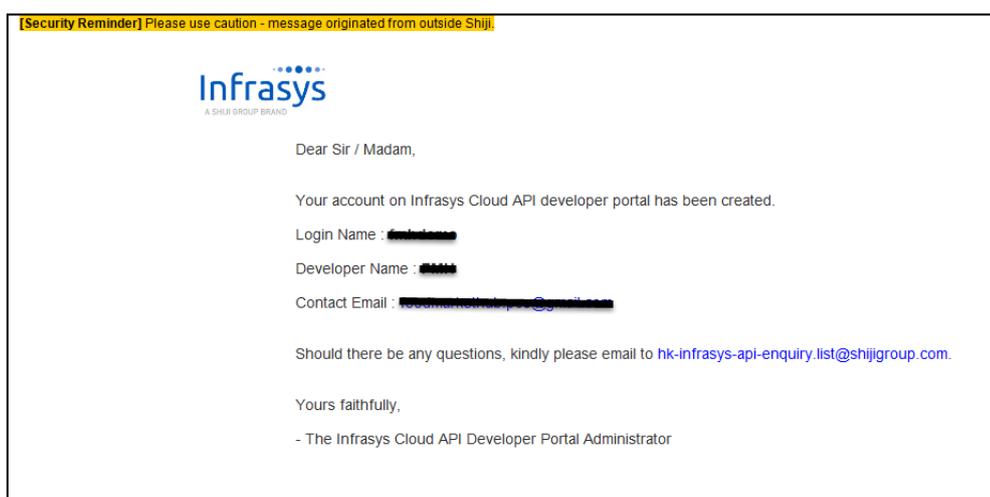


[Show Additional Details](#)

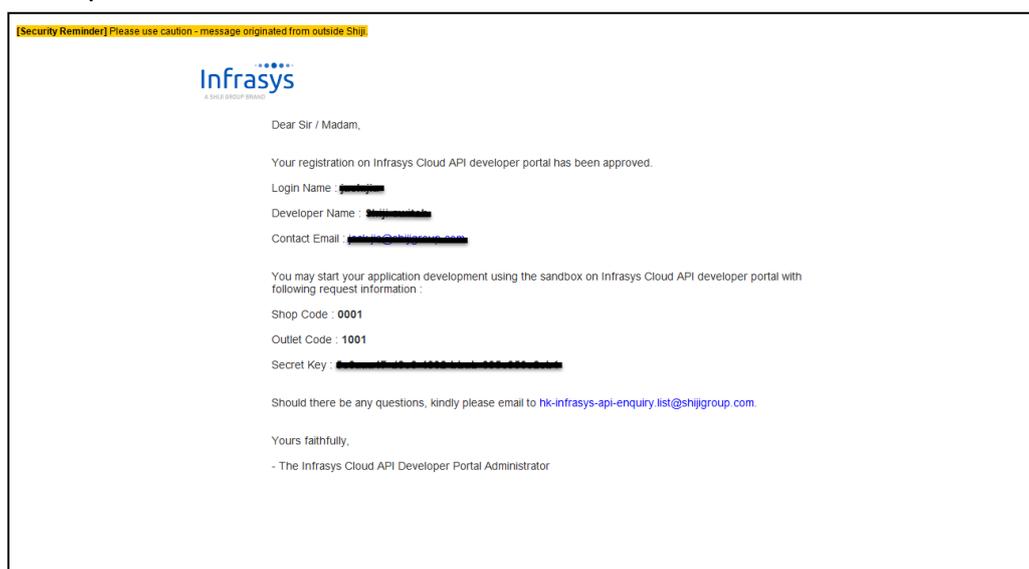
Step 5: Press “Sign Up” button to submit the application. The below message will be shown for success signup.



Please check your mail box. You should receive an Account New Registration email from our API Portal. **Please note the application process is not completed yet!**



After the account process by API operation team, a confirmation email will be issued to the Developer email. The secret key of sandbox is attached.

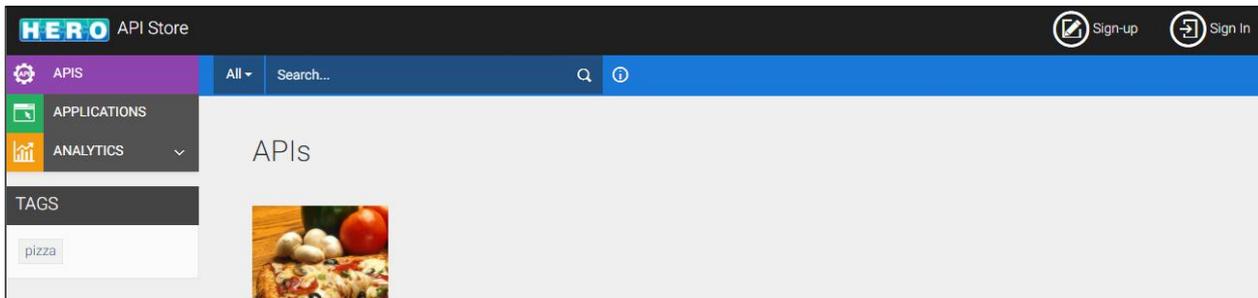


You can “Sign In” to the API Portal now!

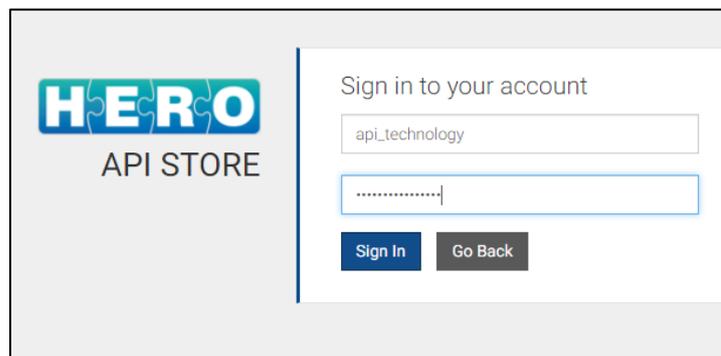
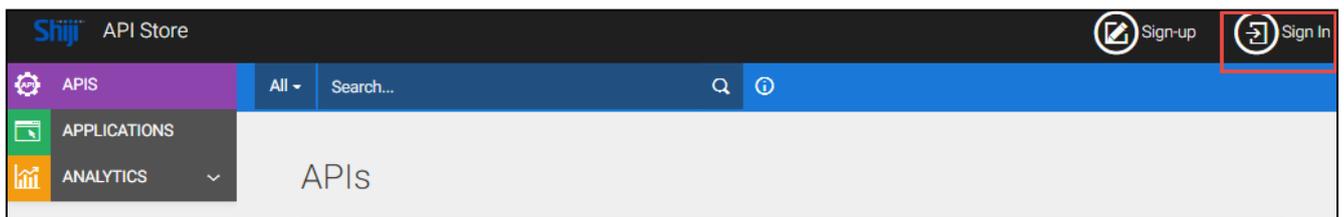
Step 1: Open the Infrasy Cloud Developer Portal

CN region: <https://api.hero-cloud.cn:9443/store>

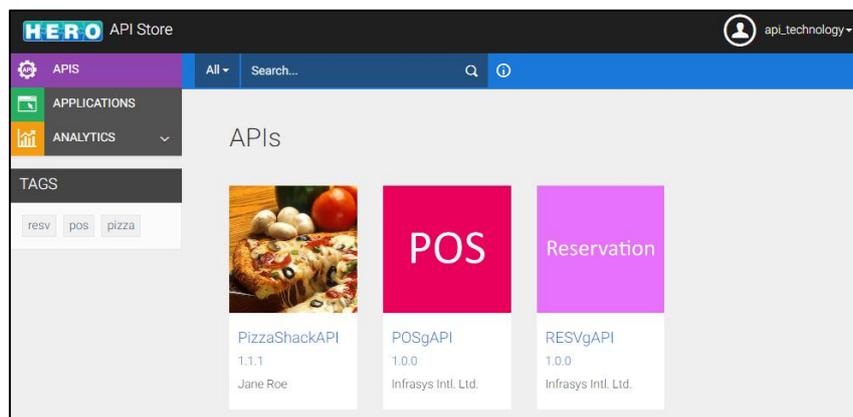
Global region: <https://api.hero-cloud.com:9443/store>



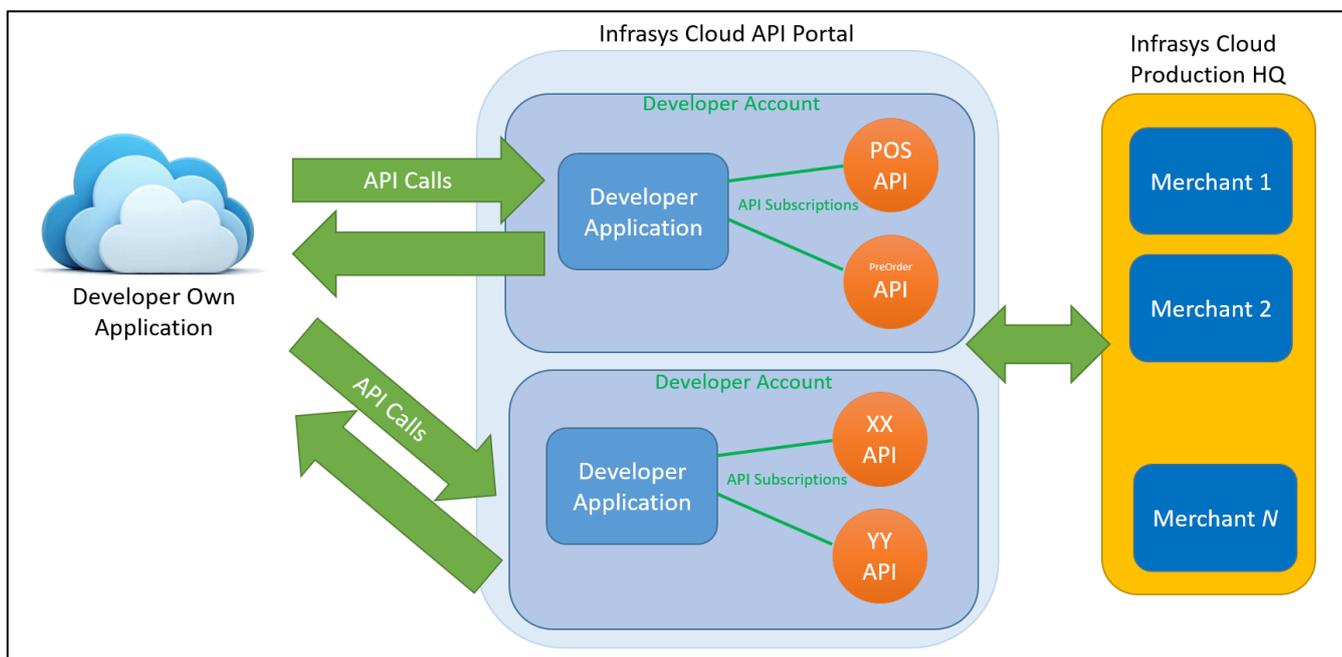
Step 2: Press “Sign In” button at the top-right corner and Type in your Username and Password



You will get your API working space.



3. CREATE DEVELOPER APPLICATION



Step 1: Login to the Developer Portal. Press “APPLICATIONS” button at the navigation bar.

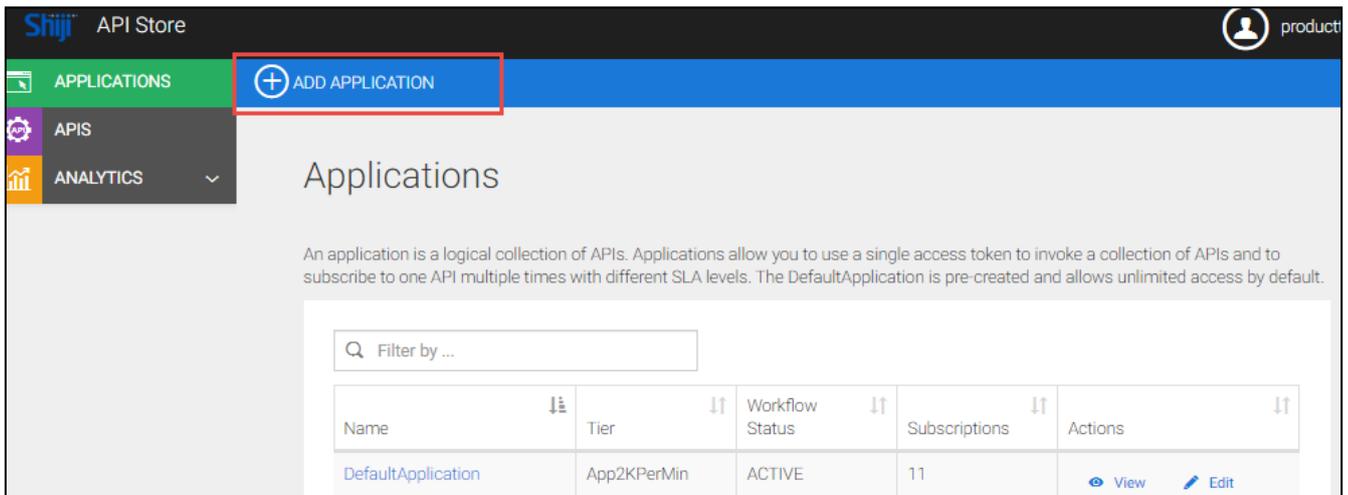
The screenshot shows the 'Applications' page in the developer portal. The navigation bar includes 'APPLICATIONS' (selected), 'ADD APPLICATION', 'APIS', and 'ANALYTICS'. The main content area has a heading 'Applications' and a descriptive paragraph: 'An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels. The DefaultApplication is pre-created and allows unlimited access by default.'

Below the text is a table with a search filter and one entry:

Name	Tier	Workflow Status	Subscriptions	Actions
DefaultApplication	Unlimited	ACTIVE	0	View Edit Delete

At the bottom, there is a 'Show 10 entries' dropdown and 'Showing 1 to 1 of 1 entries' text.

Step 2: Press “ADD APPLICATION” button.



Step 3: Fill in the details of the application and press “Add” button to complete the action

- Name : Application Name (To be shown for customer subscription)
****Please note the application name should not duplicate with “Company-APP” in sign-up**
- Per Token Quota : Select the plan for the API traffic control
- Description : Information for your application

Add Application

An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels. The DefaultApplication is pre-created and allows unlimited access by default.

Name*
Characters left: 57

Per Token Quota *Allows 50 request per minute*
 This feature allows you to assign an API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Description

The application added and details will be shown under the Application List.

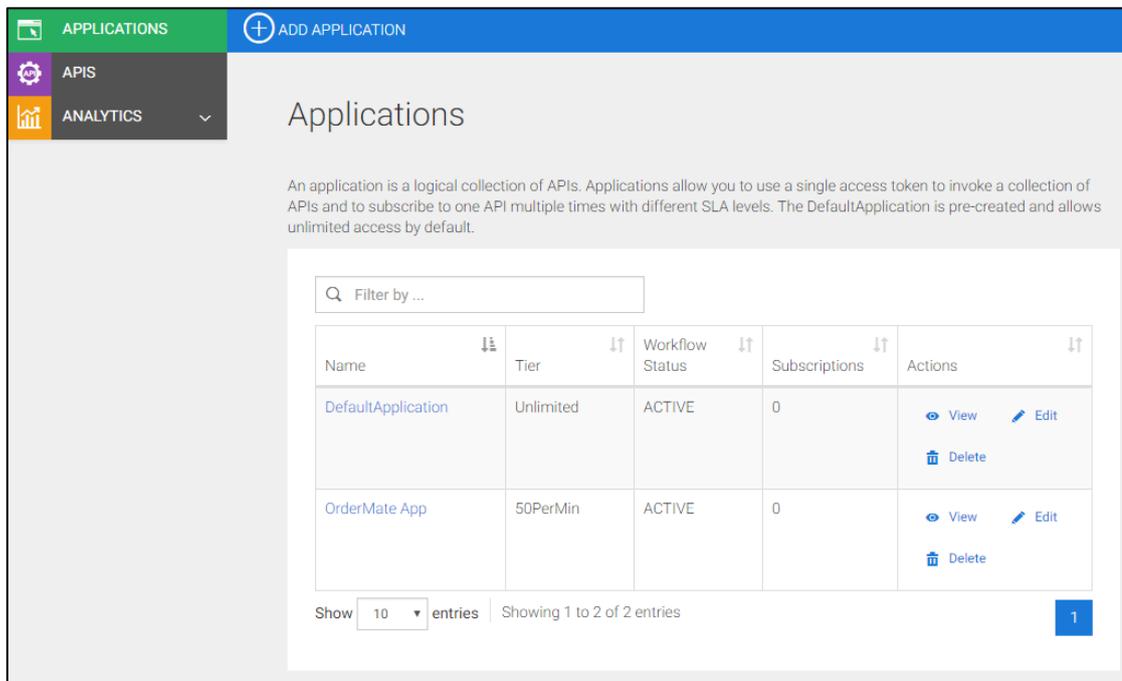
OrderMate App

- Details
- Production Keys
- Sandbox Keys
- Subscriptions

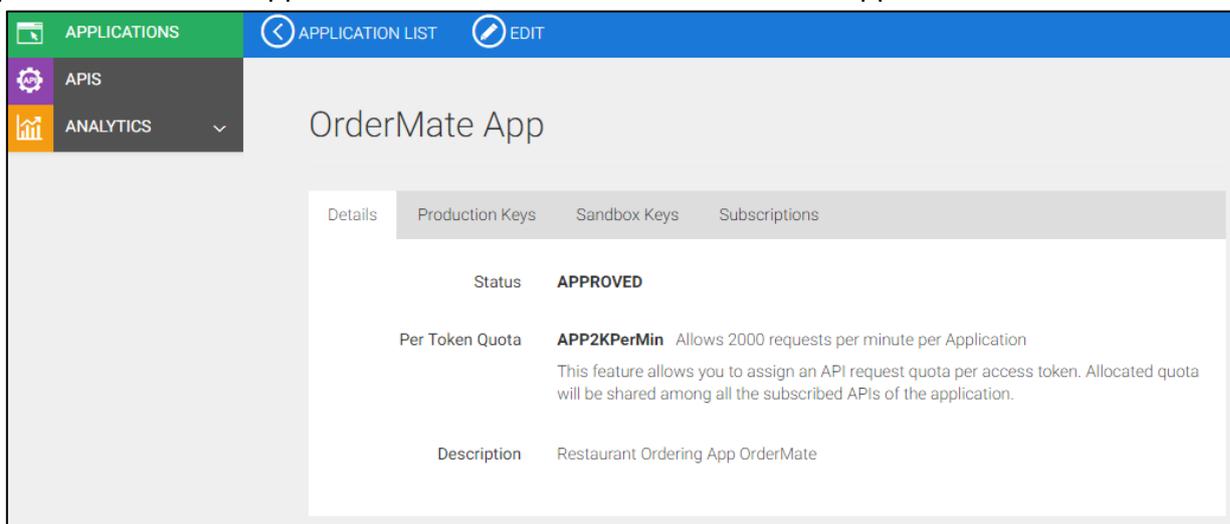
Status	APPROVED
Per Token Quota	50PerMin Allows 50 request per minute This feature allows you to assign an API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.
Description	Make restaurant item ordering by OrderMate App

4. SUBSCRIBE API FOR THE APPLICATION

Step 1: Login to the Developer Portal. Press “APPLICATIONS” button at the navigation bar.



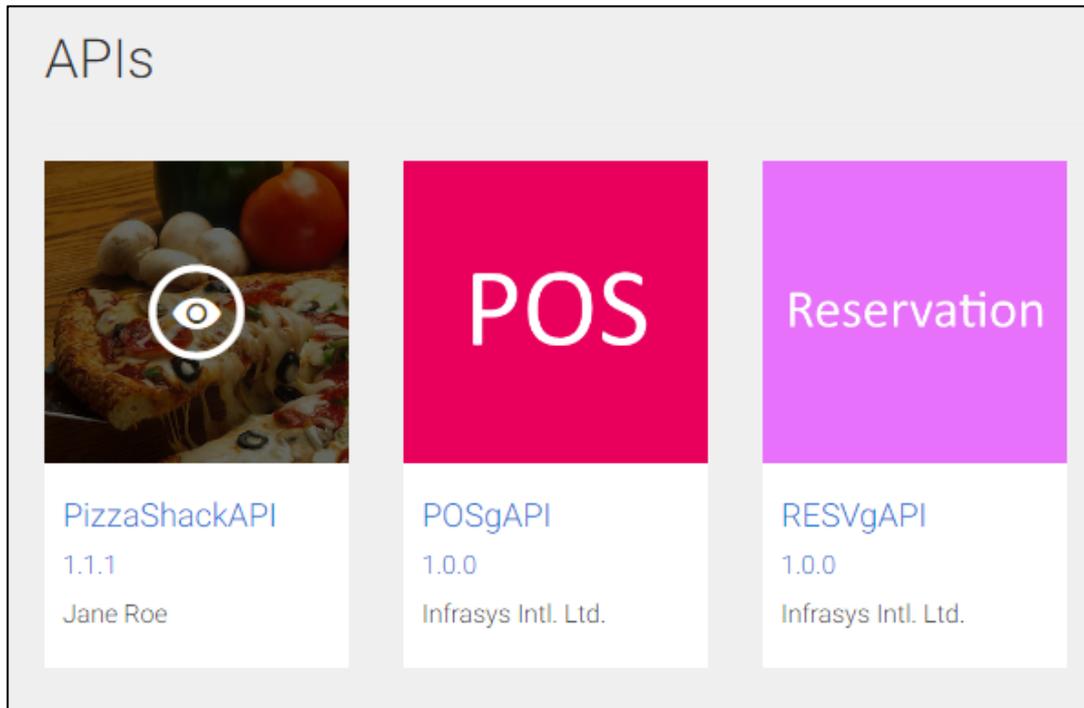
Step 2: Check the new application was created and included in the “Application List”



Step 3: Press “APIS” button at the navigation bar.



A list of arranged APIs for your account is shown on screen. If your desired API is not available, please contact our business unit to request more APIs.



Step 4: Click on the target API to view the details



Step 5: Please select the Applications to subscribe current API and select the Tier.

POSGAPI - 1.0.0

Version: 1.0.0
By: Infrasys Intl. Ltd.
Updated: 31/Aug/2017 08:27:57 AM UTC
Status: **PUBLISHED**

Applications
OrderMate App

Tiers
Bronze

Subscribe

Overview | API Console | Documentation | SDKs

Production and Sandbox Endpoints
Production and Sandbox URLs:
`https://[redacted]`

Description
This is a RESTful API for POS.

Business Information
Business Owner : Infrasys Intl. Ltd. [contact@infrasys-intl.com]
Technical Owner : HERO API Team [apiteam@hero-cloud.com]

Applications

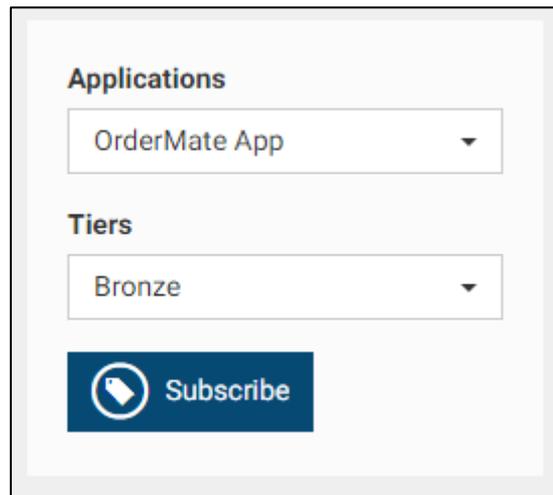
- DefaultApplication
- New Application...
- My Applications
- DefaultApplication
- OrderMate App**

Applications
OrderMate App

Tiers
Bronze

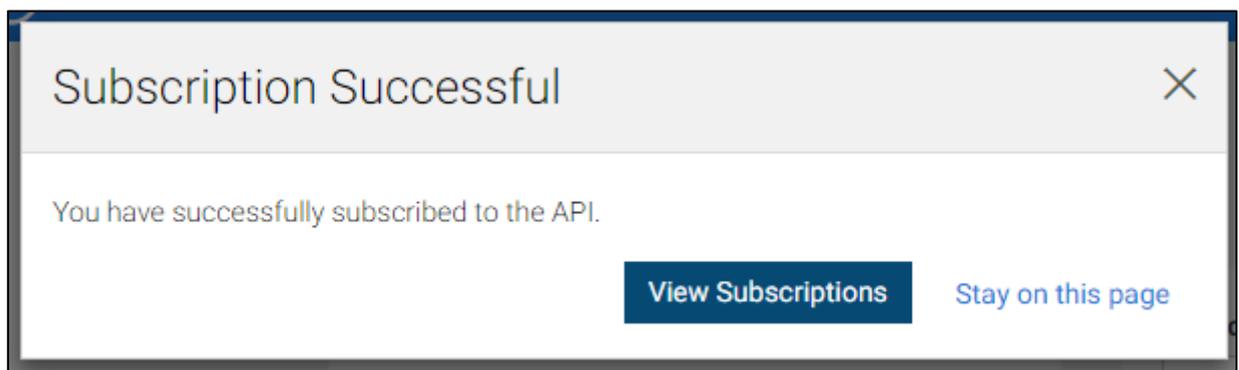
Bronze
Allows 50 requests per minute

Step 6: Then Press “Subscribe” button to confirm the Application-API subscription.



The form contains two dropdown menus. The first is labeled "Applications" and has "OrderMate App" selected. The second is labeled "Tiers" and has "Bronze" selected. Below the dropdowns is a blue button with a white play icon and the text "Subscribe".

The subscription application is submitted and approved instantly.



5. GENERATE ACCESS TOKEN FOR THE APPLICATION

Step 1: Login to the Developer Portal. Press “APPLICATIONS” button at the navigation bar.

The screenshot shows the 'Applications' page in the Shiji Developer Portal. The navigation bar includes 'APPLICATIONS' (selected), 'APIS', and 'ANALYTICS'. The main heading is 'Applications'. Below the heading is a descriptive paragraph: 'An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels. The DefaultApplication is pre-created and allows unlimited access by default.' Below this is a search filter 'Filter by ...'. A table lists the applications:

Name	Tier	Workflow Status	Subscriptions	Actions
DefaultApplication	APP2KPerMin	ACTIVE	0	View, Edit, Delete
OrderMate App	APP2KPerMin	ACTIVE	1	View, Edit, Delete

At the bottom, there are pagination controls: 'Show 10 entries | Showing 1 to 2 of 2 entries' and a page number '1'.

Step 2: Click on particular application

The screenshot shows the 'OrderMate App' details page. The page has a title 'OrderMate App' and a navigation bar with tabs: 'Details' (selected), 'Production Keys', 'Sandbox Keys', and 'Subscriptions'. The main content area displays the following details:

- Status:** APPROVED
- Per Token Quota:** APP2KPerMin. Allows 2000 requests per minute per Application. This feature allows you to assign an API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.
- Description:** Restaurant Ordering App OrderMate

Step 3: Generate both "Production Keys" and "Sandbox Keys". Press "Generate keys" to generate the keys.

Details
Production Keys
Sandbox Keys
Subscriptions

i **No Keys Found**

No keys are generated for this type in this application.

Grant Types

The application can use the following grant types to generate Access Tokens. Based on the application requirement, you can enable or disable grant types for this application.

Refresh Token

SAML2

Implicit

Password

IWA-NTLM

Client Credential

Code

Callback URL

CN region

Access token is by module. Please select the API module and click generate.

```
curl -k -d "grant_type=client_credentials" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
```

POS_LABOUR_MANAGEMENT_API : POS_LABOUR_MANAGEMENT_API.	✓
PREORDER_API_v1_0_0 : PREORDER_API_v1_0_0.	✓
KD_API_v_1_0_0 : KD_API_v_1_0_0.	✓
RESERVATION_API_v1_1_1 : RESERVATION_API_v1_1_1.	✓
PMS_API_v1_0_0 : PMS_API_v1_0_0.	✓
POS_API_v3_5_0 : POS_API_v3_5_0.	✓

POS_LABOUR_MANAGEMENT_API, PREORDER_API_v1_0_0, KD_API_v_1_0_0, RESERVATION_API_v1_1_1, PMS_API_v1_0_0, POS_API_v3_5_0

Validity period

Seconds

Regenerate

For generate access token automatically, please use the below cURL command.

Generating Access Tokens

The following cURL command shows how to generate an access token using the Password Grant type.

```
curl -k -d "grant_type=password&username=Username&password=Password&scope=Scope1 Scope2" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.hero-cloud.cn:8243/token
```

In a similar manner, you can generate an access token using the Client Credentials grant type with the following cURL command.

```
curl -k -d "grant_type=client_credentials&scope=Scope1 Scope2" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.hero-cloud.cn:8243/token
```

Global region

Access token is by application. Click Generate to get the access token.

Generate a Test Access Token

Access Token

.....
📄

Above token has a validity period of **360000** seconds.

Scopes

No Scopes Found..
▾

Validity period

Seconds

Regenerate

For generate access token automatically, please use the below cURL command.

Generating Access Tokens

The following cURL command shows how to generate an access token using the Password Grant type.

```
curl -k -d "grant_type=password&username=Username&password=Password" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.hero-cloud.com:8243/token
```

In a similar manner, you can generate an access token using the Client Credential grant type with the following cURL command.

```
curl -k -d "grant_type=client_credentials" \
-H "Authorization: Basic Base64(consumer-key:consumer-secret)" \
https://api.hero-cloud.com:8243/token
```

Keys are regenerated (Consumer Key & Consumer Secret)

Consumer Key

.....
📄

Consumer Secret

.....
📄

6. LEARN AND TEST APIS AT API CONSOLE

Step 4: Click on “Subscriptions” tab

Details
Production Keys
Sandbox Keys
Subscriptions

API Name	Subscription Tier	Status	Actions
<div style="display: flex; align-items: center;"> <div style="background-color: #e91e63; color: white; padding: 5px 10px; font-weight: bold; margin-right: 5px;">POS</div> <div> <p>POSGAPI - 1.0.0</p> <p><small>🔒 PUBLISHED</small></p> </div> </div>	Bronze	UNBLOCKED	🗑 Unsubscribe

Show entries
Showing 1 to 1 of 1 entries
1

Step 5: Select the API to open the API page → Click on “API Console” tab

Overview
API Console
Documentation
SDKs

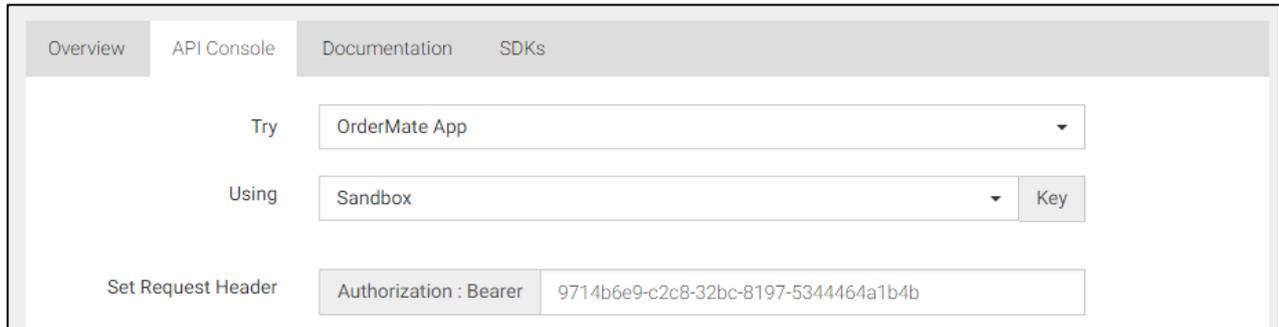
Overview
API Console
Documentation
SDKs

Try

Using Key

Set Request Header Authorization : Bearer

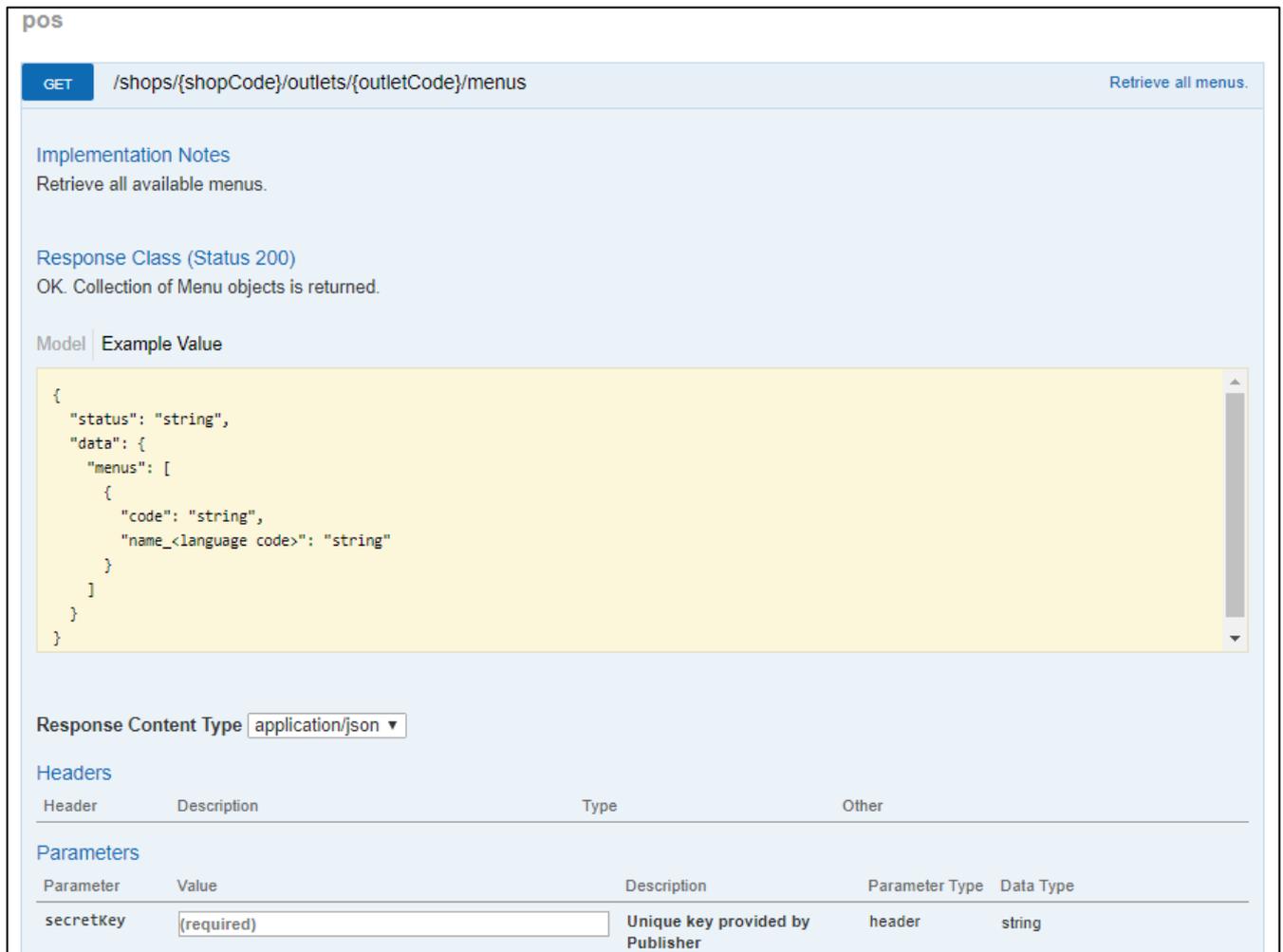
Step 6: Try the API with Sandbox: Change “Using” to “Sandbox”.



The screenshot shows the API console configuration interface. At the top, there are tabs for Overview, API Console, Documentation, and SDKs. Below the tabs, there are three main sections:

- Try:** A dropdown menu currently set to "OrderMate App".
- Using:** A dropdown menu set to "Sandbox" with a "Key" button next to it.
- Set Request Header:** A section with "Authorization : Bearer" and a text input field containing the token "9714b6e9-c2c8-32bc-8197-5344464a1b4b".

Step 7: Click on each API link to view API details



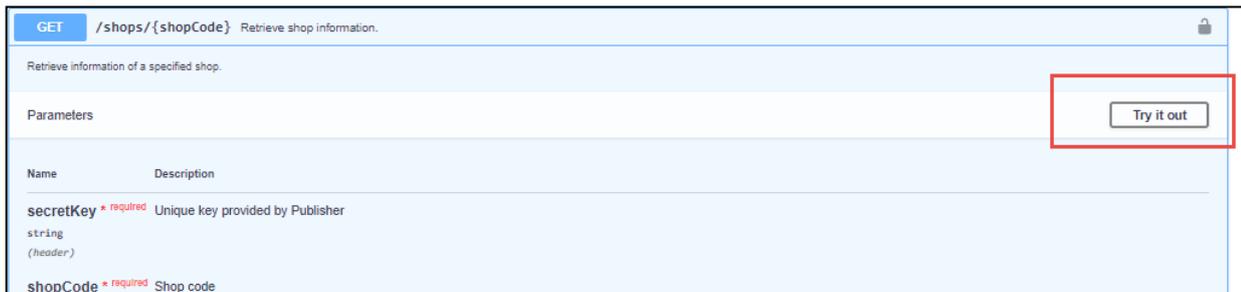
The screenshot shows the API details page for the endpoint `/shops/{shopCode}/outlets/{outletCode}/menus`. The page is titled "pos" and includes the following sections:

- GET /shops/{shopCode}/outlets/{outletCode}/menus** with a link to "Retrieve all menus."
- Implementation Notes:** Retrieve all available menus.
- Response Class (Status 200):** OK. Collection of Menu objects is returned.
- Model | Example Value:** A JSON example value is displayed in a scrollable area:


```
{
  "status": "string",
  "data": {
    "menus": [
      {
        "code": "string",
        "name_<language code>": "string"
      }
    ]
  }
}
```
- Response Content Type:** A dropdown menu set to "application/json".
- Headers:** A table with columns: Header, Description, Type, Other.
- Parameters:** A table with columns: Parameter, Value, Description, Parameter Type, Data Type.

Parameter	Value	Description	Parameter Type	Data Type
secretKey	(required)	Unique key provided by Publisher	header	string

For CN region, press “Try it out”.



GET /shops/{shopCode} Retrieve shop information.

Retrieve information of a specified shop.

Parameters

Name	Description
secretKey * required	Unique key provided by Publisher
string (header)	
shopCode * required	Shop code

Step 8: Fill in the API parameters

Parameter	Value	Description	Parameter Type	Data Type
secretKey	(required)	Unique key provided by Publisher	header	string
shopCode	(required)	Shop code	path	string
outletCode	(required)	Outlet code	path	string
languages	(required)	Define the languages of result to get. It is a list of language codes which separated by commas. E.g. zh-Hant,ja,ko. The language codes follow the standard of ISO 639-1 and the supported languages are - en, zh-Hant, zh-hk, zh-Hans, ja, ko, th and fr.	query	string

Click “Model” for the parameter detail.

Parameter	Value	Description	Parameter Type	Data Type
secretKey	<input type="text" value="(required)"/>	Unique key provided by Publisher	header	string
shopCode	<input type="text" value="(required)"/>	Shop code	path	string
outletCode	<input type="text" value="(required)"/>	Outlet code	path	string
check	<input type="text" value="(required)"/>	Check details	body	Model Example Value

Parameter content type:

```

CreateCheck {
  table_no (string, optional): The table no.,
  cover (integer): The cover no.,
  price_level (integer, optional): Target price level for all it
  (start from 0),
  meal_period (string, optional): The meal period code,
  items (Array[Item]): The list of items,
  discounts (Array[Discount], optional): The list of discount
  
```

Click "Example Value" for parameter template. And click on the content, the example will be available for use.

Parameter	Value	Description	Parameter Type	Data Type
secretKey	<input type="text" value="(required)"/>	Unique key provided by Publisher	header	string
shopCode	<input type="text" value="(required)"/>	Shop code	path	string
outletCode	<input type="text" value="(required)"/>	Outlet code	path	string
check	<pre>{ "table_no": "string", "cover": 0, "price_level": 0, "meal_period": "string", "items": [/</pre>	Check details	body	Model Example Value

Parameter content type:

```

{
  "table_no": "string",
  "cover": 0,
  "price_level": 0,
  "meal_period": "string",
  "items": [
    {
      "code": "string",
      "qty": 0,
      "open_desc": "string",
      "open_price": "string",
      "modifiers": [

```

Step 9: For Global region, press "Try it out!" button to submit the API test request.

For CN region, press "Execute" button to submit the API test request.

string (path)

languages * required Define the languages of result to get. It is a list of language codes which separated by commas. E.g. zh-Hant,ja,ko. The language codes follow the standard of ISO 639-1 and the supported languages are - en, zh-Hant, zh-hk, zh-Hans, ja, ko, th and fr.

string (query)

Execute

The response message is shown below.

Try it out!
Hide Response

Curl

```
curl -X GET --header 'Accept: application/json' --header 'secretkey: 597eee44-██████████-1516c0a86340' --header 'Authorization: Bearer ██████████' 'https://api.hero-cloud.com:8243/pos/1.0.0/shops/0001/outlets/1001/menus?languages=en'
```

Request URL

```
https://██████████/pos/1.0.0/shops/0001/outlets/1001/menus?languages=en
```

Request Headers

```
{
  "Accept": "application/json",
  "secretkey": "597eee44-██████████-1516c0a86340"
}
```

Response Body

```
{
  "status": "success",
  "data": {
    "menus": [
      {
        "code": "dinner",
        "name_en": "Dinner"
      }
    ]
  }
}
```

Response Code

```
200
```

Response Headers

```
{
  "content-type": "application/json"
}
```

The response message model and example is available under “Response Class”.

POST
/shops/{shopCode}/outlets/{outletCode}/checks

Implementation Notes
Create a new check and return check no. upon success.

Response Class (Status 200)
OK. Check is created.

Model Example Value

```
Send Check Result {
  status (string, optional),
  data (SendCheckResult, optional)
}
SendCheckResult {
  check_no (string): The check no.
}
```

Secret Key and API Specific Parameters

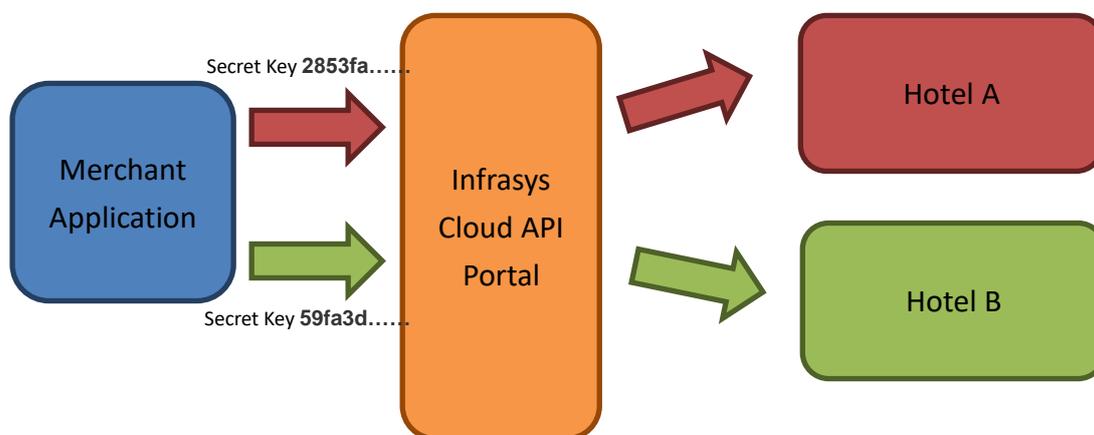
When you need to perform the testing, you need to put the Secret Key and the specific parameters to the API Console or your own testing tools.

“Secret Key” is the key granted to you to access the specific data scope.

e.g. You can use the specific Secret Key to access all outlets of “Lake Hotel”.

or You can use another specific Secret Key to access “Grill Room” and “In-Room Dining” only in “Sky Hotel”.

Therefore you can use specific Secret Key in the api to access the particular outlets.



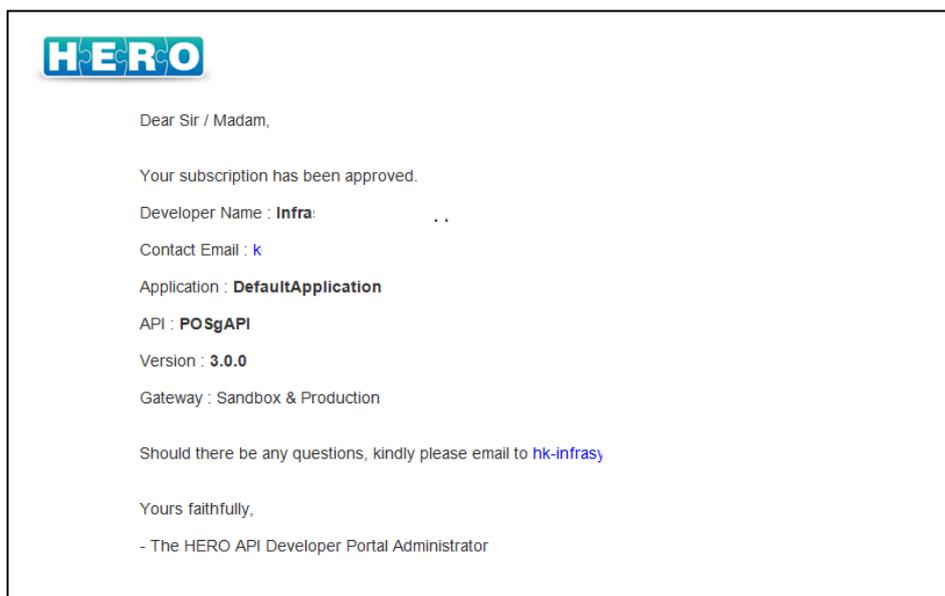
When will you get the “Secret Key”?

When your application is subscribed by any Merchant in our API Application Market, you will receive an email showing you the “Secret Key” and the outlet information. Then you can use those information in your application to access the resources through our API Portal.

First “Secret Key” you received is the one for you to access the sandbox environment for your testing. You should receive it after your “User Account” registered successfully.



Sample email after “Application Subscription” by Merchant



7. CN REGION AND GLOBAL REGION DIFFERENCES

	CN region	Global AWS
URL	https://api.hero-cloud.cn:9443/store	https://api.hero-cloud.com:9443/store
Extra Grant Types	JWT	
Access Tokens Generation - Password Grant type	<pre>curl -k -d "grant_type=password&username=Username&password=Password&scope=Scope1 Scope2" -H "Authorization: Basic aTJ0OUU1XzFBUzJoXzNSZ2ZWU2NPdEYwczN3YTpCSjhPdFRxX3pqQk5TS1U0X0VjY3B6c0ZLZ0Vh" https://api.hero-cloud.cn:8243/token</pre>	<pre>curl -k -d "grant_type=password&username=Username&password=Password" -H "Authorization: Basic ZU1faTYzSThRdGN0NFM2Qk5wZGFyVUJ3ZXpFYTpCZE1mc3NDN1lwcjdvZUITRmk4ekhIU0JwZ0lh" https://api.hero-cloud.com:8243/token</pre>
Access Tokens Generation -Client Credential grant type	<pre>curl -k -d "grant_type=client_credentials&scope=Scope1 Scope2" -H "Authorization: Basic aTJ0OUU1XzFBUzJoXzNSZ2ZWU2NPdEYwczN3YTpCSjhPdFRxX3pqQk5TS1U0X0VjY3B6c0ZLZ0Vh" https://api.hero-cloud.cn:8243/token</pre>	<pre>curl -k -d "grant_type=client_credentials" -H "Authorization: Basic ZU1faTYzSThRdGN0NFM2Qk5wZGFyVUJ3ZXpFYTpCZE1mc3NDN1lwcjdvZUITRmk4ekhIU0JwZ0lh" https://api.hero-cloud.com:8243/token</pre>
Scopes	By Module	By Application
Request URL	https://api.hero-cloud.cn:8243/	https://api.hero-cloud.com:8243

8. API INTEGRATION TO YOUR OWN APPLICATION

You are great to reach this section! You are going to integrate this API with your own application. You can use the links shown in the API Console for your application testing use.

Hints for your integration

- Prepare Secret Key and API Specific Attributes (e.g. Shop code, Outlet code)
- Prepare the “Consumer Key” and “Consumer Secret”
- Generate “Access Token”
 - It must be included in each API call.
 - Token has a validity of 3600 second. Before expiry, you can use the same Token for the API calls.
 - Generation Methods
 - Using curl as described in developer portal
 - Using HTTP POST with following configurations:
Request Url: \$api_portal_url/token

HTTP headers:

Name	Value	Explanation
Accept	application/json	
Authorization	Basic \$value	For the \$value: please use base64 to encode your consumer key and consumer secret using the following pattern: base64(consumerKey:consumerSecret)

HTTP POST Body:

```
grant_type=password&username=$devUserName&password=$devUserPw
```

\$devUesrName and \$devUserPw are the user login credentials in developer portal.

- Fire the HTTP POST request.
- The access token is generated. The response packet:

```
{"access_token":"8a7ee529-9c24-3198-a758-6142a8515efa",
"refresh_token":"95462565-a0bf-36b3-a054-ba4375ce123c",
"scope":"default","token_type":"Bearer","expires_in":3600}
```

- API Integration to your application
 - General HTTP headers required for all APIs:

Name	Value	Explanation
Accept	application/json	
secretKey	\$secretKey	For using sandbox, developers should get the secretKey from abovementioned registration approval email
Authorization	Bearer \$accessToken	Previously generated access token
Content-Type	application/json	

- Parameter Type

Example: \$api_portal_url/preorder/1.0.0/shops/**S001**/outlets/**O001**/menus?languages=en,zh-Hant

Parameter Type	Explanation	Example
header	HTTP Header	
path	RESTful format request url	shops/S001/ outlets/O001
query	HTTP request query	?languages=en,zh-Hant
body	HTTP request body	{ "items": [// item details ...] } }, "table_no": "2", // other required parameters ... }

Remarks:

Available shop codes and outlet codes can be extracted from each Merchant Subscription to your application

- Special parameter specification

Parameter	Format	Example
languages	ISO 639-1 Standard	en, zh-Hant, zh-Hans, ja, ko, th, fr
date	yyyy-MM-dd	2017-04-03
time	ISO 8601 Standard	2004-02-12T15:19:21+00:00

9. CERTIFICATION

After the integration, your application can integrate to our applications.

Please make the trial run for your application at the sandbox environment. After that, please extract the communication log and send to us (API Portal Admin) for verifications. Once it is passed, you will gain the Production environment access and start the API Integration to the real life customers.