

# 设置域名配置

## 接口地址

请求URL: domain/config (POST)

接口 请求域名: [cdn.api.baishan.com](https://cdn.api.baishan.com)

接口 版本: v2

## 接口描述

对加速域名各项功能进行设置和修改。支持功能项: 源站、回源host、缓存规则、referer黑白名单、ip黑名单、添加响应头。

注意:

接口调用频率不超过30次/分钟。

每次提交既支持单个功能设置, 也支持多个功能同时设置。当同时设置多个功能设置时, 如果某一项功能设置不合法, 那么本次提交的所有功能均无法成功设置。

如需使用请联系项目经理申请开通。

## 输入参数

公有参数: 用户的 token 可向值班同事获取。

### 请求参数(POST)

参数支持 form-data、x-www-form-urlencoded、json 等格式

参数名	数据类型	是否必须	说明
domains	string	是	指定设置功能的加速域名。最多可一次性设置10个加速域名。
config	array	是	功能项名称: origin:源站 origin_host:回源host cache_rule_list:缓存规则(新) referer:referer黑白名单 ip_black_list:ip黑名单 add_response_head:添加响应头

origin (源站)

参数名	数据类型	是否必须	说明
default_master	string	是	主回源地址, 可填多个ip或者一个域名。 多个ip以逗号(,)分隔; 主备源不能出现相同ip或域名。
default_slave	string	否	备回源地址, 可填多个ip或者一个域名。 多个ip以逗号(,)分隔; 主备源不能出现相同ip或域名。
origin_mode	string	否	回源方式: default :以用户请求的协议和端口回源 http :以http协议80端口回源 https :以https协议443端口回源 custom :自定义协议(ori_https)和端口(port)回源 不填充时, 默认值为default。
ori_https	string	否	当 origin_mode=custom 时, 该值需要设置。 https 协议回源: yes :是 no :否
port	int	否	当 origin_mode=custom 时, 该值需要设置。 回源端口, 有效值范围( 0-65535)。

#### # 修改源站

```
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
```

```
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "origin": {
      "default_master": "1.1.1.1,1.1.1.2",
      "default_slave": "2.2.2.2",
      "origin_mode": "custom",
      "ori_https": "yes",
      "port": 1040
    }
  }
}'
```

### origin\_host (回源host)

参数名	数据类型	是否必须	说明
host	string	是	回源host。 不设置时,CDN回源host与加速域名相同。

```
# 设置源站host
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json'
```

```
-d '{  
  "token": "xxx",  
  "domains": "example.com",  
  "config": {  
    "origin_host": {  
      "host": "www.example.com"  
    }  
  }  
}'
```

### cache\_rule (缓存规则)

参数名	数据类型	是否必须	说明
type	int	是	缓存类型， 1: 文件名后缀 2: 目录 3: 完整路径匹配 4: 正则

参数名	数据类型	是否必须	说明
pattern	string	是	缓存规则，多个以逗号分隔，例如： type=1时 <code>jpg,png,gif</code> type=2时 <code>/product/index,/test/index,/user/index</code> type=3时 <code>/index.html,/test/*.jpg,/user/get?index</code> type=4时 <code>例子见下面说明</code> 设置对应的正则，设置了正则后，对请求访问的url与正则做匹配，匹配成功则使用该缓存规则。 默认是带参数缓存，忽略过期时间，不忽略 不缓存头
time	int	是	缓存时间，配合timeunit使用，最大时间不超过2年， 当time=0时，不对指定pattern进行缓存(即禁止缓存)
timeunit	string	否	缓存时间单位，默认值为s，可选值有 (Y年,M月,D日,h时,i分,s秒)
<code>ignore_no_cache</code>	string	否	缓存时间time大于0时有效，忽略源站不缓存头，默认值为 <code>off</code> ，可选参数: <code>on,off</code>
<code>ignore_expired</code>	string	否	缓存时间time大于0时有效，忽略源站过期时间，默认值为 <code>on</code> ，可选参数: <code>on,off</code>
<code>ignore_query</code>	string	否	缓存时间time大于0时有效，忽略参数缓存并且忽略参数回源，默认值为 <code>off</code> ， 可选参数: <code>on,off</code>

缓存规则正则使用说明：正则规则是作为一般类型无法满足情形下的补充，使用不当会引起缓存不必要的故障，

建议有一定正则基础的开发者使用, 请理解清楚正则的使用规则后再应用到线上服务中, 设置缓存规则为正则, 会将请求的url地址(url地址: 包含http/https 协议部分、 域名、 端口[有的话]、 路径、 参数等几个部分)与正则去做匹配, 匹配成功则会使用该规则设定的缓存时间.

1. 举例1: `.*\.(gif|png|jpg)$`, 那么他将匹配所有含(gif,png,jpg)文件名后缀的url
2. 举例2: `^https?:/[^\s]+/dir1/dir2/(dir3_1|dir3_2)\.html`, 该规则匹配当前域名下/dir1/dir2/dir3\_1.html 和/dir1/dir2/dir3\_2.html 规则.

```
# 添加缓存规则(可以为多个缓存规则)
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
-d '{
  "token": "xxx",
  "domains": "aa.qingcdn.com",
  "config": {
    "cache_rule": [
      {
        "type": 1,
        "pattern": "jpg,png,gif",
        "time": 3600
      },
      {
        "type": 1,
        "pattern": "avi,mp4,mpeg",
        "time": 3600
      }
    ]
  }
}
```





参数名	数据类型	是否必须	说明
expire	int	是	缓存时间。与expire_unit组合生效, 最大缓存时间不超过2年。 当 <code>expire=0</code> 时, 对指定pattern禁止缓存。
expire_unit	string	否	缓存时间的时间单位: <code>Y</code> :年 <code>M</code> :月 <code>D</code> :日 <code>h</code> :小时 <code>i</code> :分 <code>s</code> :秒 不填充时, 默认值为s。
ignore_no_cache_headers	string	否	忽略源站响应头中的不缓存信息, 比如Cache-Control:no-cache等: <code>no</code> :不忽略 <code>yes</code> :忽略 不填充时, 默认值为no。
follow_expired	string	否	遵循源站缓存时间: <code>no</code> :不遵循 <code>yes</code> :遵循 不填充时, 默认值为no。

参数名	数据类型	是否必须	说明
query_params_op	string	否	问号后参数处理方式: do_nothing: 不处理 cache_back_source_remove: 缓存/回源均去除 不填充时, 默认值为do_nothing。
priority	int	否	优先级。数值越小的pattern 优先级越高, 优先生效。

# 设置2条缓存规则

```
curl -X POST "http://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "cache_rule_list": [
      {
        "match_method": "dir",
        "pattern": "/www/html/www/aaa",
        "case_ignore": "no",
        "priority": 13,
        "expire": 60,
        "expire_unit": "i",
        "ignore_no_cache_headers": "yes",
        "follow_expired": "no",
```

```
"query_params_op":"do_nothing"  
},  
{  
  "match_method":"ext",  
  "pattern":"php,jsp,asp,aspx",  
  "case_ignore":"no",  
  "priority":15,  
  "expire":3600,  
  "expire_unit":"s",  
  "ignore_no_cache_headers":"yes",  
  "follow_expired":"no",  
  "query_params_op":"cache_back_source_remove"  
}  
]  
}}'
```

referer (referer 黑白名单)

参数名	数据类型	是否必须	说明
type	int	是	防盗链类型 1:referer 黑名单 2:referer 白名单

参数名	数据类型	是否必须	说明
list	array	是	referrer 列表，最多可设置200个，多个以逗号分隔；不支持正则；referrer 为泛域名时，请以 * 开头，例如：*.example2.com，包括任意的匹配主机头和空主机头。
allow_empty	bool	否	允许referrer 为空的访问 true: 允许 false: 不允许 不填充时，默认为true。

# 设置referrer 黑名单

```
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "referrer": {
      "type": "1",
      "list": [
        "*.example.com",
        "www.example2.com"
      ],
      "allow_empty": true
    }
  }
}
```

```
}'
```

```
# 设置referer白名单
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "referer": {
      "type": "2",
      "list": [
        "*.example.com",
        "www.example2.com"
      ],
      "allow_empty": false
    }
  }
}'
```

ip\_black\_list (ip黑名单)

参数名	数据类型	是否必须	说明
list	array	是	ip黑名单列表。ip格式支持/8，/16，/24的网段格式，网段间的ip不能交叉重复；最多可设置500个ip。 ip黑名单不能与ip白名单共存，设置了ip黑名单，ip白名单功能将被清除。
mode	string	否	append: 追加模式 cover: 覆盖模式，默认cover

# 设置ip黑名单

```
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json' \
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "ip_black_list": {
      "list": [
        "14.12.11.0/24",
        "1.2.3.4"
      ],
      "mode": "cover"
    }
  }
}'
```

## add\_response\_head (添加响应头)

参数名	数据类型	是否必须	说明
type	string	否	<p>设置响应头的模式:</p> <ul style="list-style-type: none"> <li><b>reset</b>: 重置为本次设置的响应头</li> <li><b>add</b>: 追加本次设置的响应头。如果追加响应头的key存在, 则覆盖原有的响应头。</li> <li><b>remove</b>: 删除响应头。</li> </ul> <p>不填充时, 默认值为reset。</p>
list	array	是	<p>响应头列表</p> <ul style="list-style-type: none"> <li><b>type为reset和add时, 传入响应头和值</b>: {"name": "Content-Type", "value": "text/html"}</li> <li><b>type为remove时, 只需传入响应头的名称</b>: "Content-Type"</li> </ul>

```
#添加响应头, reset 模式
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
  -H "Content-Type: application/json"\
  -d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "add_response_head": {
```

```
"type": "reset",
"list": [
  {
    "name": "Content-Type",
    "value": "text/html; charset=ISO-8859-4"
  },
  {
    "name": "Cache-Control",
    "value": "max-age=300, must-revalidate"
  }
]
}
```

#添加响应头， add 模式

```
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H "Content-Type: application/json"\
-d '{
  "token": "xxx",
  "domains": "example.com",
  "config": {
    "add_response_head": {
      "type": "add",
      "list": [
        {
          "name": "Content-T1",
          "value": "text/html; charset=ISO-8859-4"
        }
      ]
    }
  }
}
```



```
    },  
    {  
      "name": "Cache-C1",  
      "value": "max-age=300, must-revalidate"  
    }  
  ]  
}  
}  
}'
```

#添加响应头, remove 模式

```
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\   
-H "Content-Type: application/json"\  
-d '{  
  "token": "xxx",  
  "domains": "example.com",  
  "config": {  
    "add_response_head": {  
      "type": "remove",  
      "list": [  
        "Content-T1",  
        "Cache-C1"  
      ]  
    }  
  }  
}'
```

## https (https 证书)

参数名	数据类型	是否必须	说明
cert_id	int	是	指定绑定的证书ID,可以通过证书查询接口获取。 当cert_id=0时, 将为域名解除https 服务。
http2	string	否	http2 功能 on:开启 off:关闭
force_https	string	否	请求http跳转为https 协议 0:不跳转 302:http 请求302成https 请求 301:http 请求301成https 请求 不填充时, 默认值为0。
ocsp	string	否	OCSP[on,off] 不填时不作变更

```
# 设置https 证书
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\
-H 'content-type: application/json'\
-d '{
  "token":"xxx",
```

```
"domains": "example.com",  
"config": {  
  "https": {  
    "cert_id": 10101,  
    "http2": "on",  
    "force_https": "301",  
    "ocsp": "on"  
  }  
}}
```

## 调用示例

```
# 修改多个功能  
curl -X POST "https://cdn.api.baishan.com/v2/domain/config"\  
-H "Content-Type: application/json"\  
-d '{  
  "token": "xxx",  
  "domains": "example.com,example2.com",  
  "config": {  
    "origin": {  
      "default_master": "133.233.133.33",  
      "default_slave": "12.13.41.21"  
    },  
    "origin_host": {
```

```
    "host": "test.example.com"
  },
  "referer": {
    "type": 1,
    "list": [
      "*.example.com",
      "new.vx.example.com"
    ],
    "allow_empty": false
  },
  "ip_black_list": {
    "list": [
      "14.12.11.0/24",
      "1.2.3.1"
    ]
  },
  "add_response_head": {
    "type": "add",
    "list": [
      {
        "name": "Content-Type",
        "value": "text/html; charset=ISO-8859-4"
      },
      {
        "name": "Cache-Control",
        "value": "max-age=300, must-revalidate"
      }
    ]
  }
]
```

```
    },  
    "https":{  
      "cert_id":123  
    }  
  }  
}'
```

## 返回示例

JSON 格式( 请求时传对应的参数, 设置成功后会输出对应设置成功后的配置参数)

```
{  
  "code":0,//0 操作成功, 非0时会给出对应的错误提示  
  "data": {  
    "config": {  
      "origin": {  
        "default_master": "133.233.133.33",  
        "default_slave": "12.13.41.21"  
      },  
      ...  
    }  
  }  
}
```

## 响应(Response Header)

```
HTTP/1.0 200 OK
Api-Id: 221328331
Content-Type: application/json; charset=utf-8
X-Ratelimit-Grad: minute // 接口调用频率限制粒度(分/小时/天)
X-Ratelimit-Limit: 50 // 当前每分钟请求次数上限
X-Ratelimit-Remaining: 29 // 每分钟剩余请求次数X-Retry-After: 43 //xx秒后重置当前粒度(分/小时/天)请求次数限制
```

## http 状态码说明

错误状态码	说明
404	您输入的 URL 错误
401	鉴权错误, 您的token 错误, 或者没有开通接口 权限
400	请求参数有误
200	执行成功, 返回 请求的数据
5xx	请联系值班同事