ALIBABA CLOUD

Alibaba Cloud

阿里云公共DNS

API Reference

Document Version: 20220705

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Danger:** Resetting will result in the loss of user configuration data.
🔔 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	🔔 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
🔈 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔈 **Notice:** If the weight is set to 0, the server no longer receives new requests.
ⓘ Note	A note indicates supplemental instructions, best practices, tips, and other content.	ⓘ **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview 05
2.DNS over HTTPS (DoH) 06
3.JSON API for DoH 07
4.DNS over TLS (DoT) 11
5.SDK for Android developer guide 13
6.SDK for iOS developer guide 24
7.DescribeDohSubDomainStatisticsSummary 30
8.DescribeDohDomainStatistics 33

1.Overview

Alibaba Cloud public DNS provides fast, stable, and secure DNS resolution for Internet users. Traditional DNS queries and responses are transmitted in UDP or TCP plaintext, which poses the following risks:

- Network listening: DNS queries and responses are not encrypted during transmission even if you access a website in HTTPS mode.
- DNS hijacking: Traditional DNS responses may be tampered with, and accesses may be routed to phishing websites or malicious websites.
- Interference from intermediate devices: The interference includes firewall interception or tampering, filtering of domain names, and IP fragmentation.

In response to the preceding challenges, Alibaba Cloud public DNS provides the secure transmission services of Domain Name System (DNS) in compliance with DNS over HTTPS (DoH) as specified in the RFC 8484 specifications and DNS over TLS (DoT) as specified in the RFC 7858 specifications. It also supports the DoH- and DoT-based secure transmission modes and provides HTTP- and HTTPS-based JSON APIs for DoH. The secure transmission services of DNS apply to mobile applications, browsers, operating systems, Internet of Things (IoT) devices, gateways, and routers. These services encrypt DNS queries during transmission, which improves the security, resolution stability, and privacy protection of your Internet access.

In addition, TCP or HTTP connections are used between clients and DNS servers to provide secure transmission services of DNS. On one hand, Alibaba Cloud public DNS can be used for accurate location-based DNS resolution and traffic scheduling. On the other hand, dynamic DNS changes can take effect in seconds based on the end-to-end DNS connection characteristics.

Note: Alibaba Cloud public DNS supports TLS 1.2 and TLS 1.3 for the DoH and DoT transmission services.

2.DNS over HTTPS (DoH)

Alibaba Cloud public DNS provides DNS resolution by using the TLS-encrypted HTTP connections as specified in the RFC 8484 specifications.

URI APIs for DNS over HTTPS (DoH) (only the TLS API is provided):

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

Notice: alidns_ip is the A record of dns.alidns.com and can be one of the following IP addresses: 223.5.5.5 and 223.6.6.6. The user_id is the Account ID in Alibaba Cloud Console, for example: https://9999.alidns.com/dns-query?DNS = uGkBAAABAAAAAAAAB2FsaWJhYmEDY29tAAABAAE

Request method: GET

Request parameters

Parameter	Data type	Description
dns	string	This parameter is the DNS parameter set in the process of the DoH client converting the normal DNS query message into HTTP request. The specific content is to convert the BINARY DNS request message into base64url-encoded string. Please refer to RFC8484 for the interface definition.

Sample request: https://dns.alidns.com/dns-query?dns=uGkBAAABAAAAAAAAB2FsaWJhYmEDY29tAAABAAE

The following DNS binary data is returned:

```
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
```

Note: As specified in the RFC8484 specifications, the DoH service applies to DNS HTTPS tunnels and DNS data access at the application layer. The binary format of DNS wireformat is not suitable for applications. If you use this format, the domain names in DNS messages must be compressed. The JSON API for DoH provides a more suitable access mode for applications.

3.JSON API for DoH

You can use the following URLs to call the JSON API for DNS over HTTPS (DoH). Both Transport Layer Security (TLS) and non-TLS API operations are provided.

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

https://alidns_ip/resolve?

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

http://alidns_ip/resolve?

> Notice
>
> alidns_ip is the A record of dns.alidns.com and can be one of the following IP addresses: 223.5.5.5 and 223.6.6.6.

Request method:GET

Request parameters

Parameter	Type	Description	Instance	Required/Optional and default value
name	string	The domain name in the request.	name=www.taobao.com.	Required. No default value.
type	integer	The request type.	type=1	Optional. Default value: 1.
edns_client_subnet	IP	ECS IP	edns_client_subnet=1.2.3.4/24	Used by the DNS proxy. This parameter does not apply to common clients.
short	boolean	Specifies whether to enable the short mode.	short=true or short=1	Optional. The short mode is disabled by default.
uid	string	The user ID. You can obtain the value from Account ID in the DNS console.	uid=9999	Optional.
did	string	The ID of the device.	did=afck0100	Optional.

edns_client_subnet parameter:

The edns_client_subnet parameter is designed to support EDNS Client Subnet (ECS) that is specified in the RFC 7871 specifications. ECS is a DNS extension that forwards the subnet information of users to the authoritative DNS server to achieve accurate DNS resolution and traffic scheduling. A long mask provides accurate address information, but a short mask better protects user privacy. We recommend that you use 24 as the mask length.

Note: This parameter is designed for scenarios in which the DNS proxy uses the JSON API for DoH. After the DNS proxy receives DNS queries from a user, it uses the edns_client_subnet parameter to pass the subnet information of the user to Alibaba Cloud Public DNS. Then, Alibaba Cloud Public DNS passes the subnet information to the authoritative DNS server.

For example, if the value of edns_client_subnet is 1.2.3.4/24, the authoritative DNS server selects a DNS link for the user based on the prefix of 1.2.3.4/24.

Values of the type parameter

Record type	ID	Description	Example (taobao.com and www.taobao.com used in the example)
A	1	An IPv4 record that maps a domain name to an IPv4 address.	101.37.183.171
NS	2	A name server record.	ns1.taobao.com.
CNAME	5	An alias record that associates a domain name with another domain name.	www.taobao.com.danuoyi.tbcache.com.
SOA	6	A Start of Authority (SOA) record that contains administrative information about a zone.	ns4.taobao.com. hostmaster.alibabadns.com. 2018011109 3600 1200 3600 360
TXT	16	A TXT record.	"v=spf1 include:spf1.staff.mail.aliyun.com -all"
AAAA	28	An IPv6 record that maps a domain name to an IPv6 address.	240e:e1:f300:1:3::3fa

Sample requests

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

Sample responses

```
{
    "Status":0,
    "TC":false,
    "RD":true,
    "RA":true,
    "AD":false,
    "CD":false,
    "Question": {       // Request segment
        "name":"www.taobao.com.",
        "type":1
    },
    "Answer": [          // Response segment
        {
            "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"
        }
    ]
    // The Authority segment. Data in this segment must be consistent with data in the Answer segment.
    // The Additional segment. Data in this segment must be consistent with data in the Answer segment.
    // The edns_client_subnet parameter can be set to 1.2.3.4/24.
}
```

The following table describes the return values.

Parameter	Description	Example
Status	RCODE of the DNS packet header.	0: noerror 2: servfail 3: nxdomain
TC	TrunCation (TC) of the DNS packet header. This parameter specifies whether the packet can be truncated.	false (This is the value in most cases.)
RD	Recursion Desired (RD) of the DNS packet header. This parameter specifies whether recursion is desired.	true (This is the value in most cases.)
RA	Recursion Available (RA) of the DNS packet header. This parameter specifies whether recursion is available.	true (This is the value in most cases.)
AD/CD	The identifier of the DNS packet header.	false (This is the value in most cases.)
Question	The DNS request segment.	
Answer	The DNS response segment.	
name	The domain name. Both Question and Answer contain the domain name.	www.taobao.com.
type	The request type. For more information, see **Values of the type parameter**.	
TTL	The maximum duration for which the response value is cached in the server, in seconds.	3600
data	The response result, which is related to the type parameter.	

Example of the response in short mode:

In short mode, responses that correspond to the type parameter in the request are extracted.

For example, if the DNS record type in the request for www.taobao.com is the A record, the following request Uniform Resource Locator (URL) is used:

```
http://223.5.5.5/resolve?name=www.taobao.com&type=A&short=1
```

The following code shows the response:

```
["221.229.203.214","61.155.221.227","221.229.203.213"]
```

If the DNS record type in the request for www.taobao.com is the canonical name (CNAME) record, the following request URL is used:

```
http://223.5.5.5/resolve?name=www.taobao.com&type=CNAME&short=1
```

The following code shows the response:

```
["www.taobao.com.danuoyi.tbcache.com."]
```

Failure response:

If the request fails, the HTTP status code is 4xx or 5xx and the error code for debugging or error reporting is returned. The result is displayed in the JSON format.

Failure response example:

```
{
    "code":"UrlParameterError"
}
```

The following table describes the error codes.

Error code	HTTP status code	Description
UrlParameterError	400	The error message returned because the parameter format is invalid.
NoPermission	401	The error message returned because user authentication failed.
UrlPathError	404	The error message returned because the URL is invalid.
NoResponse	500	The error message returned because no response is returned within a specified period of time.

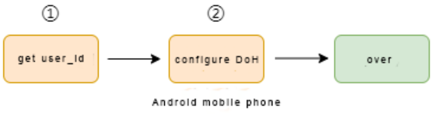
Note:You can call the JSON API for DoH on your client applications or mobile apps to resolve DNS records.

4.DNS over TLS (DoT)

This topic describes the access process of DNS over TLS (DoT).

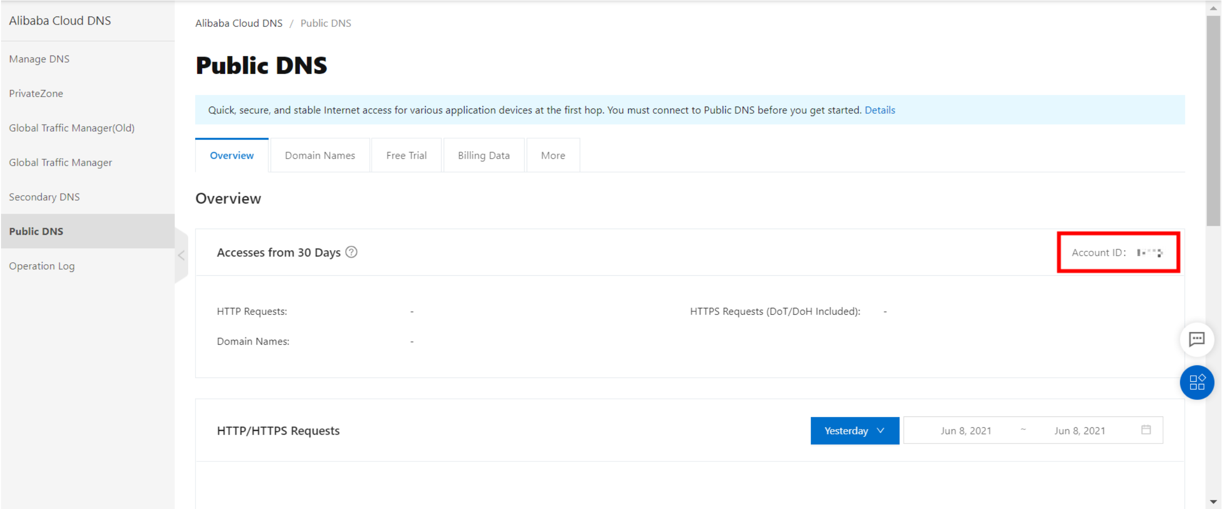
Alibaba Cloud Public DNS provides DNS resolution by using the TLS-encrypted TCP connections as specified in the RFC 7858 specifications.

The following figure shows the access process of DoT.



Obtain user_id

After Alibaba Cloud Public DNS is activated in the Alibaba Cloud DNS console, you can view the account ID on the Overview page, as shown in the following figure.



Public DNS

Quick, secure, and stable Internet access for various application devices at the first hop. You must connect to Public DNS before you get started. Details

Overview

Accesses from 30 Days

Account ID: [redacted]

HTTP Requests: -

HTTPS Requests (DoT/DoH Included): -

Domain Names: -

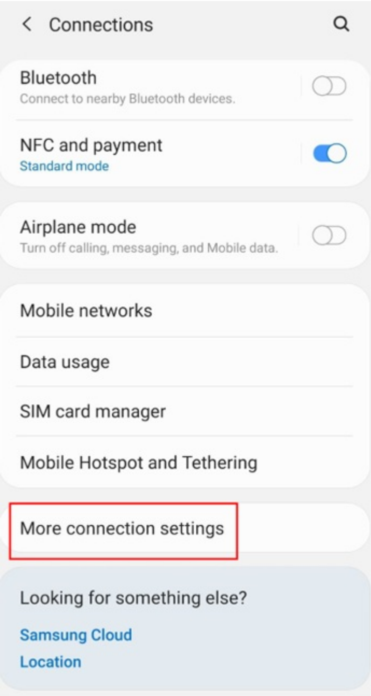
HTTP/HTTPS Requests

Jun 8, 2021 ~ Jun 8, 2021

Access example of Android mobile phones

If you are using an Android mobile phone, you can configure the domain name and IP address of Alibaba Cloud Public DNS in the settings of your mobile phone to obtain DoT-based secure transmission services.

The following example shows the configuration on a Android mobile phone. In this example, the account ID is 9999.



Connections

Bluetooth
Connect to nearby Bluetooth devices.

NFC and payment
Standard mode

Airplane mode
Turn off calling, messaging, and Mobile data.

Mobile networks

Data usage

SIM card manager

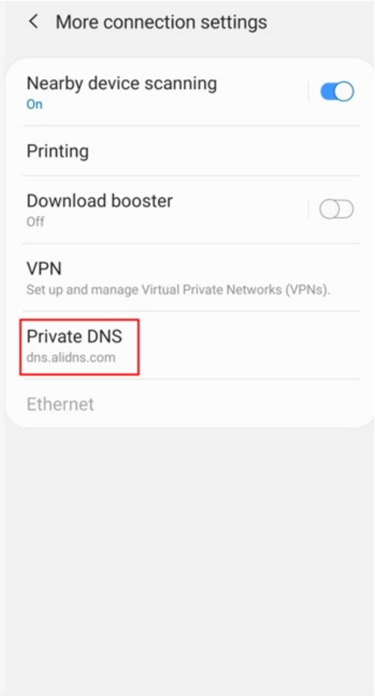
Mobile Hotspot and Tethering

More connection settings

Looking for something else?

Samsung Cloud

Location



More connection settings

Nearby device scanning
On

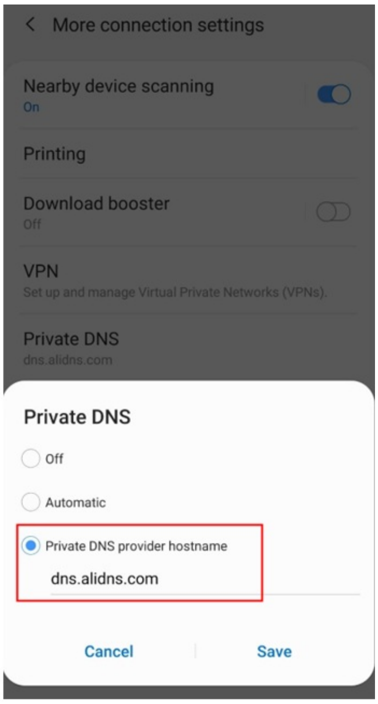
Printing

Download booster
Off

VPN
Set up and manage Virtual Private Networks (VPNs).

Private DNS
dns.alidns.com

Ethernet



Private DNS

○ Off

○ Automatic

● Private DNS provider hostname

dns.alidns.com

Cancel Save

Note:The preceding access process is suitable for the commercial edition of Alibaba Cloud Public DNS. If you need only to access the free edition of Alibaba Cloud Public DNS, see Access the free edition of Alibaba Cloud Public DNS.

5.SDK for Android developer guide

This topic describes how to develop Alibaba Cloud Public DNS SDK for Android and use the SDK to access Alibaba Cloud Public DNS.

1. Overview

Alibaba Cloud Public DNS SDK is developed by Alibaba Cloud to provide a DNS resolution service to mobile developers.

The SDK allows mobile developers to easily access Alibaba Cloud Public DNS in their Android apps. This prevents DNS resolution errors and implements precise scheduling of DNS queries at a low cost. The demo project source code provides you with an example of how to integrate the SDK with an Android app and how to use Alibaba Cloud Public DNS in the Android app.

The current version of SDK encapsulates the JSON API for DNS over HTTPS (DoH) of Alibaba Cloud Public DNS. The SDK provides Java function interfaces for Android apps to perform DNS resolution and can cache domain name resolution results based on time-to-live (TTL) and least recently used (LRU) policies in an effective manner. The SDK provides the following benefits based on the features of Alibaba Cloud Public DNS:

- **Easy to use**

  To use Alibaba Cloud Public DNS, you need to only integrate the SDK with your app. This way, you can use the resolution service in an easier and more convenient manner.

- **DNS resolution without a delay**

  The SDK implements the caching algorithm of LRU so that the IP addresses obtained from DNS resolution are cached on your local server. The SDK also automatically updates the cached data and deletes expired data based on TTL. This way, the cached data remains valid and DNS resolution can be performed without a delay.

2. Use the SDK

2.1 Integration of the SDK JAR package

1. Log on to the Alibaba Cloud DNS console. In the left-side navigation pane, click Public DNS. On the Public DNS page, activate Alibaba Cloud Public DNS. The Overview tab displays the unique account ID that is automatically generated for your Alibaba Cloud account. You can use Alibaba Cloud Public DNS in your app based on the account ID.
2. Click the link in the console to obtain the SDK JAR package of Alibaba Cloud Public DNS.
3. Decompress the alidns_android_sdk.jar file of the SDK to the libs directory of your project to install the SDK. Then, you can use the SDK.

2.2 Integration of a Gradle package by using Maven

Add the following code to the build.gradle file:

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

Add the file information that you want to reference:

```
dependencies {
     implementation 'com.alibaba.pdns:alidns-android-sdk:2.1.0'
     implementation 'com.google.code.gson:gson:2.8.5'
}
```

2.3 App initialization

```
public class DnsCacheApplication extends Application{

    private String accountID="***"; // Configure the account ID that is used to access the SDK in the Alibaba Cloud DNS console.
    private static final String TAOBAO_HOST_NAME = "www.taobao.com";// Replace www.taobao.com with the domain name that you want to preload.
    private static final String ALIYUN_HOST_NAME = "www.aliyun.com"; // Replace www.aliyun.com with the domain name that you want to preload.
    private static final int CACHE_MAX_NUMBER=100; // Specify the maximum number of domain names that can be cached. The default value of CACHE_MAX_NUMBER is 100.
    private static final int MAX_NEGATIVE_CACHE = 30; // Specify the maximum TTL for negative caching. The default value of MAX_NEGATIVE_CACHE is 30. Unit: seconds.
    private static final int MAX_TTL_CACHE = 1 * 60 * 60; // Specify the maximum TTL for caching. The default value of MAX_TTL_CACHE is 3600. Unit: seconds.

    @Override
    public void onCreate() {
       super.onCreate();
       DNSResolver.Init(this, accountID);    // Configure the account ID that is used to access the SDK in the Alibaba Cloud DNS console.
       DNSResolver.setAccessKeySecret(""); // Configure the accessKeySecret parameter that is used to access the SDK in the Alibaba Cloud DNS console.
       DNSResolver.setAccessKeyId("");      // Configure the accessKeyId parameter that is used to access the SDK in the Alibaba Cloud DNS console.
       DNSResolver.setEnableShort(false); // Specify whether to enable the short mode. The default value is false, which indicates that the short mode is not enabled.
       DNSResolver.setEnableIPv6(false);   // Specify whether to enable IPv6 access. The default value is false, which indicates that this feature is not enabled.
       DNSResolver.setEnableCache(true); // Specify whether to enable caching. The default value is true, which indicates that this feature is enabled.
       DNSResolver.setEnableSpeedTest(false); // Specify whether to enable connection speed testing of IP addresses. The default value is false, which indicates that this feature is not enabled.
       DNSResolver.setEnableSchedulePrefetch(true); // Specify whether to enable the scheduled update of expired cache data. The default value is true, which indicates that this feature is enabled.
       DNSResolver.setMaxTtlCache(MAX_TTL_CACHE); // Specify the maximum TTL for caching. The default value is 3600. Unit: seconds.
       DNSResolver.setSpeedPort(DNSResolver.PORT_80); // Specify the port number for IP socket detection. The default port number is 80.
       DNSResolver.setMaxNegativeCache(MAX_NEGATIVE_CACHE); // Specify the maximum TTL for negative caching. The default value is 30. Unit: seconds.
       DNSResolver.setSchemaType(DNSResolver.HTTP); // Specify whether the access protocol is HTTP or HTTPS.
       DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER); // Specify the maximum number of domain name resolution results that can be cached. The default value is 100.
       DNSResolver.getInstance().preLoadDomains(new String[]{TAOBAO_HOST_NAME,ALIYUN_HOST_NAME}); // Configure the domain names that you want to pre-resolve and replace the domain names that you want to preload with the domain names that you want to use Alibaba Cloud DNS to resolve.
    }
}
```

When you access Alibaba Cloud Public DNS SDK, we recommend that you integrate the SDK into the Application class of your Android app.

> Note
>
> DNSResolver is the core class of Alibaba Cloud Public DNS SDK. DNSResolver encapsulates the JSON API for DoH provided by Alibaba Cloud Public DNS to resolve the destination domain name to the related IP address.

When you use the SDK for an Android project, make sure that you have configured the following access permissions:

```
<!--Permissions that need to be configured-->
   <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"/>
```

Precautions:

(1) If a mobile phone of Android 9.0 sends an HTTP request, the error message "cleartext HTTP traffic no permitted" is returned.

- Cause: By default, plaintext network access is disabled on Android mobile phones from Android 9.0 (API level 28 for Android SDK Platform).
- Solution

Add the following information to <application> in the AndroidManifest.xml file of your app:

android:usesCleartextTraffic="true"

```
<application
        android:name=".DnsCacheApplication"
        android:icon="@mipmap/ic_launcher"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:usesCleartextTraffic="true">
```

(2) For Android 9.0, the error message "Didn't find class BasicHttpParams" is returned.

- Cause: The Apache HTTP client is no longer supported.

Google no longer supports Apache HTTP clients from Android 6.0. In Android 9.0 and later versions, org.apache.http.legacy is removed from bootclasspath.

This modification does not affect most apps whose taskVersion is earlier than Android 9.0. However, if the Apache HTTP interface is used or the referenced lib package uses this interface when you use an app whose taskVersion is later than Android 9.0, the error message "Didn't find class BasicHttpParams" is returned.

- Solution

Add the following information to <application> in the AndroidManifest.xml file of your application:

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

(3) You can use SDK 2.0 or later to enable authentication to prevent user identities from being used by unauthorized third parties.

If you do not configure the parameters that are provided in the following code, the authentication feature is not enabled. To enable this feature, create an AccessKey ID and an AccessKey secret in the Alibaba Cloud DNS console. For more information, see Service authentication. You can use one of the following methods to configure the parameters:

- Add the following configuration to the AndroidManifest.xml file of your app:

```
   <meta-data
            android:name="com.alibaba.pdns.ACCESS_KEY_ID"
            android:value="Replace the value with the AccessKey ID that is generated when you register your app in the Alibaba
Cloud Management Console"./>

   <meta-data
            android:name="com.alibaba.pdns.ACCESS_KEY_SECRET"
            android:value="Replace the value with the AccessKey secret that is generated when you register your app in the Ali
baba Cloud Management Console"./>
```

- Configure the parameters during the initialization process when your app accesses the SDK.

```
   DNSResolver.setAccessKeySecret("");   // Configure the accessKeySecret parameter that is used to access the SDK in the Alib
aba Cloud DNS console.
   DNSResolver.setAccessKeyId("");       // Configure the accessKeyId parameter that is used to access the SDK in the Alibaba
Cloud DNS console.
```

Note

You can use one of the preceding authentication configurations based on your business requirements. If you use one of the preceding configurations, you do not need to use the other configurations.

(4) To use the SDK in a more effective manner, we recommend that you set the JDK version to 1.8 when you compile the app.

2.4.1 Initialize the service

Alibaba Cloud Public DNS SDK uses the DNSResolver class to encapsulate service requests and local caches of Alibaba Cloud Public DNS. You can use DNSResolver.Init(this,accountID) to initialize services to access Alibaba Cloud Public DNS. accountID is the unique ID that the server automatically generates when you register your Alibaba Cloud account in the Alibaba Cloud Public DNS console.

2.4.2 Configure domain names that you want to pre-resolve

When you initialize your Android app, you can register domain names that you may use with Alibaba Cloud Public DNS SDK so that the SDK can resolve the domain names in advance. This reduces the delay of requests during subsequent DNS resolution. The following code shows how to configure domain names that you want to pre-resolve.

- Specify IPv6 or IPv4 domain names that you want to preload

```
// Configure the domain names that you want to pre-resolve and replace the domain names that you want to preload with the domain names that you want to use Alibaba Cloud DNS to resolve.
DNSResolver.getInstance().preLoadDomains(DNSResolver.QTYPE_IPV4,new String[]{...})

DNSResolver.QTYPE_IPV4  Obtain the IPv4 record type that corresponds to the pre-resolved domain names.
DNSResolver.QTYPE_IPV6  Obtain the IPv6 record type that corresponds to the pre-resolved domain names.
```

- Automatically match and preload IPv6 or IPv4 domain names for resolution based on the current network

```
DNSResolver.getInstance().preLoadDomains(domains)
```

> Notice
>
> When you configure a pre-resolution interface, an asynchronous network request is triggered in real time. You must make sure that the code logic has the necessary initialization settings when the pre-resolution interface is called.

2.4.3 Specify whether to use the IPv6 addresses of servers

Alibaba Cloud Public DNS supports IPv4 and IPv6 dual-stack access. You can configure DNSResolver.setEnableIPv6(boolean enable) to determine whether to use IPv4 or IPv6 addresses. If you set DNSResolver.setEnableIPv6(boolean enable) to true, IPv6 addresses are used to access the interface on the server. If you set DNSResolver.setEnableIPv6(boolean enable) to false, IPv4 addresses are used to access the interface on the server. If you do not configure DNSResolver.setEnableIPv6(boolean enable), IPv4 addresses are used. If IPv6 addresses are used for access, the system automatically switches to IPv4 addresses when Alibaba Cloud Public DNS is inaccessible. In this case, a maximum of three access retries are allowed.

2.4.4 Specify whether to enable the short mode

Data returned by the JSON API for DoH of Alibaba Cloud Public DNS can be in the JSON format or a simple array of IP addresses. You can configure DNSResolver.setEnableShort (boolean enable) to enable or disable the short mode. If you do not configure DNSResolver.setEnableShort (boolean enable), the short mode is disabled by default.

The following sample code shows the configuration of the short mode:

```
DNSResolver.setEnableShort (true); // The default value is false. If you do not configure this parameter, the default value is used.
```

> Notice
>
> In short mode, the SDK calls Alibaba Cloud Public DNS to return a simple array of IP addresses. This reduces the amount of data returned and is suitable for scenarios in which heavy network traffic is not allowed.

2.4.5 Specify the maximum number of domain name resolution results that can be cached

```
DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER); // Specify the maximum number of domain name resolution results that can be cached. The default value is 100.
```

You can configure the CACHE_MAX_NUMBER parameter based on your business requirements.

2.4.6 Specify the maximum TTL for caching

```
DNSResolver.setMaxTtlCache(MAX_TTL_CACHE);// Specify the maximum TTL for caching. The default value is 3600. Unit: seconds.
```

2.4.7 Specify the server access protocol

```
DNSResolver.setSchemaType(DNSResolver.HTTP); // Specify the server access protocol. The default access protocol is HTTP.
```

DNSResolver.HTTP indicates that HTTP is used to access the server interface.

DNSResolver.HTTPS indicates that HTTPS is used to access the server interface.

2.4.8 Specify whether to enable the caching feature

```
DNSResolver.setEnableCache(true); // Specify whether to enable the caching feature. This feature is enabled by default.
```

If setEnableCache is set to true, the caching feature is enabled. If setEnableCache is set to false, the caching feature is disabled.

2.4.9 Specify whether connection speed testing of IP addresses is enabled

```
DNSResolver.setEnableSpeedTest(false); // Specify whether to enable connection speed testing of IP addresses. This feature is disabled by default.
```

If setEnableSpeedTest is set to true, this feature is enabled. If setEnableSpeedTest is set to false, this feature is disabled.

2.4.10 Specify the port number in the socket for connection speed testing of IP addresses

```
DNSResolver.setSpeedPort(DNSResolver.PORT_80)
```

You can specify the port number in the socket for connection speed testing of IP addresses. The default port number is 80.

2.4.11 Enable domain name caching based on ISP networks

```
DNSResolver.setIspEnable(true);// Specify whether to enable domain name caching based on Internet service provider (ISP) networks.
```

You can enable domain name caching based on ISP networks. If setIspEnable is set to true, the cached domain name resolution results are separately stored based on the network environments. If setIspEnable is set to false, the same cached domain name resolution results are used in different networks.

2.4.12 Specify the maximum TTL for negative caching

```
DNSResolver.setMaxNegativeCache(MAX_NEGATIVE_CACHE);// Specify the maximum TTL for negative caching. The default value is 30. Unit: seconds.
```

You can set the maximum TTL for negative caching based on your business requirements.

2.4.13 Configure the AccessKeySecret and AccessKeyId parameters for authentication

```
  DNSResolver.setAccessKeySecret("");   // Configure the accessKeySecret parameter that is used to access the SDK in the Alibaba Cloud DNS console.
  DNSResolver.setAccessKeyId("");       // Configure the accessKeyId parameter that is used to access the SDK in the Alibaba Cloud DNS console.
```

2.4.14 Specify whether to enable the scheduled update of expired cache data

If the caching feature is enabled, you can enable the scheduled update of expired cache data. After this feature is enabled, the SDK automatically replaces the expired cache data with new data at an interval of 1 minute. This way, data in the cache can be updated in a timely manner, but the number of times DNS resolution is performed and the traffic consumption of clients may increase.

```
  DNSResolver.setEnableSchedulePrefetch(true); // Specify whether to enable the scheduled update of expired cache data. The default value is true, which indicates that this feature is enabled.
```

3. Anti-obfuscation configuration

```
   -keep class com.alibaba.pdns.** {*;}
```

4. Service API operations

```
 /**
   * Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, to obtain the resolution result of the domain name.
   *
   * @param host The domain name that you want to resolve.
   * @return Obtain the optimal array of IP addresses based on the current network environment.
   */
 public String[] getIpsByHost(String host)


  /**
   * Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, to preload the domain names that you want to resolve.
   *
   * @param domains The domain names that you want to preload.
   */
   public void preLoadDomains(final String[] domains)


 /**
   * Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, to obtain the array of IP addresses from the cache after the domain name is resolved.
   *
   * @param host The domain name that you want to query, such as www.taobao.com.
   * @param isAllowExp Specify whether to return the resolution result of the expired domain name.
   * @return Obtain the array of IP addresses from the cache after the domain name is resolved.
   */
    public String[] getIpsByHostFromCache(String host, boolean isAllowExp)


 /**
  * Obtain the DomainInfo object array in the IPv4 records that correspond to a specific URL.
  *
  * @param url Example: http://www.taobao.com
  * @return Obtain the DomainInfo object array in the IPv4 records that correspond to the URL.
  */
    public DomainInfo[] getIPsV4DInfoByUrl(String url)

  Note: The URL in the DomainInfo object is the URL in which the host field is automatically replaced by a specific IP address
. You do not need to manually replace the host field in the URL.

 /**
   * Obtain the DomainInfo object array in the IPv6 records that correspond to a specific URL.
   *
   * @param url Example: http://m.taobao.com
   * @return Obtain the DomainInfo object array in the IPv6 records that correspond to the URL.
   */
    public DomainInfo[] getIPsV6DInfoByUrl(String url)


  /**
    * Obtain a DomainInfo object in the IPv4 records that correspond to a specific URL.
    *
    * @param url Example: http://m.taobao.com
    * @return Obtain a DomainInfo object in the IPv4 records that correspond to the URL.
    */
     public DomainInfo getIPV4DInfoByUrl(String url)


 /**
    * Obtain a DomainInfo object in the IPv6 records that correspond to a specific URL.
    *
    * @param url Example: http://www.taobao.com
    * @return Obtain a DomainInfo object in the IPv6 records that correspond to the URL.
    */
    public DomainInfo getIPV6DInfoByUrl(String url)

   Note: The returned DomainInfo object encapsulates the following attributes:

  /**
   * The auto-increment ID of the access domain name.
    */
    public String id = null;

   /**
    * The available URL in which the host field is replaced by an IP address.
    */
     public String url = null;

    /**
    * The service name that needs to be included in the header of the HTTP request.
    */
    public String host = "";

   /**
    * The returned content body.
    */
   public String data = null;

   /**
    * The start time of the request.
    */
   public String startTime = null;

   /**
    * The end time of the request. If the request times out, the value is null.
    */
   public String stopTime = null;

   /**
   /**
   * The returned server status code, such as 200, 404, or 500.
   */
   public String code = null;

  /**
    * Obtain the array of IPv4 records that correspond to a specific hostname.
    * @param hostName Example: www.taobao.com
    * @return The array of IPv4 addresses that correspond to the hostname is returned.
    */
  public  String[] getIPsV4ByHost(String hostName)

/**
    * Obtain the array of IPv6 records that correspond to a specific hostname.
    * @param hostName Example: www.taobao.com
    * @return Obtain the array of IPv6 addresses that correspond to the hostname.
    */
  public String[] getIPsV6ByHost(String hostName)

/**
    * Obtain an IPv4 record that corresponds to a specific hostname.
    * @param hostName Example: www.taobao.com
    * @return A random IPv4 address in the set of IPv4 addresses that correspond to the hostname is returned. If connection sp
eed testing of IP addresses is enabled, the optimal IPv4 address is returned.
    */
  public String getIPV4ByHost(String hostName)

/**
    * Obtain an IPv6 record that corresponds to a specific hostname.
    * @param hostName Example: www.taobao.com
    * @return Obtain a random IPv6 address in the set of IPv6 addresses that correspond to the hostname. If connection speed t
esting of IP addresses is enabled, the optimal IPv6 address is returned.
    */
 public String getIPV6ByHost(String hostName)

   /**
     * Obtain the IP address array of the IPv4 record type from the cache after the domain name is resolved.
     * If no IP address array exists in the cache, null is returned. Then, an asynchronous query is triggered and the query re
sults are stored in the cache.
     *
     * @param host The domain name that you query, such as www.taobao.com.
     * @param isAllowExp Specify whether to return the resolution result of the expired domain name.
     * @return Obtain the IP address array of the IPv4 record type from the cache after the host field is resolved.
     */
    private String[] getIpv4ByHostFromCache(String host , boolean isAllowExp)

    /**
     * Obtain the IP address array of the IPv6 record type from the cache after the domain name is resolved.
     * If no IP address array exists in the cache, null is returned. Then, an asynchronous query is triggered and the query re
sults are stored in the cache.
     *
     * @param host The domain name that you query, such as www.taobao.com.
     * @param isAllowExp Specify whether to return the resolution result of the expired domain name.
     * @return Obtain the IP address array of the IPv6 record type from the cache after the host field is resolved.
     */
    private String[] getIpv6ByHostFromCache(String host , boolean isAllowExp)

   /**
     * Preload a domain name for resolution.
     *
     * @param qType The type of IP addresses that you want to preload, such as IPv4 or IPv6.
     * @param domains The domain names that you want to preload.
     */
    public void preLoadDomains(String qType, final String[] domains)

   /**
     * Obtain the statistics on successful and failed queries of Alibaba Cloud Public DNS.
     *
     * @return Obtain the JSON array string of the resolution statistics on all domain names.
     */
    public String getRequestReportInfo()
```

5. Example of SDK API operations

URl: the received access address, such as http://www.taobao.com.

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

5.1 Obtain the optimal IP addresses based on the current network environment

```
String[] ip = DNSResolver.getInstance().getIpsByHost(hostname); // Obtain the optimal IP addresses after the domain name is resolved based on the current network environment.
```

5.2 Preload domain names for DNS resolution based on the current network environment

```
DNSResolver.getInstance().preLoadDomains(domains) // Configure the domain names that you want to pre-resolve and replace the domain names that you want to preload with the domain names that you want to use Alibaba Cloud DNS to resolve.
```

5.3 Read the domain name resolution results from the cache based on the current network environment

```
String[] ip = DNSResolver.getInstance().getIpsByHostFromCache(hostname,true);// Obtain the domain name resolution results from the cache based on the current network environment.
```

5.4 Obtain IPv4 addresses

```
String IPV4 = DNSResolver.getInstance().getIPV4ByHost(hostname); // Obtain the IPv4 addresses after the domain name is resolved.
```

5.5 Obtain IPv6 addresses

```
String IPV6 =  DNSResolver.getInstance().getIPV6ByHost(hostname) // Obtain the IPv6 addresses after the domain name is resolved.
```

5.6 Obtain IPv4 addresses from the cache after the domain name is resolved

```
String[] IPV4 =  DNSResolver.getInstance().getIpv4ByHostFromCache(hostname , true) // Obtain the IPv4 addresses from the cache after the domain name is resolved.
```

5.7 Obtain IPv6 addresses from the cache after the domain name is resolved

```
String[] IPV6 =  DNSResolver.getInstance().getIpv6ByHostFromCache(hostname , true) // Obtain the IPv6 addresses from the cache after the domain name is resolved.
```

5.8 Obtain the DomainInfo object that corresponds to a specific URL

```
DomainInfo dinfo = DNSResolver.getInstance().getIPV4DInfoByUrl(url); // Obtain the URL in which the domain name is replaced by an IP address.
```

5.9 Obtain the statistics on successful queries and failed queries of Alibaba Cloud DNS

```
String reportInfo = DNSResolver.getInstance().getRequestReportInfo();// Obtain the statistics on successful queries and failed queries.
```

The following figure shows the meanings of the fields in the JSON array string of the resolution statistics of all domain names.

> Notice
>
> The resolution statistics of Alibaba Cloud Public DNS are collected based on **network environments, domain names, and query types.**

6. Example

```
public class MainActivity extends AppCompatActivity {
    private Button button;
    private TextView tvInfo;
    private TextView tvResult;
    private String hostUrl = "http://www.taobao.com"; // Replace http://www.taobao.com with the URL that corresponds to the dom
ain name that you want to resolve.
    private String hostName = "www.taobao.com"; // Replace www.taobao.com with the domain name that you want to resolve.
    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() {
                       // Call the getIPV4ByHost() method in Alibaba Cloud Public DNS SDK to obtain the IP address that is reso
lved from the specified domain name.
                       String ipv4 = DNSResolver.getInstance().getIPV4ByHost(hostName);
                       tvInfo.setText("IP address that is resolved from the domain name: "+ ipv4);
                       // Call the getIPV4DInfoByUr operation in Alibaba Cloud Public SDK to obtain the URL in the domainInfo o
bject after the specified domain name is resolved. The URL is the one in which the host field is replaced by an IP address.
                       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) {
                // Send a network request.
               String requestUrl = dinfo.url;
               HttpURLConnection conn = null;
               try {
                   URL url = new URL(requestUrl);
                   conn = (HttpURLConnection) url.openConnection();
                   // When an IP address is used for access, you must set the host field in the HTTP request header to the res
olved domain name.
                   conn.setRequestProperty("Host", url.getHost());// Set the Host field in the HTTP request header.
                   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"; // You can replace 10001 with your account ID.
   private static final String TAOBAO_HOST_NAME = "www.taobao.com";// Replace www.taobao.com with the domain name that you want to preload.
   private static final String ALIYUN_HOST_NAME = "www.aliyun.com"; // Replace www.aliyun.com with the domain name that you want to preload.
   private static final int CACHE_MAX_NUMBER = 100; // Specify the maximum number of domain name resolution results that can be cached. The default value is 100.
   private static final int MAX_NEGATIVE_CACHE = 30; // Specify the maximum TTL for negative caching. The default value of MAX_NEGATIVE_CACHE is 30. Unit: seconds.
   private static final int MAX_TTL_CACHE = 1 * 60 * 60; // Specify the maximum TTL for caching. The default value of MAX_TTL_CACHE is 3600. Unit: seconds.

   @Override
   public void onCreate() {
        super.onCreate();
        DNSResolver.Init(this, accountID);    // Configure the account ID that is used to access the SDK in the Alibaba Cloud DNS console.
        DNSResolver.setAccessKeySecret(""); // Configure the accessKeySecret parameter that is used to access the SDK in the Alibaba Cloud DNS console.
        DNSResolver.setAccessKeyId("");      // Configure the accessKeyId parameter that is used to access the SDK in the Alibaba Cloud DNS console.
        DNSResolver.setEnableShort(false); // Specify whether to enable the short mode.
        DNSResolver.setEnableIPv6(false); // Specify whether to enable IPv6 access.
        DNSResolver.setEnableCache(true); // Specify whether to enable the caching feature.
        DNSResolver.setEnableSpeedTest(false); // Specify whether to enable connection speed testing of IP addresses.
        DNSResolver.setEnableSchedulePrefetch(true); // Specify whether to enable the scheduled update of expired cache data. The default value is true, which indicates that this feature is enabled.
        DNSResolver.setIspEnable(true);// Specify whether to enable domain name caching based on ISP networks.
        DNSResolver.setMaxTtlCache(MAX_TTL_CACHE); // Specify the maximum TTL for caching. The default value is 3600. Unit: seconds.
        DNSResolver.setMaxNegativeCache(MAX_NEGATIVE_CACHE);// Specify the maximum TTL for negative caching. The default value is 30. Unit: seconds.
        DNSResolver.setSchemaType(DNSResolver.HTTP); // Specify whether the server access protocol is HTTP or HTTPS.
        DNSResolver.setSpeedPort(DNSResolver.PORT_80); // Specify the port number for IP socket detection. The default port number is 80.
        DNSResolver.getInstance().setMaxCacheSize(CACHE_MAX_NUMBER); // Specify the maximum number of resolution results of domain names that can be cached. The default value is 100.
        DNSResolver.getInstance().preLoadDomains(new String[]{TAOBAO_HOST_NAME,ALIYUN_HOST_NAME}); // Configure the domain names that you want to pre-resolve and replace the domain names that you want to preload with the domain names that you want to use Alibaba Cloud DNS to resolve.
   }
}
```

Precautions

1. After you use Alibaba Cloud Public DNS to resolve a domain name to an IP address, you can use the IP address to send business requests. You must specify the host field in the HTTP request header as the domain name that has been resolved.
2. The demo program makes it easy to learn how to use the Alibaba Cloud public DNS SDK.

   You can download the demo program for reference.

For more information, download the demo project.

6.SDK for iOS developer guide

This topic describes Alibaba Cloud Public DNS SDK for iOS and how to use the SDK.

1. Overview

Alibaba Cloud Public DNS SDK is developed by Alibaba Cloud to provide the DNS resolution service for mobile developers.

The SDK allows mobile developers to use Alibaba Cloud Public DNS in their iOS apps. This prevents DNS resolution errors and implements precise scheduling of DNS queries at a low cost. The demo project source code provides an example on how to integrate the SDK with an iOS app and use Alibaba Cloud Public DNS in the iOS app.

Two native DNS encryption protocols are supported in the iOS 14 operating system: DNS over TLS (DoT) and DNS over HTTPS (DoH). For more information about how to configure Alibaba Cloud Public DNS as the default resolver for encrypted DNS, see Native encrypted DNS in iOS 14.

The SDK encapsulates the JSON API for DoH of Alibaba Cloud Public DNS. The SDK also provides functions for iOS apps to perform DNS resolution and supports efficient domain name caching based on time-to-live (TTL) and least recently used (LRU) policies. The SDK has the following benefits based on the features of Alibaba Cloud Public DNS:

- **Easy-to-use**

  To use Alibaba Cloud Public DNS, you need to only integrate the SDK with your app. This allows you to use the DNS resolution service in an easier and more convenient manner.

- **DNS resolution without a delay**

  The SDK uses the LRU caching algorithm. This way, the IP addresses that are obtained from DNS resolution are cached on your on-premises server. The SDK also automatically updates the cache and deletes expired data based on TTL. This way, the cached data is always valid and DNS resolution can be effectively implemented without a delay.

2. SDK integration

2.1 SDK integration

1. Log on to the Alibaba Cloud DNS console. In the left-side navigation pane, click Public DNS. On the Public DNS page, click the Overview tab and view the account ID. The account ID is a unique ID that is automatically generated for your Alibaba Cloud account. You can use the account ID to use Alibaba Cloud Public DNS in your app.
2. In the SDK Download section of the Traffic Access page, click Download for iOS to download the package for Alibaba Cloud Public DNS SDK for iOS.
3. Add the `pdns-sdk-ios.framework` folder in the SDK package to your iOS app.
4. Import the following system libraries to your app:
   - Foundation.framework
   - SystemConfiguration.framework
   - CoreFoundation.framework
   - CoreTelephony.framework
5. Add `-ObjC` to Other Linker Flags in the Build Settings window of your Xcode project.
6. Call the `application:didFinishLaunchingWithOptions:` method in your app to initialize the SDK. Use the following code in the method:

```
DNSResolver *resolver = [DNSResolver share];
resolver.accountId = @"Account ID that is generated when you register your app in the console";
```

2.2 Auto-integration (CocoaPods)

1. Specify the library location (including the master branch of the library) in Podfile.

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

2. Add dependencies to the target directory of your Xcode project.

```
pod 'AlicloudPDNS'
```

3. API reference

3.1 accountId

The accountId parameter specifies the account ID that is automatically generated after you register your app in the Alibaba Cloud DNS console.

3.2 Enable IPv6 access

Alibaba Cloud Public DNS supports IPv4 and IPv6 dual-stack access. By default, the SDK uses IPv4 addresses to access the DNS server.

If you want to use IPv6 addresses to access the DNS server, make sure that the current network supports IPv6 addresses and set the ipv6Enable parameter to YES to enable IPv6 access. The following code shows an example:

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

3.3 Enable the short mode

Data that is returned by the JSON APIs for DoH of Alibaba Cloud Public DNS can be in the JSON format or an array of IP addresses. By default, the SDK uses the JSON format.

If you want the returned data to be arrays of IP addresses, configure the shortEnable parameter to enable the short mode. The following code shows an example:

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

3.4 Configure the scheme parameter

You can use the `scheme` parameter to specify whether to perform DNS resolution over HTTP or HTTPS.

By default, the SDK performs DNS resolution over HTTP. We recommend that you do not change the default configuration because DNS resolution over HTTP is faster. If you want to perform DNS resolution over HTTPS, set the scheme parameter to DNSResolverSchemeHttps. The following code shows an example:

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

3.5 Enable the caching feature

You can use the SDK to specify whether to enable the caching feature. If this feature is enabled, the data that is obtained from the first resolution is saved in the cache and the cached data is preferentially returned for subsequent resolution requests. This significantly accelerates DNS resolution.

By default, the caching feature is enabled in the SDK. To disable the caching feature, set the cacheEnable parameter to NO. The following code shows an example:

```
[DNSResolver share].cacheEnable=NO;
```

3.6 Enable the scheduled update of expired cache data

When the caching feature is enabled, you can enable the scheduled update of expired cache data. After this feature is enabled, the SDK automatically replaces the expired cache data with new data every minute. This way, data in the cache can be updated in a timely manner, but the number of DNS resolutions and the consumption of client traffic may increase.

By default, this feature is enabled. If you want to disable this feature, set the schedulePrefetchEnable parameter to NO. The following code shows an example:

```
[DNSResolver share].schedulePrefetchEnable=NO;
```

3.7 Specify the maximum number of domain names whose resolution results can be cached

If the caching feature is enabled, you can specify the maximum number of domain names whose resolution results can be cached. Valid values: 100 to 500.

By default, the resolution results for a maximum of 100 domain names can be cached by using the SDK. You can configure the cacheCountLimit parameter to specify the maximum cache size. The following code shows an example:

```
[DNSResolver share].cacheCountLimit = 200;
```

3.8 Enable connection speed testing for IP addresses

You can use the SDK to enable connection speed testing for IP addresses. If this feature is enabled, the resolution result of the IP address with the highest connection speed is preferentially returned. The resolution results are sorted in descending order of connection speeds.

By default, this feature is disabled in the SDK. To enable this feature, set speedTestEnable to YES. The following code shows an example:

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

3.9 Specify the connection speed testing mode

You can specify the connection speed testing mode. If this feature is enabled and the speedPort parameter is set to 0, Internet Control Message Protocol (ICMP) packets are used to detect the connection speed. If the speedPort parameter is set to 80, 443, or another valid value, the port that is specified in the socket is used to detect the connection speed.

The default value of this parameter in the SDK is 0. To use the port that is specified in the socket for detection, you must specify the speedPort parameter. The following code shows an example:

```
[DNSResolver share].speedPort = 80;
```

3.10 Enable domain name caching based on ISP networks

You can use the SDK to enable domain name caching based on ISP networks. If this feature is enabled, the cached domain name resolution results are separately stored based on the network environments. If this feature is disabled, the same cached domain name data is used in different networks.

By default, this feature is disabled in the SDK. To enable this feature, set the ispEnable parameter to YES. The following code shows an example:

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

3.11 Specify the maximum TTL for negative caching

You can use the SDK to specify the maximum TTL for negative caching. If you configure the maxNegativeCache parameter, the maximum TTL for negative caching cannot exceed the value of this parameter.

The default value of this parameter in the SDK is 30. Unit: seconds. The following code show an example:

```
[DNSResolver share].maxNegativeCache = 30;
```

3.12 Specify the maximum TTL for caching

You can use the SDK to specify the maximum TTL for caching. If you configure the maxCacheTTL parameter, the maximum TTL for caching cannot exceed the value of this parameter.

The default value of this parameter in the SDK is 3600. Unit: seconds. The following code shows an example:

```
[DNSResolver share].maxCacheTTL= 3600;
```

3.13 Configure the timeout parameter

The `timeout` parameter specifies the timeout period for DNS resolution. The default timeout period is 3 seconds. You can specify a custom timeout period based on your business requirements. We recommend that you set the timeout period to a value within the range of 2 to 5 seconds.

3.14 Enable authentication

You can use SDK 2.0.0 or later to enable authentication to protect user identities from being stolen by unauthorized third parties.

You can enable authentication only after you configure the following parameters. To enable authentication, create an AccessKey pair in the DNS console by following the instructions that are described in Service authentication and configure the accessKeyId and accesskeySecret parameters in your app. The following code shows an example:

```
[DNSResolver share].accessKeyId = @"Your AccessKey ID";
[DNSResolver share].accesskeySecret = @"Your AccessKey secret";
```

3.15 Preload domain names

The caching feature that is provided by the SDK accelerates the resolution of a domain name that has been resolved. We recommend that you preload the domain names that you want to resolve after you start your app.

Sample code:

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

   DNSResolver *resolver = [DNSResolver share];
   resolver.accountId = @"Account ID that was generated after you activate Alibaba Cloud Public DNS";
   resolver.cacheEnable = YES;
   // Preload the domain names that you want to resolve.
   [resolver preloadDomains:@[@"Domain name 1", @"Domain name 2", @"Domain name 3"] complete:^{
       // All specified domain names are preloaded.

   }];

   return YES;
}
```

4. Service API operations

Sample code:

```
/// Pre-resolve an array of domain names. You can call this operation after you start your app to accelerate subsequent DNS resolution.
/// Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack. Then, resolve the array of the domain names to the IP addresses that are supported by the current network environment.
/// @param domainArray The array of domain names that you want to resolve.
/// @param complete     Return resolution results.
- (void)preloadDomains:(NSArray<NSString *> *)domainArray complete:(void(^)(void))complete;

/// Obtain the array of IP addresses after DNS resolution. Then, automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, to obtain the IP addresses that are supported by the current network environment.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return all the IP addresses to which the domain name is resolved.
- (void)getIpsDataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray))complete;

/// Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, and obtain the array of IP addresses that are supported by the current network environment from the cache without a delay. If no data exists in the cache, or the data in the cache has expired and the enable parameter is set to NO, nil is returned.
/// @param domain   The domain name that you want to resolve.
/// @param enable   Specify whether to return expired IP addresses.
- (NSArray<NSString *> *)getIpsByCacheWithDomain:(NSString *)domain andExpiredIPEnabled:(BOOL)enable;

/// Obtain an array of information about the IPv4 addresses to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return information about all the IPv4 addresses to which the domain name is resolved.
- (void)getIpv4InfoWithDomain:(NSString *)domain complete:(void(^)(NSArray<DNSDomainInfo *> *domainInfoArray))complete;

/// Obtain an array of information about the IPv6 addresses to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return information about all the IPv6 addresses to which the domain name is resolved.
- (void)getIpv6InfoWithDomain:(NSString *)domain complete:(void(^)(NSArray<DNSDomainInfo *> *domainInfoArray))complete;

/// Obtain information about an IPv4 address to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return information about an IPv4 address to which the domain name is resolved.
- (void)getRandomIpv4InfoWithDomain:(NSString *)domain complete:(void(^)(DNSDomainInfo *domainInfo))complete;

/// Obtain information about an IPv6 address to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return information about an IPv6 address to which the domain name is resolved.
- (void)getRandomIpv6InfoWithDomain:(NSString *)domain complete:(void(^)(DNSDomainInfo *domainInfo))complete;

/// Obtain the array of IPv4 addresses to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return all IPv4 addresses to which the domain name is resolved.
- (void)getIpv4DataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray))complete;

/// Obtain the array of IPv6 addresses to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return all IPv6 addresses to which the domain name is resolved.
- (void)getIpv6DataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray))complete;

/// Obtain an IPv4 address to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return one of the IPv4 addresses to which the domain name is resolved.
- (void)getRandomIpv4DataWithDomain:(NSString *)domain complete:(void(^)(NSString *data))complete;

/// Obtain an IPv6 address to which a domain name is resolved.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return one of the IPv6 addresses to which the domain name is resolved.
- (void)getRandomIpv6DataWithDomain:(NSString *)domain complete:(void(^)(NSString *data))complete;

/// Pre-resolve an array of domain names to IPv4 addresses. You can call this operation after you start your app to accelerate subsequent DNS resolution.
/// @param domainArray  The array of domain names that you want to resolve.
/// @param complete         Return resolution results.
- (void)preloadIpv4Domains:(NSArray<NSString *> *)domainArray complete:(void(^)(void))complete;

/// Pre-resolve an array of domain names to IPv6 addresses. You can call this operation after you start your app to accelerate subsequent DNS resolution.
/// @param domainArray  The array of domain names that you want to resolve.
/// @param complete         Return resolution results.
- (void)preloadIpv6Domains:(NSArray<NSString *> *)domainArray complete:(void(^)(void))complete;

/// Obtain the IPv4 addresses that are resolved from the domain names without a delay. If no data exists in the cache, or the data in the cache has expired and the enable parameter is set to NO, nil is returned.
/// @param domain   The domain name that you want to resolve.
/// @param enable   Specify whether to return expired IP addresses.
- (NSArray<NSString *> *)getIpv4ByCacheWithDomain:(NSString *)domain andExpiredIPEnabled:(BOOL)enable;

/// Obtain the IPv6 addresses that are resolved from the domain names without a delay. If no data exists in the cache, or the data in the cache has expired and the enable parameter is set to NO, nil is returned.
/// @param domain   The domain name that you want to resolve.
/// @param enable   Specify whether to return expired IP addresses.
- (NSArray<NSString *> *)getIpv6ByCacheWithDomain:(NSString *)domain andExpiredIPEnabled:(BOOL)enable;

/// Collect statistical information.
-(NSArray *)getRequestReportInfo;
```

5. Sample API calls

5.1 Configure basic information

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

   // The only method that can be used for initialization.
   DNSResolver *resolver = [DNSResolver share];
   // The required parameter.
   resolver.accountId = @"Account ID that is generated after you activate Alibaba Cloud Public DNS";

   return YES;
}
```

5.2 Resolve a domain name by calling an operation

The SDK provides various DNS resolution methods. For more information, see the header file `DNSResolver.h`. The following example describes a DNS resolution method that automatically detects the IP address type supported by the network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack.

Operation declaration:

```
/// Obtain the array of IP addresses after DNS resolution. Then, automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, to obtain the IP addresses that are supported by the current network environment.
/// @param domain           The domain name that you want to resolve.
/// @param complete       Return all addresses to which the domain name is resolved.
- (void)getIpsDataWithDomain:(NSString *)domain complete:(void(^)(NSArray<NSString *> *dataArray))complete;
```

The following sample code shows how to call the operation:

```
[[DNSResolver share] getIpsDataWithDomain:@"www.taobao.com" complete:^(NSArray<NSString *> *dataArray) {
    // dataArray is an array of IP addresses that correspond to the domain name www.taobao.com.
   if (dataArray.count > 0) {
       // TODO: Use the IP address to send a business request.
   }
}];
```

5.3 Obtain the resolution results from the cache

Operation declaration:

```
/// Automatically detect the IP address type that is supported by the current network environment, such as IPv4-only, IPv6-only, or IPv4/IPv6 dual-stack, and obtain the array of the IP addresses that are supported by the current network environment from the cache without a delay. If no data exists in the cache, or the data in the cache has expired and the enable parameter is set to NO, nil is returned.
/// @param domain   The domain name that you want to resolve.
/// @param enable   Specify whether to return expired IP addresses.
- (NSArray<NSString *> *)getIpsByCacheWithDomain:(NSString *)domain andExpiredIPEnabled:(BOOL)enable;
```

Sample code:

```
NSArray *result = [[DNSResolver share] getIpsByCacheWithDomain:@"Domain name" andExpiredIPEnabled:YES];
// Obtain an IP address that maps the domain name from the cache.
if (result.count > 0) {
   // TODO: Use the IP address to send a business request.

}
```

Note: It is faster to obtain resolution results from the cache than to perform DNS resolution. If the domain name that you want to resolve has not been resolved before, or its resolution result in the cache has expired and the enable *parameter is set to NO, nil is returned.*

5.4 Collect statistical information

Operation declaration:

```
/// Collect statistical information.
-(NSArray *)getRequestReportInfo;
```

Sample code:

```
NSArray *array = [[DNSResolver share] getRequestReportInfo];
```

Data format:

```
 (
      {
         avgRtt = "1";                                // Average DNS resolution duration. Unit: milliseconds.
         cacheDnsNum = 0;                             // Number of cache hits.
         domain = "www.taobao.com"; // The domain name that you want to resolve.
         gobackLocaldnsNum = 0;                       // Number of times DNS resolution is downgraded to the on-premises DNS s
erver.
         localErro = 0;                               // Number of resolution failures on the on-premises DNS server.
         maxRtt = "60";                               // Maximum DNS resolution duration. Unit: milliseconds.
         noPermissionErro = 0;                        // Number of authentication failures.
         noResponseErro = 0;                          // Number of times that no response is returned.
         requestPDnsNum = 1;                          // Number of recursive queries.
         sp = China Mobile;                                // Name of the ISP.
         successNum = 1;                              // Number of successful resolutions.
         timeoutErro = 0;                             // Number of network timeout errors.
         type = 28;                                   // If the value of the type parameter is 1, IPv4 addresses are used. If
the value of the type parameter is 28, IPv6 addresses are used.
         urlParameterErro = 0;                        // Number of errors that are returned because the request parameter is i
nvalid.
         urlPathErro = 0;                             // Number of errors that are returned because the URL is invalid.
      }
         ......
 );
```

6. Precautions

1. The `pdns-sdk-ios.framework` folder supports only iOS 9.0 or later.
2. If you perform DNS resolution over HTTP, you must set `Allow Arbitrary Loads` in App Transport Security Settings to `YES` in the `Info.plist` file.
3. After you use Alibaba Cloud Public DNS to resolve a domain name to an IP address, you can use the IP address to send business requests. You must specify the hostname field in the HTTP request header as the domain name that has been resolved.

   Example:

   ```
   // Set the ip parameter to an IP address to which a domain name is resolved.
   NSURL *url = [NSURL URLWithString:[NSString stringWithFormat:@"https://%@", ip]];
   NSMutableURLRequest *mutableReq = [NSMutableURLRequest requestWithURL:url cachePolicy:NSURLRequestUseProtocolCachePolicy t
   imeoutInterval: 10];
   // Set the host field to the domain name.
   [mutableReq setValue:@"Domain name" forHTTPHeaderField:@"host"];
   ```

4. A demo project helps you learn how to use Alibaba Cloud Public DNS SDK.

   You can download the demo program for reference. For more information, download the demo project.

7.DescribeDohSubDomainStatisticsSummary

Obtains the query volumes of DNS over HTTPS (DoH) subdomains.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDohSubDomainStatisticsSummary	The operation that you want to perform. Set the value to DescribeDohSubDomainStatisticsSummary.
DomainName	String	Yes	example.com	The domain name.
PageNumber	Integer	Yes	1	The number of the page to return. Pages start from page 1. Default value: 1.
StartDate	String	Yes	2019-07-04	The beginning of the time range to query. Specify the time in the YYYY-MM-DD format. You can query only the DNS records of the last 90 days. `The value of StartDate must be greater than or equal to the difference between the current date and 90` .
Lang	String	No	en	The language type.
PageSize	Integer	No	20	The number of entries to return on each page. Maximum value: 100. Default value: 20.
EndDate	String	No	2019-07-04	The end of the time range to query. Specify the time in the YYYY-MM-DD format. The default value is the day when you query the required data.
SubDomain	String	No	www.example.com	The subdomain.

Response parameters

Parameter	Type	Example	Description
PageNumber	Integer	1	The page number of the returned page.
PageSize	Integer	10	The number of entries returned per page.
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	The ID of the request.
Statistics	Array of Statistic		The statistics list.
HttpCount	Long	3141592653	The number of HTTP requests.
HttpsCount	Long	3141592653	The number of HTTPS requests.
IpCount	Long	20	The number of IP addresses.

Parameter	Type	Example	Description
SubDomain	String	www.example.com	The subdomain.
TotalCount	Long	14141592653	Total number of requests.
V4HttpCount	Long	3141592653	The number of IPv4-based HTTP requests.
V4HttpsCount	Long	3141592653	The number of IPv4-based HTTPS requests.
V6HttpCount	Long	3141592653	The number of IPv6-based HTTP requests.
V6HttpsCount	Long	3141592653	The number of IPv6-based HTTPS requests.
TotalItems	Integer	100	Total number of entries returned.
TotalPages	Integer	50	Total number of pages returned.

Examples

Sample requests

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohSubDomainStatisticsSummary
&DomainName=example.com
&PageNumber=1
&StartDate=2019-07-04
&<Common request parameters>
```

Sample success responses

`XML` format

```
<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` format

```
{
 "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"
 }]
}
```

Error codes

For a list of error codes, visit the API Error Center.

8.DescribeDohDomainStatistics

Obtains the query volume of DNS over HTTPS (DoH) domain names.

Debugging

OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call this operation in OpenAPI Explorer. OpenAPI Explorer dynamically generates the sample code of the operation for different SDKs.

Request parameters

Parameter	Type	Required	Example	Description
Action	String	Yes	DescribeDohDomainStatistics	The operation that you want to perform. Set the value to DescribeDohDomainStatistics.
DomainName	String	Yes	example.com	The domain name.
EndDate	String	Yes	2019-07-04	The end of the time range to query. Specify the time in the YYYY-MM-DD format. The default value is the day when you perform the operation.
StartDate	String	Yes	2019-07-04	The beginning of the time range to query. Specify the time in the YYYY-MM-DD format. You can query only the DNS records of the latest 90 days. `The value of StartDate must be greater than or equal to the difference between the current date and 90`.
Lang	String	No	en	The language type.

Response parameters

Parameter	Type	Example	Description
RequestId	String	0F32959D-417B-4D66-8463-68606605E3E2	The ID of the request.
Statistics	Array of Statistic		The statistics list.
Timestamp	Long	1544976000000	The timestamp.
TotalCount	Long	3141592653	The total number of requests.
V4HttpCount	Long	3141592653	The number of IPv4-based HTTP requests.
V4HttpsCount	Long	3141592653	The number of IPv4-based HTTPS requests.
V6HttpCount	Long	3141592653	The number of IPv6-based HTTP requests.
V6HttpsCount	Long	3141592653	The number of IPv6-based HTTPS requests.

Examples

Sample requests

```
http(s)://alidns.aliyuncs.com/?Action=DescribeDohDomainStatistics
&DomainName=example.com
&EndDate=2019-07-04
&StartDate=2019-07-04
&<Common request parameters>
```

Sample success responses

`XML` format

```
<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` format

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

Error codes

For a list of error codes, visit the API Error Center.