ALIBABA CLOUD

阿里云

IP地理位置库
离线版SDK使用说明

文档版本：20210723

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.SDK（Python版本）说明 05
1.1. Linux环境Python版本 05
1.2. Windows环境Python版本 08
2.SDK（C++版本）说明 12
2.1. Linux环境C++版本 12
2.2. Windows环境C++版本 16
3.SDK（Go版本）说明 22
3.1. Linux环境Golang版本 22
4.SDK（Java版本）说明 26

1.SDK（Python版本）说明

1.1. Linux环境Python版本

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

SDK说明

基于安全和性能方面的综合考虑，IP 地址库的核心逻辑采用了 C++版 SDK 生成的动态链接库，再用 Python 对接口进行封装。

- 因为需要配置动态链接库，所以对使用 SDK 的操作系统环境会有一定的要求。
- SDK 中已包含了部分操作系统环境下编译成功后的动态链接库。

环境准备

- 目前支持Linux x86_64 (64-bit)平台(Redhat/CentOS/Ubuntu)
- 支持的 python 版本（查看版本 python -V ）

python2.7.*

python3.4.*

python3.5.*

python3.6.*

python3.7.*

python3.8.*

- Botan2安装

> **注意**
>
> 版本要求：2.13.0或以上版本。

通过包管理工具安装：

- centos：yum install -y botan2

通过源码安装（推荐）：

```
wget https://github.com/randombit/botan/archive/2.13.0.tar.gz
tar -xvzf 2.13.0.tgz
./configure.py
make
make instal
```

下载SDK

如已购买IP地理位置库（离线版）商品，可进入实例详情，下载以下三个文件

1. 离线数据库（通常是 **.dex）

2. 授权证书文件（通常是 **.lic）
3. Python语言SDK（通常是 alibaba-geoip-python-sdk-{version}.tgz）
4. {version} 代表SDK版本例如：alibaba-geoip-python-sdk-1.0.tgz

> **注意**
>
> 续费、升级商品套餐一定要同步更新license证书文件，如果license证书过期，则SDK会直接不可用。

安装SDK

将下载的alibaba-geoip-python-sdk-{version}.tgz 放到指定的目录，然后运行以下命令进行安装

```
tar -zxvf alibaba-geoip-python-sdk-{version}.tgz
cd alibaba-geoip-python-sdk-{version}
python setup.py install
```

> **注意**
>
> 因为安装后 SDK 等于集成到了python Lib 标准库，所以 alibaba-geoip-python-sdk-{version} 文件夹可以删掉。

快速入门

下面以查询某个IPV4地址为例演示如何使用：

目录结构如下

```
.
|-- data
|    |-- license-ipv4.dex
|    `-- license-ipv4.lic
|-- example.py
`-- tests.py
```

```
from geoipclient import GeoIpClient
import os

current_file_dir = os.path.abspath(os.path.dirname(__file__))
# license file path
LICENSE_FILE_PATH = os.path.join(current_file_dir, 'data', 'license-ipv4.lic')
# ipdata file path
IPDATA_FILE_PATH = os.path.join(current_file_dir, 'data', 'license-ipv4.dex')

client = GeoIpClient(LICENSE_FILE_PATH, IPDATA_FILE_PATH)
print('init finish')
print(client.search('47.116.2.4'))
client.close()
```

```
geoip-python-sdk# python example.py
finish loading data, taken: 1 seconds
init finish
{"country":"中国","province":"上海市","city":"上海市","county":"浦东新区","isp":"阿里云","country_code":"CN","country_en":"China","province_en":"Shanghai","city_en":"Shanghai","longitude":"121.567706","latitude":"31.245944_5","isp_code":"1000323","routes":"中国电信/中国联通/中国移动/中国铁通/中国教育网"}
```

附件

`properties` 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang
county	区（四级）中文	阳明区
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645

字段	含义	示例值
latitude	纬度	44.596328

异常处理

出现异常会通过geoipclient.GeoIpException抛出

一般是文件路径异常，离线数据错误异常

ip查找失败，会返回空字符串

1.2. Windows环境Python版本

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

SDK说明

基于安全和性能方面的综合考虑，IP 地址库的核心逻辑采用了 C++版 SDK 生成的动态链接库，再用 Python 对接口进行封装。

- 因为需要配置动态链接库，所以对使用 SDK 的操作系统环境会有一定的要求。
- SDK 中已包含了部分操作系统环境下编译成功后的动态链接库。

环境准备

以下代码在Windows 10 、Visual Studio 2019、python3.5 编译测试通过

目前支持Windows x86_64 (64-bit)平台

支持的 python 版本（查看版本 python -V ）

python2.7.*

python3.4.*

python3.5.*

python3.6.*

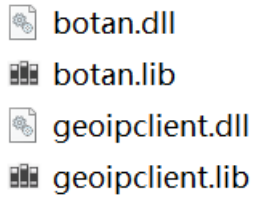
python3.7.*

python3.8.*

为了方便演示SDK的使用，Windows版本提供了编译后的 C++ 动态链接库 botan.dll、botan.lib，默认情况下可以直接进行测试

> **注意**
>
> 如果编译了Botan2其它版本，需要替换Python安装目录文件。

替换 C:\Python35\Lib\site-packages\alibaba_geoip_python_sdk-1.1-py3.5.egg\geoipclient\lib\win 目录中的botan.dll和botan.lib



下载SDK

如已购买IP地理位置库（离线版）商品，可进入实例详情，下载以下三个文件

1. 离线数据库（通常是**.dex）
2. 授权证书文件（通常是 **.lic）
3. Python语言SDK（通常是 alibaba-geoip-python-sdk-{version}.tgz）
4. {version} 代表SDK版本例如：alibaba-geoip-python-sdk-1.0.tgz

> **注意**
>
> 续费、升级商品套餐一定要同步更新license证书文件，如果license证书过期，则SDK会直接不可用。

安装SDK

将下载的alibaba-geoip-python-sdk-{version}.tgz 放到指定的目录，然后运行以下命令进行安装

```
解压文件 alibaba-geoip-python-sdk-{version}.tgz
进入目录 alibaba-geoip-python-sdk-{version}
执行命令 python setup.py install
```

> **注意**
>
> 因为安装后 SDK 等于集成到了python Lib 标准库，所以 alibaba-geoip-python-sdk-{version} 文件夹可以删掉。

快速入门

下面以查询某个IPV4地址为例演示如何使用：

目录结构如下

```
.
|-- data
|    |-- license-ipv4.dex
|    `-- license-ipv4.lic
|-- example.py
`-- tests.py
```

```
from geoipclient import GeoIpClient
import os

current_file_dir = os.path.abspath(os.path.dirname(__file__))
# license file path
LICENSE_FILE_PATH = os.path.join(current_file_dir, 'data', 'license-ipv4.lic')
# ipdata file path
IPDATA_FILE_PATH = os.path.join(current_file_dir, 'data', 'license-ipv4.dex')

client = GeoIpClient(LICENSE_FILE_PATH, IPDATA_FILE_PATH)
print('init finish')
print(client.search('47.116.2.4'))
client.close()
```

```
geoip-python-sdk# python example.py
finish loading data, taken: 1 seconds
init finish
{"country":"中国","province":"上海市","city":"上海市","county":"浦东新区","isp":"阿里云","country_code":"CN","country_en":"China","province_en":"Shanghai","city_en":"Shang
hai","longitude":"121.567706","latitude":"31.245944_5","isp_code":"1000323","routes":"中国电信/中国联通/中国移动/中国铁通/中国教育网"}
```

附件

properties 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang
county	区（四级）中文	阳明区

字段	含义	示例值
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645
latitude	纬度	44.596328

异常处理

出现异常会通过geoipclient.GeoIpException抛出

一般是文件路径异常，离线数据错误异常

ip查找失败，会返回空字符串

2.SDK（C++版本）说明

2.1. Linux环境C++版本

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

环境准备

- 目前只支持Linux x86_64 (64-bit)平台（Redhat/Centos）
- 需要支持c++11标准的c++编译器
- Botan2安装

> **注意**
>
> 版本要求：2.13.0或以上版本。

  - 通过包管理工具安装：
    - centos：yum install -y botan2
  - 通过源码安装（推荐）：

```
wget https://github.com/randombit/botan/archive/2.13.0.tar.gz
tar -xvzf 2.13.0.tar.gz
./configure.py
make
make instal
```

下载SDK

如已购买IP地理位置库（离线版）商品，可进入实例详情，下载以下三个文件

- 离线数据库（数据文件）
- 授权证书文件（license）
- SDK（C++语言）
  - 头文件 `geoipclient.hpp`
  - 动态链接库 `libgeoipclient.so`
  - 静态链接库 `libgeoipclient.a`

> **注意**
>
> 续费、升级商品套餐一定要同步更新license证书文件，如果license证书过期，则SDK会直接不可用。

安装SDK

1.将下载的头文件geoipclient.hpp拷贝到项目的头文件目录下

2.根据需要把 `libgeoipclient.so` 或 `libgeoipclient.a` 拷贝到 `/usr/local/lib` 下

3.设置环境变量 `export LD_LIBRARY_PATH=/usr/local/lib`

> **注意**
>
> 环境变量一定要设置好，不然无法查找到动态链接库
>
> `libgeoipclient.so`
>
> 或静态链接库
>
> `libgeoipclient.a`

快速入门

重要类介绍：

- ○ `license_path`

  ：license路径

  ○ `ipdata_path`

  ：data数据路径

- ○ `ip`

  ：被查询IP

使用方法：

1. 将加载字段、license文件路径、数据文件路径封装成一个

   `geoipclient`

2. 实例化

   `GeoIPClient`

3. 通过

   `client.search(String ip)`

   方法来检索ip的地址位置等相关信息，返回的数据为json格式，值不在时用空字符串 "" 代替。

例子程序

- libgeoipclient主要封装了c++类 `GeoIPClient` ，暴漏 `search` 方法，该方法返回对应ip地址的信息，以 `json` 的格式返回。
- 将下载的头文件

`geoipclient.hpp`

、离线数据库（数据文件）、授权证书文件（license）拷到

`example.cpp`

相同的目录下

。

异常说明：

1.如果license文件和ip地址数据文件有问题，会抛出异常

2.如果查询频率超过限速，会直接返回空字符串

调用实例：

```
#include <iostream>
#include "geoipclient.hpp"
using namespace alibaba::dns;
int main() {
  try {
    //第一个参数是授权文件的路径，第二个参数是离线数据库文件路径
    GeoIPClient client("client.lic", "ipv4.dex");
    std::cout << client.search("47.116.2.4") << std::endl;;
  }
  catch (std::exception& e) {
    std::cout << "get error:" << e.what() << std::endl;
  }
  return 0;
}
```

- 在目录下执行

```
[root@localhost ~]#g++ -o testgeoipclinet example.cpp -I. -lgeoipclient -lbotan-2 -std=c++11
```

- 执行及查看查询结果

```
[root@localhost ~]#./testgeoipclinet
```

```
{
  "country": "中国",
  "city": "上海市",
  "isp": "阿里云",
  "province_en": "Shanghai",
  "latitude": "31.2459066",
  "county": "浦东新区",
  "country_en": "China",
  "city_en": "Shanghai",
  "country_code": "CN",
  "routes": "中国电信/中国联通/中国移动/中国铁通/中国教育网",
  "province": "上海市",
  "isp_code": "1000323",
  "longitude": "121.5677066"
}
```

附件

`properties` 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang
county	区（四级）中文	阳明区

字段	含义	示例值
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645
latitude	纬度	44.596328

2.2. Windows环境C++版本

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

环境准备

以下代码在Windows 10 、 Visual Studio 2019、python3.5 编译测试通过：

目前支持Windows x86_64 (64-bit)平台

Botan2安装

> **注意**
>
> 版本要求：2.13.0或以上版本。

通过源码安装Botan2（推荐）：

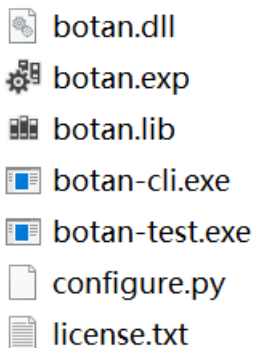
下载 https://github.com/randombit/botan/archive/2.13.0.tar.gz

解压缩文件 2.13.0.tar.gz 通过命令行进入解压后的文件夹botan-2.13.0

执行命令：python configure.py --cc=msvc --cpu=x86_64

> **注意**
>
> 编译Botan2需要依赖python请自行安装并设置环境变量

编译成功后的文件如图所示：



可在window命令行工具执行 botan-test.exe 观察测试结果。

下载SDK

如已购买IP地理位置库（离线版）商品，可进入实例详情，下载以下三个文件

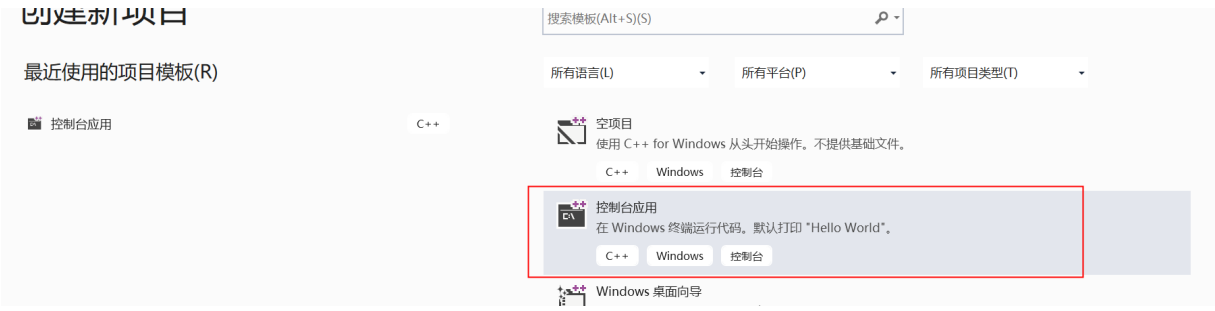
- 离线数据库（通常是**.dex）
- 授权证书文件（通常是**.lic）
- SDK（C++语言）

1. 头文件 `geoipclient.hpp`
2. 动态链接库文件 `geoipclient.dll`
3. 外部链接库文件 `geoipclient.lib`

> **注意**
>
> 续费、升级商品套餐一定要同步更新license证书文件，如果license证书过期，则SDK会直接不可用。

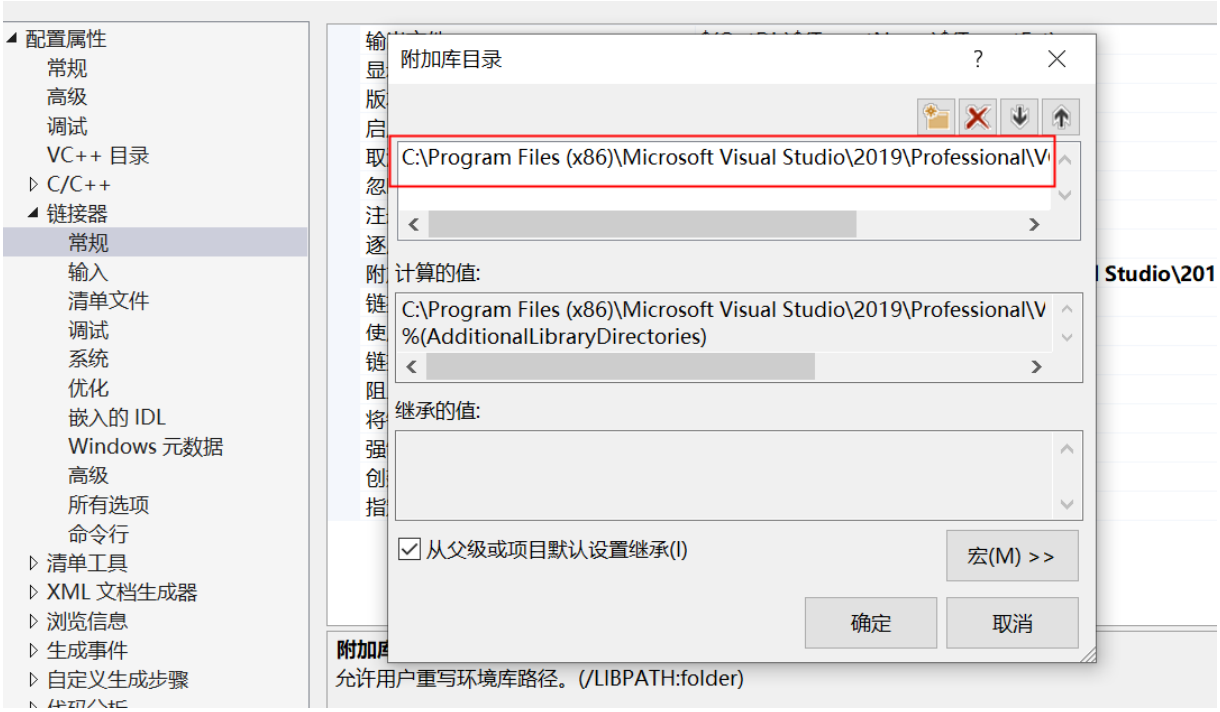
安装SDK

1. 安装 Visual Studio 2019 创建演示项目 HelloWorld



2.通过Visual Studio 2019加载引入botan.lib、geoipclient.lib（此步骤重要）

> **注意**
>
> 需要自行将botan.lib、geoipclient.lib复制到下面的参考路径



参考路径：C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\VC\Tools\MSVC\14.27.29110\lib\x64

快速入门

重要类介绍：

- license_path ：license路径
- ipdata_path ：data数据路径
- ip ：被查询IP

使用方法：

1. 将加载字段、license文件路径、数据文件路径封装成一个 geoipclient
2. 实例化 GeoIPClient
3. 通过 client.search(String ip) 方法来检索ip的地址位置等相关信息，返回的数据为json格式，值不在时用空字符串 "" 代替。

演示代码

- geoipclient主要封装了c++类 GeoIPClient ，暴漏 search 方法，该方法返回对应ip地址的信息，以 json 的格式返回。
- **异常说明：**

  1.如果license文件和ip地址数据文件有问题，会抛出异常

2.如果查询频率超过限速，会直接返回空字符串

- 修改HelloWorld.cpp代码为以下演示代码作为查询ipv4示例：

```
#include <iostream>
#include "geoipclient.hpp"
using namespace alibaba::dns;
int main() {
  try {
    //第一个参数是授权文件的路径，第二个参数是离线数据库文件路径
    GeoIPClient client("D:\\ben\\client.lic", "D:\\ben\\ipv4.dex");
    std::cout << client.search("47.116.2.4") << std::endl;;
  }
  catch (std::exception& e) {
    std::cout << "get error:" << e.what() << std::endl;
  }
  return 0;
}
```

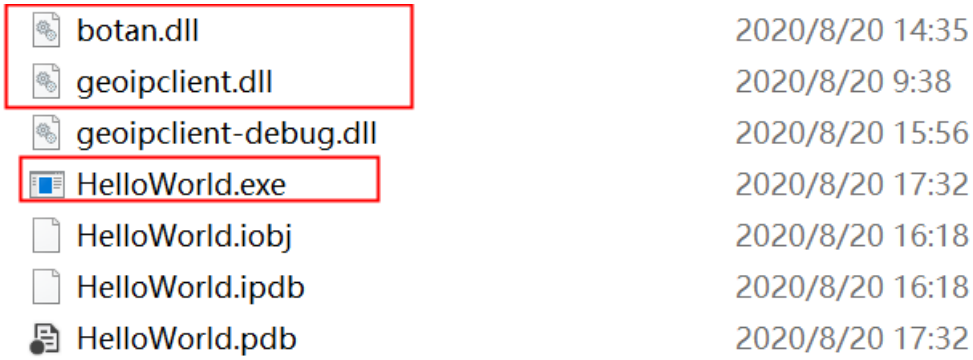
1.设置Release为x64

2.配置好代码中的client.lic、ipv4.dex 文件

3.按F5键进行调试

4.打开 source\repos\HelloWorld\x64\Release 可以看到 HelloWorld.exe 文件生成

5.复制botan.dll、geoipclient.dll 到HelloWorld.exe文件的同一路径如下图所示



6.在此目录新建命令窗口执行HelloWorld.exe会有如下输出内容，如果命令行有乱码可以执行以下命令尝试解决chcp 65001或Active code page: 65001

```
[redacted]e\repos\HelloWorld\x64\Release>HelloWorld.exe
finish loading data, taken: 2 seconds
{"country":"中国","province":"上海市","city":"上海市","county":"","isp":"阿里巴巴","country_code":"CN","country_en":"China","province_en":"Shanghai","city_en
":"Shanghai","longitude":"121.4726000","latitude":"31.2317000"}
```

附件

properties 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang
county	区（四级）中文	阳明区
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645

字段	含义	示例值
latitude	纬度	44.596328

3.SDK（Go版本）说明

3.1. Linux环境Golang版本

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

SDK说明

基于安全和性能方面的综合考虑，IP地址库的核心逻辑使用C++离线版SDK编译生成的动态链接库，再用 Go 对接口进行封装。

> **注意**
>
> 由于使用了cgo，所以使用这个sdk的项目，在编译的时候不能设置CGO_ENABLED=0，并且对运行环境有一定要求，windows、mac均不支持，建议centos7*，gcc 4*或使用Docker，使用文档、sdk、cgo代码、Dockerfile等请钉钉联系：behladu 获取。

环境准备

- Linux x86_64 (64-bit)平台(Redhat / CentOS)
- Go（gomod、gopath方式均支持）
- Botan2

> **注意**
>
> Botan2 版本要求：2.13.0或以上版本。

通过源码安装（推荐）：

```
wget https://github.com/randombit/botan/archive/2.13.0.tar.gz
tar -xvzf 2.13.0.tgz
./configure.py
make
make instal
```

通过包管理工具安装：

centos：yum install -y botan2

下载SDK

如已购买IP地理位置库（离线版）商品，可进入云产品控制台，下载以下两个文件

1. 离线数据库（通常是 **.dex）
2. 授权证书文件（通常是 **.lic）
3. Go语言SDK（请钉钉联系：behladu 获取）

安装SDK

1、将alibaba-geoip-go-sdk-1.0.tgz 解压后放到指定的目录，结构如图所示：

```
.
├── geoipclient
│   ├── client.go
│   ├── geoipclient.h
│   └── libgeoipclient.so
├── go.mod
└── main.go
```

> **注意**
>
> go的包名为geoipclient，client是线程安全的，可以在多个goroutine中使用。go.mod和main.go为演示代码，可以忽略。

2、将动态链接库libgeoipclient.so文件拷贝或移动到 /usr/local/lib目录下

3、设置环境变量 export LD_LIBRARY_PATH=/usr/local/lib

快速入门

下面以查询某个IPV4地址为例演示如何使用：

go.mod示例代码

```
module demo

go 1.14
```

main.go示例代码，lic和dex文件是我们通过阿里云控制台下载的离线库和授权文件

```
package main

import (
 "fmt"
 "log"
 "demo/geoipclient"
)

func main() {
 //client是线程安全，可以被多个goroutine使用
 client, err := geoipclient.NewClient("20210222123044.lic", "20210310075715.dex")
 if err != nil {
  log.Println(err)
  return
 }
 //查询
 search := client.Search("221.206.131.10")
 fmt.Println(search)
}
```

查结果如下：

```
{
  "country": "中国",
  "city": "牡丹江市",
  "county_code": "231003",
  "isp": "中国联通",
  "province_en": "Heilongjiang",
  "latitude": "44.596328",
  "county": "阳明区",
  "country_en": "China",
  "city_code": "231000",
  "province_code": "230000",
  "city_en": "Mudanjiang",
  "country_code": "CN",
  "routes": "中国联通",
  "province": "黑龙江省",
  "isp_code": "100026",
  "longitude": "129.634645"
}
```

附件

`properties` 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang

字段	含义	示例值
county	区（四级）中文	阳明区
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645
latitude	纬度	44.596328

异常处理

1、加载文件异常可在NewClient后捕获

2、查询ip异常会返回空字符串

4.SDK（Java版本）说明

环境准备

目前该产品在做内部调整，不支持新购、续费、升级等操作，订单有效期内可继续正常使用，本文档仅供订单期内用户参考。请留意服务到期时间，做好备选方案，感谢您的支持理解。

- 环境要求使用Java 1.8及以上版本。
- 查看Java版本执行命令 `java -version` 。

下载SDK

如已购买IP地理位置库（离线版）商品，可进入实例详情，下载以下三个文件

- 离线数据库（数据文件）
- 授权证书文件（license）
- SDK（Java语言）

> **注意**
>
> 续费、升级商品套餐一定要同步更新license证书文件，如果license证书过期，则SDK会直接不可用。

安装SDK

方式一：在 Intellij IDEA 项目中导入JAR包

1. 下载SDK（Java语言）JAR。
2. 将SDK JAR文件拷贝到您的项目中。
3. 在 Intellij IDEA 中选择您的工程，选择**File > Project Structure > Modules > Dependencies > + > JARs or directories**。
4. 选中您在第2步拷贝的SDK JAR文件，导入到External Libraries中。

注意：

1. SDK JAR包未发布到线上仓库，仅支持直接引入使用
2. SDK JAR包内仅包含SDK 的开发代码，未包含需要的其他第三方JAR包代码，在使用时需要手动加入相应的第三方依赖如下（使用时可根据实际情况做调整，其中commons-codec的版本必须为1.11或以上版本）：

```
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.10</version>
</dependency>
<!--1.0.4版本已经移除-->
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>28.2-jre</version>
</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.11</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.68</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.26</version>
</dependency>
```

快速入门

重要类介绍：

- `com.alibaba.sec.domain.FastGeoConf` 目前有如下字段，分别是：
  - `properties` ：用户自定加载的字段，不填写则加载所有字段，在使用时可选择性加载其中需要的属性，这样能显著降低内存的占用空间。具体字段含义参考下文附件
  - `licenseFilePath` ：读取授权文件存放路径
  - `dataFilePath` ： 读取离线数据文件存放路径

  1.0.4及以上版本支持通过非文件路径的方式加载离线许可证和离线数据

  - `licenseBytes` ：读取授权文件字节数组
  - `dataBytes` ： 读取离线数据文件字节数组
  - `licenseInput` ：读取授权文件字节流
  - `dataInput` ： 读取离线数据文件字节流
  - `filterEmptyValue` ：用于控制查询结果是否返回内容为空的字段（默认不启用此功能）

> **注意**
>
> properties与filterEmptyValue()区别，properties设置返回的字段，无论是否为空值，filterEmptyValue启用后字段为空值时，不在结果集中显示该空值字段。

- `com.alibaba.sec.client.FastIPGeoClient` 是用户使用SDK的客户端对象，需要通过构造函数实例化

使用方法：

1. 将加载字段、license文件路径、数据文件路径封装成一个 `com.alibaba.sec.domain.FastGeoConf`
2. 通过静态方法获取单例（能保证在数据文件和license文件路径一样的情况下的实例只会在内存里面生成一份） `com.alibaba.sec.client.FastIPGeoClient.getSingleton(com.alibaba.sec.domain.FastGeoConf)`
3. 通过 `FastIPGeoClient.search(String ip)` 方法来检索ip的地址位置等相关信息，返回的数据为json格式，键值对都是 `FastGeoConf.properties` 里指定的字段（若未指定，则加载所有字段。在明确实际要用到字段的情况下，生产环境建议固定字段，避免因为新数据字段增加导致内存占用增加），值不在时用空字符串 "" 代替。
4. 在使用时需要显式的捕获异常： `LicenseException` ， `FastIPGeoException` ，根据异常的编码做相应处理，具体异常列表参考下文附件。

调用实例：

```
import com.alibaba.sec.client.FastIPGeoClient;
import com.alibaba.sec.domain.FastGeoConf;
import com.alibaba.sec.exception.FastIPGeoException;
import com.alibaba.sec.license.exception.LicenseException;

import java.util.Arrays;
import java.util.HashSet;

public class GeoClientTest {
  private static final String DATA_FILE_PATH = "${path}/ipv4.dex";
  private static final String LICENSE_FILE_PATH = "${path}/license.lic";

  public static void main(String[] args) throws Exception {
    FastGeoConf geoConf = new FastGeoConf();
    geoConf.setDataFilePath(DATA_FILE_PATH);
    geoConf.setLicenseFilePath(LICENSE_FILE_PATH);
    //指定sdk返回字段，不指定字段默认返回全部，可节省内存开销
    HashSet<String> set = new HashSet<>(Arrays.asList(
        "country", "province", "province_code",
        "city", "city_code", "county", "county_code",
        "isp", "isp_code", "routes",
        "longitude", "latitude"
    ));
    geoConf.setProperties(set);
    geoConf.filterEmptyValue(); //空值字段，不在结果中返回
    //通过构造方法初始化（非单例，旧版SDK中使用）
    //FastIPGeoClient fastIpGeoClient = new FastIPGeoClient(geoConf);
    //通过静态方法（单例）,推荐使用该方法，sdk需升级到最新版本
    FastIPGeoClient fastIpGeoClient = FastIPGeoClient.getSingleton(geoConf);
    String result;
    try {
      result = fastIpGeoClient.search("221.206.131.10");
      System.out.println(result);
    } catch (LicenseException e) {
      e.printStackTrace();
    } catch (FastIPGeoException e) {
      e.printStackTrace();
    }
  }
}
```

LicenseException 错误码：

```
UNKNOWN(500, "未知异常"),
ALREADY_EXIST(100, "文件已经存在"),
NOT_ABS_PATH(101, "不是绝对路径"),
DIR_NOT_EXIST(102, "文件夹不存在"),
LICENSE_INVALID(501, "证书异常"), //证书不合法
SYSTEM_TIME_ERR(502, "系统时间异常"),
LICENSE_WILL_EXPIRE(503, "证书即将过期，请提前申请，过期后将无法使用"),
LICENSE_EXPIRE(504, "证书过期, 请重新申请"),
LICENSE_DELAY_ERR(505, "证书延期异常"),
LICENCE_ERR_ECHO(506, "非法回应"), //证书不合法，证书内部校验返回值失败
LICENSE_ERR_RATELIMIT(507, "超过购买的qps阈值");
```

FastIPGeoException 错误编码：

```
UNKNOWN(1500, "未知异常"),
INVALID_DAT(1501, "文件内容不合法"),
NOT_MATCH(1503, "文件不匹配"); //用户的license类型和文件类型不匹配
```

附件

`properties` 字段含义：

字段	含义	示例值
country	国家（或地区）中文	中国
country_code	国家（或地区）两位字母	CN
country_en	国家（或地区）英文	China
province	省（二级）中文	黑龙江省
province_code	省（二级）数字代码	230000
province_en	省（二级）英文/拼音	Heilongjiang
city	市（三级）中文	牡丹江市
city_code	市（三级）数字代码	231000
city_en	市（三级）英文/拼音	Mudanjiang

字段	含义	示例值
county	区（四级）中文	阳明区
county_code	区（四级）数字代码	231003
isp	运营商	中国联通
isp_code	运营商数字代码	100026
routes	运营商线路	中国联通
longitude	经度	129.634645
latitude	纬度	44.596328