**SỞ THÔNG TIN VÀ TRUYỀN THÔNG THÀNH PHỐ HỒ CHÍ MINH**

**--------------**

**MÔ HÌNH LIÊN THÔNG NỀN TẢNG HCM LGSP TPHCM**

**DỊCH VỤ CHIA SẺ**

**DỊCH VỤ CUNG CẤP DANH MỤC TỈNH THÀNH**

**TP.HCM, 8/2019**

**MỤC LỤC**

[THUẬT NGỮ VÀ TỪ VIẾT TẮT 2](#_Toc13485964)

[DANH MỤC HÌNH VẼ, BẢNG BIỂU 3](#_Toc13485965)

[I. TỔNG QUAN 4](#_Toc13485966)

[1.1. Mô tả nghiệp vụ 4](#_Toc13485967)

[1.2. Mục đích tài liệu 4](#_Toc13485968)

[1.3. Phạm vi và đối tượng áp dụng 4](#_Toc13485969)

[1.4. Thông tin kết nối 4](#_Toc13485970)

[II. MÔ HÌNH KẾT NỐI 5](#_Toc13485971)

[2.1. Mô hình kết nối lấy danh mục tỉnh thành 5](#_Toc13485972)

[2.2. Quá trình trao đổi gói tin 6](#_Toc13485973)

[III. ĐẶC TẢ API LẤY DANH MỤC TỈNH THÀNH 6](#_Toc13485974)

[3.1. Thông tin kết nối 6](#_Toc13485975)

[3.2. API lấy danh mục tỉnh thành 7](#_Toc13485976)

[3.3. Gói tin danh mục tỉnh thành 9](#_Toc13485977)

[IV. CODE EXAMPLE 9](#_Toc13485978)

[4.1. Code mẫu .Net 9](#_Toc13485979)

[4.2. Code mẫu java 10](#_Toc13485980)

[V. Tham khảo 12](#_Toc13485981)

# THUẬT NGỮ VÀ TỪ VIẾT TẮT

| STT | Thuật ngữ | Ý Nghĩa |
| --- | --- | --- |
|  | LGSP | Nền tảng tích hợp và chia sẻ dữ liệu |
|  | CSDL | Cơ sở dữ liệu |
|  | HCM | Thành phố Hồ Chí Minh |
|  | JSON | JavaScript Object Notation: một kiểu dữ liệu mở trong JavaScript |

# DANH MỤC HÌNH VẼ, BẢNG BIỂU

| STT | Hình vẽ/ Bảng | Ý nghĩa |
| --- | --- | --- |
|  | Hình 1 | Mô hình kết nối lấy bộ danh mục tỉnh thành |
|  | Hình 2 | Quy trình trao đổi gói tin |
|  | Bảng 1 | Thông tin kết nối |
|  | Bảng 2 | Tham số trả về của chuỗi json |
|  | Bảng 3 | Gói tin danh mục tỉnh thành |

# TỔNG QUAN

## Mô tả nghiệp vụ

* Phần mềm ISO điện tử khi có nhu cầu lấy bộ danh mục tỉnh thành thì sẽ gọi api lấy bộ danh mục tỉnh thành lên HCM LGSP (APILayDanhMucTinhThanh).
* HCM LGSP sẽ lấy thông tin bộ danh mục tỉnh thành từ CSDL dùng chung của Thành phố và trả về thông tin danh mục tỉnh thành cho đơn vị.

## Mục đích tài liệu

* Tài liệu này nhằm hướng dẫn đơn vị lấy thông tin danh mục tỉnh thành từ CSDL tập trung của Thành phố.

## Phạm vi và đối tượng áp dụng

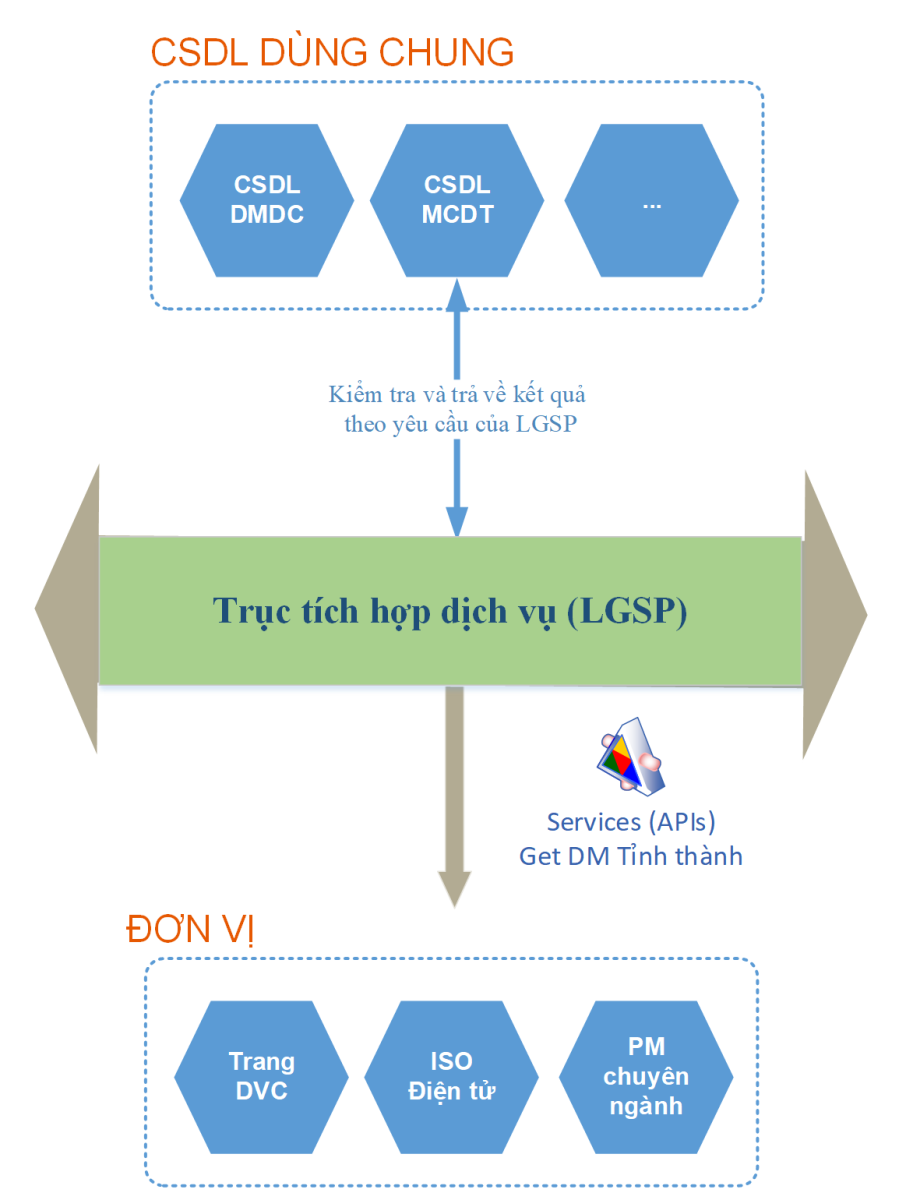
* Tài liệu này được sử dụng bởi các kỹ thuật viên của các đơn vị cần lấy thông tin danh mục tỉnh thành từ chuẩn CSDL dùng chung của Thành phố.

## Thông tin kết nối

* Đơn vị cần kết nối liên hệ sở Thông tin và Truyền thông để được cung cấp thông tin kết nối bao gồm:
  + AppName
  + AccessKey
  + SecretKey

# MÔ HÌNH KẾT NỐI

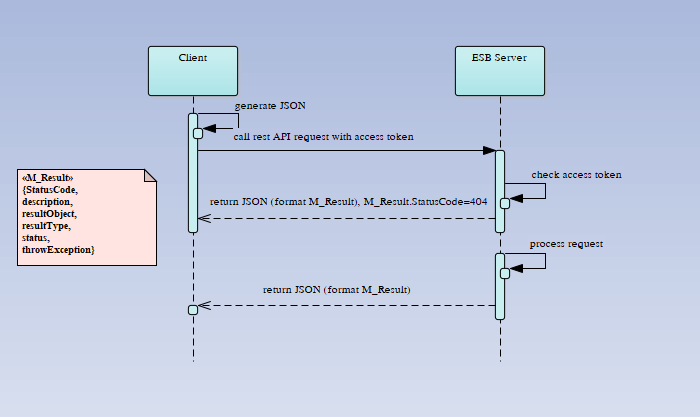
## Mô hình kết nối lấy danh mục tỉnh thành



*Hình 1: Mô hình kết nối lấy bộ danh mục tỉnh thành*

* CSDL Dùng chung là nơi tập trung toàn bộ kho dữ liệu dùng chung của TPHCM. Cơ quan, tổ chức có thể thực hiện việc khai thác và đồng bộ dữ liệu về CSDL Dùng chung của Thành phố thông qua Nền tảng HCM LGSP.
* Nền tảng HCM LGSP đảm nhiệm vai trò chính trong việc cung cấp các thông tin kết nối đến các phần mềm tại đơn vị, khai thác và đồng bộ dữ liệu từ CSDL tập trung của thành phố.
* APILayDanhMucTinhThanh hỗ trợ phần mềm kết nối vào Nền tảng HCM LGSP lấy bộ danh mục Tỉnh thành chuẩn từ CSDL dùng chung của Thành phố.

## Quá trình trao đổi gói tin



*Hình 2: Quy trình trao đổi gói tin*

* Bước 1: Phần mềm của đơn vị gọi đến Nền tảng HCM LGSP thông qua api kèm theo access token.
* Bước 2: Nền tảng HCM LGSP kiểm tra access token của đơn vị. Nếu thất bại sẽ trả về gói tin JSON lỗi cho phần mềm của đơn vị.
* Bước 3: Nền tảng HCM LGSP xử lý dữ liệu và trả về gói tin JSON kết quả cho phần mềm đơn vị.

# ĐẶC TẢ API LẤY DANH MỤC TỈNH THÀNH

## Thông tin kết nối

|  |  |  |
| --- | --- | --- |
| **Kiểu** | **Diễn giải** | **Ví dụ** |
| Địa chỉ adapter Nền tảng HCM chính | Địa chỉ cổng dịch vụ chính thức | https://hcmlgsp.tphcm.gov.vn |
| Địa chỉ adapter Nền tảng HCM thử nghiệm | Địa chỉ cổng dịch vụ thử nghiệm | https://hcmesb-test.tphcm.gov.vn |
| AccessKey | Khóa | rTkhYCBwHM |
| SecretKey | Khóa bí mật | DWkQgY1YSS |
| AppName | Đơn vị kết nối | TPHCM |
| PartnerCode | Mã đơn vị cung cấp ứng dụng | 000.00.01.H29 |
| PartnerCodeCus | Mã đơn vị sử dụng dịch vụ | 000.00.01.H29 |
| Authorization | Khai báo trong Header, Là chuỗi gồm 5 tham số dạng json được mã hóa thành StringBase64 | ewoiQWNjZXNzS2V5IjoiclRraFlDQndITSIsCiJTZWNyZXRLZXkiOiJEV2tRZ1kxWVNTIiwKIkFwcE5hbWUiOiAiVFBIQ00iLAoiUGFydG5lckNvZGUiOiAiMDAwLjAwLjAxLkgyOSIsCiJQYXJ0bmVyQ29kZUN1cyI6ICIwMDAuMDAuMDEuSDI5Igp9Cg==  được stringbase64 từ:  {  "AccessKey":"rTkhYCBwHM",  "SecretKey":"DWkQgY1YSS",  "AppName": "TPHCM",  "PartnerCode": "000.00.01.H29",  "PartnerCodeCus": "000.00.01.H29"  } |

*Bảng 1: bảng thông tin kết nối*

## API lấy danh mục tỉnh thành

* Đường dẫn: /GetDanhMucTinhThanh
* Method: GET

#### Đầu vào:

* + Header: Authorization

#### Đầu ra:

* + chuỗi json chứa thông tin trả về, resultObject là chuỗi json array chứa thông tin danh mục, chi tiết dữ liệu tại ***mục 3.3***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tham số | Thuộc tính tham số | Mô tả |
| 1 | StatusCode | Int | Mã trạng thái gói tin  - 200: Request đã được tiếp nhận và xử lý thành công  - 400: Request bị lỗi hoặc không thể xử lý  - 404: Tài nguyên không tồn tại  - 500: Lỗi trong quá trình xử lý |
| 2 | Description | String | Nội dung lỗi trả về |
| 3 | ResultObject | Boolean | Kết quả trả về |
| 4 | ResultType | String | Type của gói tin trả về Mặc định là “JSON” |
| 5 | Status | String | Trạng thái gói tin“SUCCESS” hoặc “FAIL” |
| 6 | ThrowException | bool | Có lỗi xảy ra hay không |

Bảng 2: tham số trả về của chuỗi json

Ví dụ:

|  |  |
| --- | --- |
| Thành công | Lỗi |
| {  "StatusCode": 200,  "Description": null,  "ResultObject": [{  "MaTinhThanh": "001",  "MoTa": " ",  "TenTinhThanh": "TP Hồ Chí Minh",  "MaQuocGia": "046",  "Used": true  }],  "ResultType": "JSON",  "Status": "SUCCESS",  "ThrowException": false  } | {  "StatusCode": 500,  "Description": "Tham số đầu vào không đúng",  "ResultObject": null,  "ResultType": null,  "Status": "FAIL",  "ThrowException": true  } |

## Gói tin danh mục tỉnh thành

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tham số | Thuộc tính tham số | Mô tả | Yêu cầu |
| 1 | MaTinhThanh | String | Mã tỉnh thành | Bắt buộc |
| 2 | TenTinhThanh | String | Tên tỉnh thành | Bắt buộc |
| 3 | MoTa | String | Mô tả | Không bắt buộc |
| 4 | MaQuocGia | String | Mã quốc gia | Bắt buộc |
| 5 | Used | Boolean | Được sử dụng | Bắt buộc |

*Bảng 3: Gói tin danh mục tỉnh thành*

**Ví dụ:**

{

"MaTinhThanh": "001",

"MoTa": " ",

"TenTinhThanh": "TP Hồ Chí Minh",

"MaQuocGia": "046",

"Used": true

}

# CODE EXAMPLE

## **Code mẫu .Net**

using RestSharp;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Text;

* API Method Get

public static String HTTP\_GET(String url, String urlSerivce, String sercet, Dictionary<string, string> param)

{

var client = new RestClient(url);

var request = new RestRequest(urlSerivce, Method.GET);

request.AddHeader("Authorization", sercet);

//Add Parameter vào request

foreach (var item in param)

{

request.AddParameter(item.Key, item.Value);

}

var response = client.Execute(request);

if(response.StatusCode == System.Net.HttpStatusCode.OK)

{

return response.Content;

}

else

{

//Xử lý trường hợp kết nối lỗi

return "";

}

}

* API Method POST

//jsonObj : object có cấu trúc theo như mô tả

public static String HTTP\_POST(String url, String urlSerivce, String sercet,object jsonObj)

{

var client = new RestClient(url);

var request = new RestRequest(urlSerivce, Method.GET);

request.AddHeader("Authorization", sercet);

request.AddHeader("Content-Type", "application/json;charset=UTF-8");

request.AddBody(jsonObj); // jsonObj sẽ được tự động chuyển thành chuỗi json tương ứng

var response = client.Execute(request);

if(response.StatusCode == System.Net.HttpStatusCode.OK)

{

return response.Content;

}

else

{

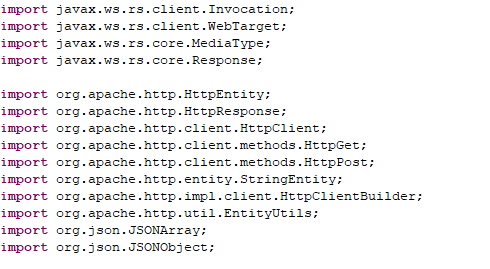
//Xử lý trường hợp kết nối lỗi

return "";

}

}

## **Code mẫu java**

****

* API Method Get

**public** **static** String HTTP\_GET(String url, **boolean** authorization, String sercet) **throws** UnsupportedEncodingException {

HttpClient httpClient = HttpClientBuilder.*create*().build();

HttpGet httpGet = **new** HttpGet(url);

**if** (authorization) {

httpGet.setHeader("Authorization", sercet);

}

String responseString = "";

**try** {

HttpResponse response = httpClient.execute(httpGet);

HttpEntity entity = response.getEntity();

**if** (entity != **null**) {

responseString = EntityUtils.*toString*(entity, "UTF-8");

}

} **catch** (IOException e) {

e.printStackTrace();

}

System.***out***.println(responseString);

**return** responseString;

}

* API Method POST

**public** **static** String HTTP\_POST(String url, **boolean** authorization, String sercet, String dataJson) **throws** UnsupportedEncodingException {

HttpClient httpClient = HttpClientBuilder.*create*().build();

HttpPost httpPost = **new** HttpPost(url);

**if** (authorization) {

httpPost.setHeader("Authorization", sercet);

}

httpPost.setHeader("Content-Type", "application/json;charset=UTF-8");

StringEntity httpEntity = **new** StringEntity(dataJson, "UTF-8");

httpPost.setEntity(httpEntity);

String responseString = "";

**try** {

HttpResponse response = httpClient.execute(httpPost);

HttpEntity entity = response.getEntity();

**if** (entity != **null**) {

responseString = EntityUtils.*toString*(entity, "UTF-8");

}

} **catch** (IOException e) {

e.printStackTrace();

}

System.***out***.println(responseString);

**return** responseString;

}

# Tham khảo

* Thông tin tài liệu được cập nhật tại địa chỉ: <https://api.tphcm.gov.vn/ChiTietLgsp/id/37/idPa/15/mota/MoTa>