ALIBABA CLOUD

阿里云

阿里云公共DNS
API文档

文档版本：20200916

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.概述

阿里公共DNS致力于为广大的互联网用户提供快速、稳定和安全的DNS解析。传统的DNS查询和应答采用UDP和TCP明文传输，存在网络监听、DNS劫持、中间设备干扰的风险：

- 网络监听风险：即便用户采用HTTPS加密的方式访问站点，DNS查询应答并没有采用加密传输
- DNS劫持：传统DNS应答数据会被篡改，用户的访问会被路由到钓鱼网站和恶意站点
- 中间设备干扰：主要包括防火墙的拦截或篡改，针对域名的过滤，以及大包MTU分片等

为了应对以上挑战，阿里公共DNS遵守DoH（RFC8484）和DoT(RFC7858) 标准对外提供DNS的安全传输服务，支持DNS over HTTPs，DNS over TLS两种安全传输模式，以及基于 HTTP 和 HTTPS 的 DoH JSON API 接口。DNS的安全传输服务可以适用于移动应用程序、浏览器、操作系统、物联网设备，网关和路由器等多个场景。通过传输加密的方式发送DNS查询，加强了用户访问互联网的安全性、解析稳定和隐私保护。

除了隐私加密以外，用户端和DNS服务器间可以使用TCP或HTTP连接来提供安全传输服务，一方面可以服务精准的基于位置的DNS解析和流量调度，另一方面基于DNS端到端的连接特性，DNS的动态变更可以实现秒级端到端生效。

注意：在DoH和DoT传输服务中，阿里公共DNS支持 TLS 1.2 和TLS 1.3。

2.DNS over HTTPs(DoH)

阿里公共DNS通过RFC 8484指定的经过TLS加密的HTTP连接提供DNS解析

DNS over HTTPS（DoH）的URI接口：（仅提供TLS API）

- https://dns.alidns.com/dns-query?
- https://alidns_ip/dns-query?
- https://user_id.alidns.com/dns-query?

> 注意 其中alidns_ip是dns.alidns.com 的A记录，可以是以下两个地址之一：223.5.5.5，223.6.6.6。其中user_id是阿里云控制台的 Account ID（请参考控制台总览），例如：https://9999.alidns.com/dns-query?dns=uGkBAAABAAAAAAAAB2FsaWJhYmEDY29tAAABAAE

请求方式：GET

请求参数：

参数	类型	描述
dns	string	该参数是DoH客户端将正常的DNS查询报文转化成HTTP请求过程中设置的dns参数，具体内容是将DNS的请求二进制报文转变为base64url编码的字符串。接口定义具体请参考RFC8484

请求示例：https://dns.alidns.com/dns-query?dns=uGkBAAABAAAAAAAAB2FsaWJhYmEDY29tAAABAAE

返回的DNS二进制数据：

```
b869 8180 0001 0004 0000 0001 0761 6c69 6261 6261 0363 6f6d 0000 0100 0107
616c 6962 6162 6103 636f 6d00 0001 0001 0000 012c 0004 6a0b d097 0761 6c69 6261 6261 0363 6f6d 0000 0100
0100 0001 2c00 04cb 77d7 5207 616c 6962 6162 6103 636f 6d00 0001 0001 0000 012c 0004 6a0b df65 0761 6c69
6261 6261 0363 6f6d 0000 0100 0100 0001 2c00 04cb 7781 6d00 0029 1000 0000 0000 000c 0008 0008 0001 2018
6a0b 22e6
```

注：根据RFC8484的定义，DoH服务适用于两类场景：一种是DNS HTTPS隧道，一种是应用层访问DNS数据。

3.DoH JSON API

DoH JSON API的URL 接口 （提供TLS和非TLS API）

https://dns.alidns.com/resolve?

https://alidns_ip/resolve?

http://dns.alidns.com/resolve?

http://alidns_ip/resolve?

> 注意
>
> 其中alidns_ip是dns.alidns.com 的A记录，可以是以下两个地址之一： 223.5.5.5，223.6.6.6。

请求方式： GET

请求参数：

参数	类型	描述	实例	使用方法和默认值
name	string	请求域名	name=www.taobao.com.	必选，无默认值
type	number	请求类型	type=1	可选，1
edns_client_subnet	IP	ECS IP	edns_client_subnet=1.2.3.4/24	DNS代理使用，普通客户端不适用
short	boolean	是否开启简洁模式	short=true or short=1	可选，默认关闭
uid	string	用户ID，即控制台上的AccountID	uid=6666	可选

关于edns_client_subnet参数：

edns_client_subnet是为了支持DNS ECS功能（RFC7871），将用户的子网信息传递给权威DNS，做更精确的DNS解析和流量调度。其中掩码越长地址信息越精确，掩码越短用户隐私效果越好。建议使用“/24”掩码长度

注：该参数是特地为DNS代理（proxy）使用DoH JSON API场景设计，即用户发送DNS查询给DNS代理，DNS代理通过该参数携带用户的子网信息传递给阿里公共DNS，最后传递到权威DNS服务器。

例如edns_client_subnet=1.2.3.4/24，权威服务器会收到基于1.2.3.0/24地址前缀信息来帮助用户选择DNS链路

关于type参数支持类型：

记录类型	ID	意义	示例（以 taobao.com，www.taobao.com 为例）
A	1	IPv4地址	101.37.183.171
NS	2	NS记录	ns1.taobao.com.
CNAME	5	域名 CNAME 记录	www.taobao.com.danuoyi.tbcache.com.
SOA	6	ZONE 的 SOA 记录	ns4.taobao.com. hostmaster.alibabadns.com. 2018011109 3600 1200 3600 360
TXT	16	TXT 记录	"v=spf1 include:spf1.staff.mail.aliyun.com -all"
AAAA	28	IPv6 地址	240e:e1:f300:1:3::3fa

请求示例：

http://dns.alidns.com/resolve?name=www.taobao.com.&type=1

返回示例：

```
{ "Status": 0, "TC": false, "RD": true, "RA": true, "AD": false, "CD": false, "Question": { // 请求段 "name": "www.taobao.com.", "type": 1 }, "Answer": [ // 应答段 { "name": "www.taobao.com.", "TTL": 45, "type": 5, "data": "www.taobao.com.danuoyi.tbcache.com." }, { "name": "www.taobao.com.danuoyi.tbcache.com.", "TTL": 45, "type": 1, "data": "47.246.24.234" }, { "name": "www.taobao.com.danuoyi.tbcache.com.", "TTL": 45, "type": 1, "data": "47.246.24.233" } ] //"Authority" 为权威段, 如果有数据，需要与Answer字段一致 //"Additional" 为附加段, 如果有数据，需要与Answer字段一致 // 可配置"edns_client_subnet":"1.2.3.4/24" }
```

注：用户可以在客户端应用或手机应用中调用DoH和DoH JSON API解析DNS 。

4.DNS over TLS （DoT）

阿里公共DNS通过RFC 7858指定的经过TLS加密的TCP连接提供DNS解析。提供两种模式接入：域名方式和IP方式。基本流程是：

1. 终端设备配置DoT的解析服务器dns.alidns.com 、alidns_ip或者user_id.alidns.com

2. 如果配置的是dns.alidns.com或user_id.alidns.com，客户端先解析dns.alidns.com或user_id.alidns.com获得地址alidns_ip

3. 获得DoT解析服务器IP地址之后，终端设备再建立与DoT解析服务器在端口853 的TCP连接

4. 经过TLS握手协商，终端设备与DoT解析服务器建立TLS连接

5. 通过该TLS连接，终端设备可以发送DNS查询到DoT解析服务器

> 注意
>
> 其中alidns_ip是dns.alidns.com 的A记录，可以是以下两个地址之一：223.5.5.5，223.6.6.6。
>
> 其中user_id是阿里云控制台的 Account ID（请参考控制台总览），例如：9999.alidns.com。

对使用安卓手机的用户来说，可以设置在手机设置界面设置阿里公共DNS的域名和地址来获取DNS的DoT安全传输服务。

第一步：

进入设置页面，点击【无线和网络】

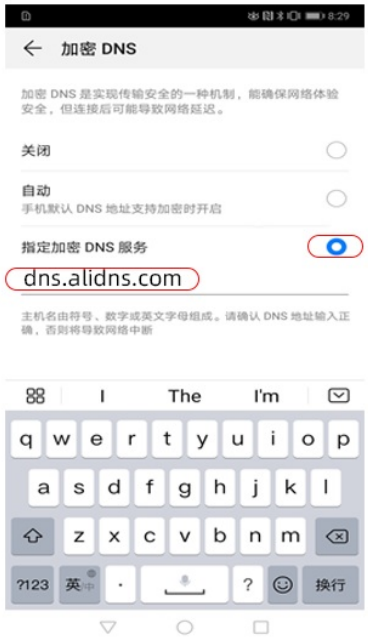
第二步：

点击【加密DNS】

第三步：

选中【指定加密DNS服务】，

并输入阿里云公共DNS服务接口地址。



注：上图是华为手机的的DoT网络设置

5.Android SDK开发指南

本文档介绍了阿里云公共DNS Android SDK的接入和开发方式。

1.概述

阿里云公共DNS SDK是阿里云面向广大移动开发者提供DNS域名解析服务的开发工具包。

开发者利用本SDK，可以在自己的Android APP中轻松接入阿里云公共DNS，解决域名解析异常的问题，低成本实现域名解析精准调度。您可以参考Demo示例工程源码了解如何使用本SDK。

SDK当前版本封装了阿里公共DNS的DoH JSON API，提供Java函数接口给Android APP进行域名解析，并且提供了基于TTL和LRU策略的高效域名缓存功能。在公共DNS原有功能的基础上，SDK还可以为用户带来以下优势：

- **简单易用：**

  用户仅需集成我们提供的SDK，便可接入阿里云公共DNS业务。接入方法简单易用，为用户提供更为轻松便捷的解析服务。

- **零延迟：**

  SDK内部实现了LRU的缓存机制，将每次域名解析后的IP缓存到本地；并且主动更新TTL过期缓存，保证缓存及时有效，从而帮助用户达到域名解析零延迟的效果。

2.如何使用SDK

2.1 jar集成

1. 您需先在控制台注册自己的应用，获取应用的唯一标识Account ID
2. 然后通过控制台的链接获取阿里公共DNS SDK
3. 获取到的SDK的pdns-sdk-android.jar集成到自己工程项目libs目录中即可轻松使用

2.2 gradle集成maven

在 build.gradle 文件中加入以下代码：

```
allprojects {
 repositories {
  maven {
   url 'https://maven.aliyun.com/repository/public/'
  }
  mavenLocal()
  mavenCentral()
 }
}
```

加入你要引用的文件信息：

```
dependencies {
  compile 'com.alibaba.pdns:pdns-android-sdk:1.0.1'

}
```

2.3 应用程序初始化

```
public class DnsCacheApplication extends Application{

    private String accountID="10001"; //可以更换成您的accountID
    private static final String TAOBAO_URL="www.taobao.com";
    private static final String M_TAOBAO_URL="m.taobao.com";
    private static final String ALIYUN_URL="aliyun.com";
    private static final int CACHE_MAX_NUMBER=100; //缓存的域名最大个数

    @Override
    public void onCreate() {
      super.onCreate();
      DNSResolver.Init(this, accountID);
      DNSResolver.setEnableShort (false);
      DNSResolver.setEnableIPv6 false);
      DNSResolver.setSchemaType(DNSResolver.HTTP);
      DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER);
      DNSResolver.getInstance().preLoadDomains(newString[]{TAOBAO_URL,M_TAOBAO_URL,ALIYUN_URL});
    }
}
```

接入阿里公共DNS SDK时建议在Application子类里集成。

> **说明**
>
> DNSResolver为阿里公共DNS SDK的核心类，其内部封装了阿里公共DNS提供的DoH JSON API，将用户的目标域名解析成为对应的IP。

另外，Android工程在使用SDK时，需要保证提供以下访问权限的配置：

```
<!--需要配置的权限-->
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
  <uses-permission android:name="android.permission.WAKE_LOCK"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

注意事项：

Android9.0会报异常：Didn't find class BasicHttpParams。

- 原因：Apache Http客户端弃用 。

因为早在Android 6.0中，谷歌取消了对Apache Http客户端的支持。从Android 9.0开始，org.apache.http.legacy将从bootclasspath中删除。

该修改对于大多数 taskVersion < 9.0的应用没有影响，对所有taskVersion > 9.0的应用，如果继续使用Apache Http接口或者引用的lib包中用到该接口时，都会出现Apache Http接口找不到的异常。

- 解决方案

在应用的AndroidManifest.xml文件的<application>中添加：

```
<uses-library android:name="org.apache.http.legacy" android:required="false"/>
```

2.3.1 服务初始化

阿里公共DNS SDK是通过DNSResolver类封装了阿里公共DNS服务请求和本地缓存实现的。用户通过DNSResolver.init(this,accountID)初始化，即可统一接入阿里公共DNS服务。(accountID为用户在控制台注册时，server端自动生成用户唯一标识)

2.3.2 设置预解析域名

在您初始化程序时，可以选择性地预先向阿里公共DNS SDK中注册您后续可能会使用到的域名，以便SDK提前解析，减少后续解析域名时请求的时延。调用以下方法设置预解析域名：

```
DNSResolver.getInstance().preLoadDomains(newString[]{TAOBAO_URL,M_TAOBAO_URL,ALIYUN_URL});
```

> **注意**
>
> 预解析接口设置的同时会实时触发异步网络请求，应该在代码逻辑上确保调用预解析接口时，已经进行了必备的初始化设置。

2.3.3 设置是否使用服务端IPV6地址

阿里公共DNS服务支持IPV4、IPV6双栈访问，通过设置DNSResolver.setEnableIPv6(boolean enable)的方法选择使用那种接入协议。当设置为ture表示使用IPV6的IP访问服务端接口，false表示使用IPV4的IP访问服务端接口。如不显式设置，默认为使用IPV4地址访问。另外当设置为IPV6时访问阿里公共DNS服务不通时则自动切换为IPV4，并且支持重试3次策略。

2.3.4 设置是否开启Short模式

阿里公共DNS的DoH JSON API返回数据类型分为全量JSON和简要IP数组格式，可以通过调用DNSResolver.setEnableShort (boolean enable)来开启或关闭short模式。如不显式设置，默认为关闭short模式。

如下方式：

```
DNSResolver.setEnableShort (true); //默认值是false,该参数用户可以不需设置，不设置则默认为false。
```

> 注意
>
> short模式为SDK调用阿里公共DNS服务返回的比较简单IP数组，可以减少回复的数据量，适用于对网络流量敏感的场景。

2.3.5 设置缓存模块最大缓存数量

```
DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER);
private static final int CACHE_MAX_NUMBER =100;//缓存域名最大count默认是100。
```

用户可以自定义最大count的值。

2.3.6 设置访问服务端协议

```
DNSResolver.setSchemaType(DNSResolver.HTTP); 默认访问为http的模式。
```

DNSResolver.HTTP，以HTTP协议访问服务端接口；

DNSResolver.HTTPS，以HTTPS协议访问服务端接口。

3.服务API

```
/**
 * 获取url对应IPv4记录的DomainInfo对象数组
 *
 * @param url例如(http://www.taobao.com)
 * @return目标url对应IPV4类型的DomainInfo对象数组
 */
 public DomainInfo[] getIPsV4DInfoByUrl(String url)

注意：DomainInfo对象中的url为替换host为IP拼接后的url,用户不需再手动url的替换拼接。

/**
  * 获取url对应IPv6记录的DomainInfo对象数组
  *
  * @param url 例如(http://m.taobao.com)
  * @return 目标url对应IPV6类型的DomainInfo对象数组
  */
```

```
    public DomainInfo[] getIPsV6DInfoByUrl(String url)


  /**
    * 获取url对应IPV4记录的DomainInfo对象
    *
    * @param url 例如(http://m.taobao.com)
    * @return 目标url对应IPV4类型的DomainInfo对象集合中随机的一个
    */
    public DomainInfo getIPV4DInfoByUrl(String url)


  /**
    * 获取url对应IPV6记录的DomainInfo对象
    *
    * @param url 例如(http://www.taobao.com)
    * @return 目标url对应IPV6类型的DomainInfo对象集合中随机的一个
    */
   public DomainInfo getIPV6DInfoByUrl(String url)


 /**
    * 获取hostName对应IPV4记录数组
    * @param hostName例如(www.taobao.com)
    * @return 返回目标hostName对应的IPV4地址的数组
    */
   public  String[] getIPsV4ByHost(String hostName)


 /**
    * 获取hostName对应IPV6记录数组
    * @param hostName例如(www.taobao.com)
    * @return 返回目标hostName对应的IPV6地址的数组
    */
   public String[] getIPsV6ByHost(String hostName)


 /**
    * 获取hostName对应IPV4记录
    * @param hostName例如(www.taobao.com)
    * @return 返回目标hostName对应的IPV4地址集合中的一个随机IPV4地址
```

```
    */
  public String getIPV4ByHost(String hostName)


/**
    * 获取hostName对应IPV6记录
    * @param hostName例如(www.taobao.com)
    * @return 返回目标hostName对应的IPV6地址集合中的一个随机IPV6地址
    */
  public String getIPV6ByHost(String hostName)


/**
 * 预加载逻辑
 *
 * @param domains
 */
public void preLoadDomains(final String[] domains)

 说明：以上返回的domainInfo对象，主要是封装了如下属性

 /**
   * id访问域名自增的id编号
    */
 public String id = null;

   /**
    * 可以直接使用的url 已经替换了host为ip后的url
    */
public String url = null;


    /**
    * 需要设置到http header里面的目标服务名称
    */
    public String host = "";


   /**
    * 返回的内容体
    */
```

```
    public String data = null;


    /**
     * 开始请求的时间
     */
    public String startTime = null;


    /**
     * 请求结束的时间，如果请求超时，该值为null
     */
    public String stopTime = null;


   /**
    * server返回的状态值200 \ 404 \ 500等
    */
  public String code = null;
```

4.SDK API使用示例

URl：传进来的访问地址，例如：http://www.taobao.com。

```
String hostname = "www.taobao.com";
String url = "http://www.taobao.com";
```

获得IPv4地址：

```
String IPV4 = DNSResolver.getInstance().getIPV4ByHost(hostname);
```

获得IPv6地址：

```
String IPV6 =  DNSResolver.getInstance().getIPV6ByUrl (url);
```

获得url对应的DomainInfo对象：

```
DomainInfo dinfo = DNSResolver.getInstance().getIPV4DInfoByUrl(url); //获取替换后地址
```

示例：

```
public class MainActivity extends AppCompatActivity {
  private Button button;
  private TextView tvInfo;
  private TextView tvResult;
  private String hostUrl = "http://www.taobao.com";
  private String hostName = “www.taobao.com” ;
  private static final String TAG = "PDnsDemo";
  private static ExecutorService pool = Executors.newSingleThreadExecutor();
  private static final String PDNS_RESULT = "pdns_result";
  private static final int SHOW_CONSOLE_TEXT = 10000;
  private Handler mHandler;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.demo_activity_main);
     init();
     initHandler();
  }
```

```
private void init() {
    tvInfo = findViewById(R.id.tv_respons_info);
    tvResult = findViewById(R.id.tv_respons);
    button = findViewById(R.id.btn_onclik);
    button.setOnClickListener(new View.OnClickListener() {
      public void onClick(View view) {
        new Thread(new Runnable() {
            @Override
            public void  run() {
       //调用阿里公共sdk里的getIPV4ByHost()方法获得目标域名解析后的IP
        String ipv4 = DNSResolver.getInstance().getIPV4ByHost(hostName);
        tvInfo.setText("您解析域名的IP是："+ ipv4);
      //调用阿里公共sdk里getIPV4DInfoByUr获取目标域名解析后的domainInfo对象中的URL，该URL为已替换原来url中host为IP的url
         DomainInfo dinfo = DNSResolver.getInstance().getIPV4DInfoByUrl(hostUrl);
             if (dinfo != null) {
                 showResponse(dinfo);
              }
           }
        }).start();
      }
    });
  }
```

```
  private void initHandler() {
    mHandler = new Handler() {
      @Override
      public void handleMessage(Message msg) {
        switch (msg.what)  {
          case SHOW_CONSOLE_TEXT:
            tvResult.setText(msg.getData().getString(PDNS_RESULT) + "\n");
            break;
        }
      }
    };
  }
```

```
private void showResponse(final DomainInfo dinfo) {
        //发送网络请求
        String requestUrl = dinfo.url;
        HttpURLConnection conn = null;
        try {
            URL url = new URL(requestUrl);
            conn = (HttpURLConnection) url.openConnection();
            //使用IP的方式进行访问时，需要设置HTTP请求头的HOST字段为原来的域名
            conn.setRequestProperty("Host", url.getHost());//设置HTTP请求头HOST字段
            DataInputStream dis = new DataInputStream(conn.getInputStream());
            int len;
            byte[] buff = new byte[4096];
            StringBuilder response = new StringBuilder();
            while ((len = dis.read(buff)) != -1) {
                response.append(new String(buff, 0, len));
            }
            Log.d(TAG, "Response: " + response.toString());
            dis.close();
            sendMessage(response.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }finally {
            if (conn != null) {
                conn.disconnect();
            }
        }
    }
```

```
    private void sendMessage(String message) {
        if (mHandler != null) {
            Message msg = mHandler.obtainMessage();
            Bundle bundle = new Bundle();
            bundle.putString(PDNS_RESULT, message);
            msg.setData(bundle);
            msg.what = SHOW_CONSOLE_TEXT;
            mHandler.sendMessage(msg);
        }
    }

}

public class DnsCacheApplication extends Application {

    private String accountID = "10001"; //可以更换成您的accountID
    private static final String TAOBAO_URL = "www.taobao.com";
    private static final String M_TAOBAO_URL = "m.taobao.com";
    private static final String ALIYUN_URL = "aliyun.com";
    private static final String BAIDU_URL = "www.baidu.com";
    private static final int CACHE_MAX_NUMBER = 100; //缓存的域名最大个数

    @Override
    public void onCreate() {
        super.onCreate();
        DNSResolver.Init(this, accountID);
        DNSResolver.setEnableShort(false);
        DNSResolver.setEnableIPv6(false);
        DNSResolver.setSchemaType(DNSResolver.HTTP);
        DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER);
        DNSResolver.getInstance().preLoadDomains(new String[]{TAOBAO_URL, M_TAOBAO_URL, ALIYUN_
URL,BAIDU_URL});
    }
}
```

注意事项

1. 通过阿里公共DNS获得域名的IP地址后，客户端可以使用这个IP发送业务请求，HTTP请求头的Host字段需改为原来的域名。
2. 为帮助用户更快的使用阿里公共DNS SDK，我们为读者提供Demo程序，读者可以

下载到本地作为参考。

请点击这里，下载Demo程序。

6.iOS SDK开发指南

本文档介绍了阿里云公共DNS iOS SDK的接入和开发方式。

1.概述

阿里云公共DNS SDK是阿里云面向广大移动开发者提供DNS域名解析服务的开发工具包。

开发者利用本SDK，可以在自己的iOS APP中轻松接入阿里云公共DNS，解决域名解析异常的问题，低成本实现域名解析精准调度。您可以参考Demo示例工程源码了解如何使用本SDK。

SDK当前版本封装了阿里公共DNS的DoH JSON API，提供接口函数给iOS APP进行域名解析，并且提供了基于TTL和LRU策略的高效域名缓存功能。在公共DNS原有功能的基础上，SDK还可以为用户带来以下优势：

- **简单易用：**

  用户仅需集成我们提供的SDK，便可接入阿里云公共DNS业务。接入方法简单易用，为用户提供更为轻松便捷的解析服务。

- **零延迟：**

  SDK内部实现了LRU的缓存机制，将每次域名解析后的IP缓存到本地；并且主动更新TTL过期缓存，保证缓存及时有效，从而帮助用户达到域名解析零延迟的效果。

2.SDK集成

2.1 SDK集成

1. 您需先在控制台注册自己的应用，获取应用的唯一标识Account ID。
2. 通过控制台的链接获取阿里公共DNS iOS SDK。
3. 获取到的SDK的 `pdns-sdk-ios.framework` 后，手动集成到自己工程中。
4. 引入系统库：
   - Foundation.framework
   - SystemConfiguration.framework
   - CoreFoundation.framework
5. 在 `application:didFinishLaunchingWithOptions:` 初始化SDK。

```
DNSResolver *resolver = [DNSResolver share];
resolver.accountId = @"您在控制台注册的应用的Account ID";
```

2.2 自动集成（Cocoapods）

1.Podfile中指定仓库位置：（Master仓库不要遗漏）

```
source 'https://github.com/CocoaPods/Specs.git'
source 'https://github.com/aliyun/aliyun-specs.git'
```

2.为工程target添加依赖：

```
pod 'AlicloudPDNS'
```

3.API介绍

3.1 accountId

必传参数，您在控制台注册自己的应用后，控制台会为此应用的生成唯一标识Account ID。

3.2 设置是否使用服务端IPV6地址

阿里云公共DNS服务支持IPv4、IPv6双栈访问。SDK默认使用IPv4地址访问DNS服务器进行解析。

如果需要通过IPv6地址访问DNS服务器进行解析（当前网络需要支持IPv6），则需要通过以下代码设置：

```
[DNSResolver share].ipv6Enable = YES;
```

3.3 short模式

阿里公共DNS的DoH JSON API返回数据分为全量JSON和简要IP数组格式。SDK默认为全量JSON格式。

若要设置为简要IP数组格式，则需要通过以下代码设置：

```
[DNSResolver share].shortEnable = YES;
```

3.4 scheme

在访问DNS服务器进行解析时，是通过http路解析，还是通过https路解析，可通过 `scheme` 属性进行设置。

SDK默认使用http路进行解析，建议使用http路进行解析，解析速度更快。若更要使用https路进行解析，则需要如下设置：

```
[DNSResolver share].scheme = DNSResolverSchemeHttps;
```

3.5 设置缓存数量

SDK自带缓存功能，第一次解析过域名后，后续再解析时，可大大提高解析速度。

SDK默认的缓存数量为100个域名。若要自定义缓存数量，可通过 `cacheCountLimit` 属性进行设置：

```
//自定义缓存数量
[DNSResolver share].cacheCountLimit = 200;
```

3.6 timeout

`timeout` 属性为域名解析的超时时间。默认超时时间为3s，用户可自定义超时时间，建议设置在2~5s之间。

3.7 预解析

由于SDK自带缓存功能，在第一次解析完域名后，后续再次解析此域名可大大提高解析速度，所以建议在app启动后，可对app中可能要解析的域名进行预加载。

代码示例：

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  // Override point for customization after application launch.

  DNSResolver *resolver = [DNSResolver share];
  resolver.accountId = @"你的Accound ID";

  //对后续可能要解析的域名进行预加载
  [resolver preloadIpv4Domains:@[@"域名1", @"域名2", @"域名3"] complete:^{
     //所有域名预加载完成

  }];

  return YES;
}
```

4.服务API

代码示例：

```
/// 获取域名解析后的IPv4信息数组
/// @param domain          域名
/// @param complete      回调(所有域名信息)
- (void)getIpv4InfoWithDomain:(NSString *)domain complete:(void(^)(NSArray<DNSDomainInfo *> *domainInfoArray))complete;

/// 获取域名解析后的IPv6信息数组
/// @param domain          域名
/// @param complete      回调(所有域名信息)
- (void)getIpv6InfoWithDomain:(NSString *)domain complete:(void(^)(NSArray<DNSDomainInfo *> *domainInfoArray))complete;

/// 获取域名解析后的IPv4信息
/// @param domain          域名
/// @param complete      回调(所有域名信息中随机一个)
- (void)getRandomIpv4InfoWithDomain:(NSString *)domain complete:(void(^)(DNSDomainInfo *domainInfo))complete;

/// 获取域名解析后的IPv6信息
/// @param domain          域名
/// @param complete      回调(所有域名信息中随机一个)
```

```
- (void)getRandomIpv6InfoWithDomain:(NSString *)domain complete:(void(^)(DNSDomainInfo *domainI
nfo))complete;

/// 获取域名解析后的IPv4地址数组
/// @param domain        域名
/// @param complete      回调(所有ip地址)
- (void)getIpv4DataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray
))complete;

/// 获取域名解析后的IPv6地址数组
/// @param domain        域名
/// @param complete      回调(所有ip地址)
- (void)getIpv6DataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray
))complete;

/// 获取域名解析后的IPv4地址
/// @param domain        域名
/// @param complete      回调(所有ip地址中随机一个)
- (void)getRandomIpv4DataWithDomain:(NSString *)domain complete:(void(^)(NSString *data))complet
e;

/// 获取域名解析后的IPv6地址
/// @param domain        域名
/// @param complete      回调(所有ip地址中随机一个)
- (void)getRandomIpv6DataWithDomain:(NSString *)domain complete:(void(^)(NSString *data))complet
e;

/// 预解析域名IPv4信息，可在程序启动时调用，加快后续域名解析速度
/// @param domainArray  域名数组
/// @param complete      解析完成后回调
- (void)preloadIpv4Domains:(NSArray<NSString *> *)domainArray complete:(void(^)(void))complete;

/// 预解析域名IPv6信息，可在程序启动时调用，加快后续域名解析速度
/// @param domainArray  域名数组
/// @param complete      解析完成后回调
- (void)preloadIpv6Domains:(NSArray<NSString *> *)domainArray complete:(void(^)(void))complete;

/// 直接从缓存中获取IPv4解析结果，无需等待. 如无缓存，或有缓存但已过期，返回 nil
/// @param domain   域名
- (NSArray<NSString *> *)getIpv4ByCacheWithDomain:(NSString *)domain;
```

```
/// 直接从缓存中获取IPv6解析结果，无需等待. 如无缓存，或有缓存但已过期，返回 nil
/// @param domain   域名
- (NSArray<NSString *> *)getIpv6ByCacheWithDomain:(NSString *)domain;
```

5. API使用示例

5.1 设置基本信息

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  // Override point for customization after application launch.

  //唯一初始化方式
  DNSResolver *resolver = [DNSResolver share];
//必传参数
  resolver.accountId = @"您在控制台注册的应用的Account ID";

  return YES;
}
```

5.2 域名解析接口

SDK提供了IPv4和IPv6两种解析域名的服务，可在 `DNSResolver.h` 头文件中查看。 下面以其中一个IPv4解析方法为例，进行说明。

接口声明：

```
/// 获取域名解析后的IPv4地址
/// @param domain        域名
/// @param complete      回调(所有ip地址中随机一个)
- (void)getRandomIpv4DataWithDomain:(NSString *)domain complete:(void(^)(NSString *data))complete;
```

接口调用示例：

```
[[DNSResolver share] getRandomIpv4DataWithDomain:@"www.taobao.com" complete:^(NSString *data) {
  //data为域名www.taobao.com所对应的IPv4地址
  if (data.length > 0) {
     //TODO: 使用IP地址进行url连接

  }
}];
```

5.3 直接从缓存中拿解析结果

接口声明：

```
/// 直接从缓存中获取IPv4解析结果，无需等待. 如无缓存，或有缓存但已过期，返回 nil
/// @param domain  域名
- (NSArray<NSString *> *)getIpv4ByCacheWithDomain:(NSString *)domain;

/// 直接从缓存中获取IPv6解析结果，无需等待. 如无缓存，或有缓存但已过期，返回 nil
/// @param domain  域名
- (NSArray<NSString *> *)getIpv6ByCacheWithDomain:(NSString *)domain;
```

调用示例：

```
NSArray *result = [[DNSResolver share] getIpv4ByCacheWithDomain:@"域名"];
//拿到缓存结果
if (result.count > 0) {
  //TODO:使用ip地址进行url连接

}
```

注意：直接从缓存中拿缓存结果，速度较快，但无缓存、或者有缓存但缓存的解析结果已过期时，返回结果会为nil。

6.注意事项

1. `pdns-sdk-ios.framework` 支持最低版本为iOS9.0。
2. 使用http路请求时，需要在 `Info.plist` 中设置 `App Transport Security Settings->Allow Arbitrary Loads` 为 `YES` 。
3. 通过阿里公共DNS获得域名的IP地址后，客户端可以使用这个IP发送业务请求，HTTP请求头的Host字段需改为原来的域名。

   例如：

   ```
   //ip为原域名解析后的ip地址
   NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"https://%@", ip]];
   NSMutableURLRequest *mutableReq = [NSMutableURLRequest requestWithURL:url cachePolicy:NS
   URLRequestUseProtocolCachePolicy timeoutInterval: 10];
   //设置host
   [mutableReq setValue:@"原域名" forHTTPHeaderField:@"host"];
   ```

4. 为帮助用户更快的使用阿里公共DNS iOS SDK，我们为开发者提供Demo程序，您可以下载到本地作为参考。请点击这里，下载Demo程序。

7.获取DoH用户基本信息

调用DescribeDohUserInfo获取DoH用户基本信息

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohUserInfo	系统规定参数。取值：DescribeDohUserInfo。
Lang	String	否	en	语言
StartDate	String	否	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即： `StartDate &gt;= Now - 90` 。
EndDate	String	否	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。

返回数据

名称	类型	示例值	描述
DomainCount	Integer	123	已接入域名数量
PdnsId	Long	12345678	公共DNS用户ID
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码

名称	类型	示例值	描述
SubDomainCount	Integer	123	已接入子域名数量

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohUserInfo
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<SubDomainCount>123</SubDomainCount>
<PdnsId>12345678</PdnsId>
<DomainCount>123</DomainCount>
```

`JSON` 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "SubDomainCount": "123",
 "PdnsId": "12345678",
 "DomainCount": "123"
}
```

错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。

8.获取DoH账户请求量统计

调用DescribeDohAccountStatistics获取DoH账户请求量统计概览

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohAccountStatistics	系统规定参数。取值：DescribeDohAccountStatistics。
EndDate	String	是	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。
StartDate	String	是	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即：`StartDate >= Now - 90`。
Lang	String	否	en	语言

返回数据

名称	类型	示例值	描述
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码
Statistics	Array of Statistic		统计数据列表
Timestamp	Long	1544976000000	时间戳
TotalCount	Long	3141592653	请求总量
V4HttpCount	Long	3141592653	IPv4 HTTP 请求量

名称	类型	示例值	描述
V4HttpsCount	Long	3141592653	IPv4 HTTPS 请求量
V6HttpCount	Long	3141592653	IPv6 HTTP 请求量
V6HttpsCount	Long	3141592653	IPv6 HTTPS 请求量

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohAccountStatistics
&EndDate=2019-07-04
&StartDate=2019-07-04
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<Statistics>
  <TotalCount>3141592653</TotalCount>
  <V4HttpCount>3141592653</V4HttpCount>
  <V6HttpCount>3141592653</V6HttpCount>
  <Timestamp>1544976000000</Timestamp>
  <V4HttpsCount>3141592653</V4HttpsCount>
  <V6HttpsCount>3141592653</V6HttpsCount>
</Statistics>
```

`JSON` 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "Statistics": [{
  "TotalCount": "3141592653",
  "V4HttpCount": "3141592653",
  "V6HttpCount": "3141592653",
  "Timestamp": "1544976000000",
  "V4HttpsCount": "3141592653",
  "V6HttpsCount": "3141592653"
 }]
}
```


错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。

9.获取DoH子域名请求量统计

调用DescribeDohSubDomainStatistics获取DoH子域名请求量统计

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohSubDomainStatistics	系统规定参数。取值：DescribeDohSubDomainStatistics。
EndDate	String	是	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。
StartDate	String	是	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即：`StartDate >= Now - 90`。
SubDomain	String	是	www.example.com	要查看统计数据的子域名
Lang	String	否	en	语言

返回数据

名称	类型	示例值	描述
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码
Statistics	Array of Statistic		统计数据列表
Timestamp	Long	1544976000000	时间戳
TotalCount	Long	3141592653	请求总量

名称	类型	示例值	描述
V4HttpCount	Long	3141592653	IPv4 HTTP 请求量
V4HttpsCount	Long	3141592653	IPv4 HTTPS 请求量
V6HttpCount	Long	3141592653	IPv6 HTTP 请求量
V6HttpsCount	Long	3141592653	IPv6 HTTPS 请求量

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohSubDomainStatistics
&EndDate=2019-07-04
&StartDate=2019-07-04
&SubDomain=www.example.com
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<Statistics>
  <TotalCount>3141592653</TotalCount>
  <V4HttpCount>3141592653</V4HttpCount>
  <V6HttpCount>3141592653</V6HttpCount>
  <Timestamp>1544976000000</Timestamp>
  <V4HttpsCount>3141592653</V4HttpsCount>
  <V6HttpsCount>3141592653</V6HttpsCount>
</Statistics>
```

`JSON` 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "Statistics": [{
  "TotalCount": "3141592653",
  "V4HttpCount": "3141592653",
  "V6HttpCount": "3141592653",
  "Timestamp": "1544976000000",
  "V4HttpsCount": "3141592653",
  "V6HttpsCount": "3141592653"
 }]
}
```


错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。

10.获取DoH域名的请求量统计列表

调用DescribeDohDomainStatisticsSummary获取DoH域名的请求量统计列表

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohDomainStatisticsSummary	系统规定参数。取值：DescribeDohDomainStatisticsSummary。
PageNumber	Integer	是	1	当前页数，起始值为1，默认为1
StartDate	String	是	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即：`StartDate >= Now - 90`。
Lang	String	否	en	语言
PageSize	Integer	否	20	分页查询时设置的每页行数，最大值100，默认为20
EndDate	String	否	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。
DomainName	String	否	example.com	域名

返回数据

名称	类型	示例值	描述
PageNumber	Integer	1	当前页码
PageSize	Integer	20	本次查询获取的数量

名称	类型	示例值	描述
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码
Statistics	Array of Statistic		统计数据列表
DomainName	String	example.com	域名
HttpCount	Long	3141592653	HTTP请求量
HttpsCount	Long	3141592653	HTTPS请求量
IpCount	Long	20	IP数量
TotalCount	Long	14141592653	请求总量
V4HttpCount	Long	3141592653	IPv4 HTTP 请求量
V4HttpsCount	Long	3141592653	IPv4 HTTPS 请求量
V6HttpCount	Long	3141592653	IPv6 HTTP 请求量
V6HttpsCount	Long	3141592653	IPv6 HTTPS 请求量
TotalItems	Integer	300	总数据条数
TotalPages	Integer	50	总页数

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohDomainStatisticsSummary
&PageNumber=1
&StartDate=2019-07-04
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<PageSize>20</PageSize>
<PageNumber>1</PageNumber>
<TotalPages>50</TotalPages>
<TotalItems>300</TotalItems>
<Statistics>
  <TotalCount>14141592653</TotalCount>
  <IpCount>20</IpCount>
  <DomainName>example.com</DomainName>
  <V4HttpCount>3141592653</V4HttpCount>
  <V6HttpCount>3141592653</V6HttpCount>
  <HttpCount>3141592653</HttpCount>
  <HttpsCount>3141592653</HttpsCount>
  <V4HttpsCount>3141592653</V4HttpsCount>
  <V6HttpsCount>3141592653</V6HttpsCount>
</Statistics>
```

JSON 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "PageSize": "20",
 "PageNumber": "1",
 "TotalPages": "50",
 "TotalItems": "300",
 "Statistics": [{
  "TotalCount": "14141592653",
  "IpCount": "20",
  "DomainName": "example.com",
  "V4HttpCount": "3141592653",
  "V6HttpCount": "3141592653",
  "HttpCount": "3141592653",
  "HttpsCount": "3141592653",
  "V4HttpsCount": "3141592653",
  "V6HttpsCount": "3141592653"
 }]
}
```

错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。

11.获取DoH子域名的请求量统计列表

调用DescribeDohSubDomainStatisticsSummary获取DoH子域名的请求量统计列表

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohSubDomainStatisticsSummary	系统规定参数。取值：DescribeDohSubDomainStatisticsSummary。
DomainName	String	是	example.com	域名
PageNumber	Integer	是	1	当前页数，起始值为1，默认为1
StartDate	String	是	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即：`StartDate >= Now - 90`。
Lang	String	否	en	语言
PageSize	Integer	否	20	分页查询时设置的每页行数，最大值100，默认为20
EndDate	String	否	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。
SubDomain	String	否	www.example.com	子域名

返回数据

名称	类型	示例值	描述
PageNumber	Integer	1	当前页码

名称	类型	示例值	描述
PageSize	Integer	10	本次查询获取的数量
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码
Statistics	Array of Statistic		统计数据列表
HttpCount	Long	3141592653	HTTP请求量
HttpsCount	Long	3141592653	HTTPS请求量
IpCount	Long	20	IP数量
SubDomain	String	www.example.com	子域名
TotalCount	Long	14141592653	总请求量
V4HttpCount	Long	3141592653	IPv4 HTTP 请求量
V4HttpsCount	Long	3141592653	IPv4 HTTPS 请求量
V6HttpCount	Long	3141592653	IPv6 HTTP 请求量
V6HttpsCount	Long	3141592653	IPv6 HTTPS 请求量
TotalItems	Integer	100	总数据条数
TotalPages	Integer	50	总页数

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohSubDomainStatisticsSummary
&DomainName=example.com
&PageNumber=1
&StartDate=2019-07-04
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<PageSize>10</PageSize>
<PageNumber>1</PageNumber>
<TotalPages>50</TotalPages>
<TotalItems>100</TotalItems>
<Statistics>
  <SubDomain>www.example.com</SubDomain>
  <TotalCount>14141592653</TotalCount>
  <IpCount>20</IpCount>
  <V4HttpCount>3141592653</V4HttpCount>
  <V6HttpCount>3141592653</V6HttpCount>
  <HttpCount>3141592653</HttpCount>
  <HttpsCount>3141592653</HttpsCount>
  <V4HttpsCount>3141592653</V4HttpsCount>
  <V6HttpsCount>3141592653</V6HttpsCount>
</Statistics>
```

`JSON` 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "PageSize": "10",
 "PageNumber": "1",
 "TotalPages": "50",
 "TotalItems": "100",
 "Statistics": [{
  "SubDomain": "www.example.com",
  "TotalCount": "14141592653",
  "IpCount": "20",
  "V4HttpCount": "3141592653",
  "V6HttpCount": "3141592653",
  "HttpCount": "3141592653",
  "HttpsCount": "3141592653",
  "V4HttpsCount": "3141592653",
  "V6HttpsCount": "3141592653"
 }]
}
```


错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。

12.获取DoH域名的请求量统计

调用DescribeDohDomainStatistics获取DoH域名请求量统计概览

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDohDomainStatistics	系统规定参数。取值：DescribeDohDomainStatistics。
DomainName	String	是	example.com	域名
EndDate	String	是	2019-07-04	查询的结束时间，格式：YYYY-MM-DD。 默认为查询当天的时间。
StartDate	String	是	2019-07-04	查询的开始时间，格式：YYYY-MM-DD。 只能查询最近90天的记录，即：`StartDate >= Now - 90`。
Lang	String	否	en	语言

返回数据

名称	类型	示例值	描述
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	唯一请求识别码
Statistics	Array of Statistic		统计数据列表
Timestamp	Long	1544976000000	时间戳
TotalCount	Long	3141592653	请求总量

名称	类型	示例值	描述
V4HttpCount	Long	3141592653	IPv4 HTTP 请求量
V4HttpsCount	Long	3141592653	IPv4 HTTPS 请求量
V6HttpCount	Long	3141592653	IPv6 HTTP 请求量
V6HttpsCount	Long	3141592653	IPv6 HTTPS 请求量

示例

请求示例

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohDomainStatistics
&DomainName=example.com
&EndDate=2019-07-04
&StartDate=2019-07-04
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<RequestId>0F32959D-417B-4D66-8463-68606605E3E2</RequestId>
<Statistics>
  <TotalCount>3141592653</TotalCount>
  <V4HttpCount>3141592653</V4HttpCount>
  <V6HttpCount>3141592653</V6HttpCount>
  <Timestamp>1544976000000</Timestamp>
  <V4HttpsCount>3141592653</V4HttpsCount>
  <V6HttpsCount>3141592653</V6HttpsCount>
</Statistics>
```

`JSON` 格式

```
{
 "RequestId": "0F32959D-417B-4D66-8463-68606605E3E2",
 "Statistics": [{
  "TotalCount": "3141592653",
  "V4HttpCount": "3141592653",
  "V6HttpCount": "3141592653",
  "Timestamp": "1544976000000",
  "V4HttpsCount": "3141592653",
  "V6HttpsCount": "3141592653"
 }]
}
```


错误码

访问错误中心查看更多错误码。

访问错误中心查看更多错误码。