Customer Identity and Access Management Development Guide

Product Documentation

Contents

Development Guide
Overview

Last updated：2023-12-22 11:42:07

This guide introduces how to access Customer Identity Access Management (CIAM) and APIs in the application system. You can use CIAM to quickly implement features such as user login, logout, and sign-up, and take full advantage of CIAM's flexible and powerful configuration and management capabilities.

Applications

CIAM supports the following types of applications:

Web applications

Web applications are application systems on the backend web server. Web applications are developed in languages or frameworks such as Java, .NET, PHP, Node.js, and Express. Users can access them through a browser. Generally, backend applications are running on protected servers. Therefore, these applications can securely store sensitive information such as application passwords, and protect tokens from being leaked.

Single page applications

Single page applications (SPAs) are frontend applications running in the browser. SPAs are developed using HTML, CSS and JavaScript technologies based on React, Vue, and Angular frameworks. SPAs can directly initiate requests to service APIs without forwarding by backend applications. But the applications and data exist in "user-agent" (such as browsers), so SPA applications are not suitable for storing or processing sensitive data that requires protection.

Mobile applications

Mobile applications are applications installed and run on user devices (such as mobile phones, tablets, PCs, and smart devices). Mobile applications are generally developed using specialized application development languages, such as Objective-C, Swift, and Kotlin. These applications are not suitable for storing sensitive information such as application passwords, but can protect tokens from being leaked.

Machine-to-machine applications

Machine-to-machine (M2M) applications are applications running on the backend (such as backend services, command line programs, and daemons). M2M applications implement services by calling APIs without user participation. This way, they can better protect sensitive information from being leaked.

Integration Guide

Web, single page, and mobile applications can quickly access CIAM as OpenID Connect (OIDC) or OAuth clients. OIDC and OAuth protocols are widely used on the Internet. Leveraging their abundant development resources, you can find the desired development libraries for various applications for quick integration.

M2M applications get the Access Token via client credential mode, and then carry the Access Token to access APIs.

Note:

The APIs in this guide are accessed through the HTTPS protocol. The prefix of the access address isyour user directory's domain name, which can be viewed on the Domain Settings page. In this guide, `https://sample.portal.tencentciam.com` is used as the prefix of the access address.

Access via Authentication API
User Sign-up

Last updated：2023-12-22 11:42:07

API Description

This API is used to sign up a new user. This API is suitable for **automatic application sign-up development scenarios**. If your application uses the Customer Identity Access Management (CIAM) authentication portal, see Sign up via Authentication Portal.

Before calling this API, make sure that the application sign-up process has been configured and enabled. Input parameters of the API need to follow the rules of the application configured for the sign-up process. For example, if the mobile number is set as a authentication attribute and the username as the required general attribute for the sign-up process, the input parameters must contain the mobile number and the username. **If the mobile number is not set as a authentication attribute for sign-up, the input parameters cannot contain the mobile number**.

When the sign-up information contains a mobile number or email address, you need to call the API for sending one-time password (OTP) verification code to send the verification code to the user.

Password is an optional parameter, which can be set as needed.

If users only log in via SMS OTP, email OTP or social authentication, no password is required.

If users are allowed to log in by account and password authentication, a password should be set.

If you need to set a password, make sure that the **authentication source of account and password is associated** in the login process of the application. The API will verify the passed password according to the password policies of the authentication source. If the password does not meet the policy requirements, the sign-up will fail.

Note:

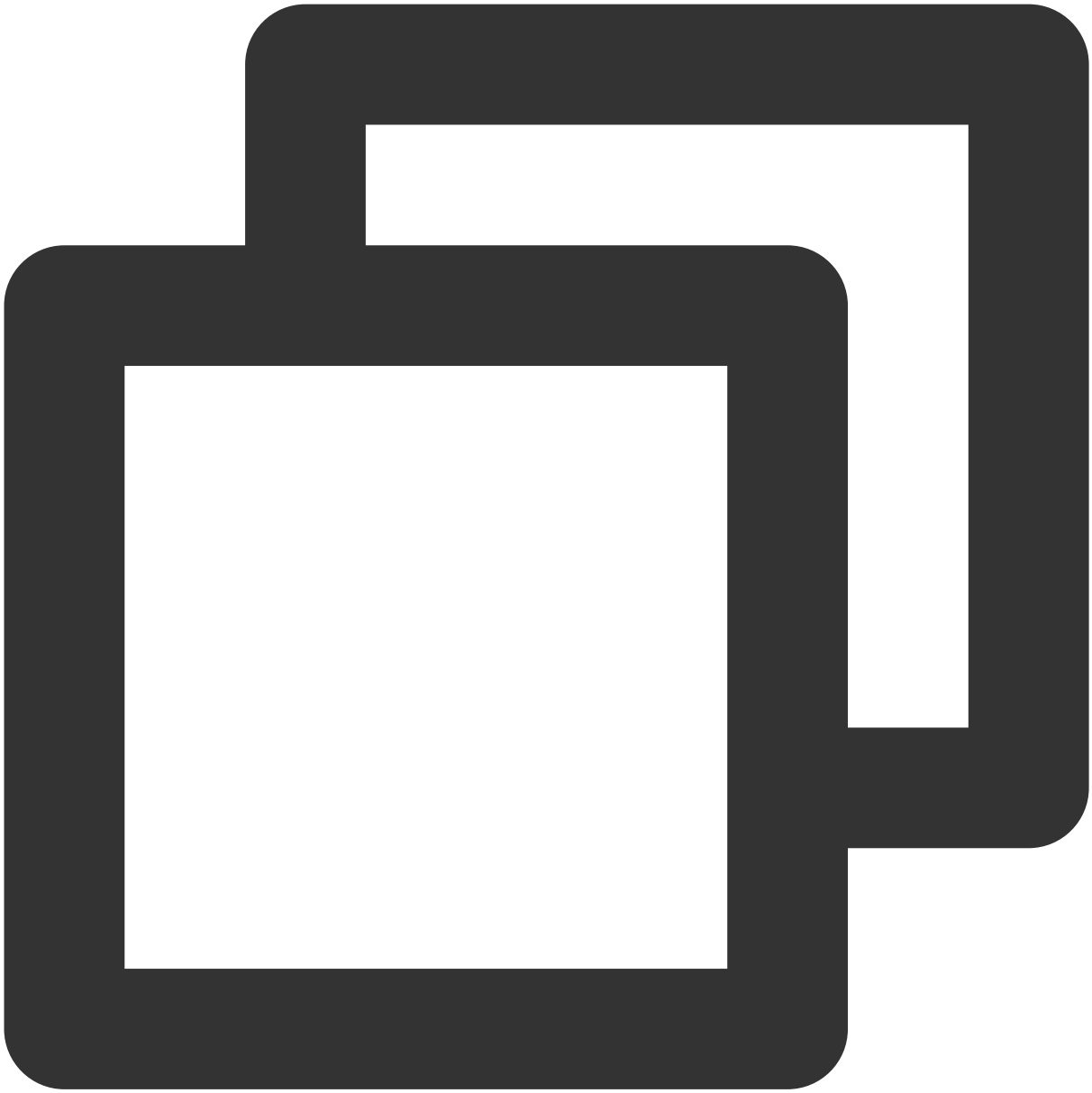
The user groups cannot be set via this API. Users who has signed up are included in the user group configured in the sign-up process by default.

This API does not handle automatic login and identity verification logic. The configurations of automatic login and identity verification in the sign-up process do not take effect on this API.

Supported Applications

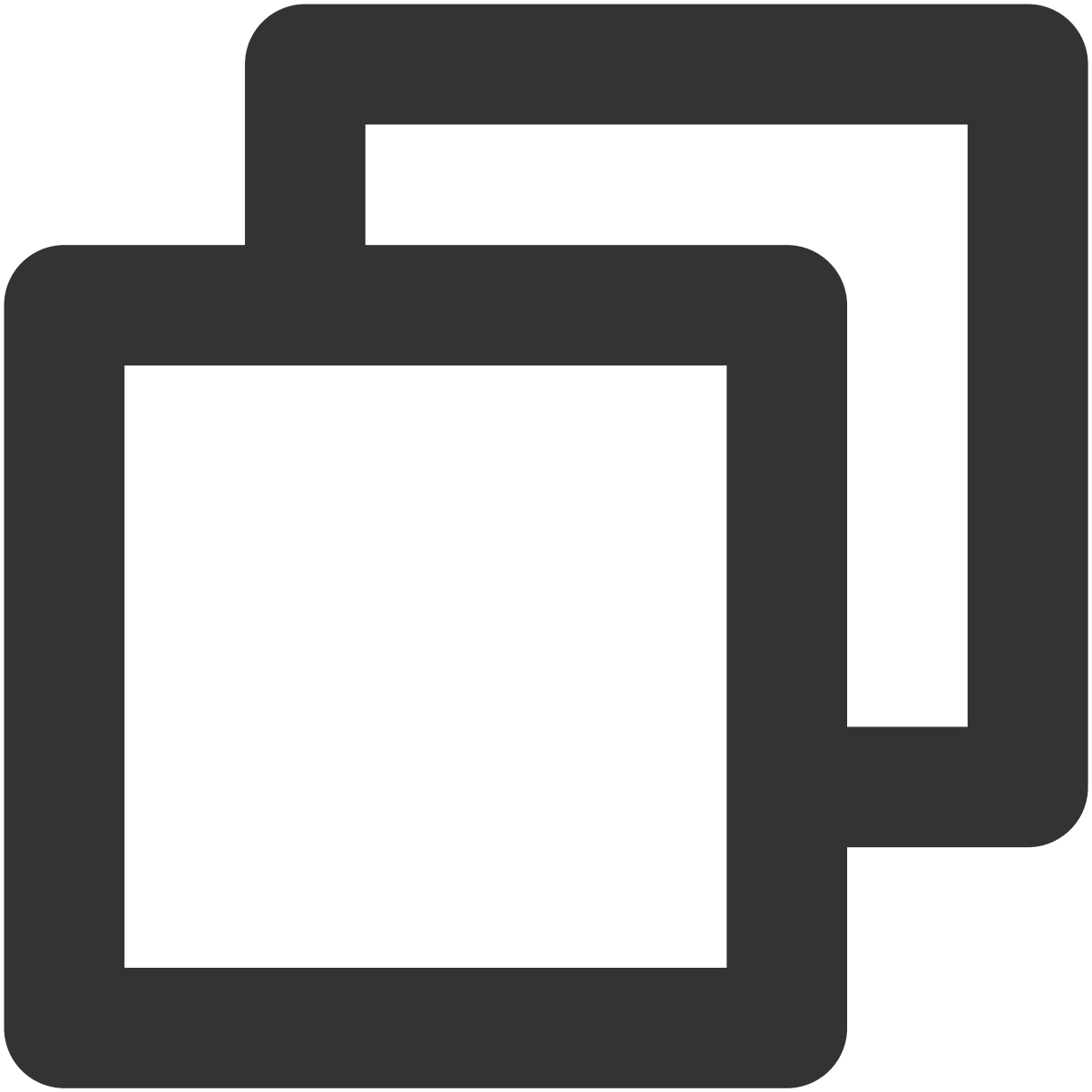
Web applications.

Request Method



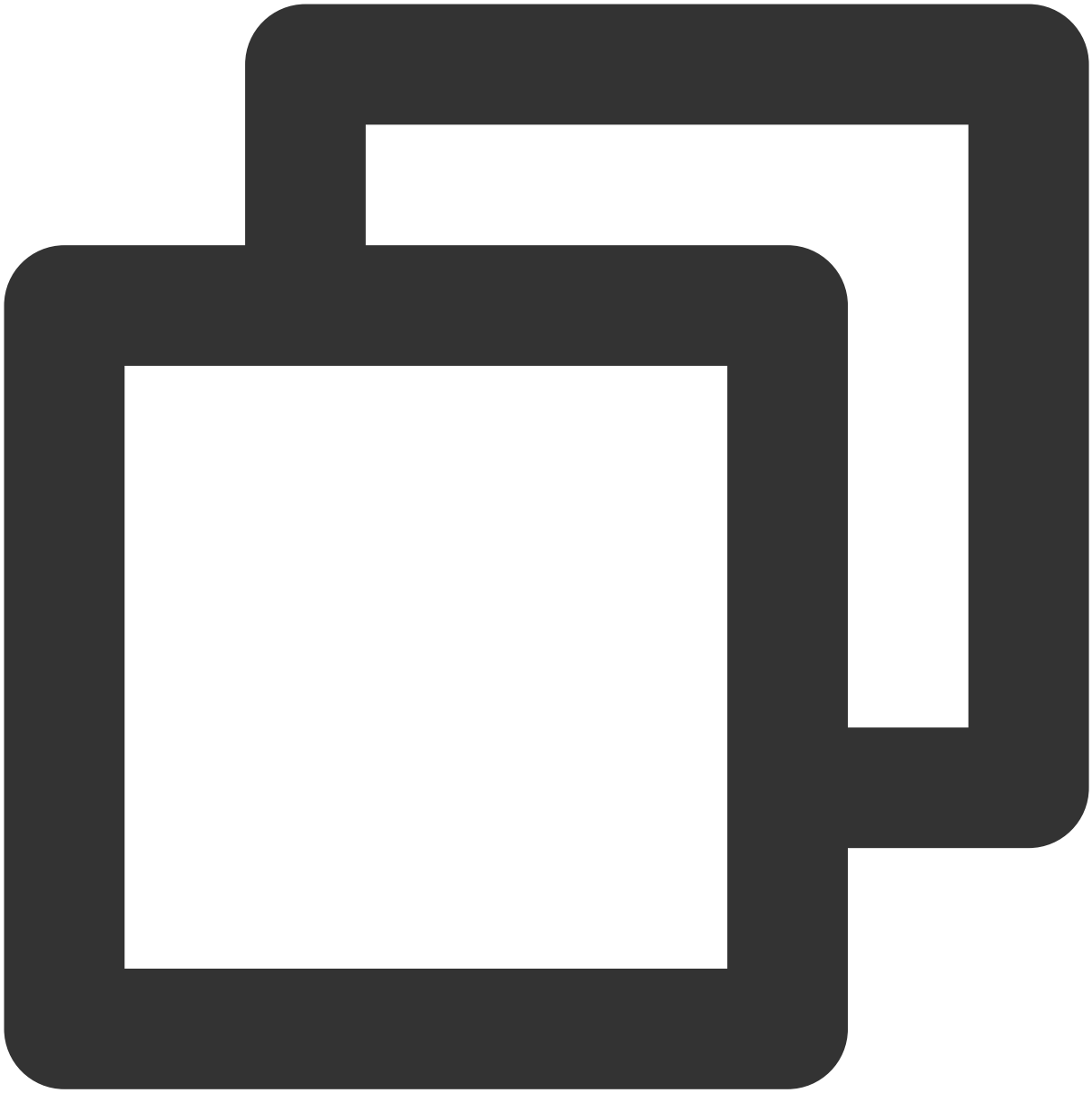
```
POST
```

Request Path



```
/signup
```

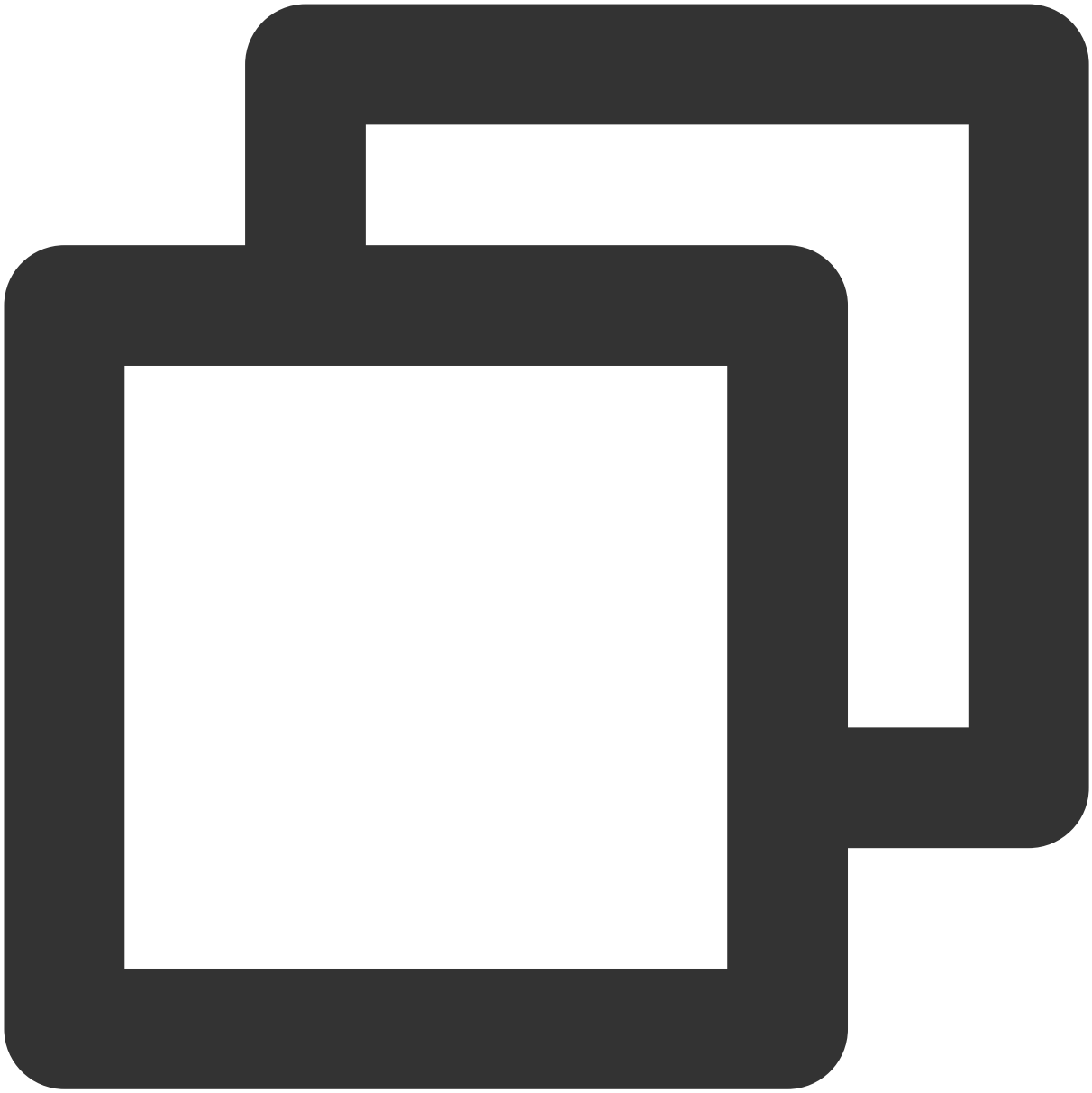
Request Content-Type



```
application/json
```

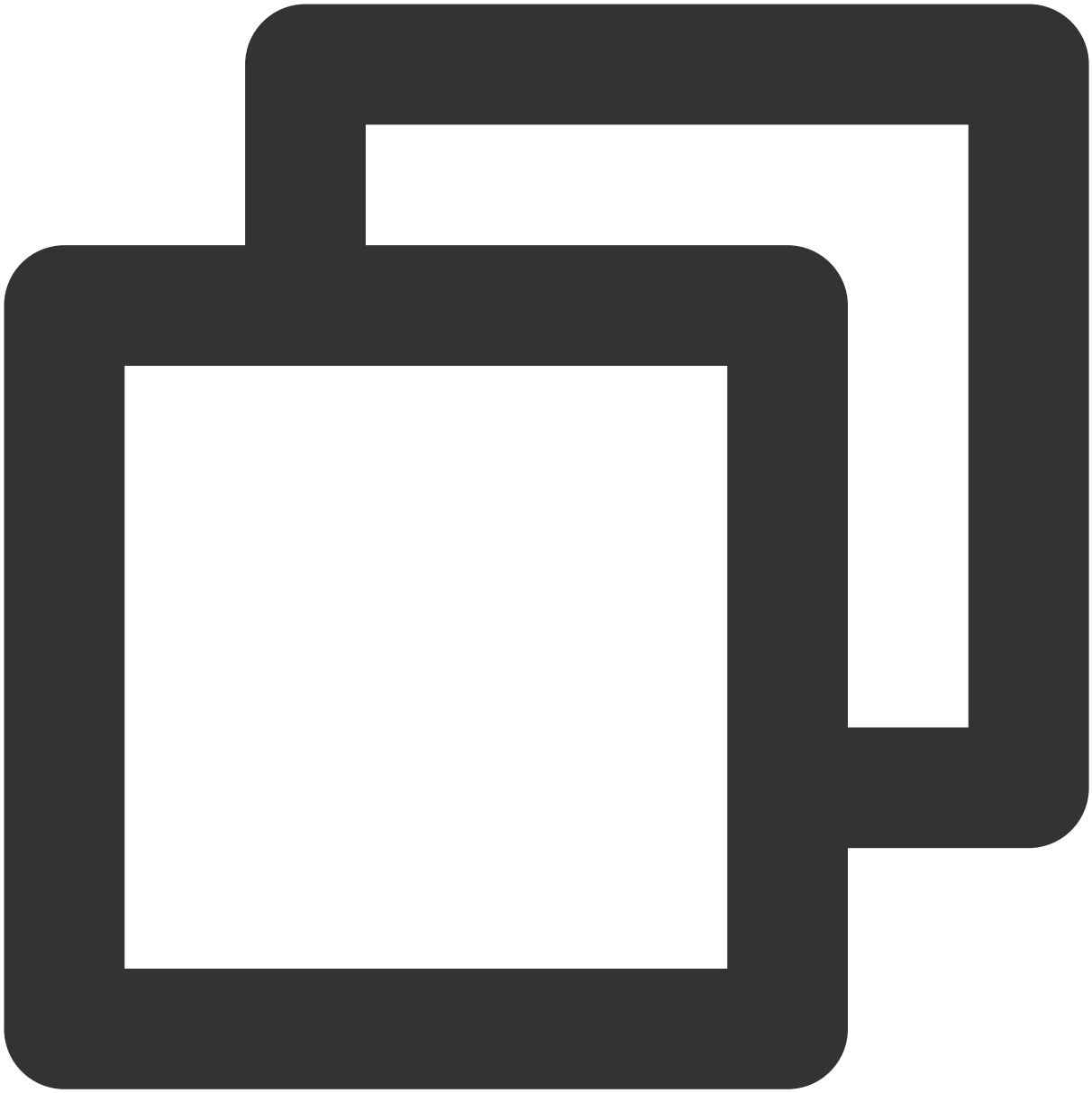
Sample Requests

Sign up with a username and set a password.



```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"username" : "MOCK_USERNAME",
"password" : "MOCK_PASSWORD"
}
```

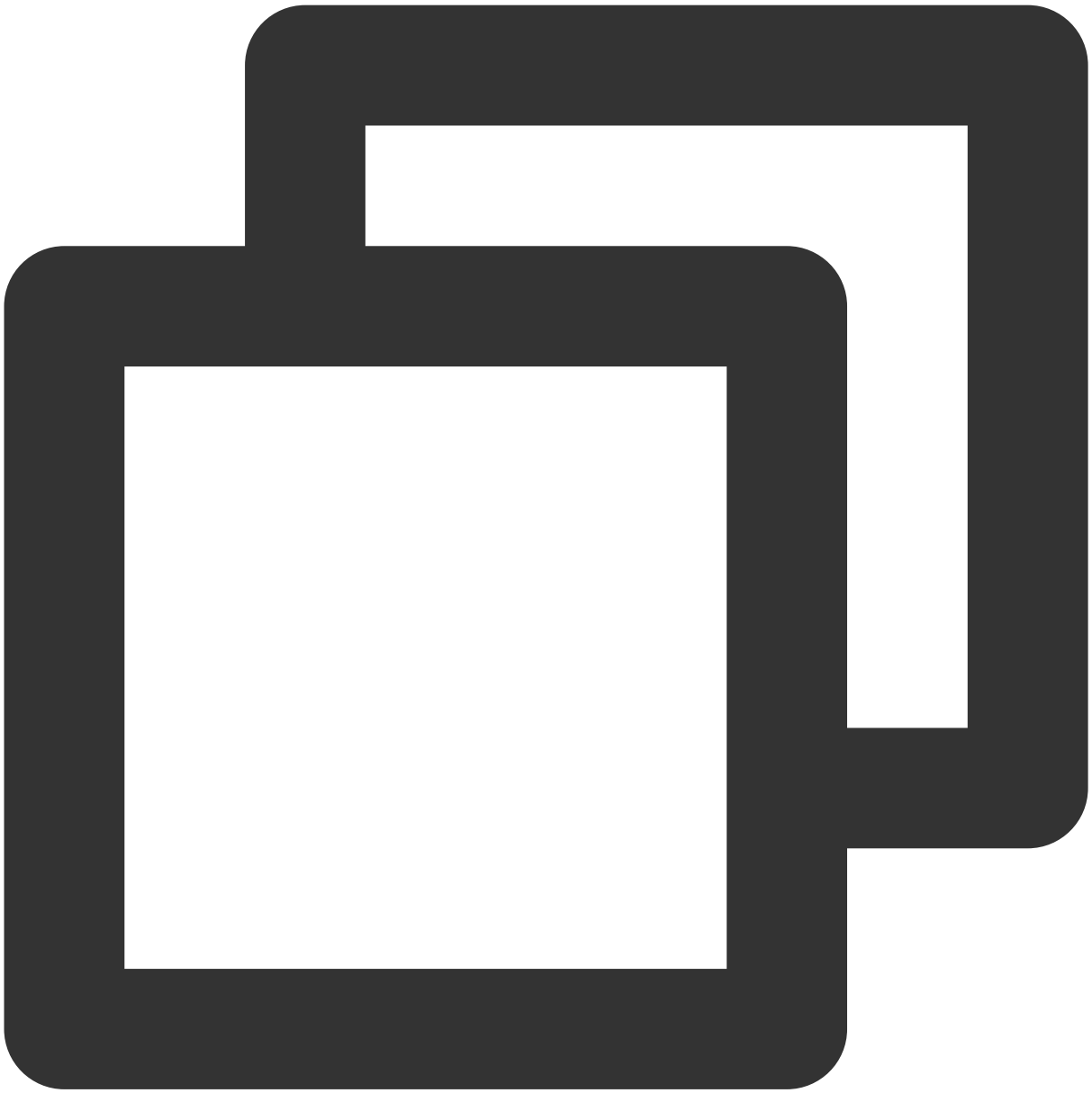
Sign up with an email and an alias and set a password.



```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"email" : "MOCK_USERNAME@example.com",
"email_otp_token" : "MOCK_EMAIL_OTP_TOKEN",
"email_otp" : "MOCK_EMAIL_OTP",
"password" : "MOCK_PASSWORD",
"nickname" : "MOCK_NICKNAME"
```

```
}
```

Sign up with a mobile number without setting a password.



```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"phone_number" : "13612345678",
"phone_number_otp_token" : "MOCK_PHONE_NUMBER_OTP_TOKEN",
```

```
"phone_number_otp" : "MOCK_PHONE_NUMBER_OTP"
}
```

Request Headers

Parameter	Description
Authorization	HTTP `Basic` authentication request header. The format is `Basic <credentials>`, where `Basic` is a fixed string and `<credentials>` is calculated by `base64(url_encode(client_id) + ":" + url_encode( client_secret))`. `Basic` and `<credentials>` are separated by a space.

Request Parameters in JSON Format

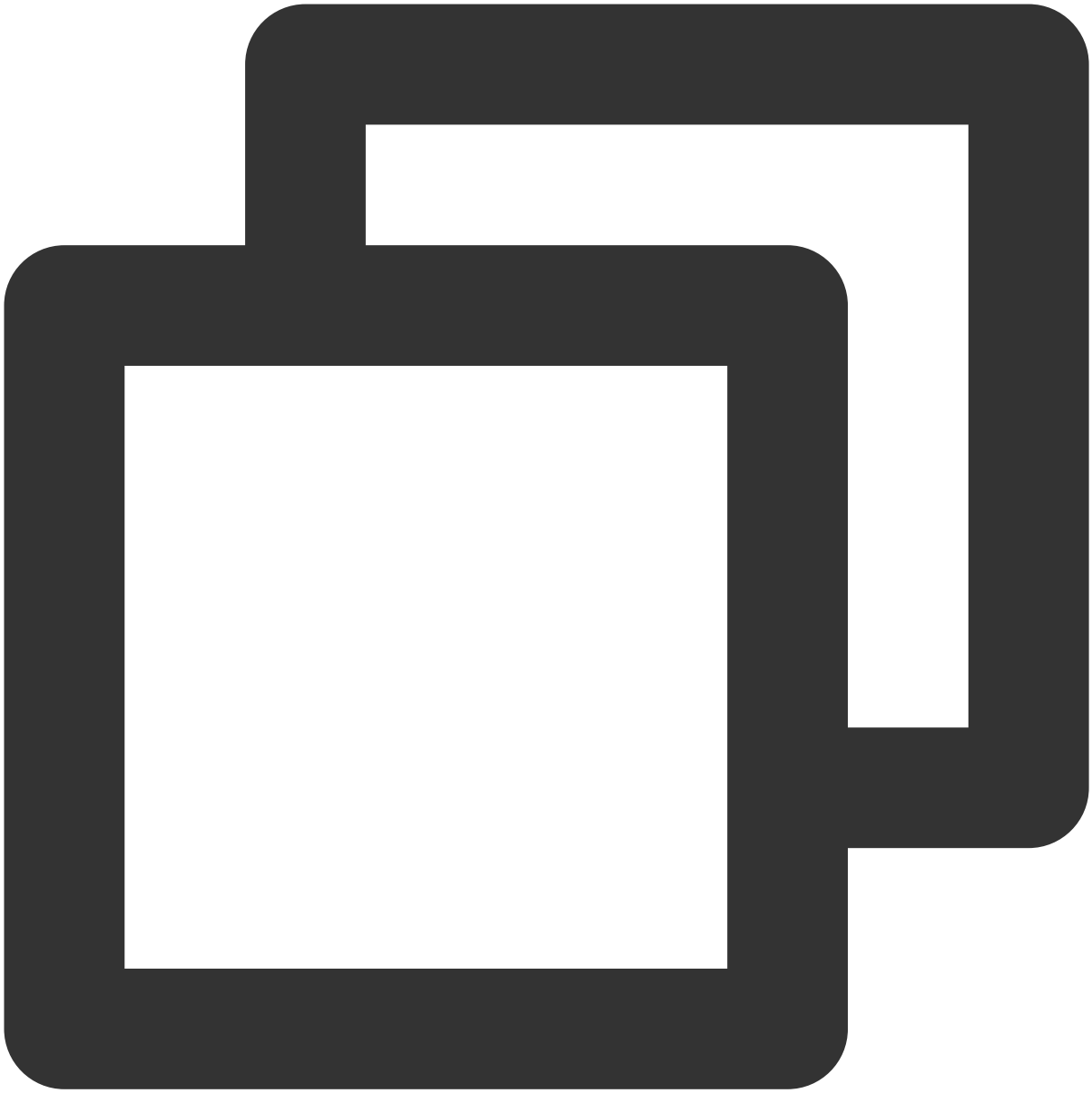
JSON Path	Data Type	Description
username	String	Username. It can contain up to 32 characters of letters, numbers and underscores. It must start with a letter.
password	String	User password. The password you set must comply with the policies of the account and password authentication source associated with the application.
phone_number	String	The user's mobile number, which should be an 11-digit mobile number of the three major carriers in Chinese mainland. When this parameter is specified, both `phone_number_otp_token` and `phone_number_otp` are required.
hone_number_otp_token	String	The `otp_token` returned by the server after the SMS verification code is sent.
phone_number_otp	String	The OTP verification code received by the user's mobile phone.
email	String	The user's email address. When this parameter is specified, both `email_otp_token` and `email_otp` are required.
email_otp_token	String	The `otp_token` returned by the server after the email verification code is sent.
email_otp	String	The OTP verification code received by the user's email.

Name	String	Username
nickname	String	The user's alias.
zoneinfo	String	The user's time zone, such as `Asia/Shanghai` or `Europe/Paris`.
locale	String	The user `locale` information, such as `zh-CN` or `en-US`.

Note:

The values of other parameters are user attribute identifiers. The attribute identifiers can be viewed on the Custom Attributes page.

Sample Success Responses of Sign-up



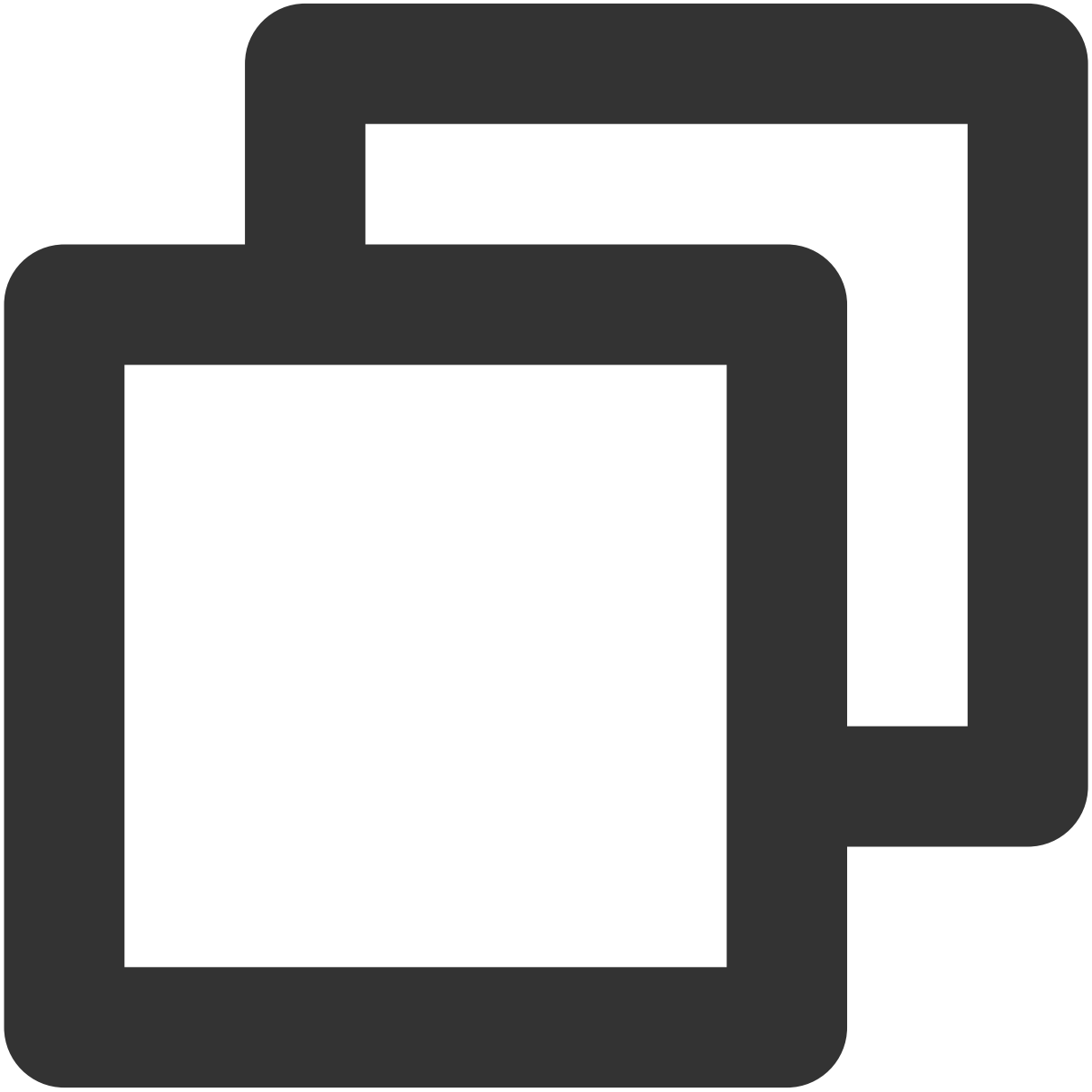
```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "sub" : "MOCK_USER_ID"
}
```

Response Parameters

Parameter	Data Type	Description
sub	String	Unique identifier of the user.

Sample Error Responses of Sign-up

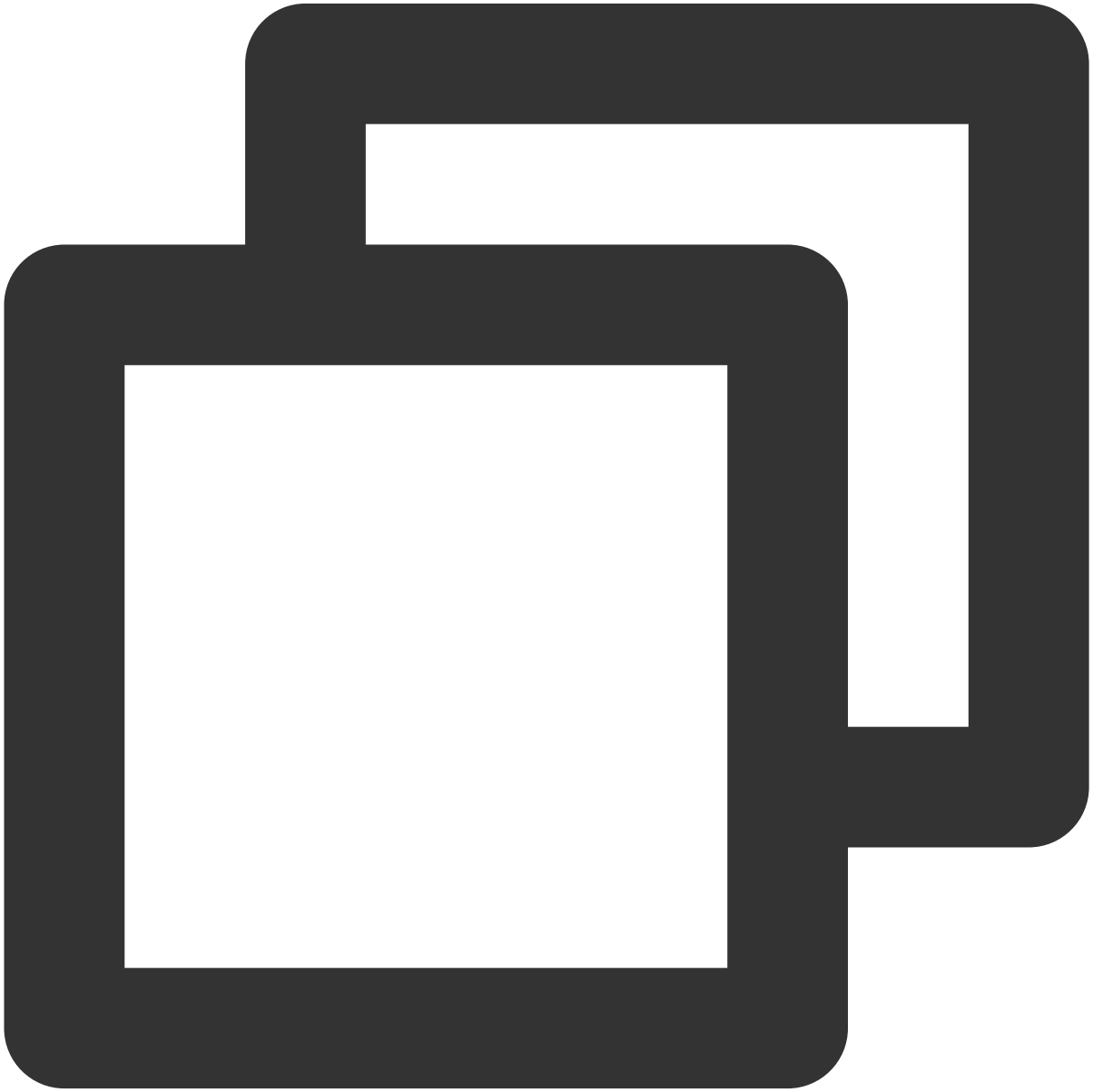
The sign-up process of the application is not enabled.



```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "misconfigured",
"error_description" : "Sign up flow of the application is not enabled."
}
```

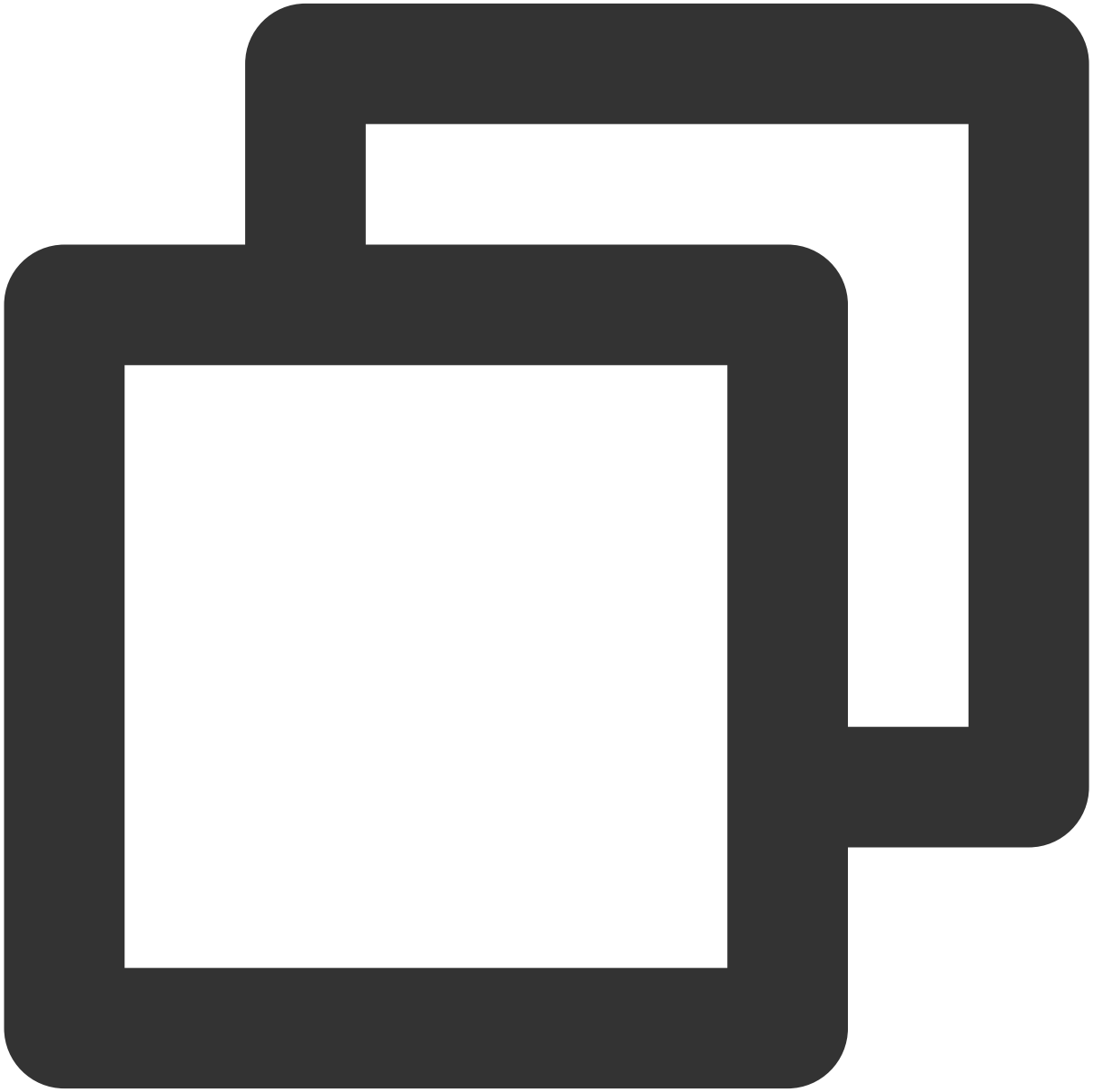
The input parameters do not contain the authentication attributes or required general attributes configured in the sign-up process.



```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Missing required sign-up attribute(s)."
}
```

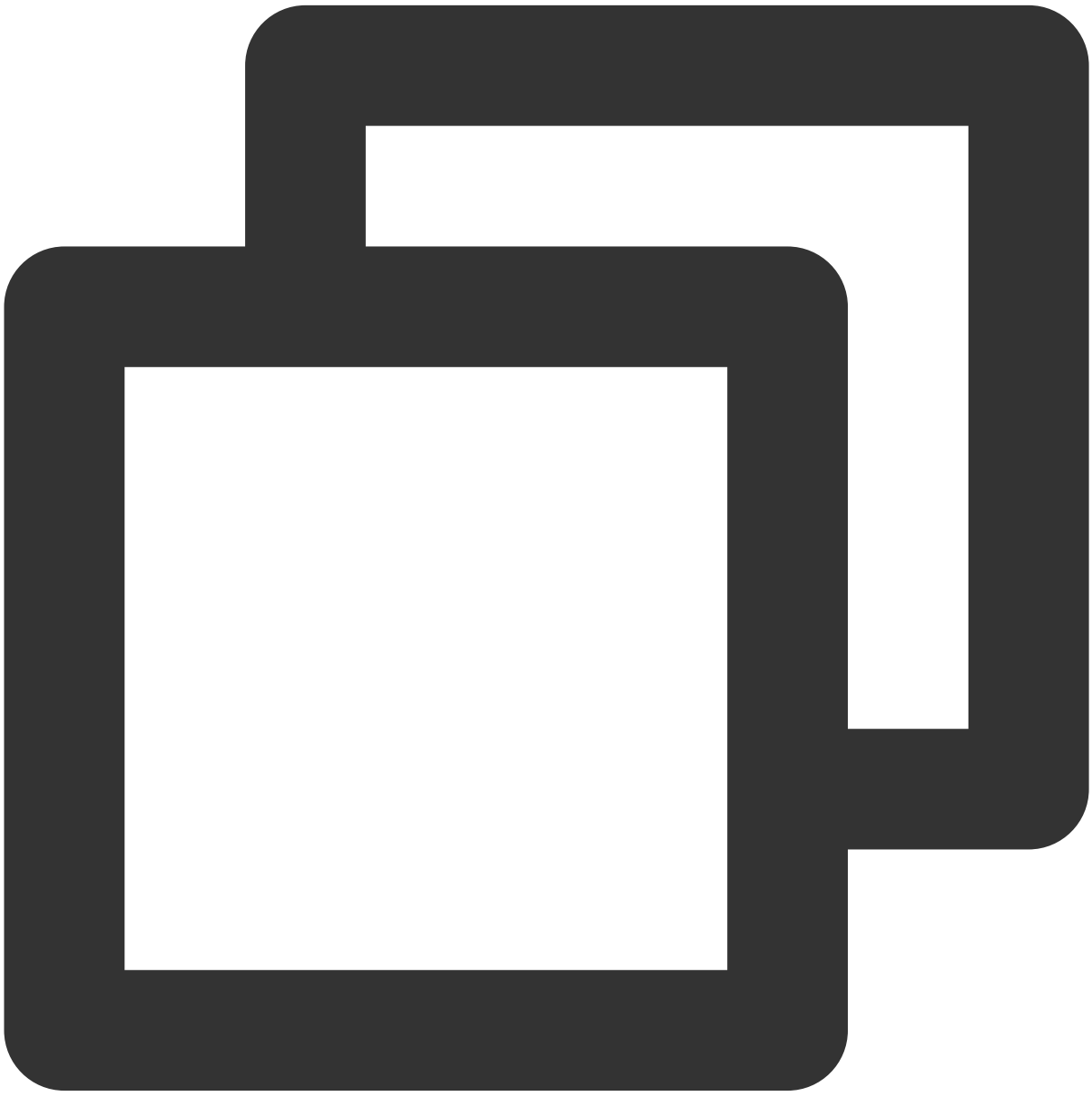
The input parameters contain authentication attributes or general attributes that are not configured in the sign-up process.



```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unconfigured sign-up attribute(s) found."
}
```

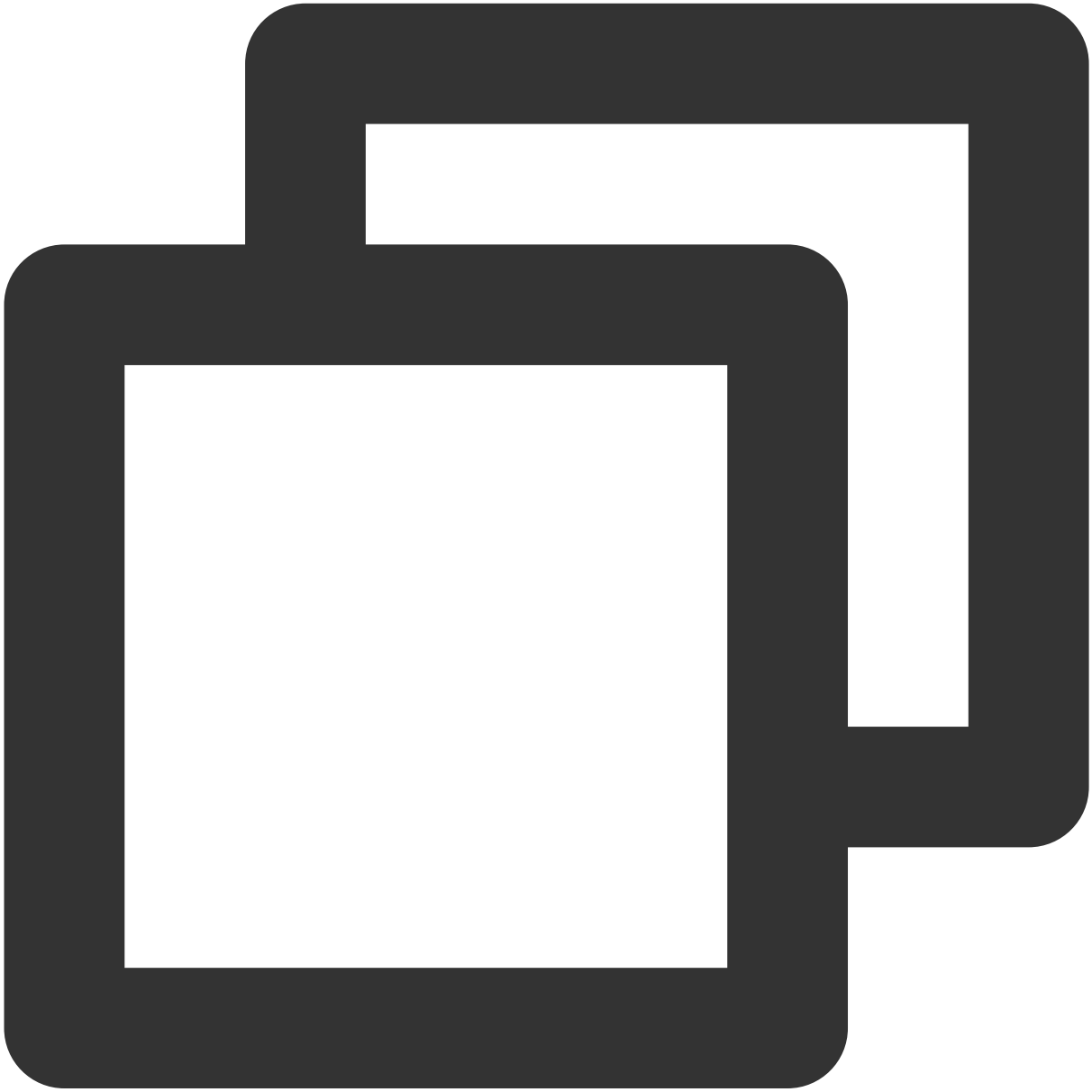
The input parameters contain unknown attributes.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
```

```
{
"error" : "invalid_request",
"error_description" : "Unknown attribute(s) found."
}
```

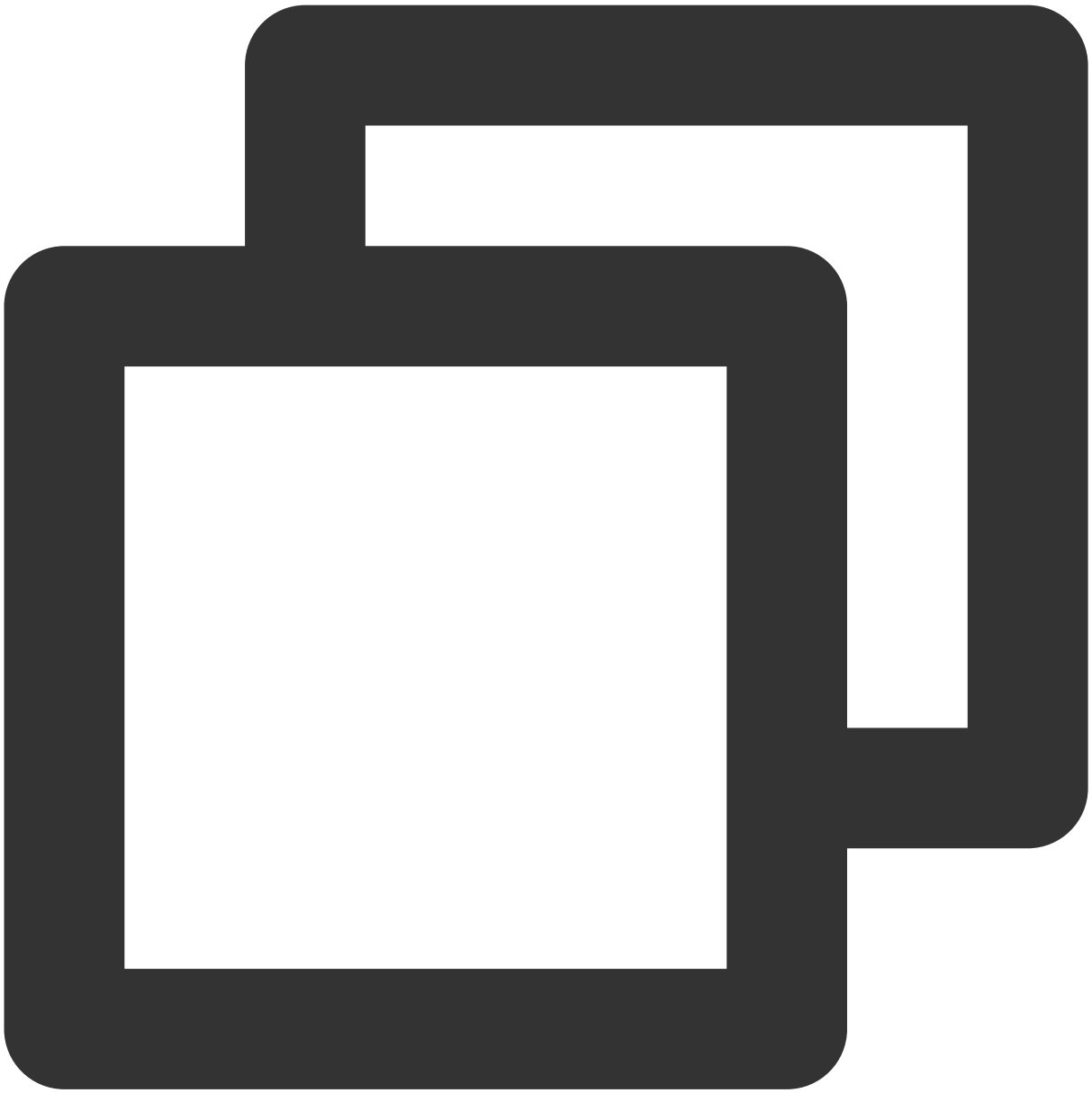
The username format is invalid.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_username"
```

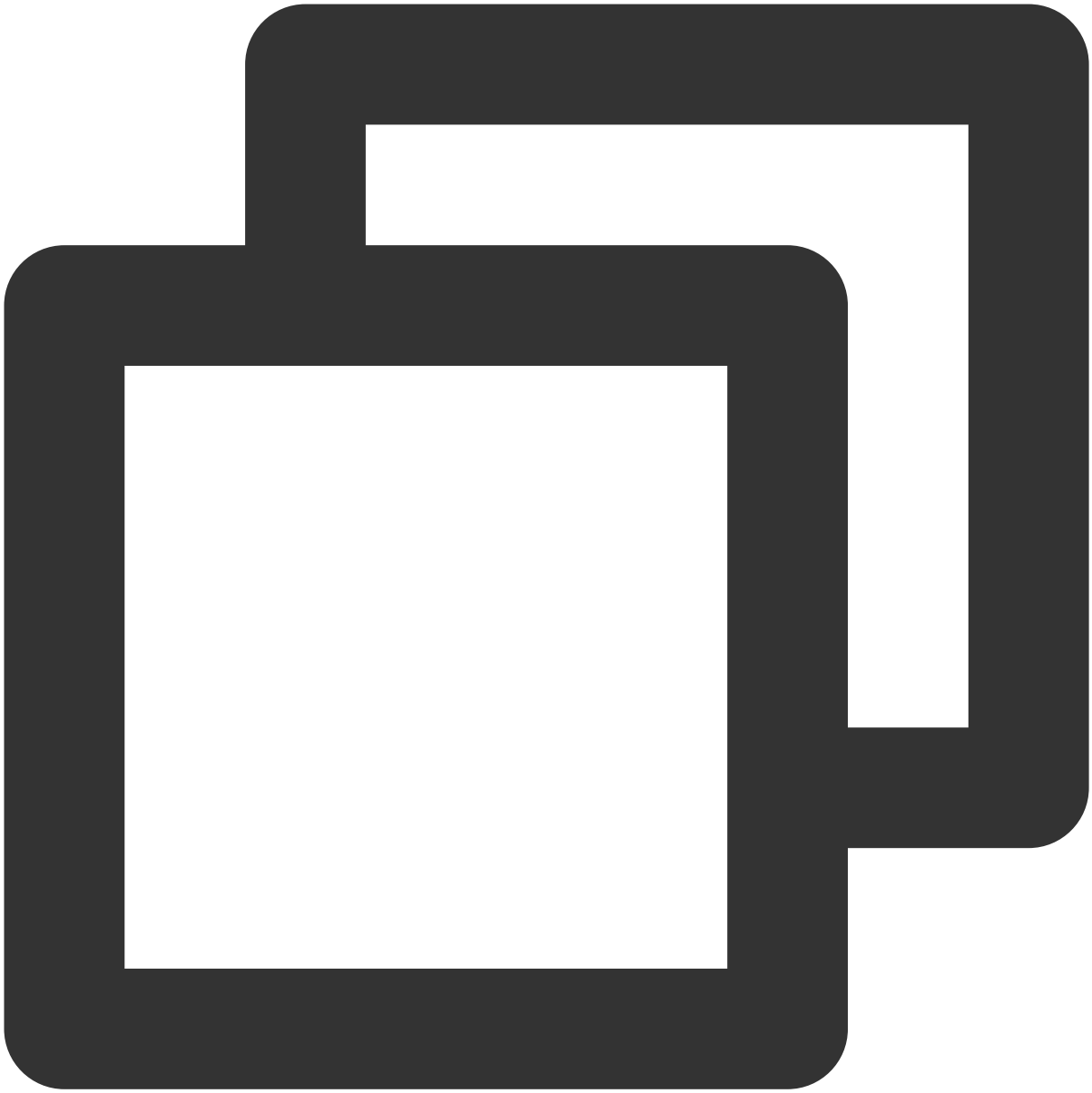
```
}
```

The username already exists.



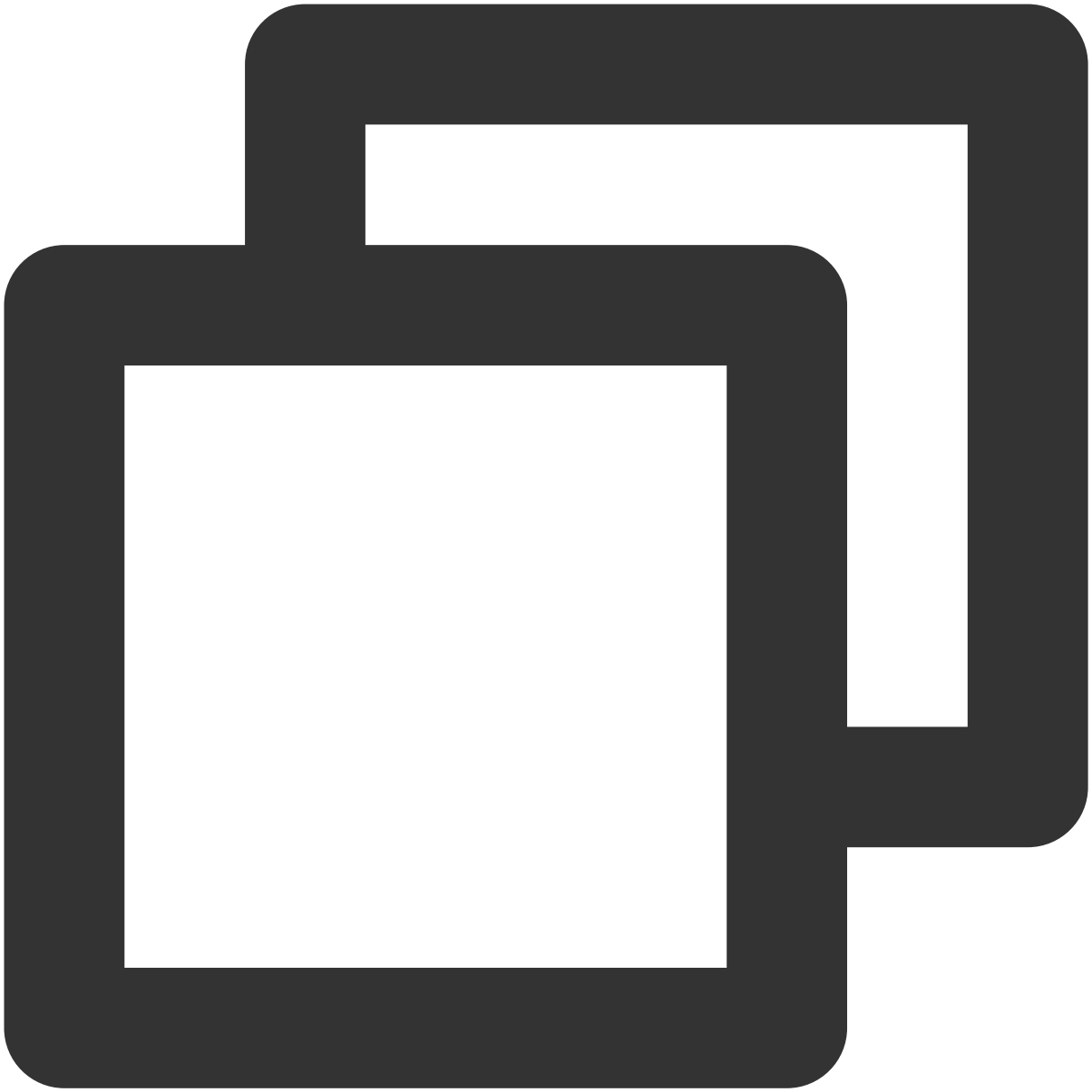
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_username"
}
```

The mobile number format is invalid.



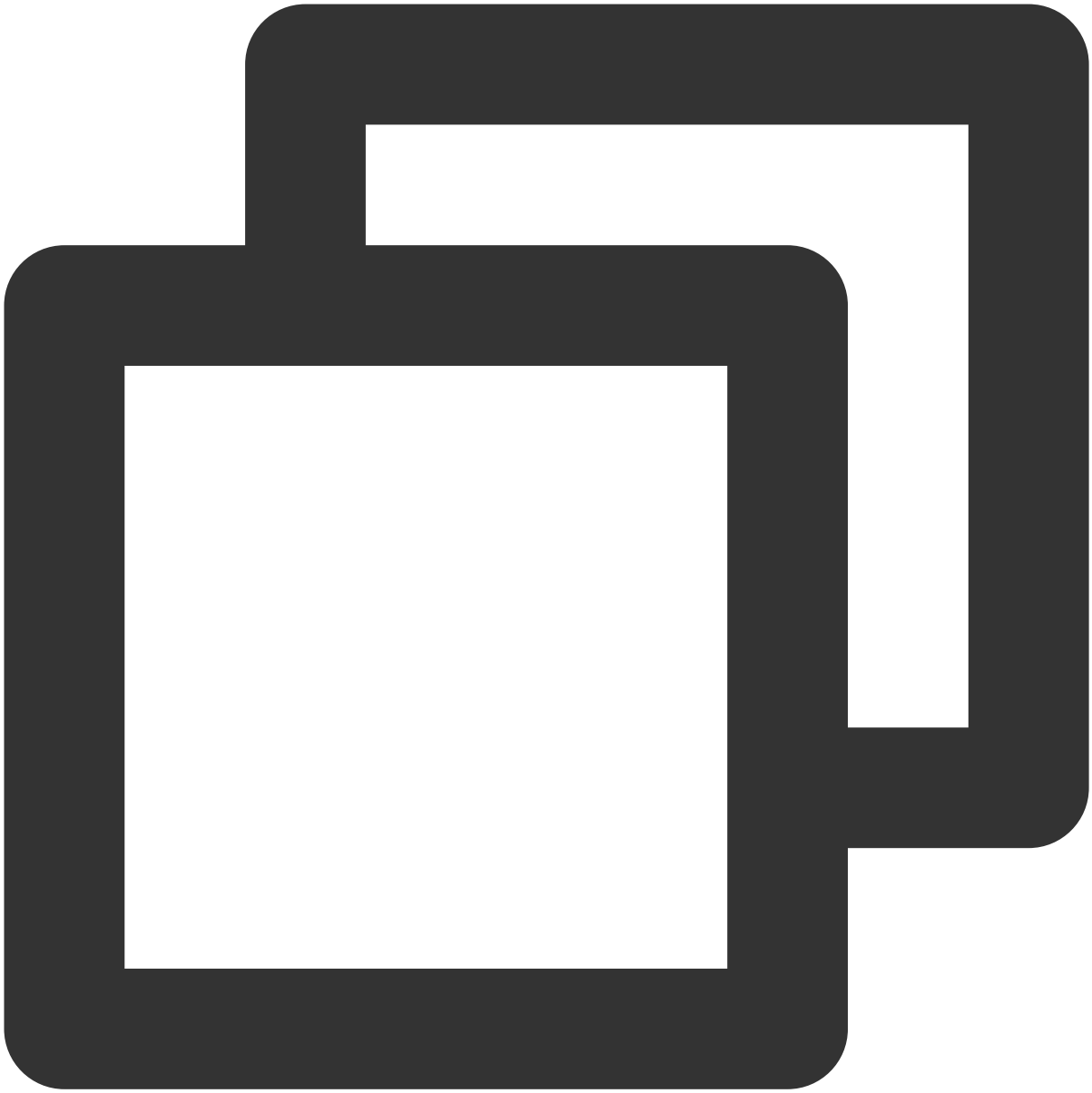
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

The mobile number already exists.



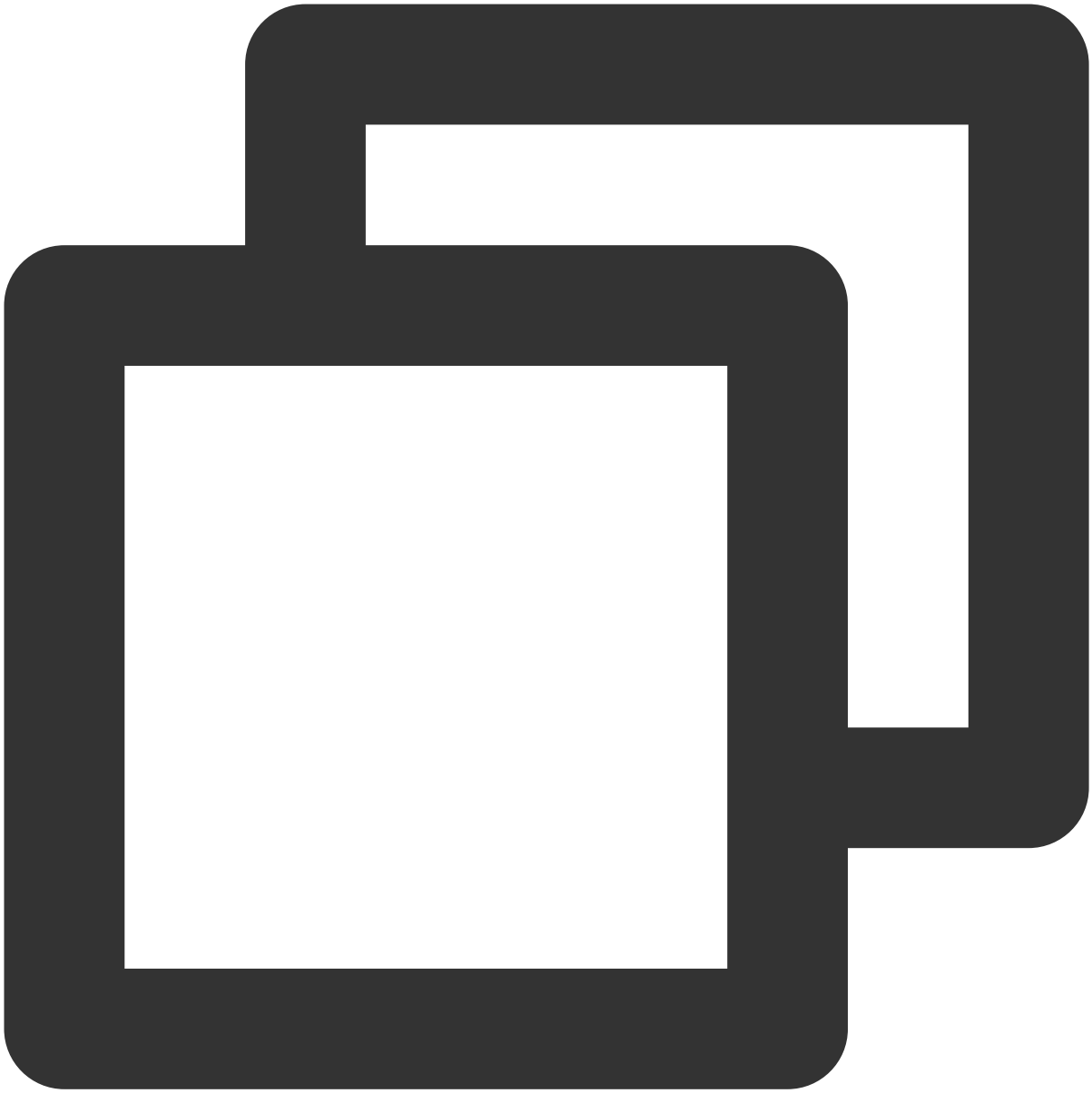
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_phone_number"
}
```

The `phone_number_otp_token` parameter is incorrect or has expired, or the parameter used for sign-up is not the same as the one for sending the verification code. For example, the mobile numbers are different.



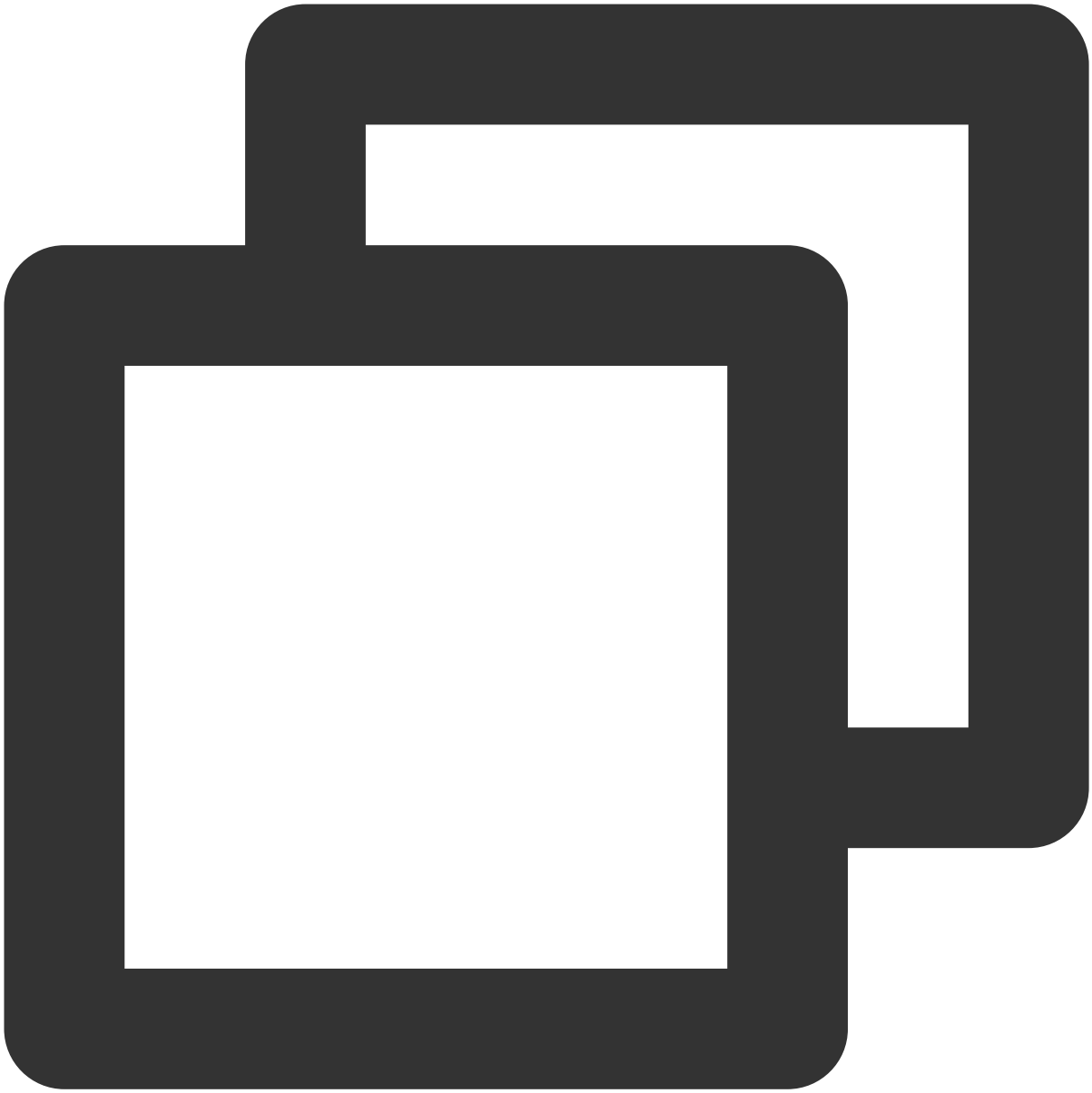
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp_token"
}
```

The `phone_number_otp` parameter is incorrect or has expired.



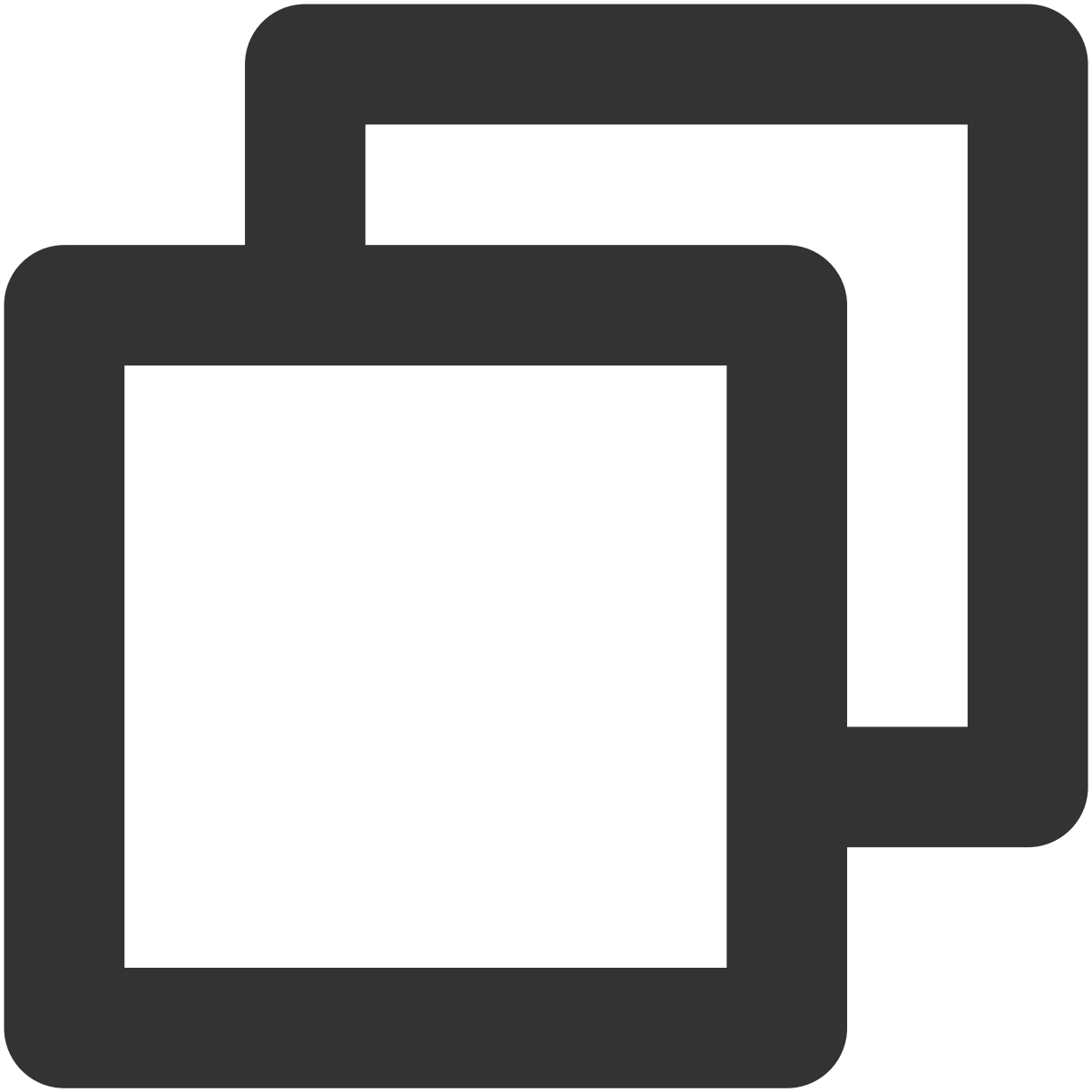
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp"
}
```

The email address format is invalid.



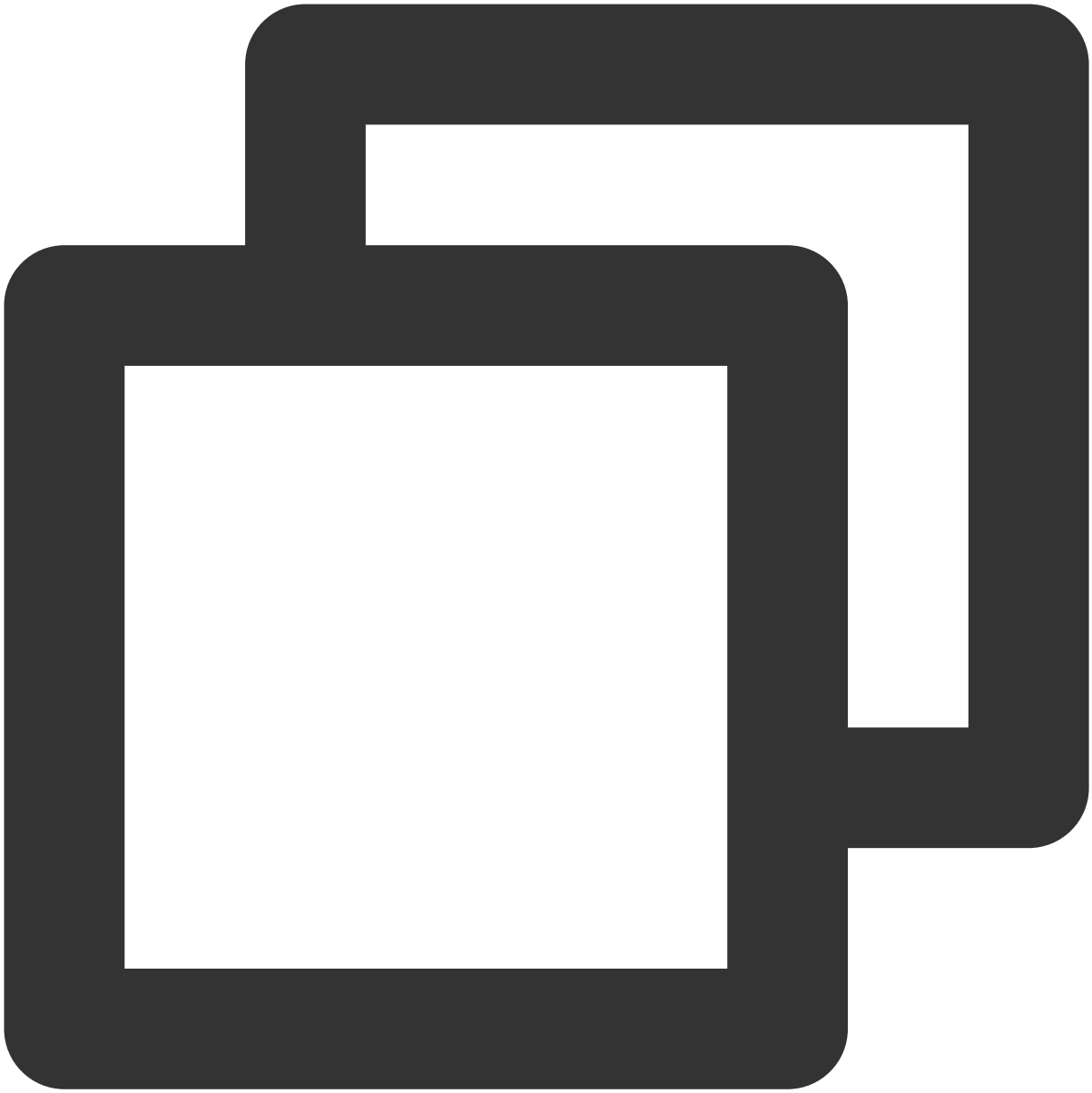
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

The email address already exists.



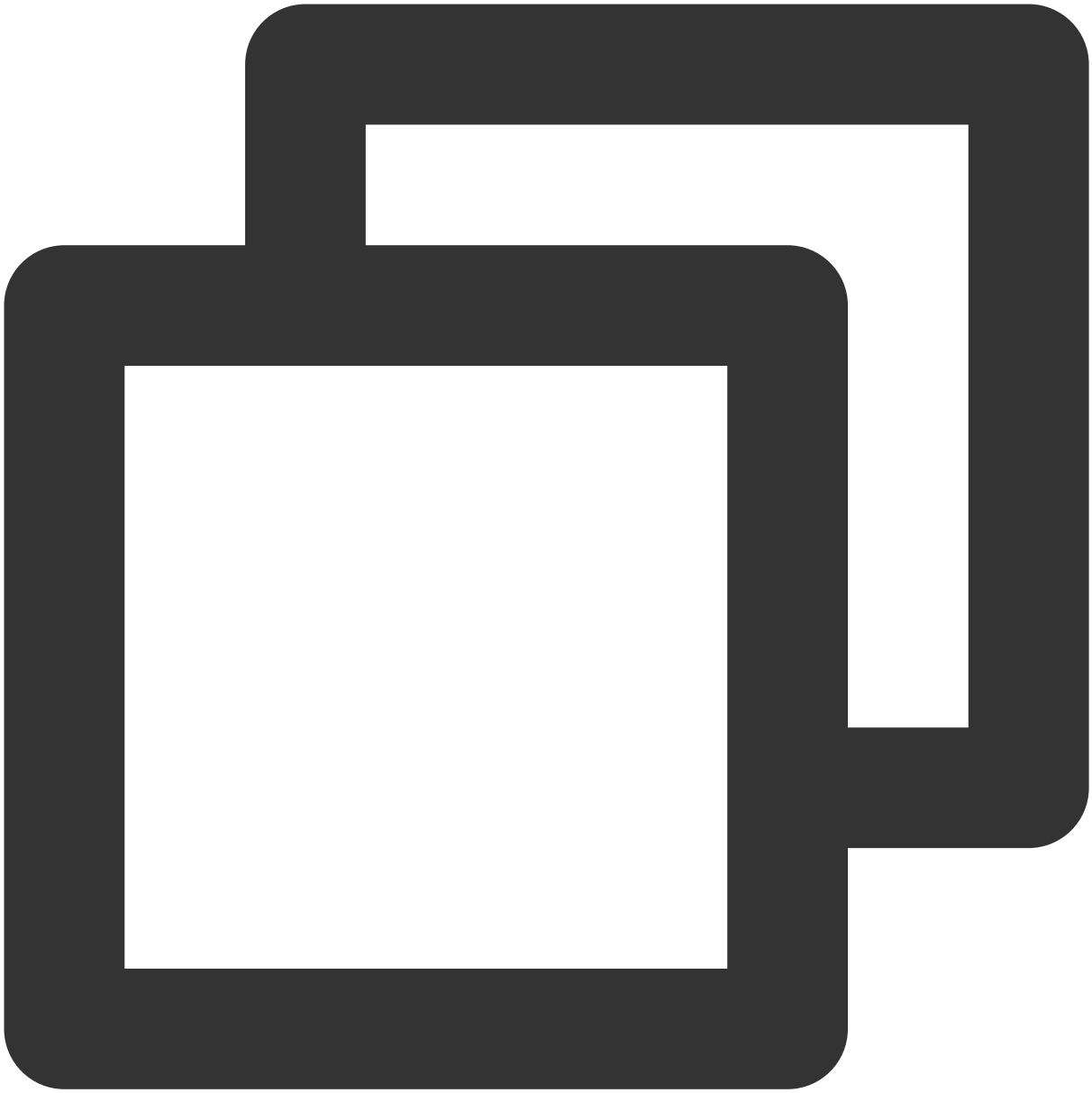
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_email"
}
```

The `email_otp_token` parameter is incorrect or has expired, or the parameter value used for sign-up is not the same as the one used for sending the verification code. For example, the email addresses are different.



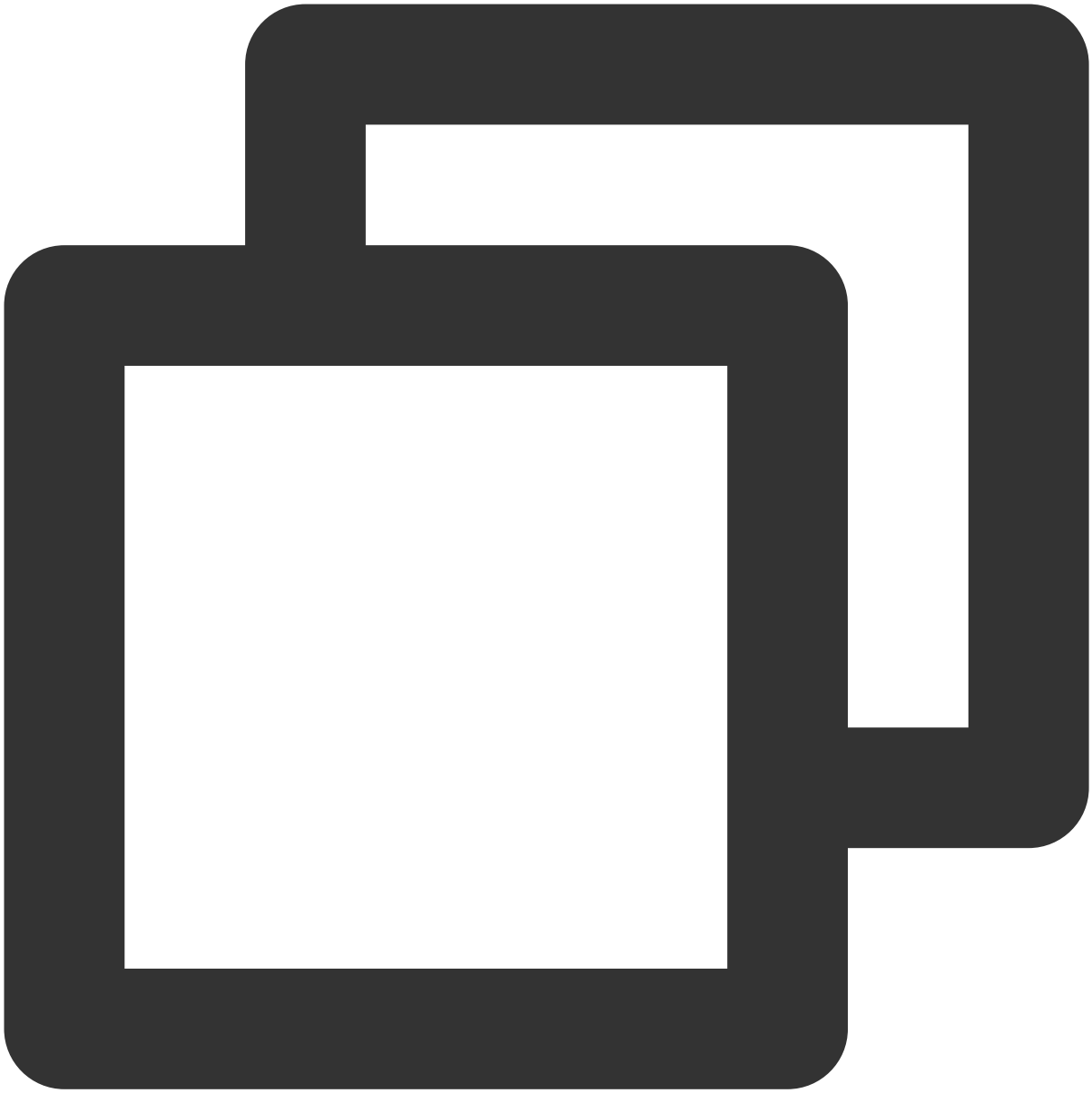
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

The `email_otp` parameter is incorrect or has expired.



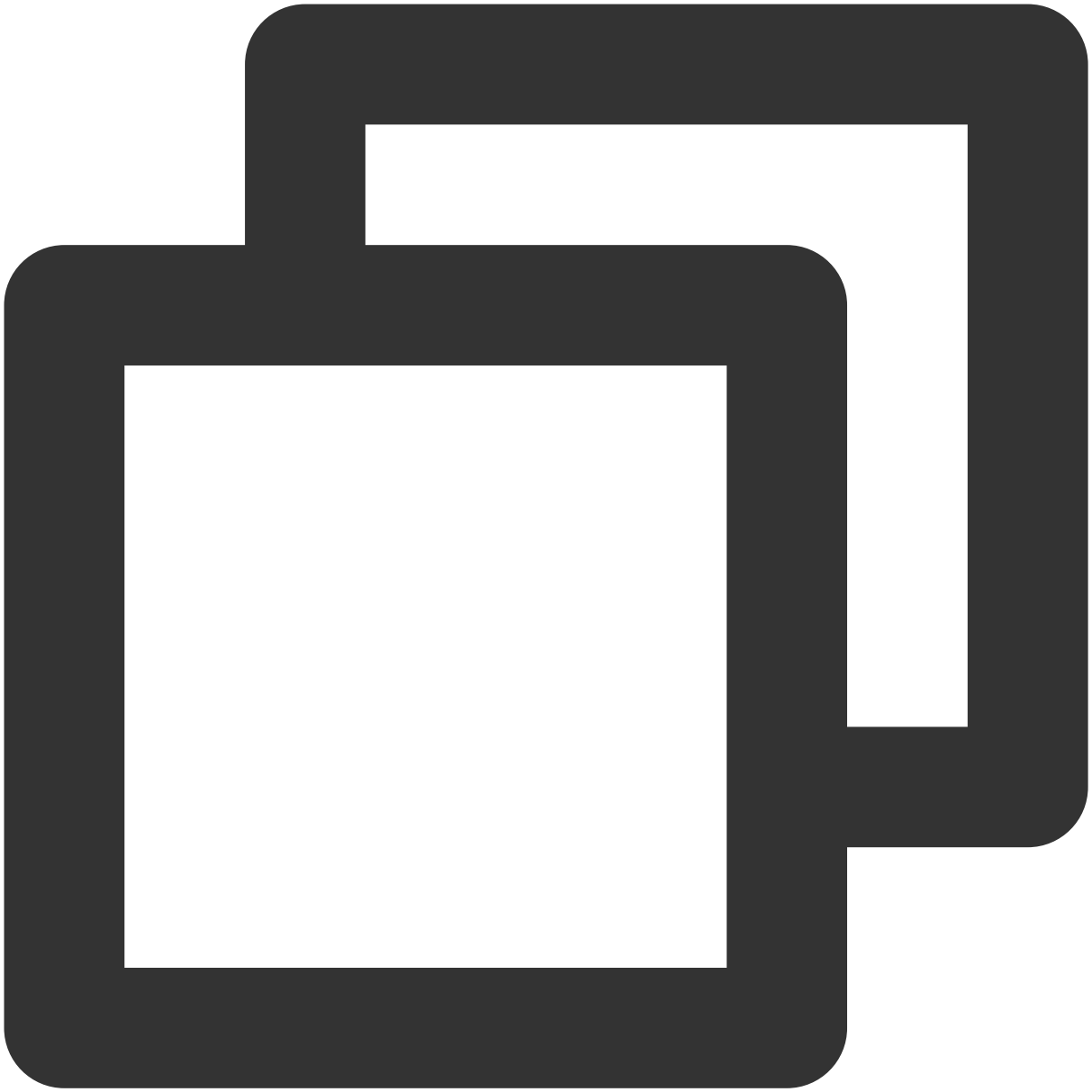
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

A password is passed as an input parameter, but the account and password authentication source is not associated in the application login process.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "misconfigured",
"error_description" : "No password auth source is associated with the application."
}
```

The password does not meet the policy requirements.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_password"
}
```

Account and Password Authentication

Last updated：2023-12-22 11:42:07

API Description

This API is used to verify the username and password and get the Access Token and ID Token for login. This API adopts the Resource Owner Password Credentials mode of the OAuth 2.0 protocol for authentication.

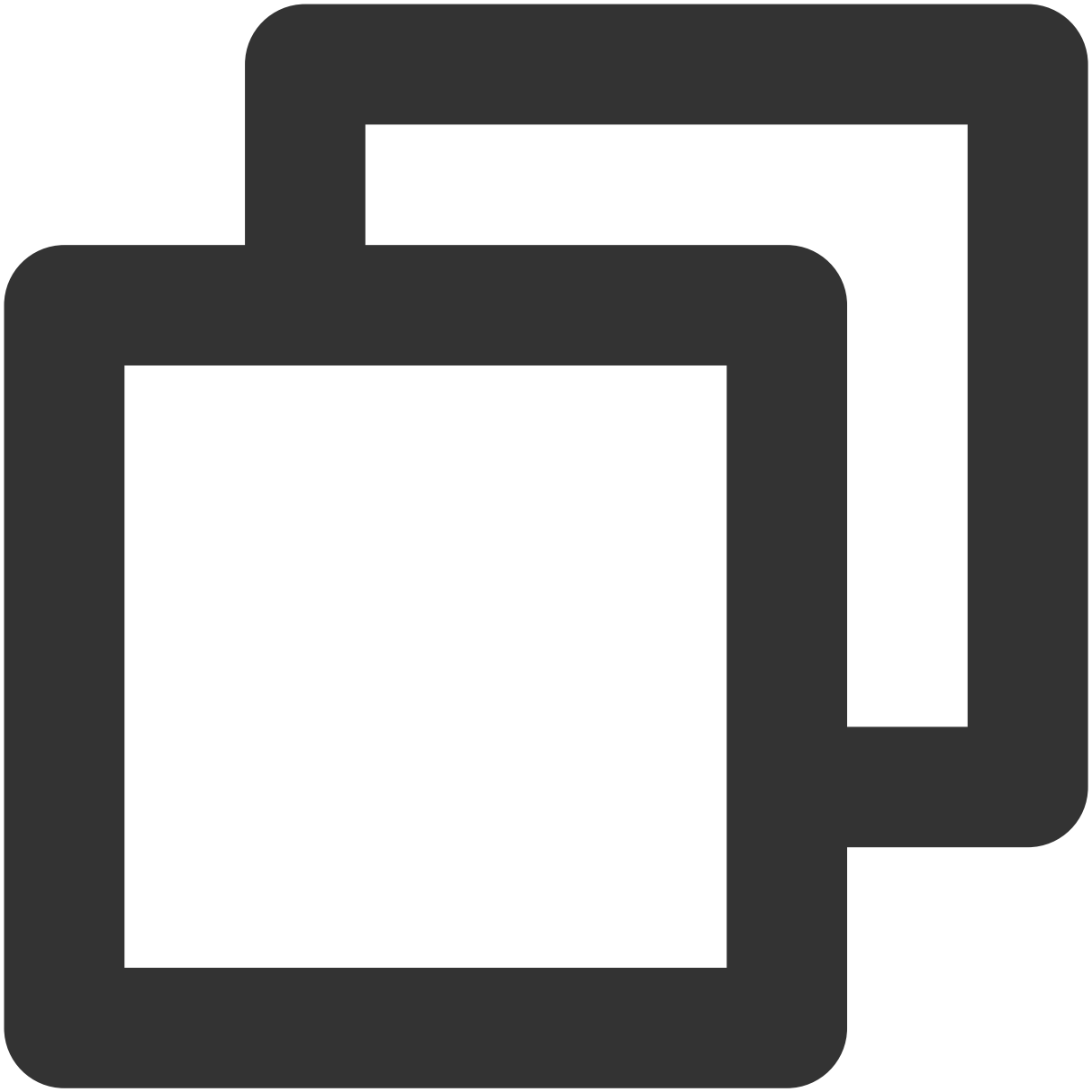
Note:

The password is passed between the user's client and the application. Make sure that you use a trusted application to call this API, and properly transmit the password. For example, make sure that the HTTPS protocol is used. We recommend that you use Login via Authentication Portal first if possible.

Supported Applications

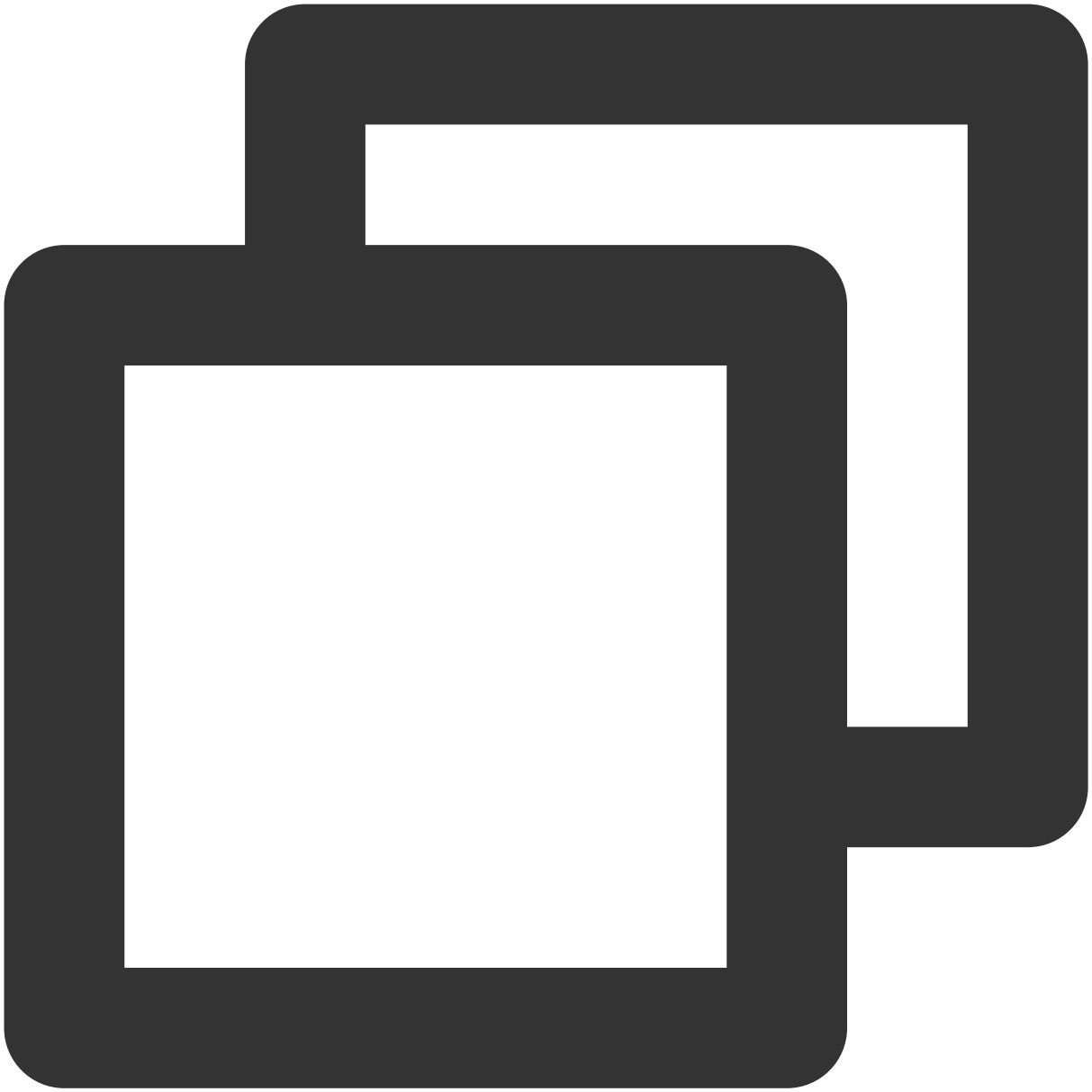
Web applications, single-page applications (SPA), and mobile applications.

Request Method



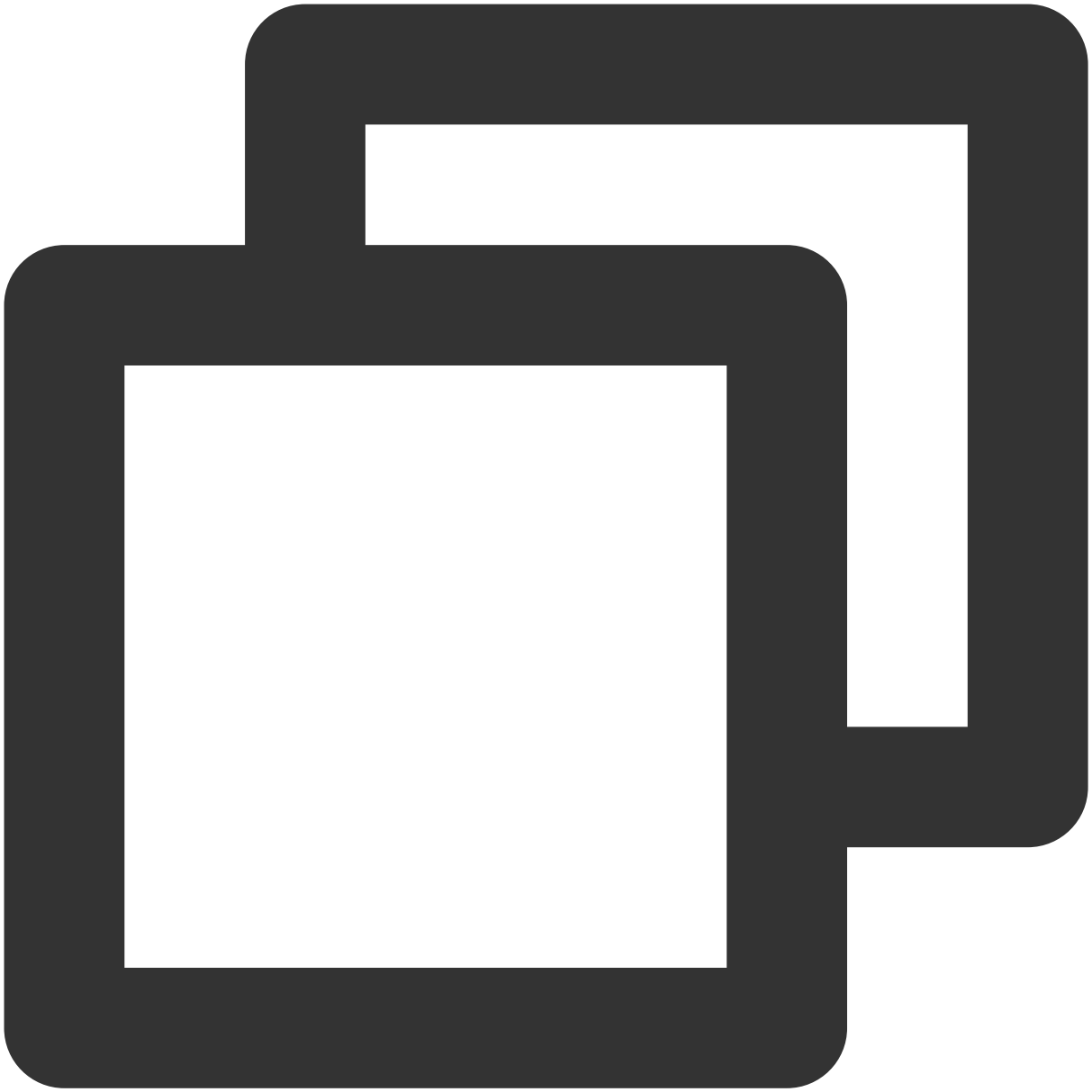
```
POST
```

Request Path



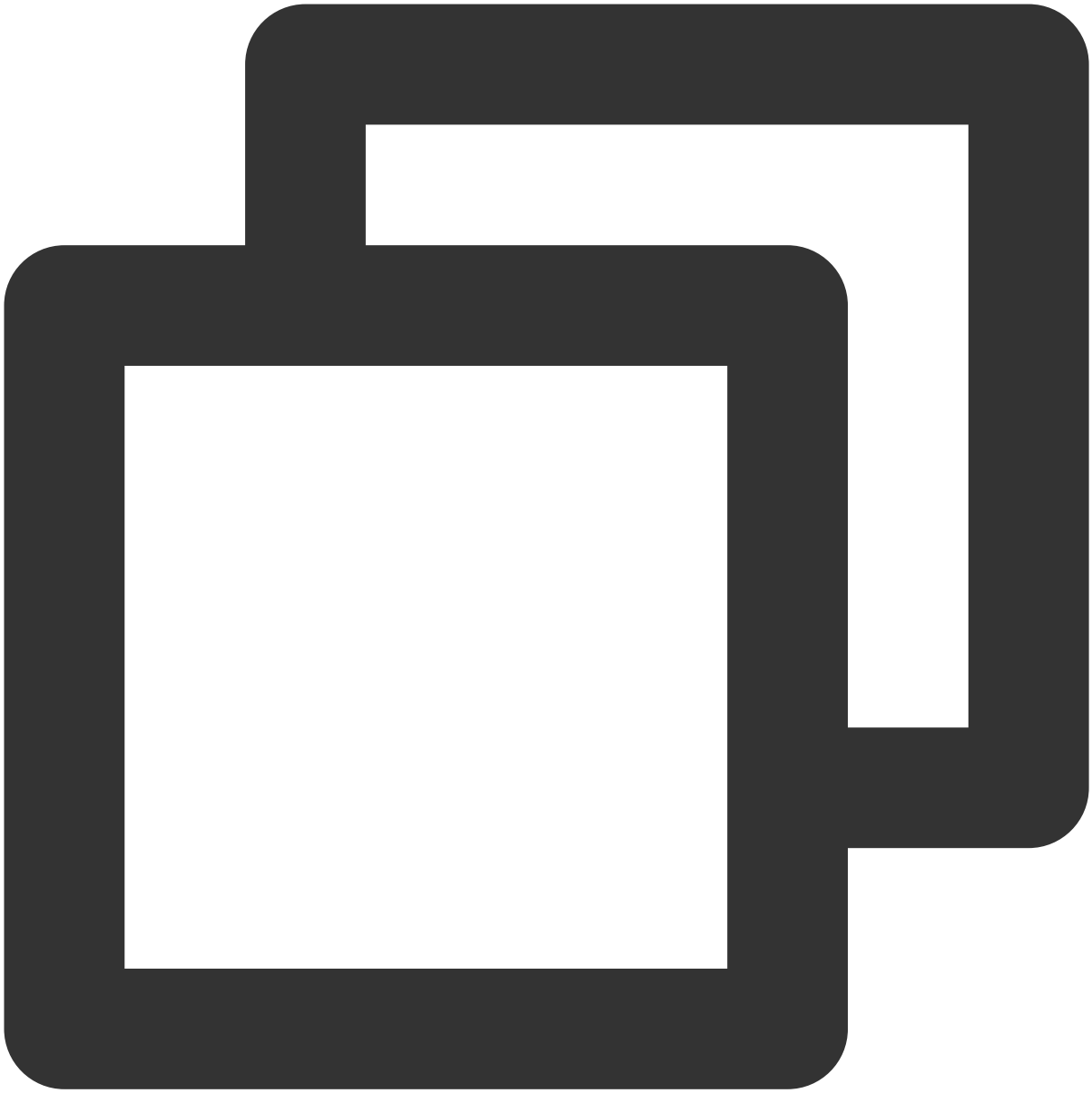
```
/oauth2/token
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests



```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&a
```

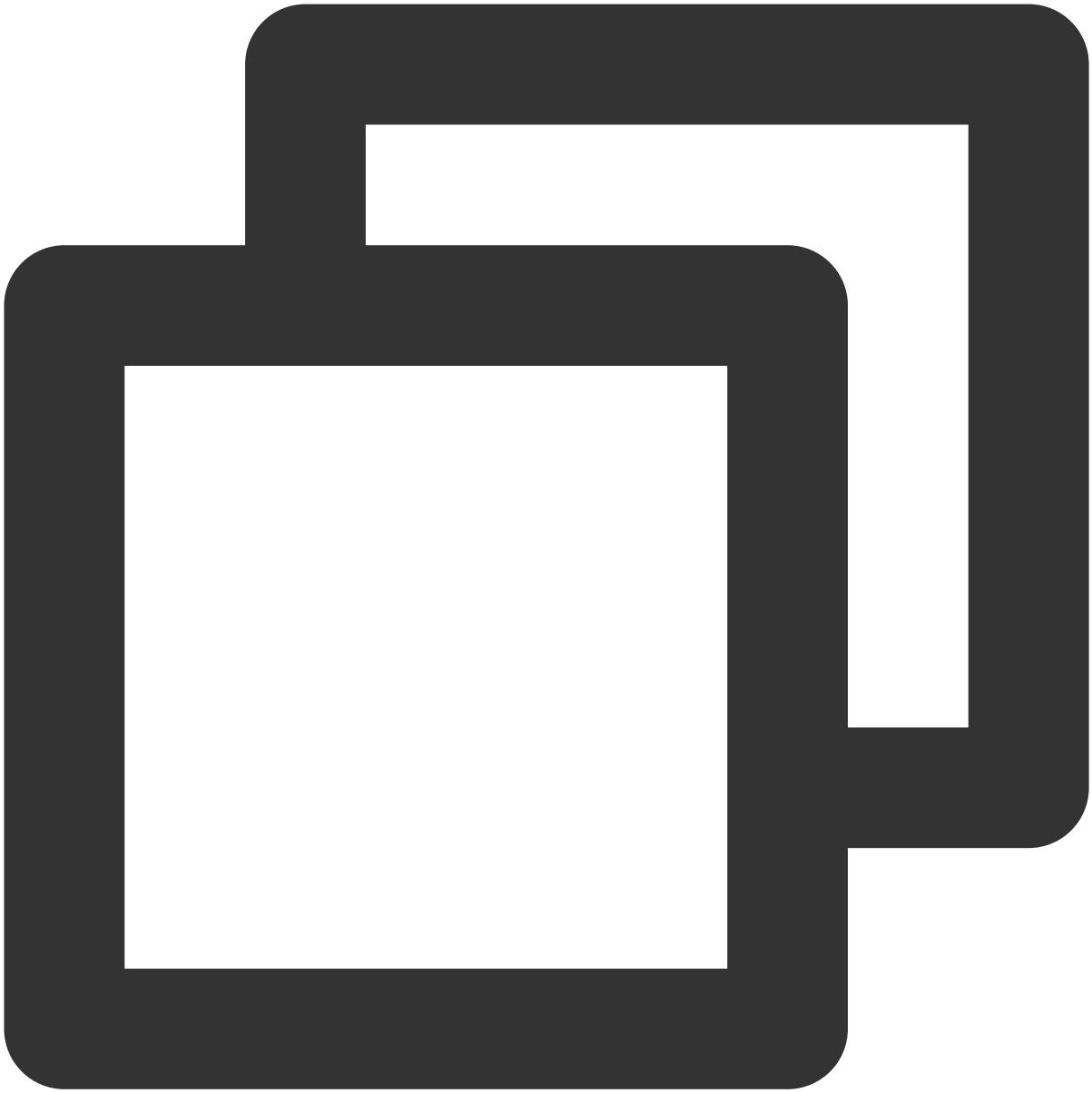
Request Parameters

Parameter	Optional	Description

grant_type	false	Fixed value: `password` .
client_id	false	The `client_id` of the application. Go to the **application management page** and **select the application**, and then click **Application Configuration** to find the **Client ID**.
client_secret	true	The `client_secret` of the application. Go to the **application management page** and **select the application**, and then click **Application Configuration** to find the **client_secret**. This parameter is required for web applications, yet it is not needed for SPA and mobile applications.
auth_source_id	false	ID of the account and password authentication source. You can view it on the General Authentication Source page on the console.
username	false	Username. The authentication source attributes that need to be configured with the account and password authentication source include the username, mobile number, and email address.
password	false	User password.
scope	true	This parameter is not required. If this parameter is used, the value should be `openid` .

Sample Success Responses

Authentication successful



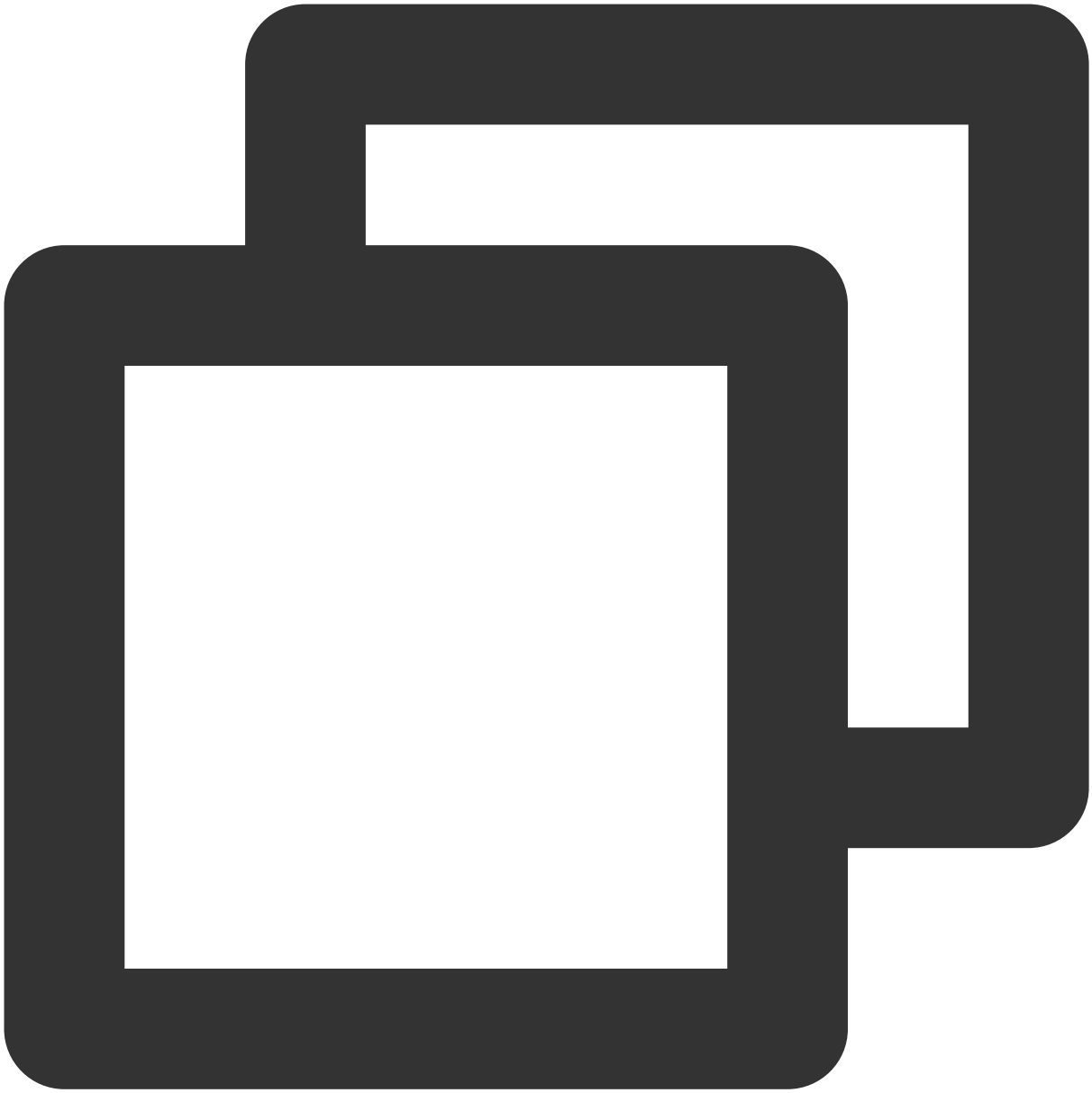
```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiI1MzQyOGU3ZS1kOTJiLTQ3OTAtOGIwMC0wMmEyZjc4NjUxNzMiLCJ
  "refresh_token" : "7uqTlthTQrzIZx8joT20chQbakZp81_iv39GTyCpsEyYpWoquNhuB3s6qEQHGe
  "scope" : "openid",
  "id_token" : "eyJraWQiOiI1MzQyOGU3ZS1kOTJiLTQ3OTAtOGIwMC0wMmEyZjc4NjUxNzMiLCJ0eXA
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

Response parameters

Parameter	Data Type	Description
access_token	String	OAuth 2.0 Access Token (JSON Web Token, or JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)
scope	String	Access Token scope.
refresh_token	String	OAuth 2.0 Refresh Token.
id_token	String	OpenID Connect (OIDC) ID Token (JWT).

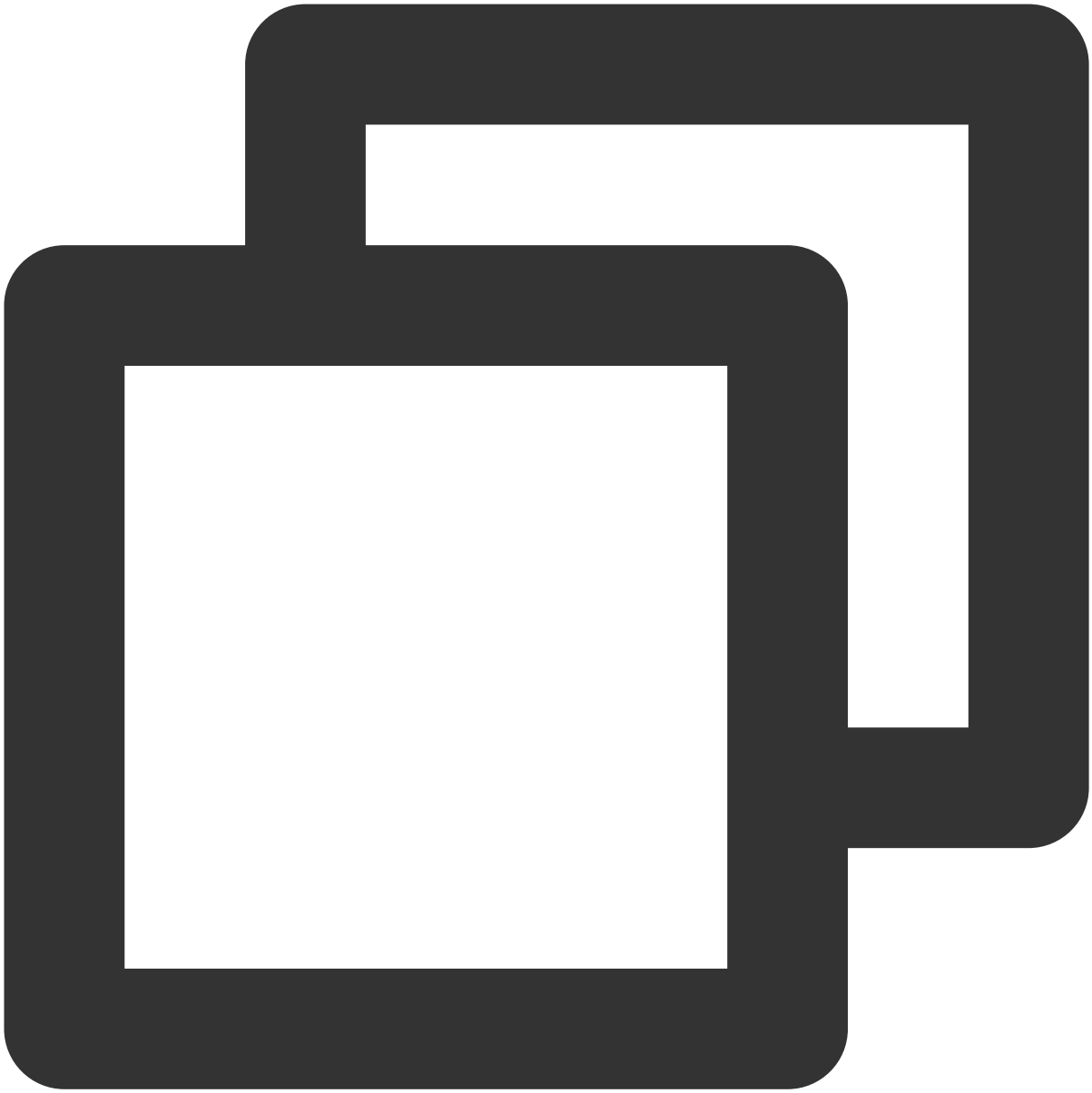
Sample Error Responses

Incorrect username or password.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Wrong username or password"
}
```

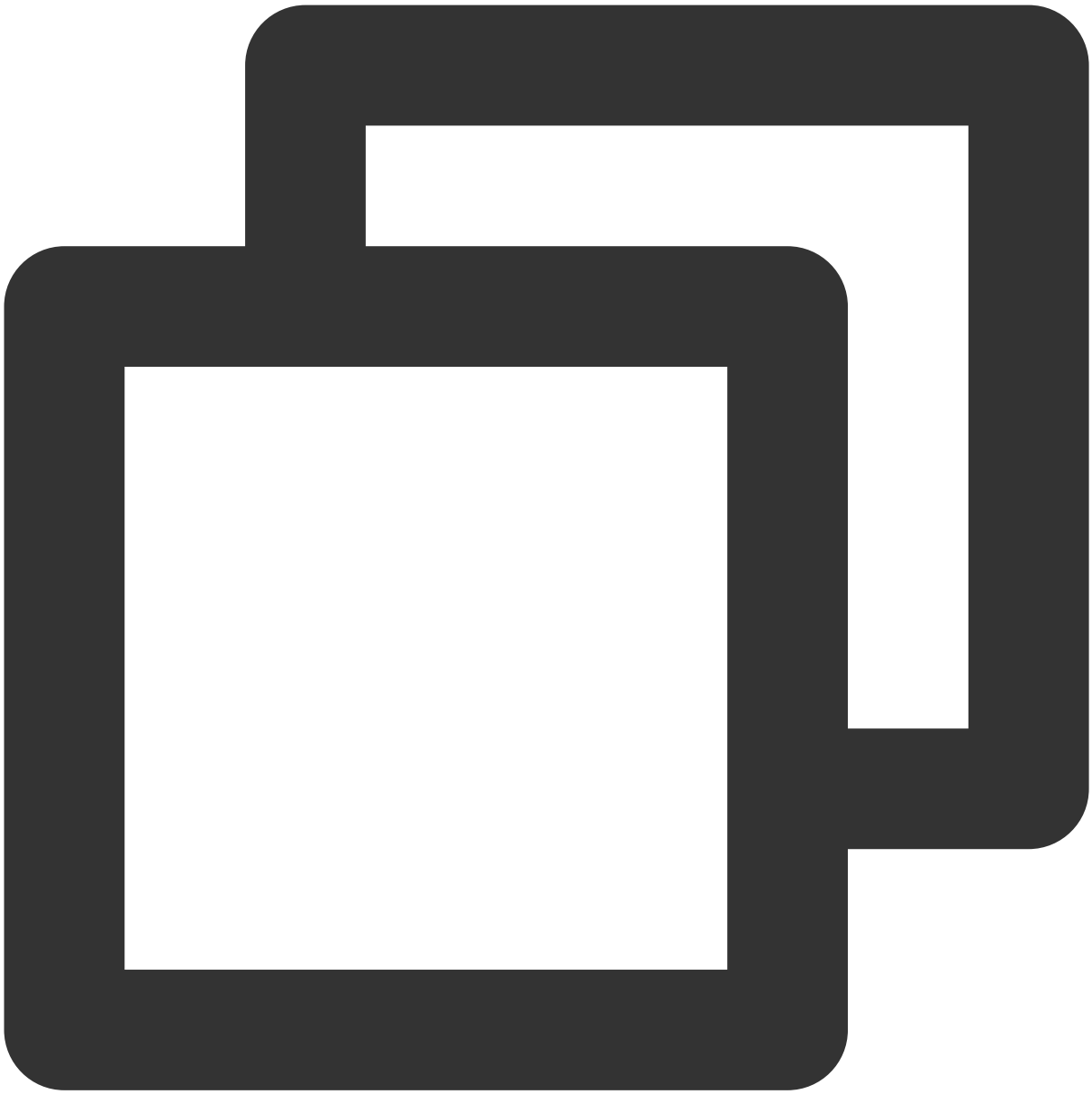
Abnormal user status. For example, the account is locked or frozen.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
}
```

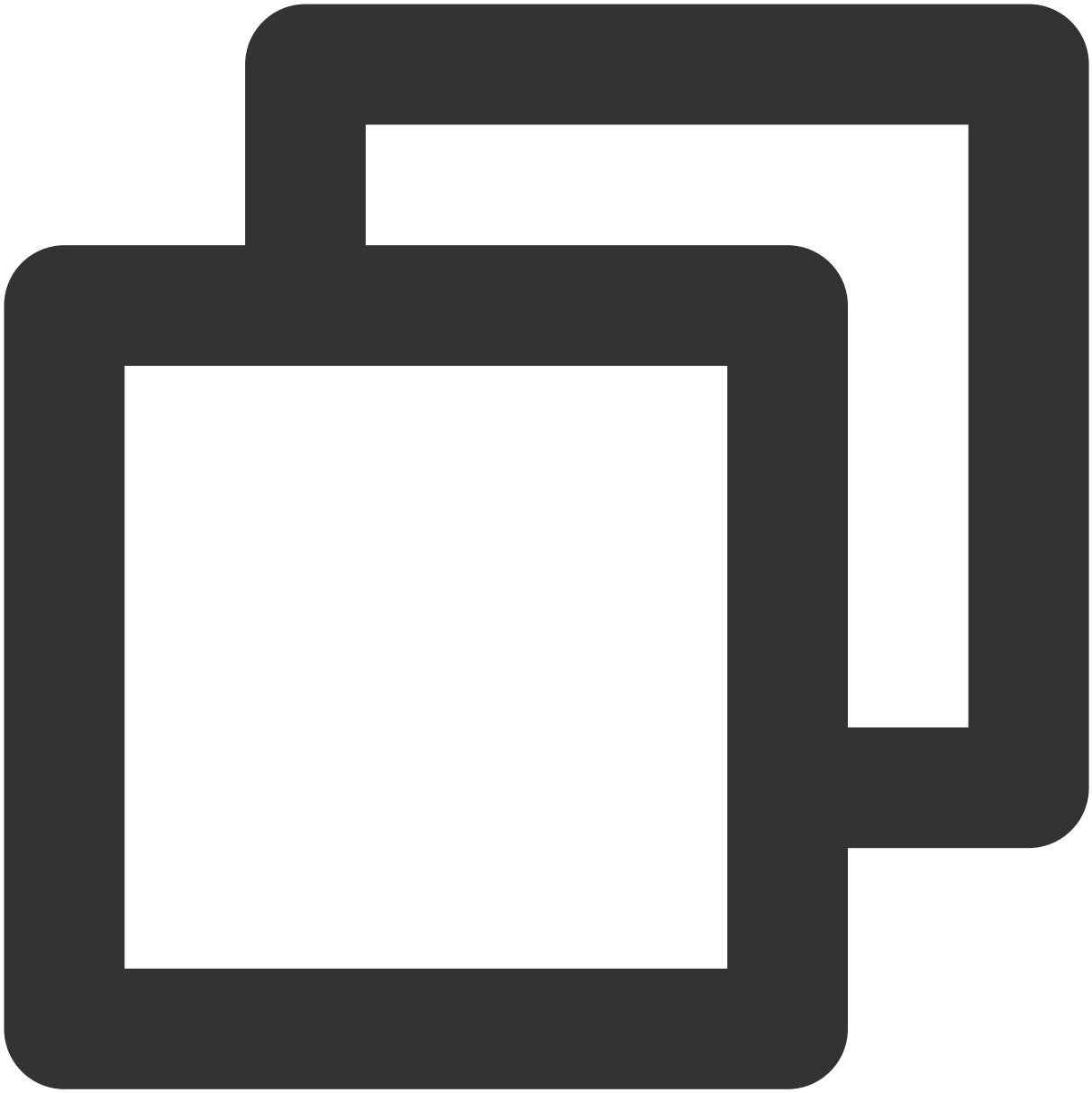
An attribute that is not supported by the authentication source is used as the username. For example, an email address is passed as the username, but the authentication source of the account and password does not configure the

email address as a authentication source attribute.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unsupported username identifier"
}
```

The authentication source is not the preferred one or is not associated with the application.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

OTP Authentication by SMS and Email

Last updated：2023-12-22 11:42:07

API Description

This API is used to verify the SMS or email one-time password (OTP) verification code, get the Access Token and ID Token for login. Before calling this API, you need to call the API for sending OTP verification code to send a verification code to user.

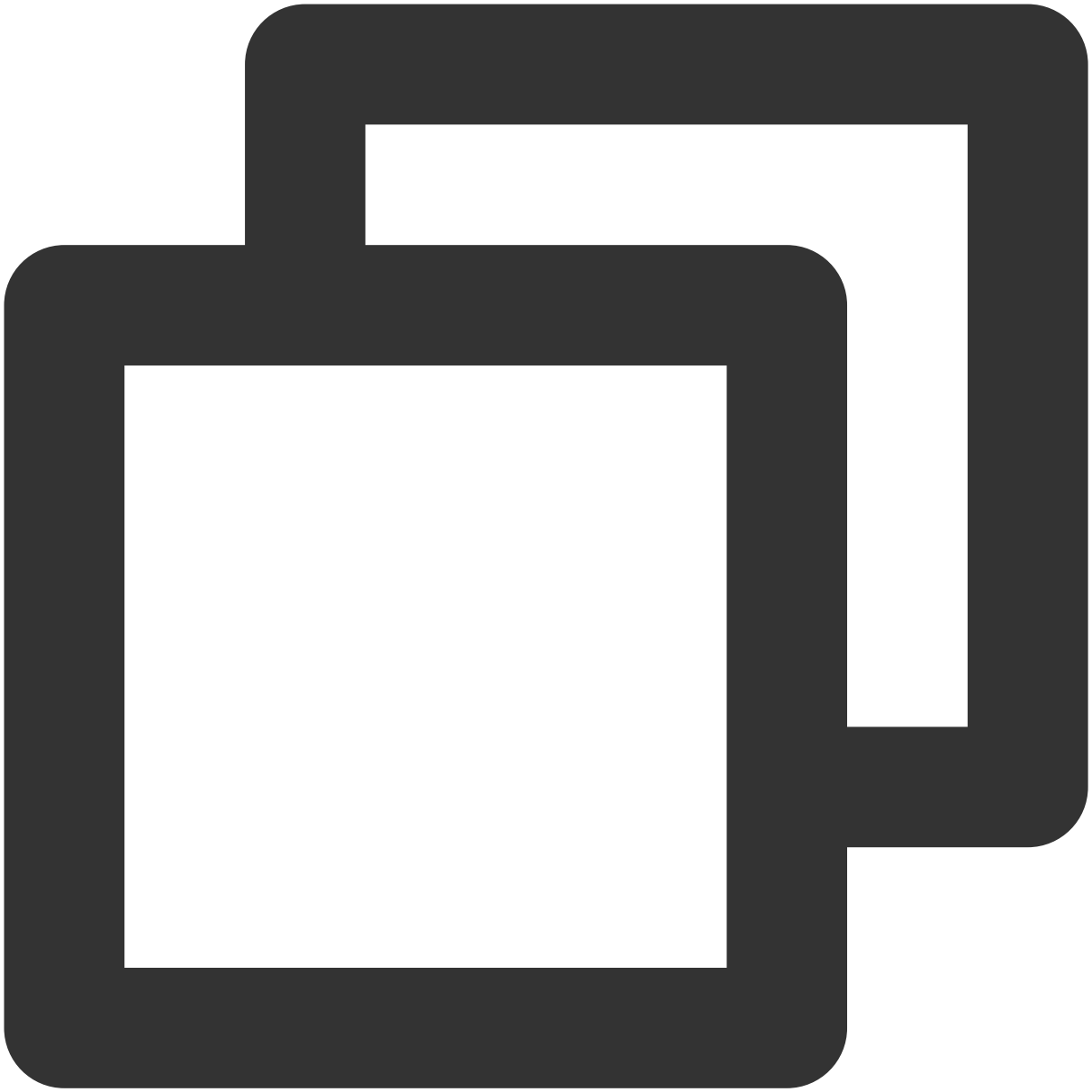
Note:

You can set `auto_signup=true` to enable automatic sign-up of users.

Supported Applications

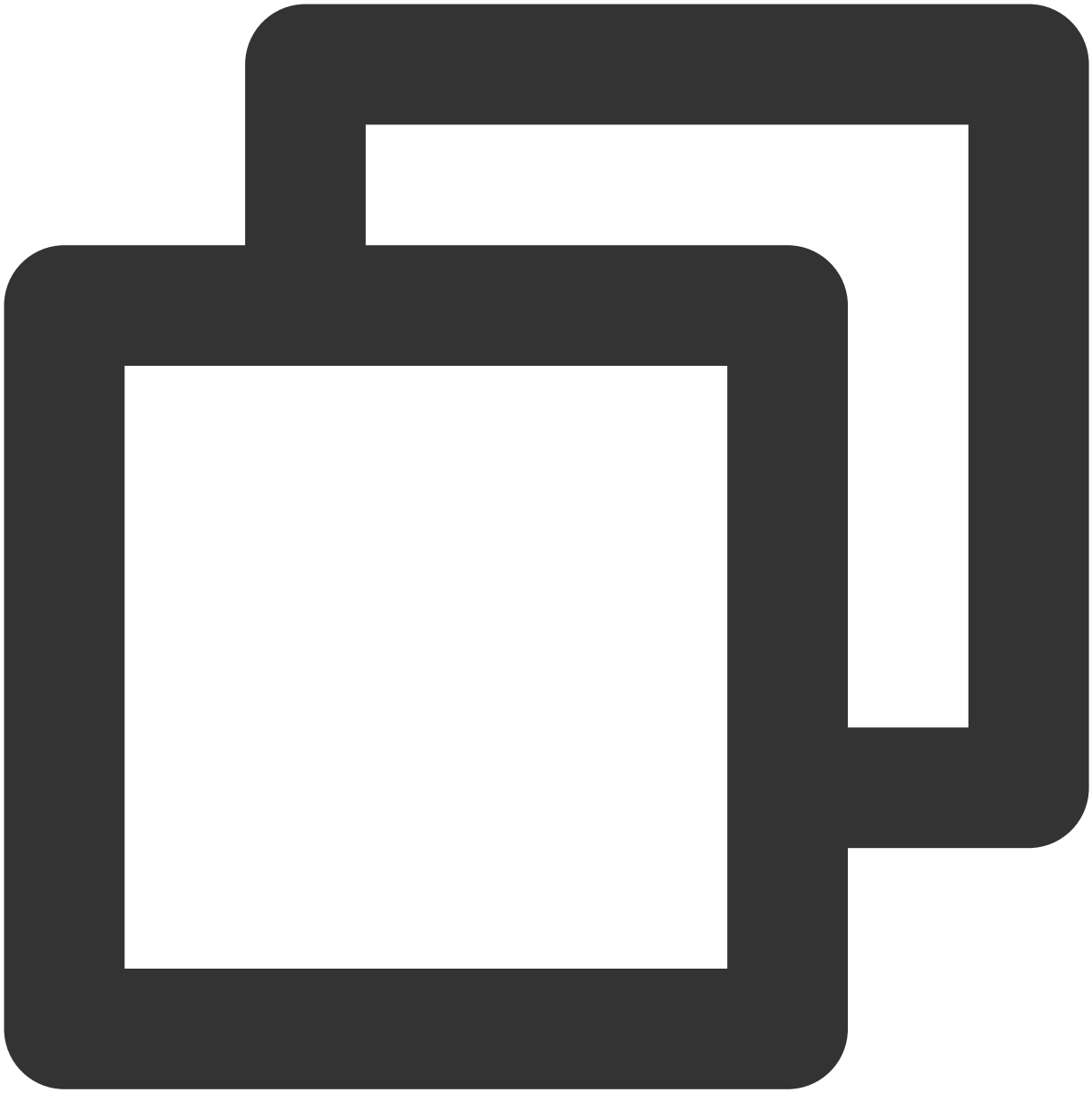
Web applications, single-page applications (SPA), and mobile applications.

Request Method



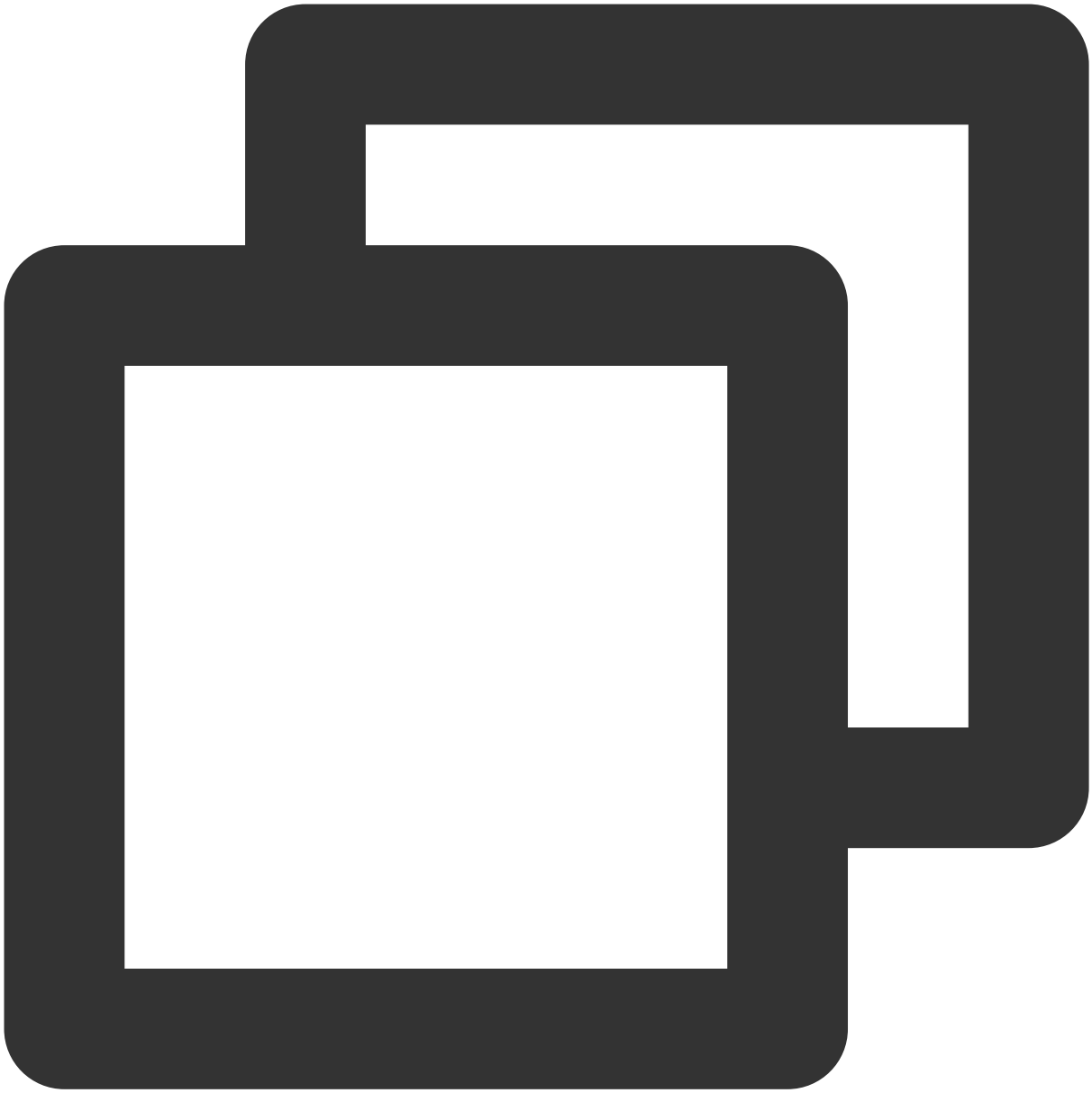
```
POST
```

Request Path



```
/oauth2/token
```

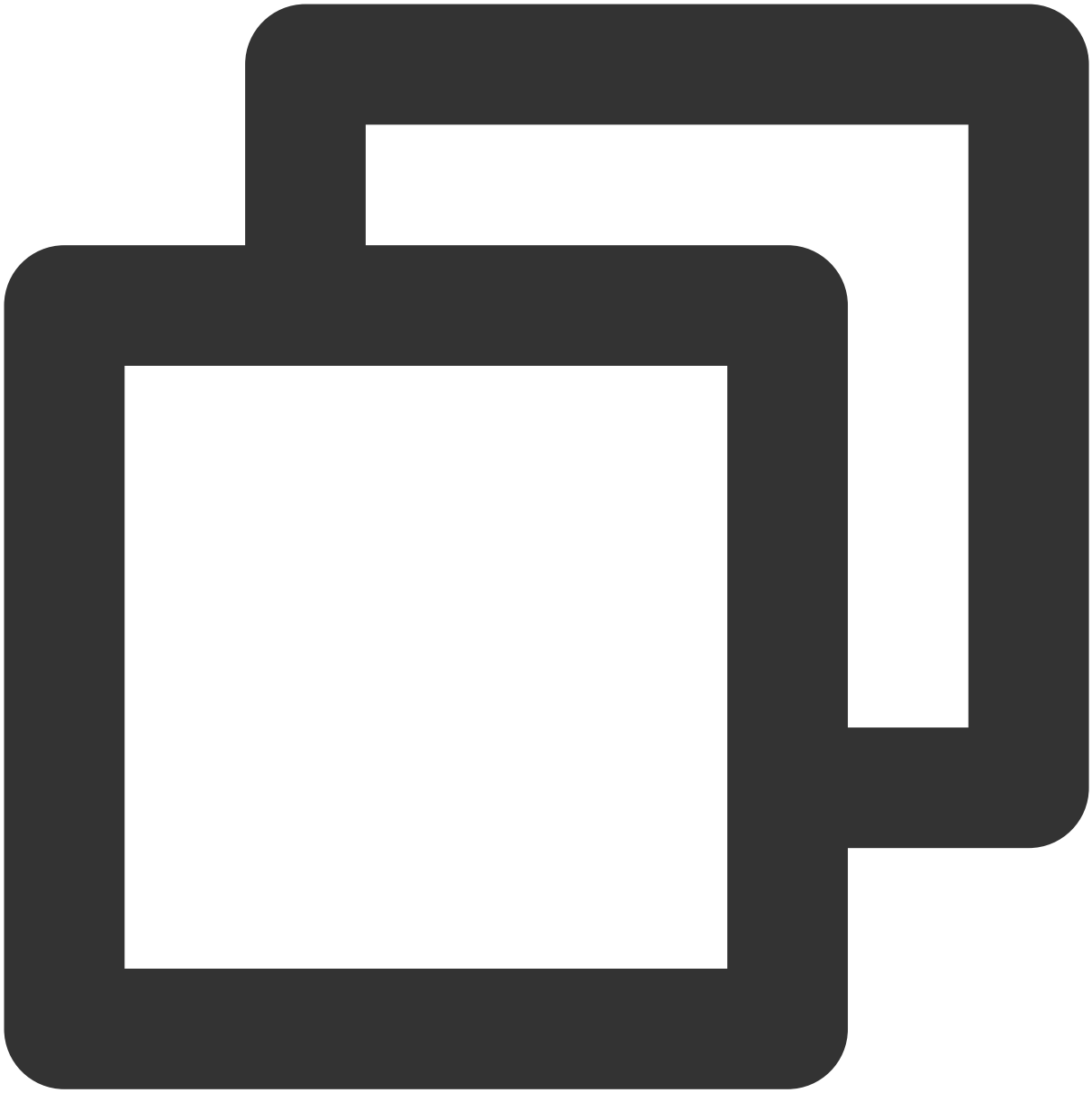
Request Content-Type



```
application/json
```

Sample Requests

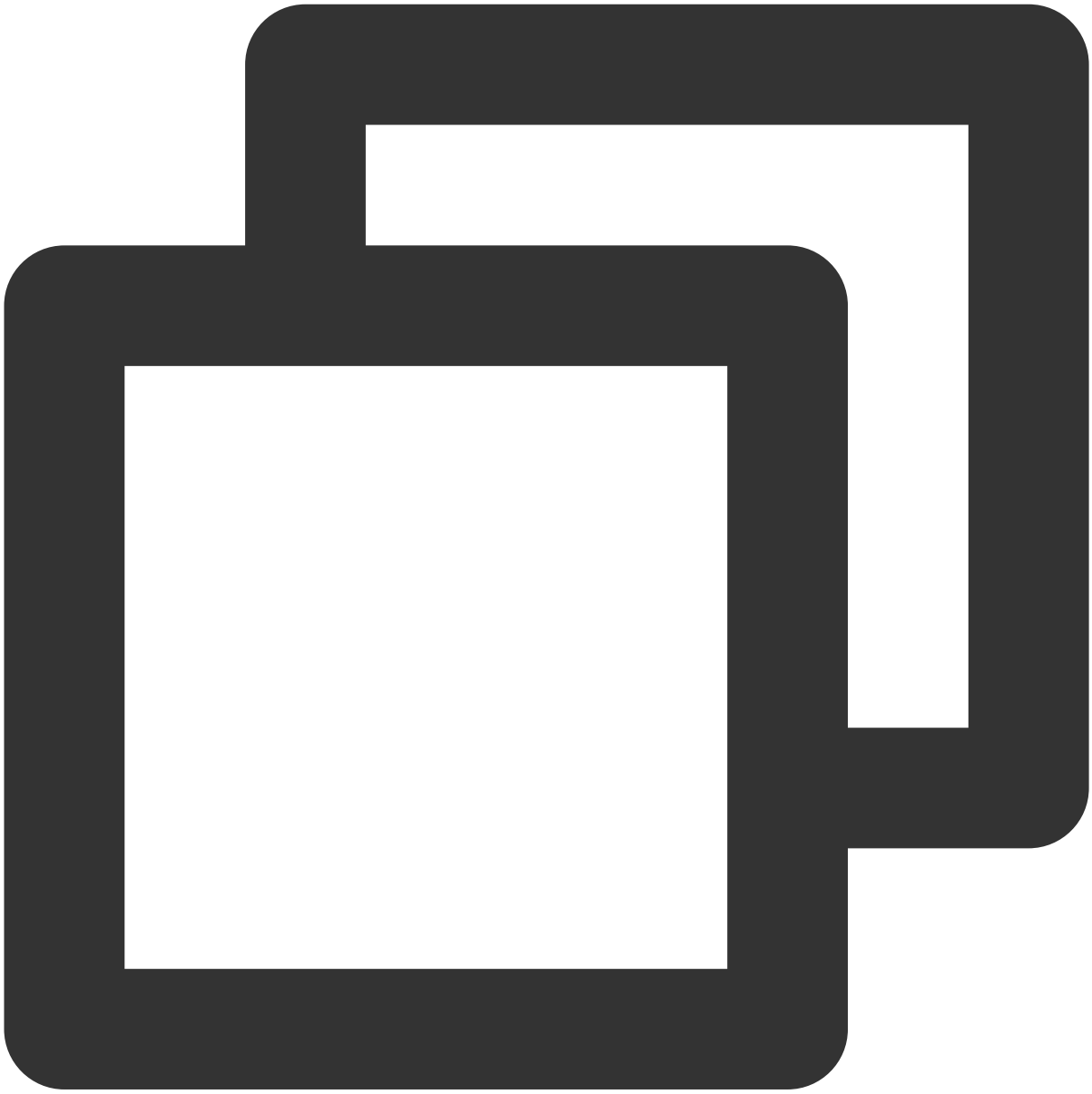
OTP login by SMS



```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
  "grant_type" : "http://tencentciam.com/oauth2/grant-type/otp/sms",
  "client_id" : "TENANT_CLIENT_ID",
  "client_secret" : "TENANT_CLIENT_SECRET",
  "auth_source_id" : "MOCK_SMS_OTP_AUTH_SOURCE_ID",
  "phone_number" : "13612345678",
  "otp_token" : "MOCK_OTP_TOKEN",
```

```
  "otp" : "123456"
}
```

OTP login by email



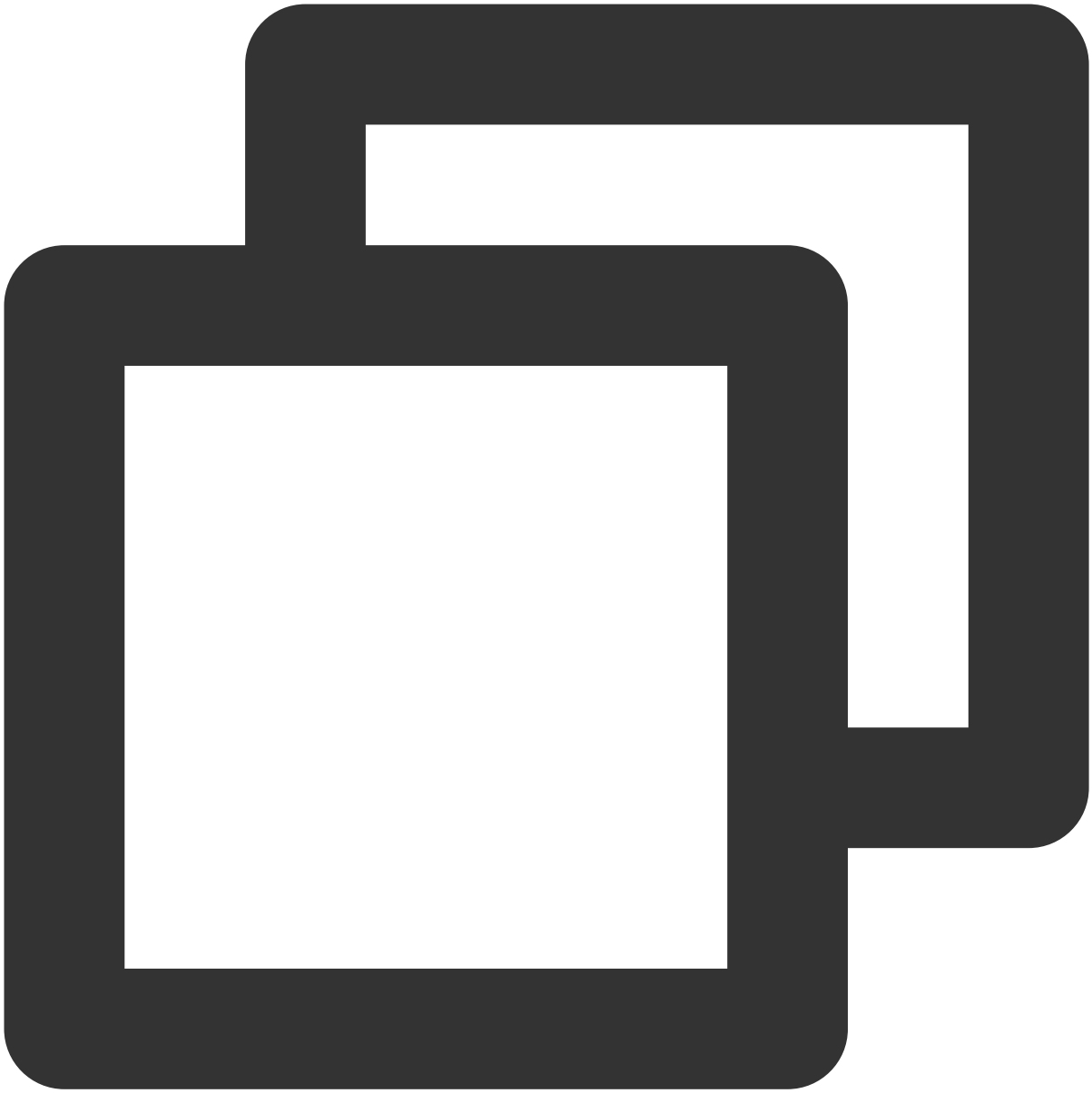
```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
  "grant_type" : "http://tencentciam.com/oauth2/grant-type/otp/email",
```

```
  "client_id" : "TENANT_CLIENT_ID",
  "client_secret" : "TENANT_CLIENT_SECRET",
  "auth_source_id" : "MOCK_EMAIL_OTP_AUTH_SOURCE_ID",
  "email" : "MOCK_USERNAME@example.com",
  "otp_token" : "MOCK_EMAIL_OTP_TOKEN",
  "otp" : "123456"
}
```

Request Parameters in JSON Format

JSON Path	Data Type	Description
grant_type	String	OTP login by SMS: `http://tencentciam.com/oauth2/grant-type/otp/sms` OTP login by email: `http://tencentciam.com/oauth2/grant-type/otp/email`
client_id	String	The `client_id` of the application. This should be the same as that used for sending verification code.
client_secret	String	The `client_secret` of the application. This parameter is required for web applications, yet it is not needed for SPA and mobile applications.
auth_source_id	String	The ID of the authentication source for OTP by SMS or email. This should be the same as that used for sending verification code.
phone_number	String	The user's mobile number. This should be the same as that used for sending verification code. This parameter is required for OTP login by SMS.
email	String	The user's email address. This should be the same as that used for sending verification code. This parameter is required for OTP login by email.
otp_token	String	The `otp_token` returned by the server after the verification code is sent.
otp	String	The OTP verification code received by the user's mobile number or email.
auto_signup	Boolean	To enable automatic sign-up of users, pass "true". Otherwise, this parameter is not required.

Sample Success Responses



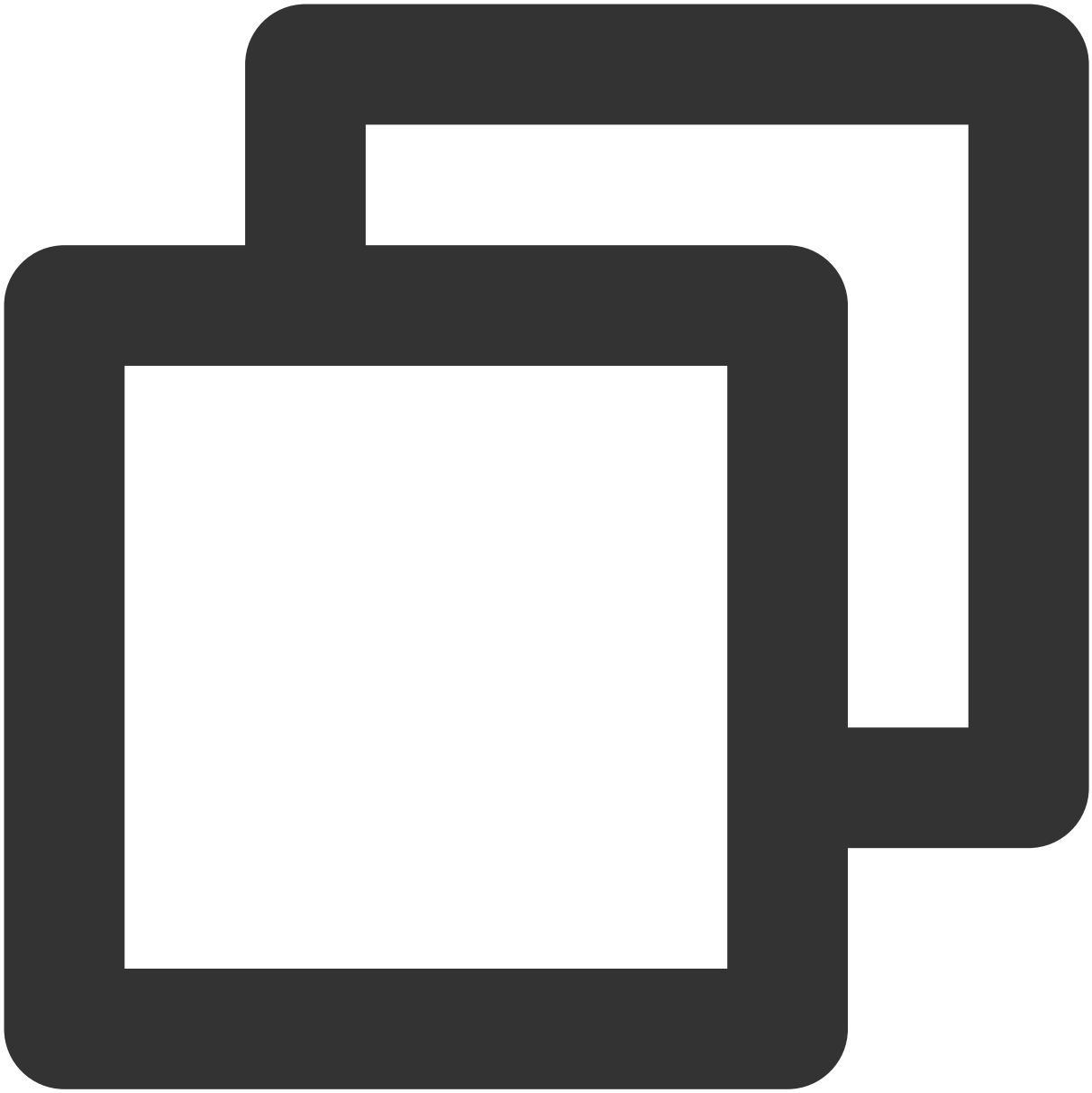
```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiJmZTQ4YTJjYS1lNGU3LTQyMGEtOThjOS01OGM5NmI2NzUwZjIiLCJ
  "refresh_token" : "B-72VlkQa3jQNuo9Xbbl-muoh4w7nYu-7Q3Wb-qmPgyftN1CgXPov2aWsOBWee
  "scope" : "openid",
  "id_token" : "eyJraWQiOiJmZTQ4YTJjYS1lNGU3LTQyMGEtOThjOS01OGM5NmI2NzUwZjIiLCJ0eXA
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

Response Parameters

Parameter	Data Type	Description
access_token	String	OAuth 2.0 Access Token (JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)
scope	String	Access Token scope.
refresh_token	String	OAuth 2.0 Refresh Token.
id_token	String	OpenID Connect (OIDC) ID Token (JSON Web Token, or JWT).

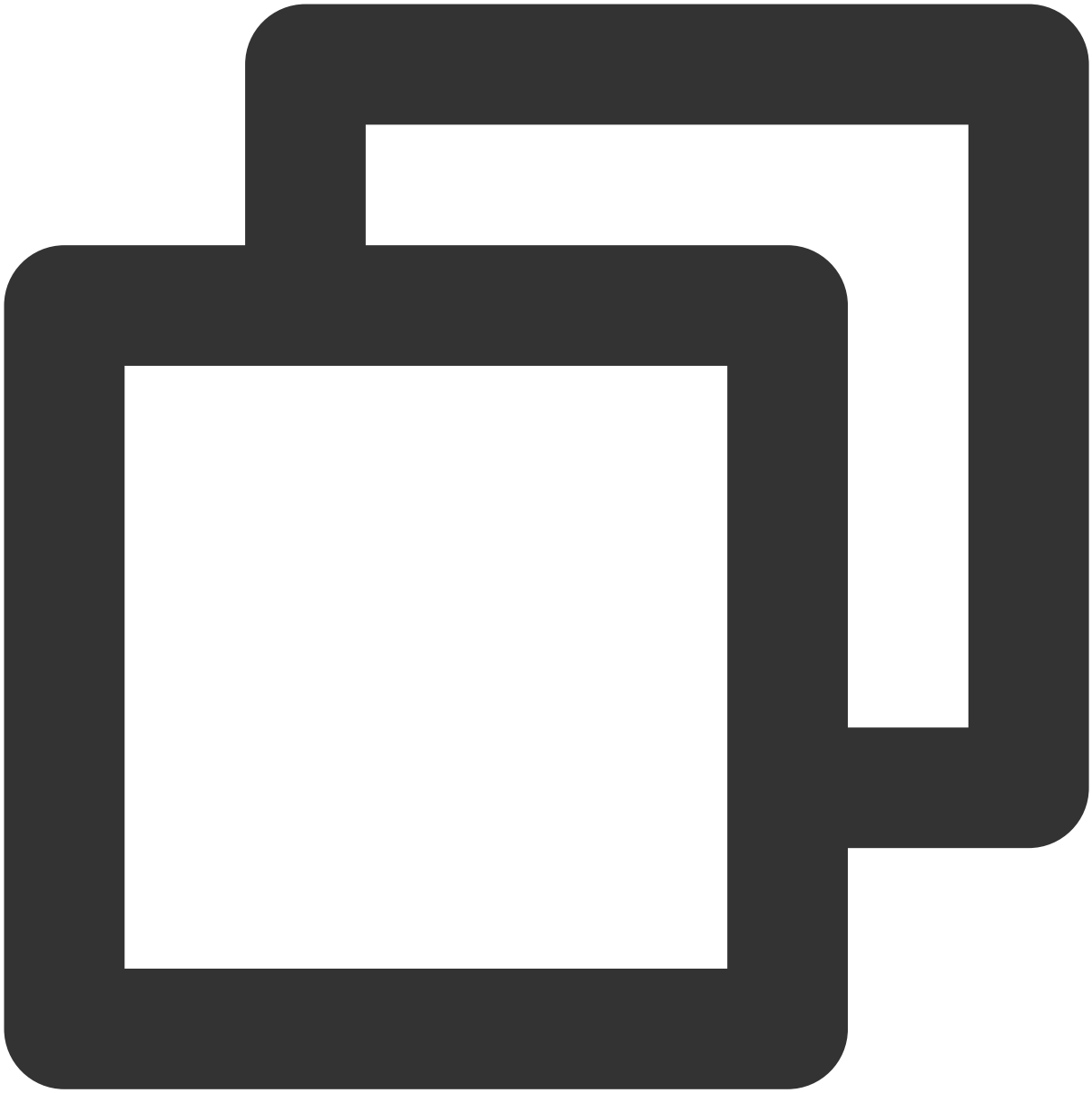
Sample Error Responses

`otp_token` is incorrect or has expired.



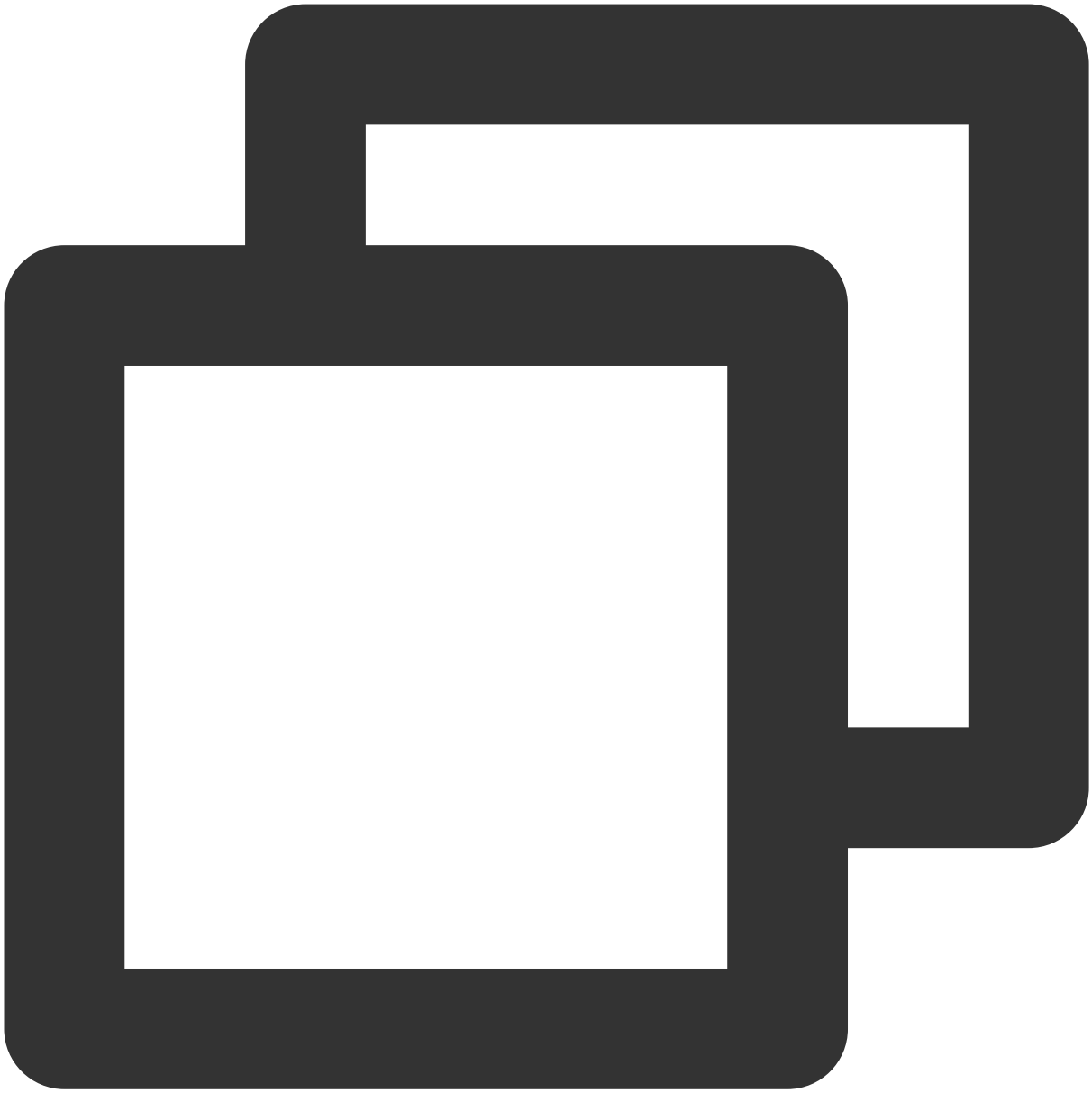
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unknown or expired otp_token"
}
```

`otp` is incorrect or has expired.



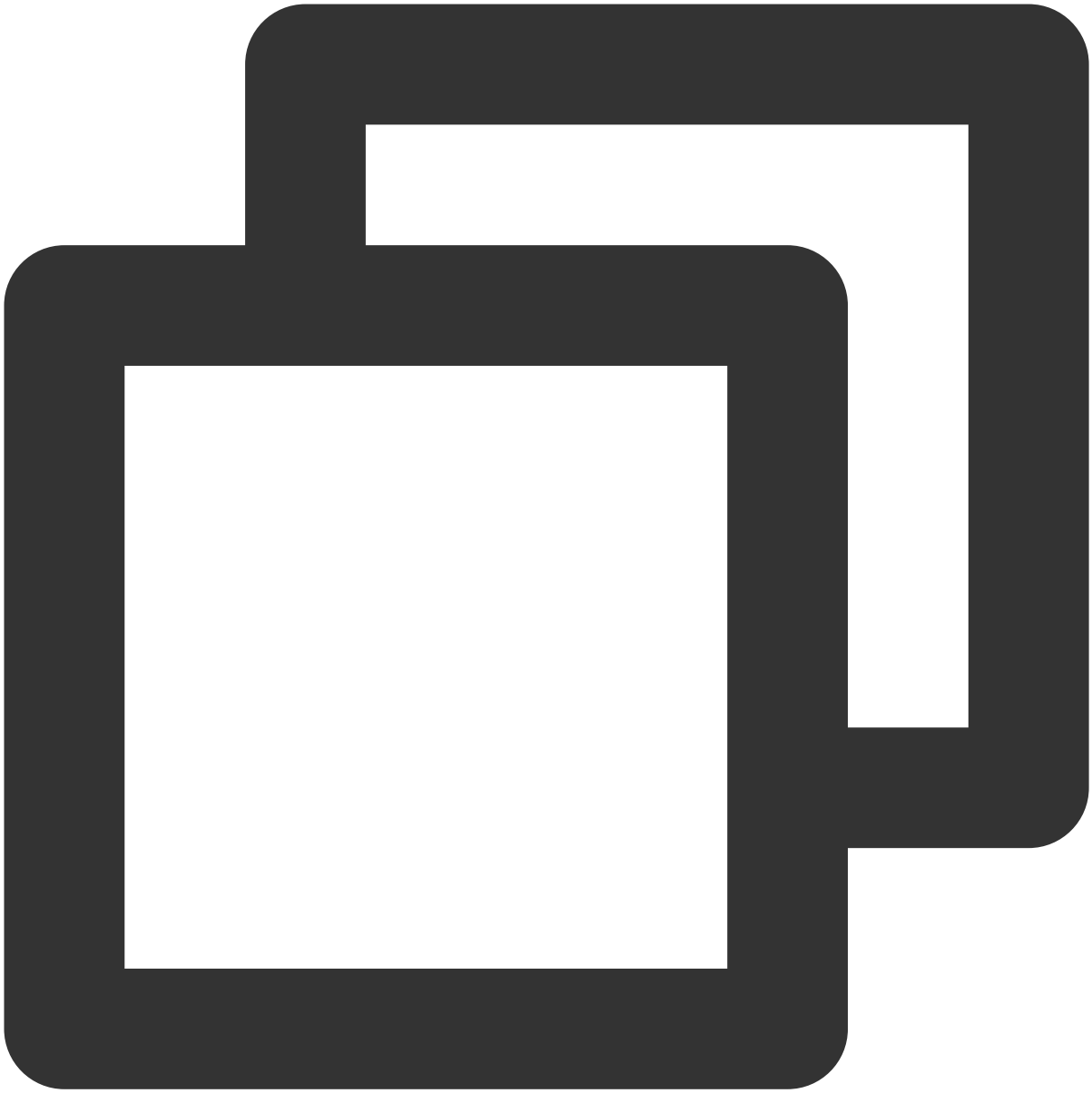
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unknown or expired OTP"
}
```

The parameter used is not the same as the one for sending the verification code. For example, the mobile numbers are different.



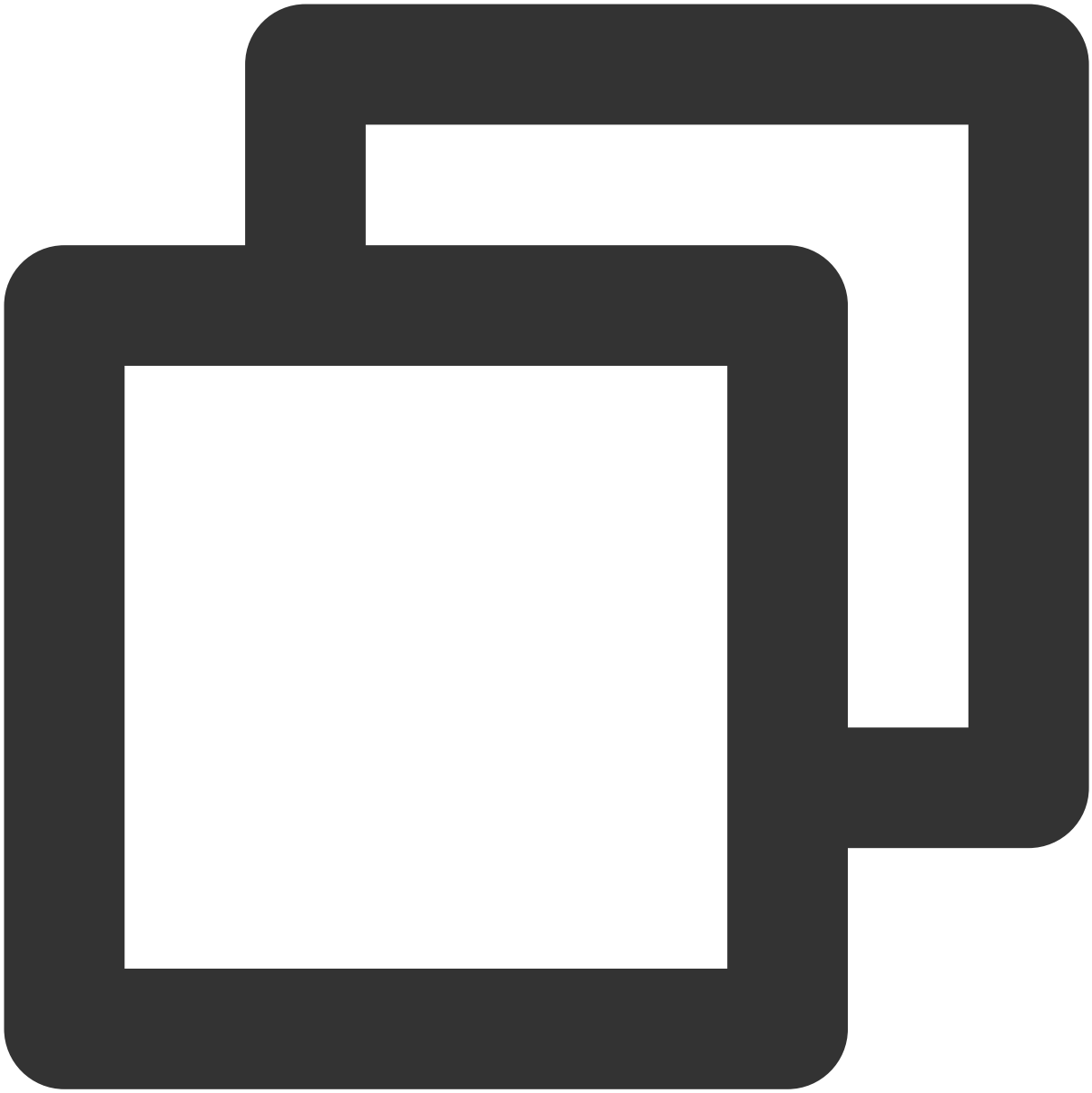
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Mismatched OTP token and OTP sending parameters"
}
```

The user corresponding to the mobile number or email address cannot be found. This occurs when automatic sign-up of users is not allowed.



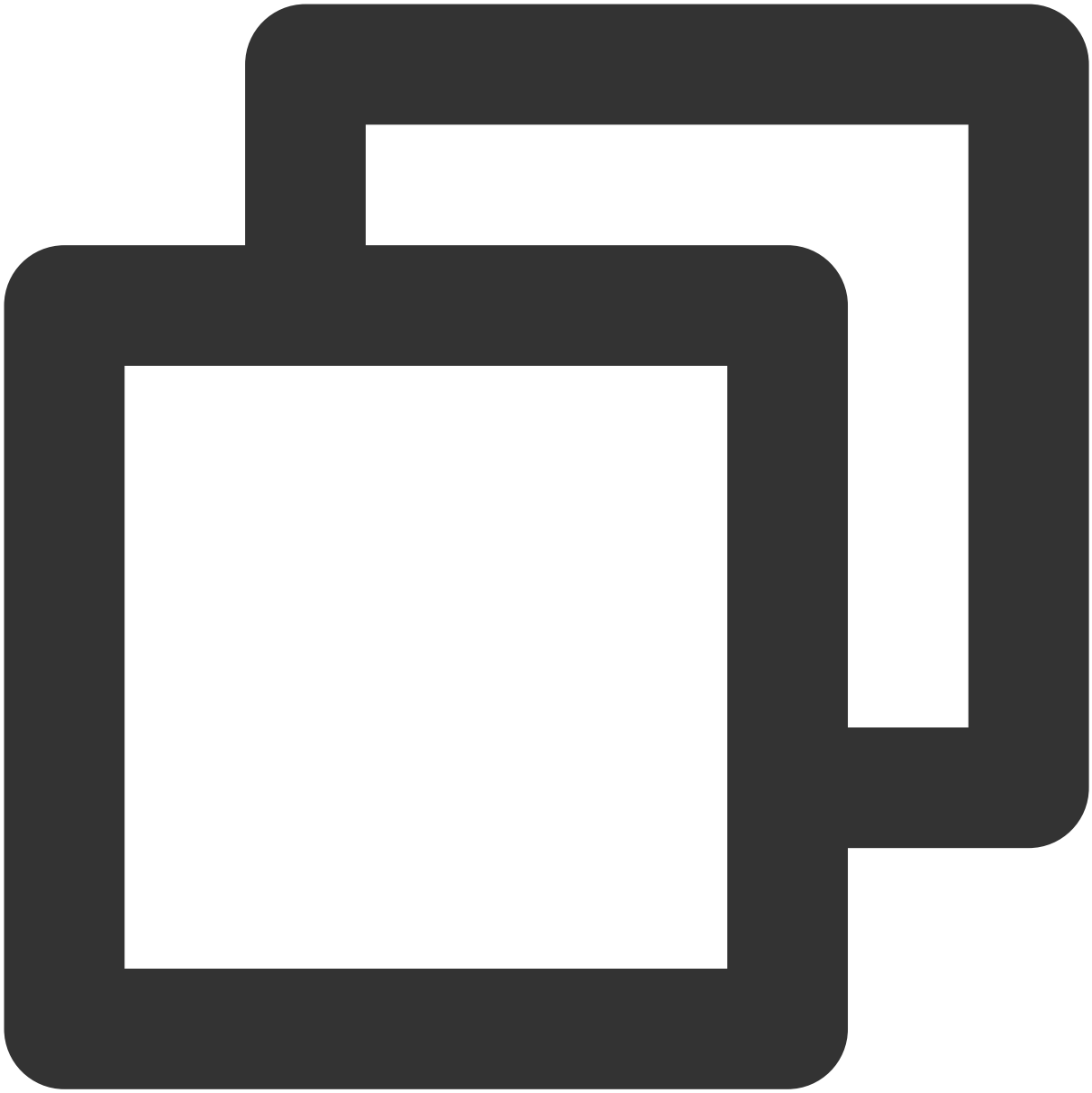
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "User not found"
}
```

The status of the user corresponding to the mobile number or email address is abnormal. For example, the account is locked or frozen.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
}
```

The authentication source is not the preferred one or is not associated with the application.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

Send OTP Verification Code

Last updated：2023-12-22 11:42:07

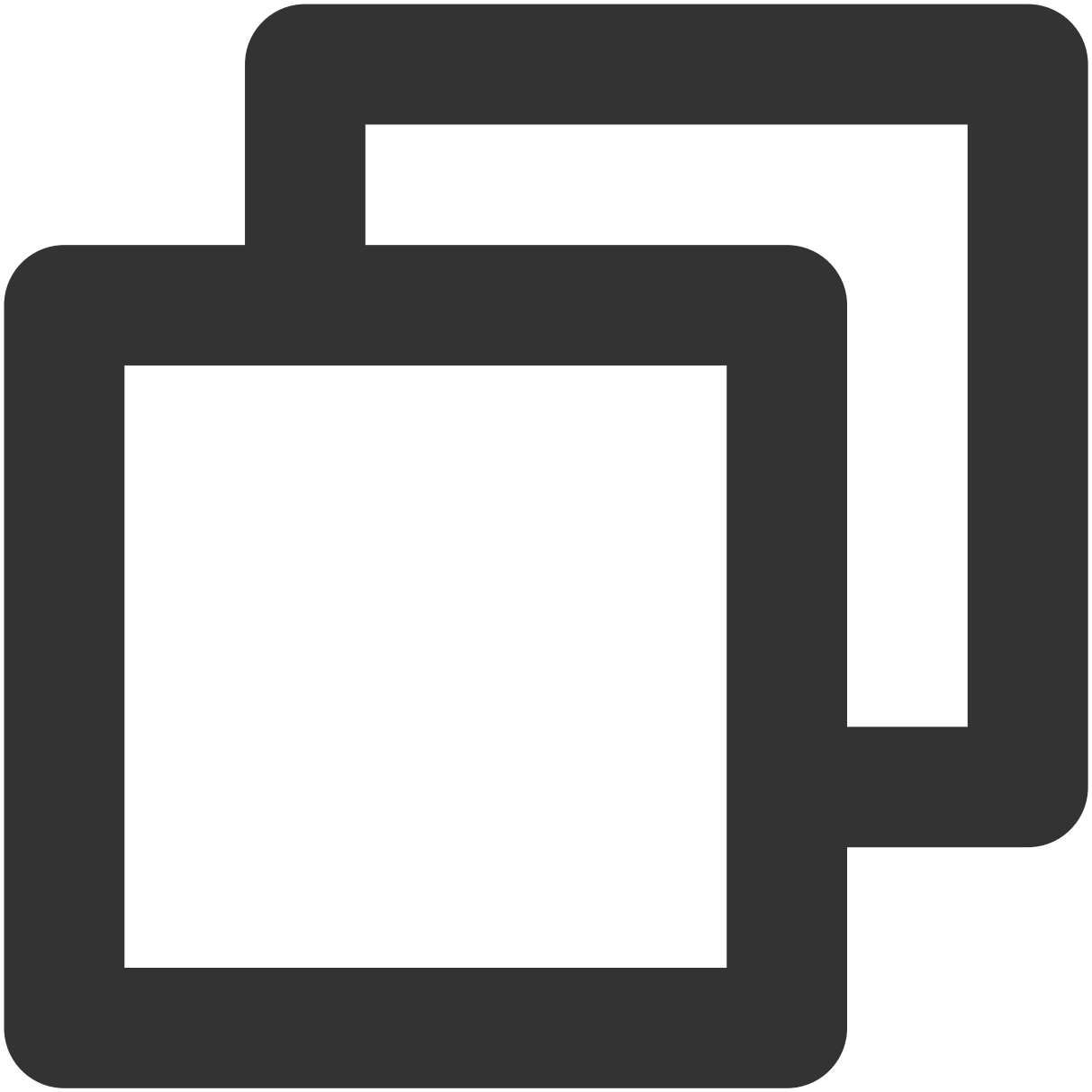
API Description

This API is used to send SMS or email one-time password (OTP) verification code to users for login, sign-up or user information update.

Supported Applications

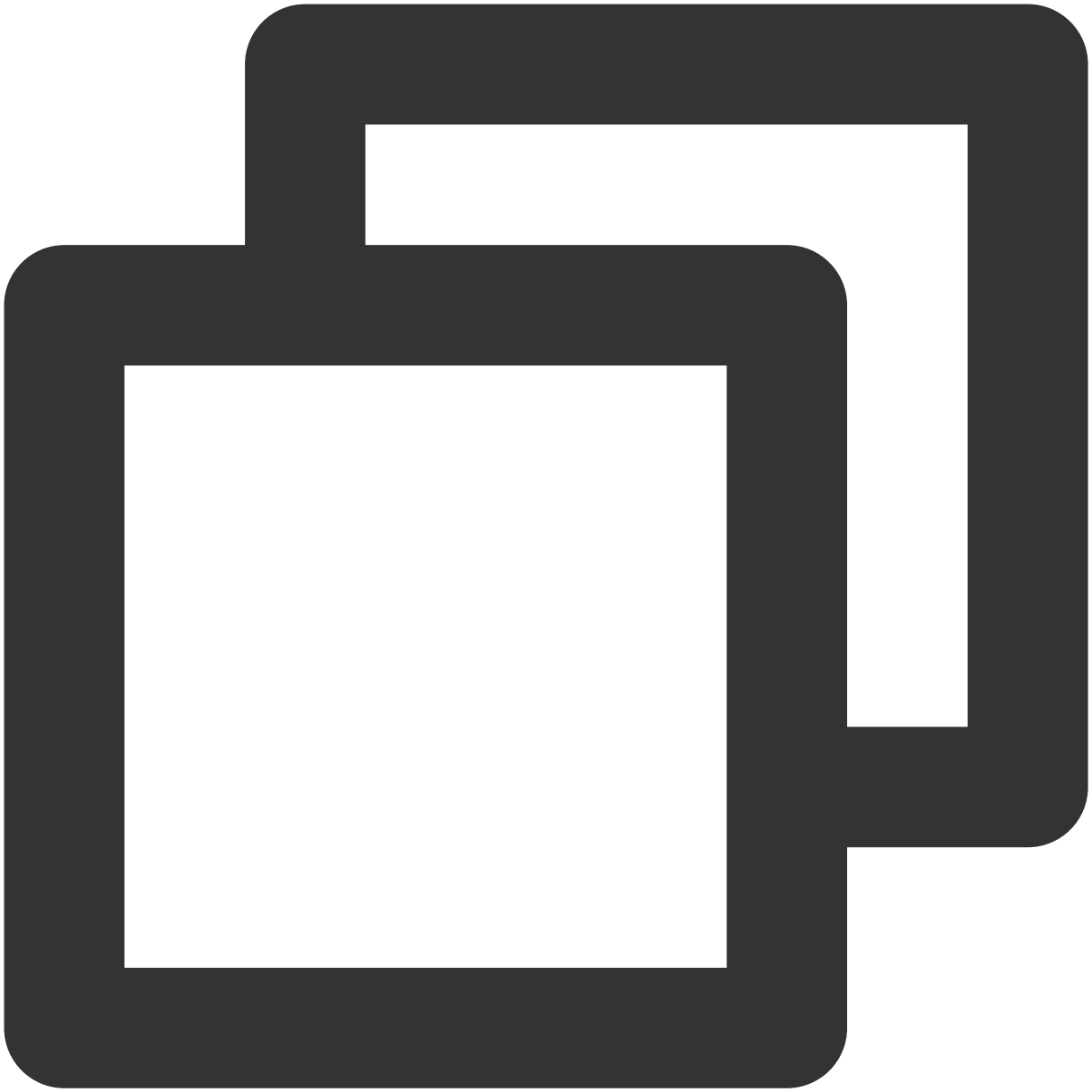
Web applications and machine-to-machine (M2M) applications.

Request Method



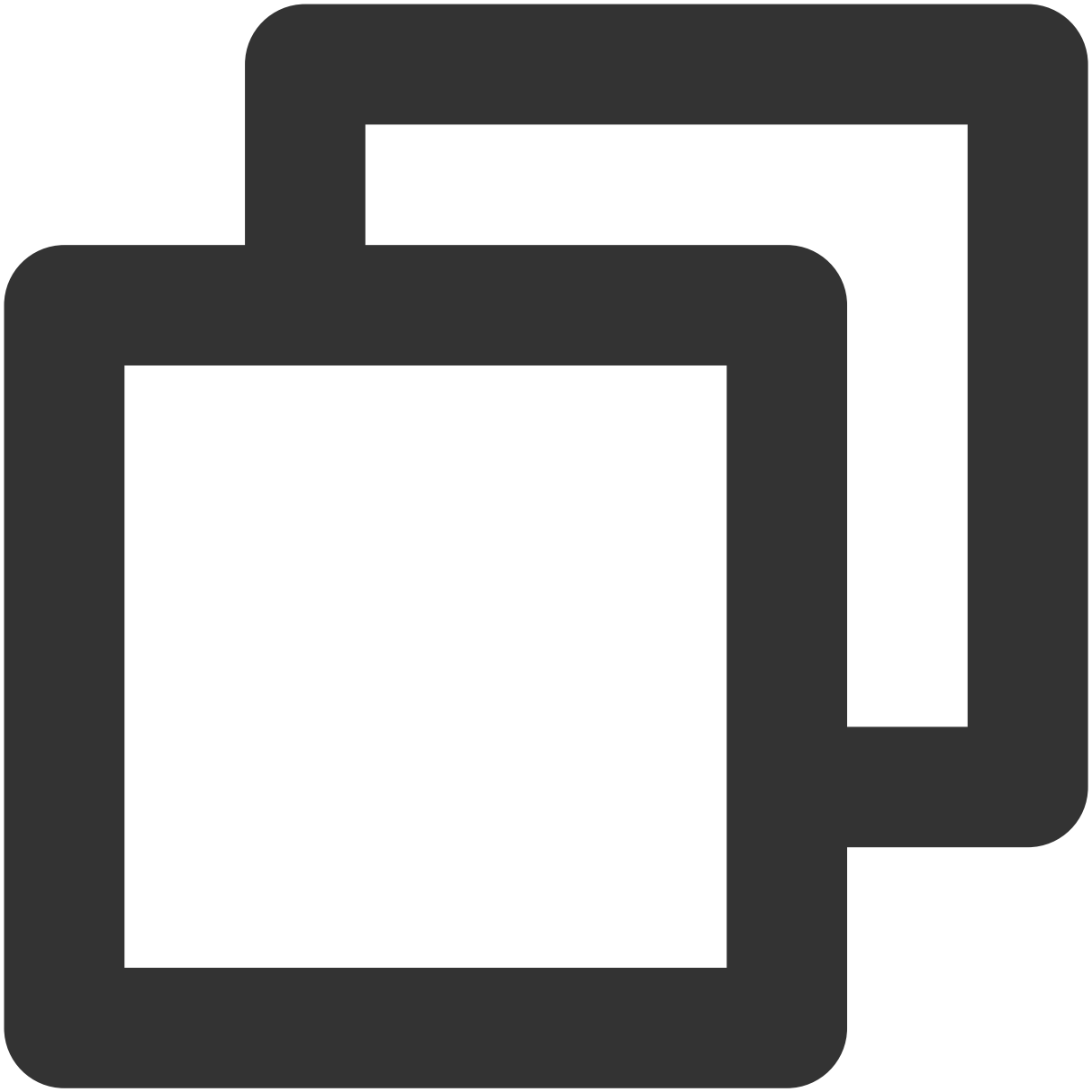
```
POST
```

Request Path



```
/otp/send
```

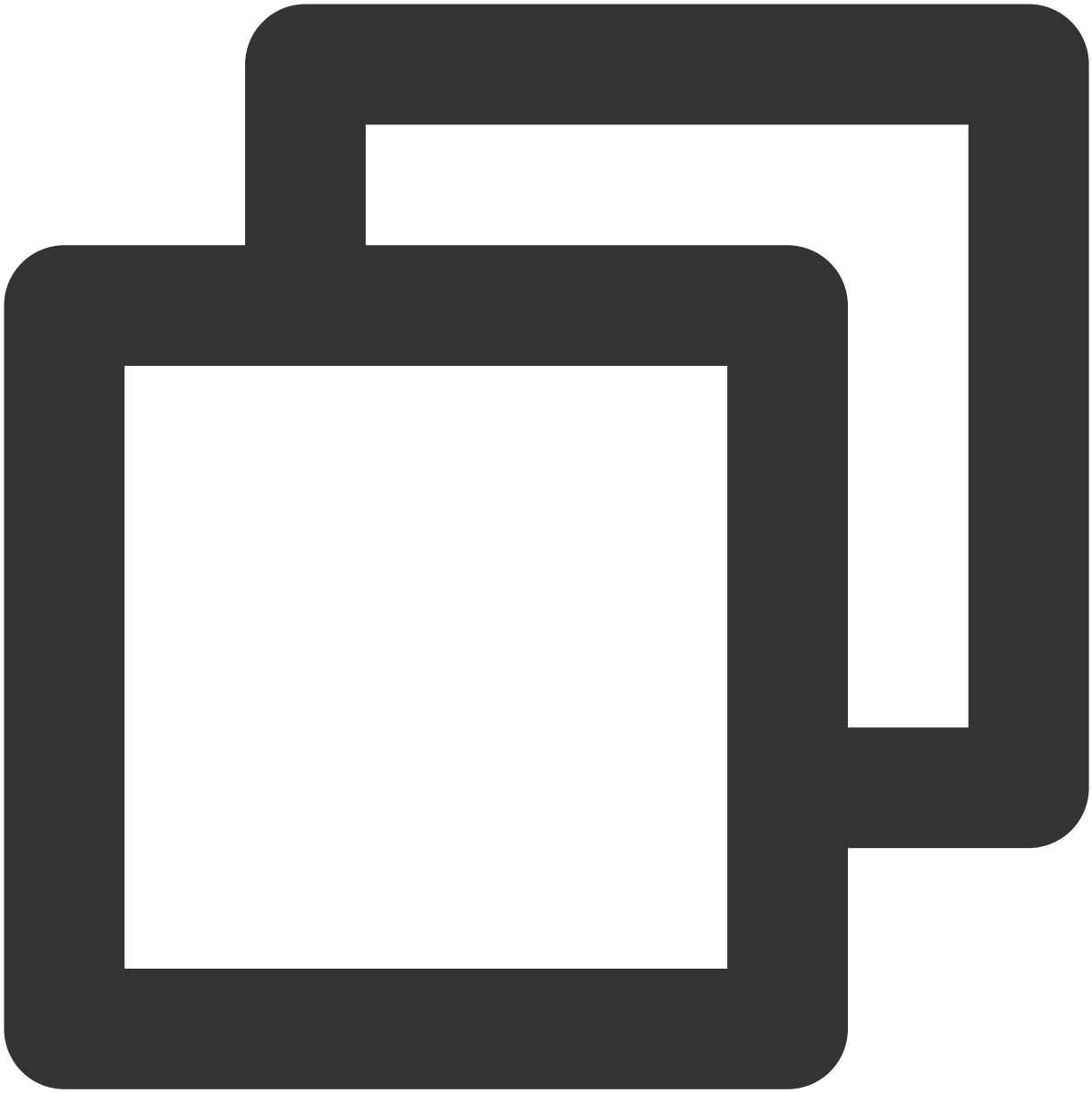
Request Content-Type



```
application/json
```

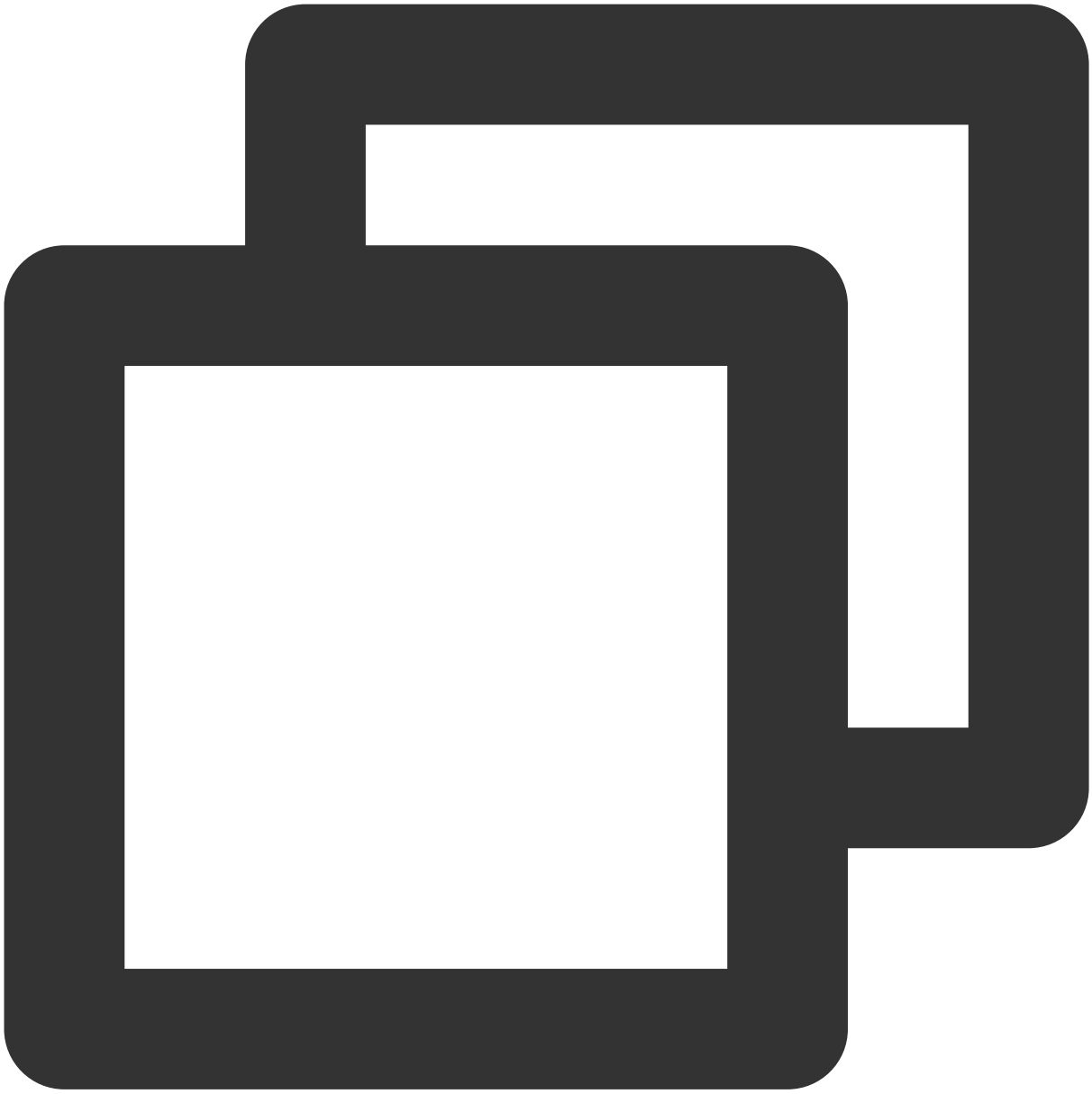
Sample Requests

Send SMS verification code for OTP login by SMS.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"usage" : "login",
"phone_number" : "13612345678",
"auth_source_id" : "MOCK_SMS_OTP_AUTH_SOURCE_ID"
}
```

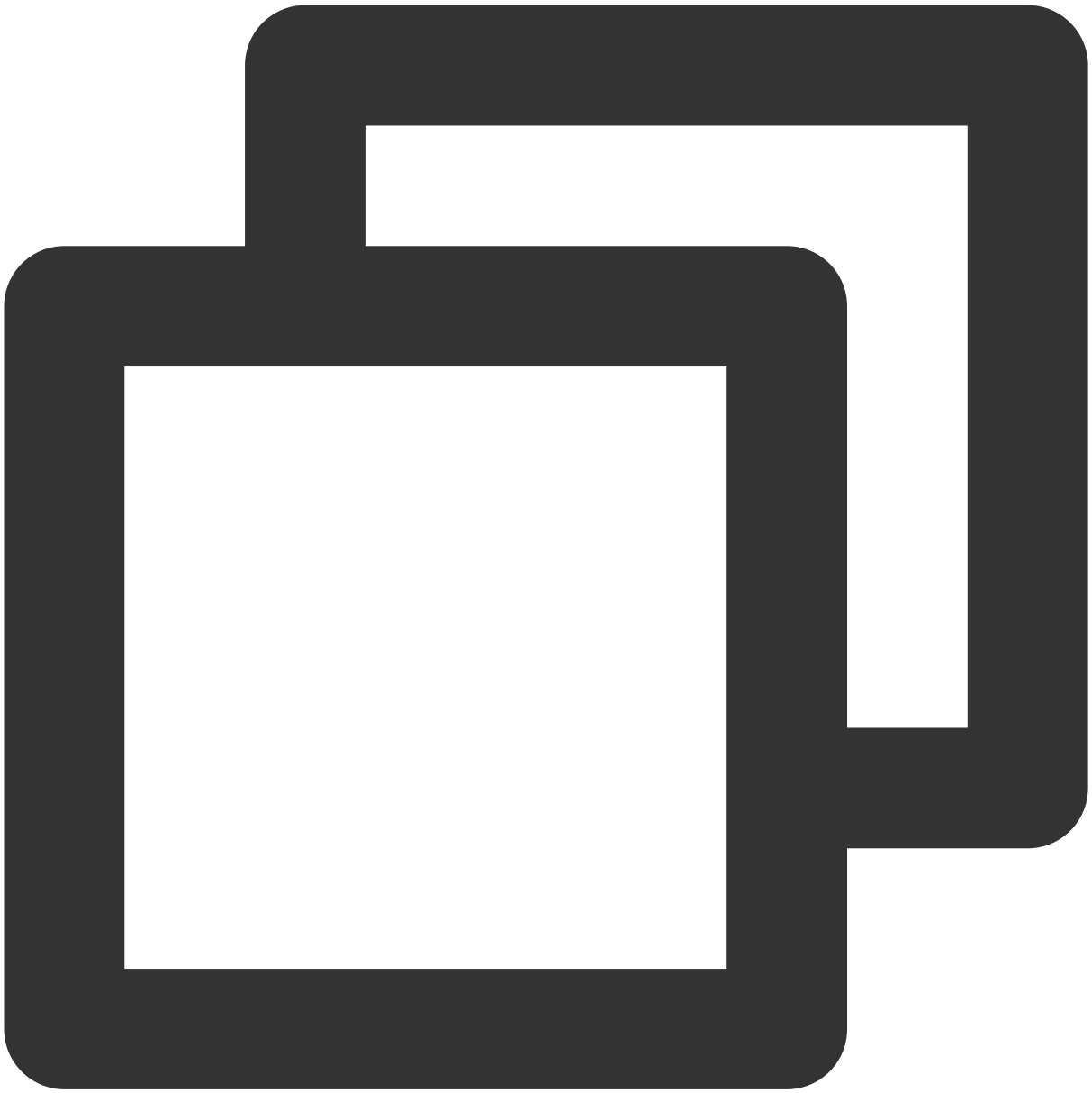
Send email verification code for OTP login by email.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "login",
"email" : "MOCK_USERNAME@example.com",
"auth_source_id" : "MOCK_EMAIL_OTP_AUTH_SOURCE_ID"
```

```
}
```

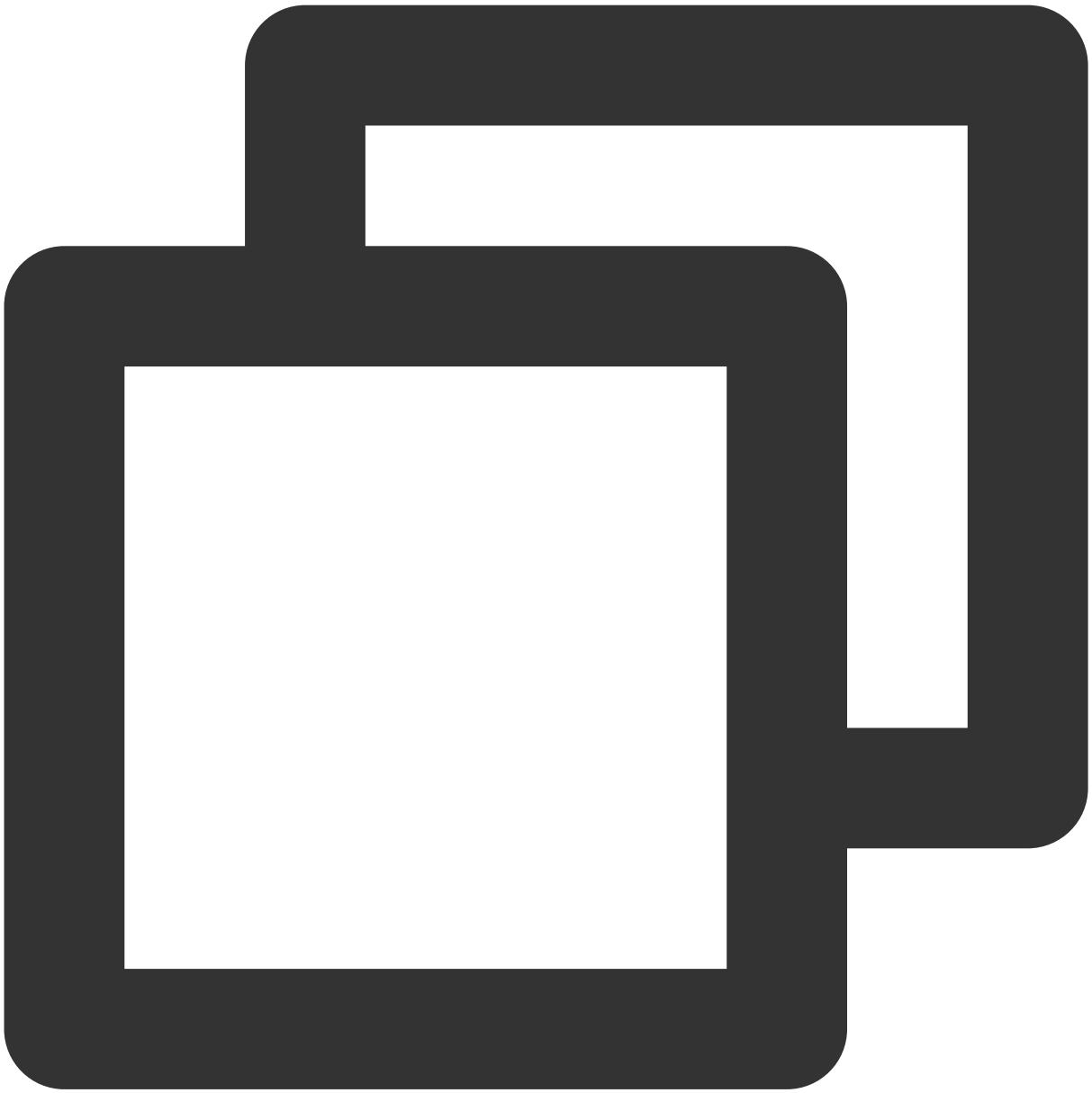
Send SMS verification code for binding mobile number during sign-up.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "signup",
"phone_number" : "13612345678"
```

```
}
```

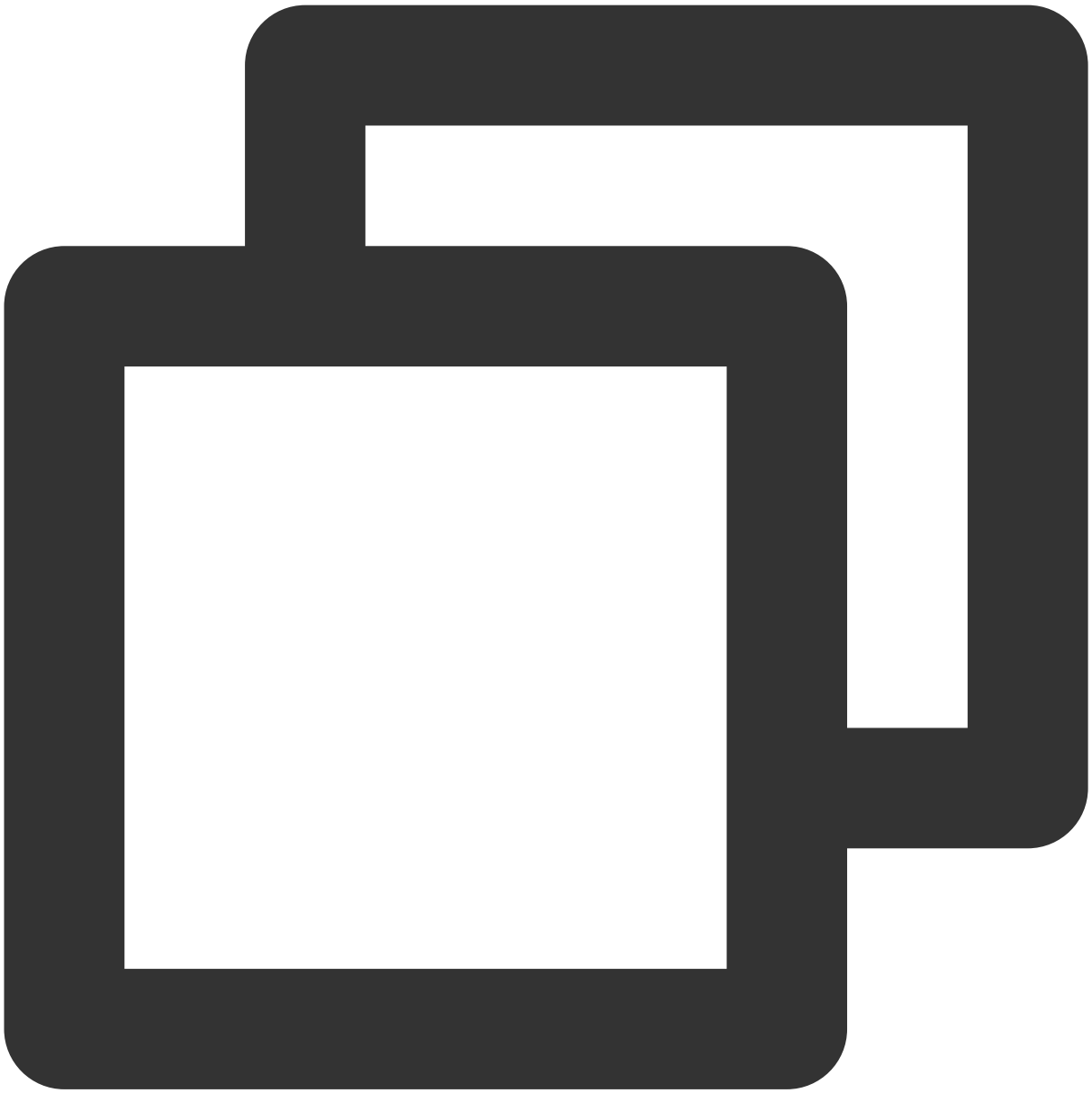
Send email verification code for binding email address during sign-up.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "signup",
"email" : "MOCK_USERNAME@example.com"
```

```
}
```

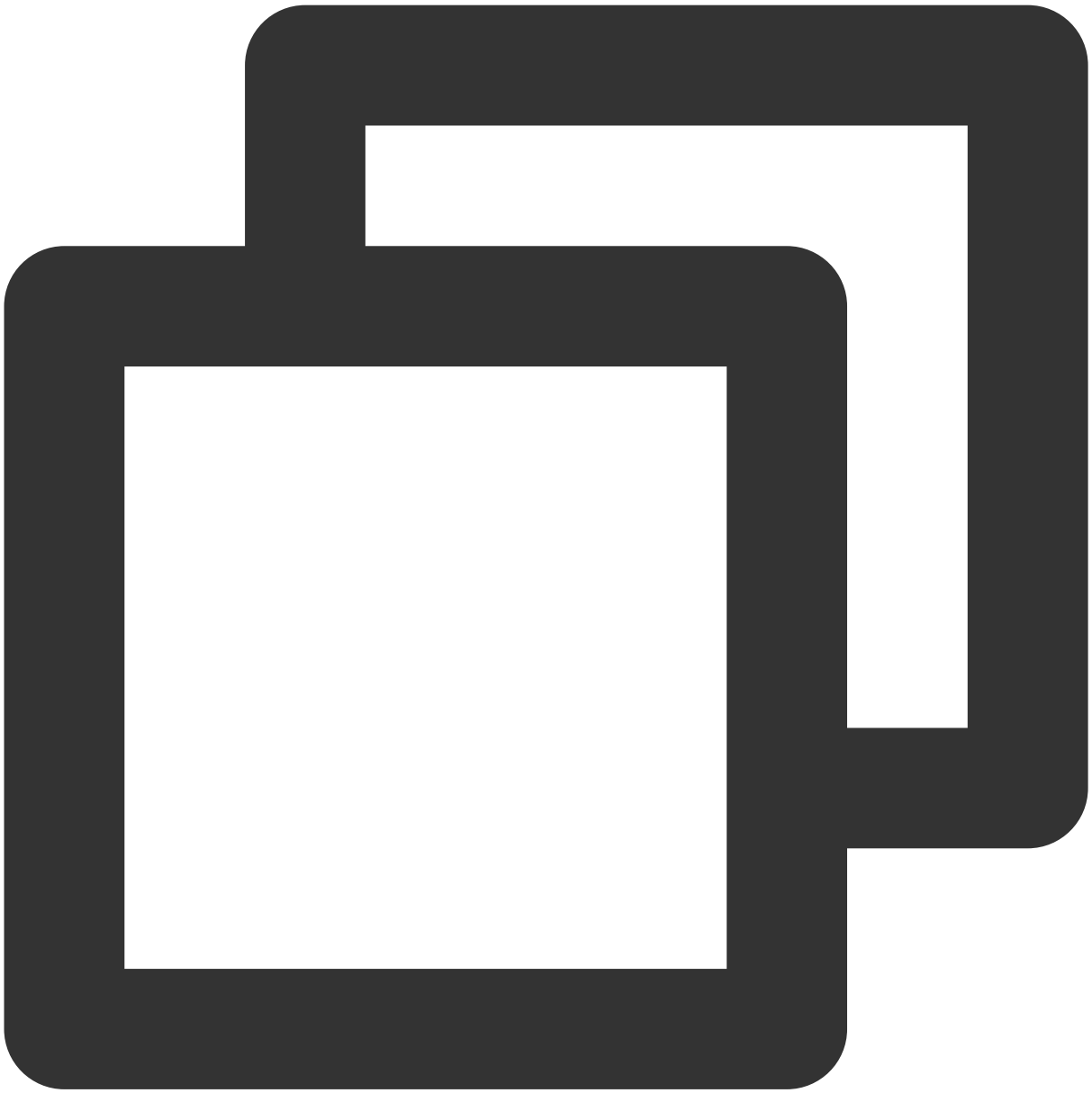
Send email verification code for binding or changing mobile number to update user information.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "update_userinfo",
"phone_number" : "13612345678"
```

```
}
```

Send email verification code for resetting password.



```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "reset_password",
"email" : "MOCK_USERNAME@example.com"
```

```
}
```

Request Headers

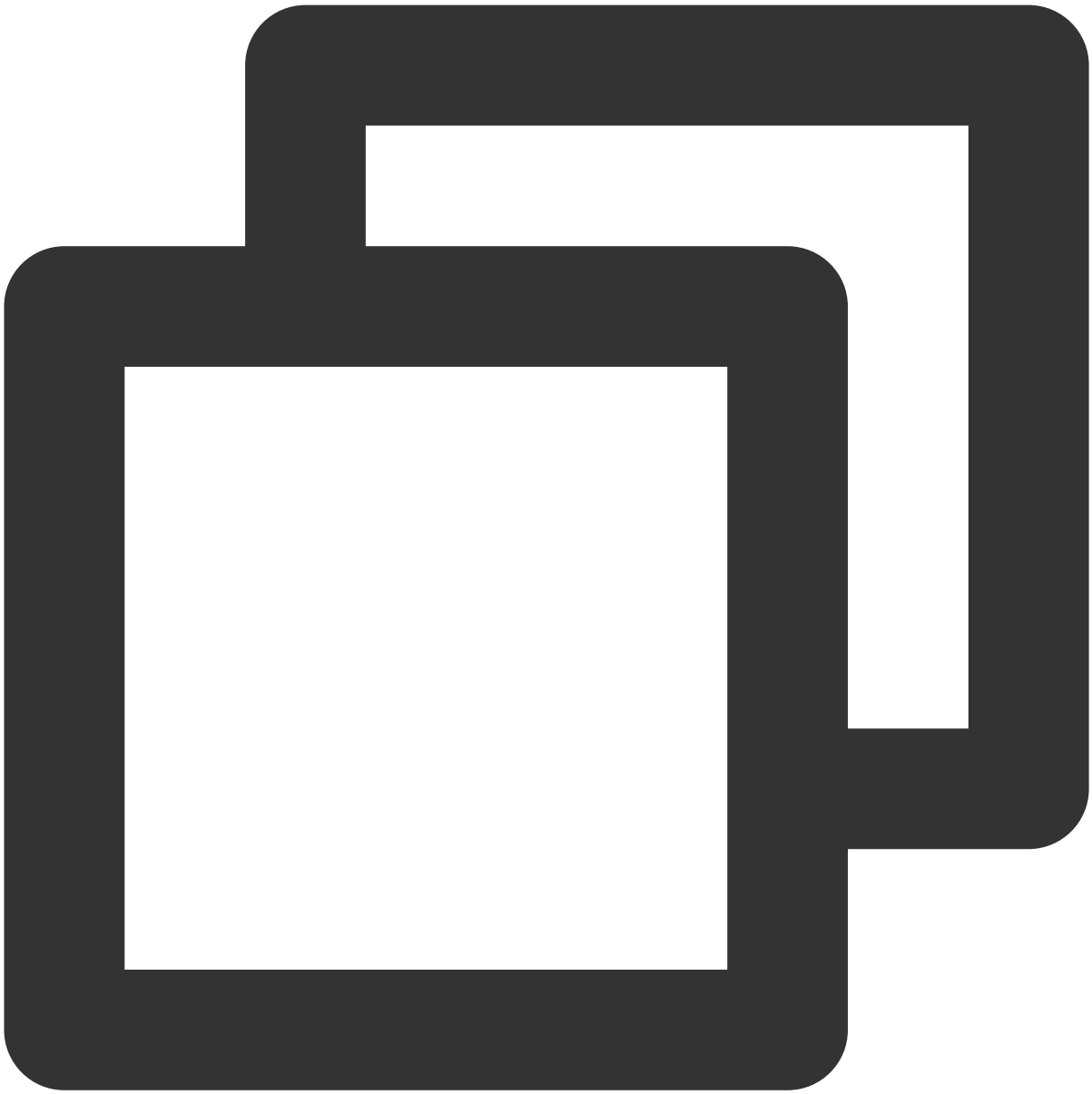
Parameter	Description
Authorization	HTTP `Basic` authentication request header. The format is `Basic <credentials>`, where `Basic` is a fixed string and `<credentials>` is calculated by `base64(url_encode(client_id) + ":" + url_encode( client_secret))`. `Basic` and `<credentials>` are separated by a space.

Request Parameters in JSON Format

JSON Path	Data Type	Description
usage	String	The use case of the OTP verification code. Enter `login` for OTP login by SMS or email. Enter `signup` for user sign-up. Enter `update_userinfo` for updating user information. Enter `reset_password` for resetting user password. If this parameter is not specified, it defaults to login.
phone_number	String	The user's mobile number, which should be an 11-digit mobile number of the three major carriers in Chinese mainland. This parameter is required for sending SMS OTP verification code.
email	String	The user's email address. This parameter is required for sending email OTP verification code.
auth_source_id	String	The ID of the authentication source for OTP by SMS or email. It can be viewed on the general authentication source list of the console. This parameter is required for OTP login by SMS or email. The length and validity period of the verification code in the authentication source configuration will be used. In other scenarios, this parameter is left empty. A 6-digit verification code will be used by default, which is valid for 60 seconds.

Sample Success Responses

The verification code is sent successfully.



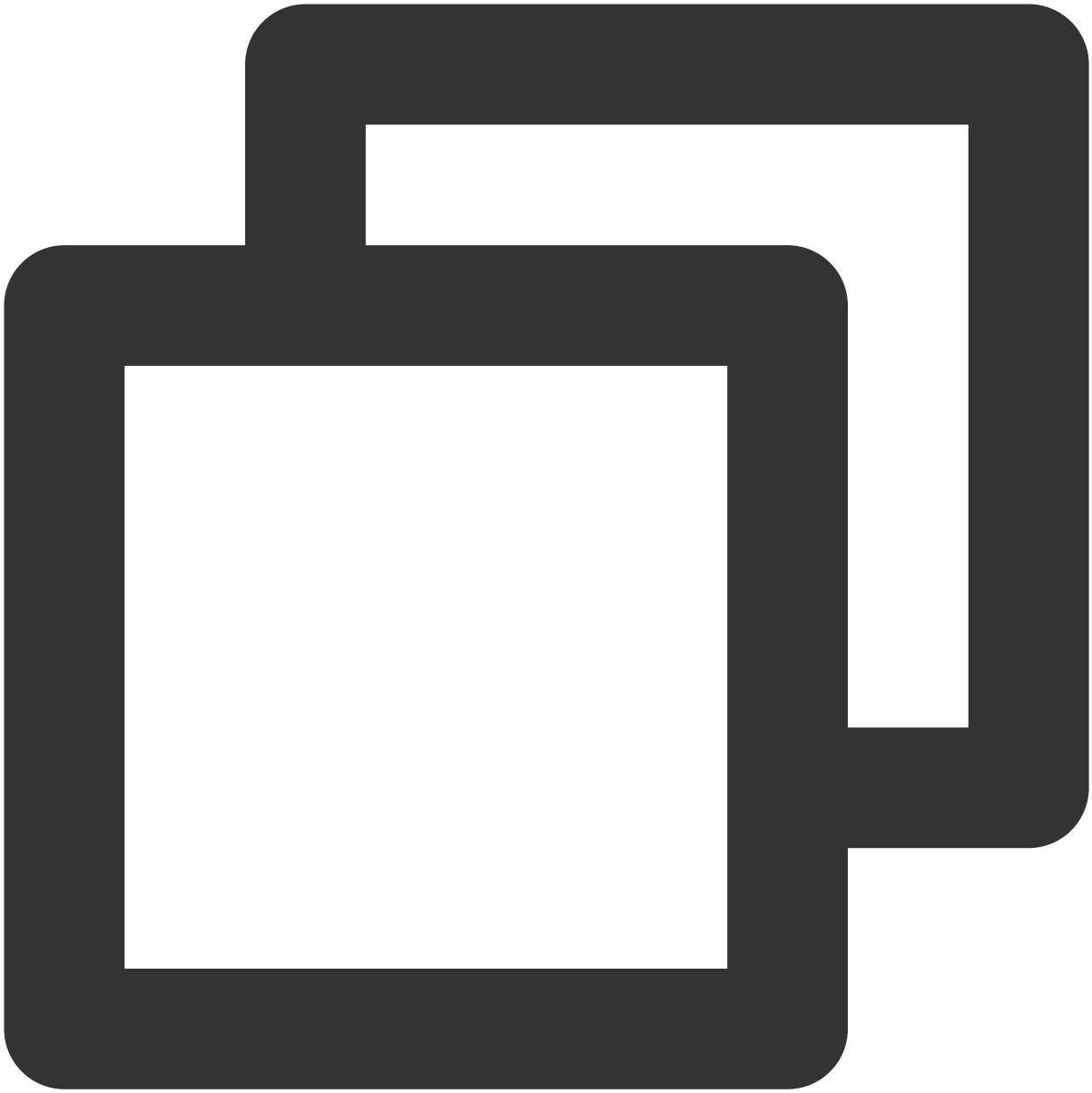
```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "otp_token" : "MOCK_OTP_TOKEN"
}
```

Response Parameters

Parameter	Data Type	Description
otp_token	String	OTP token, which is used in OTP verification. This is valid for 5 minutes.

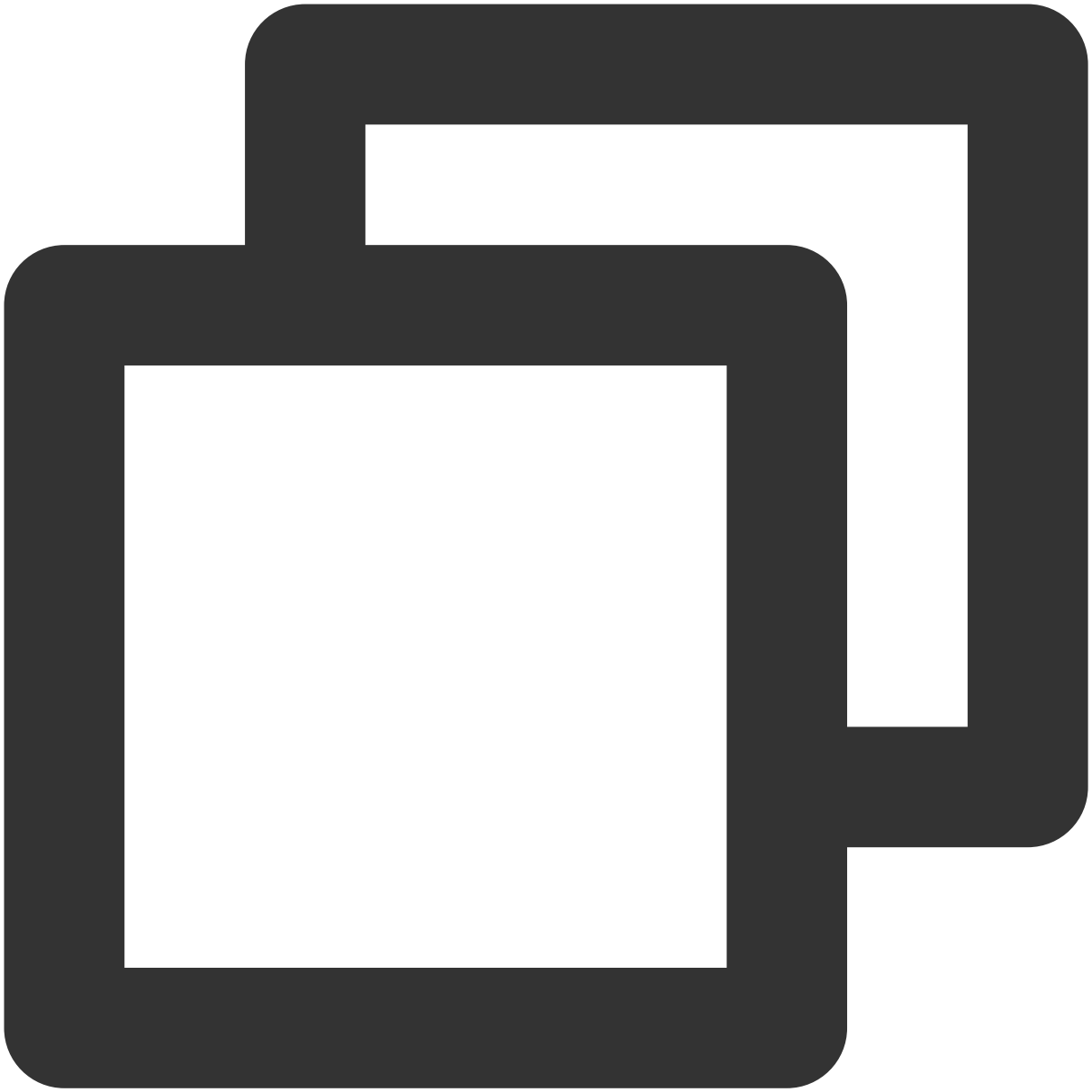
Sample Error Responses

Incorrect mobile number format.



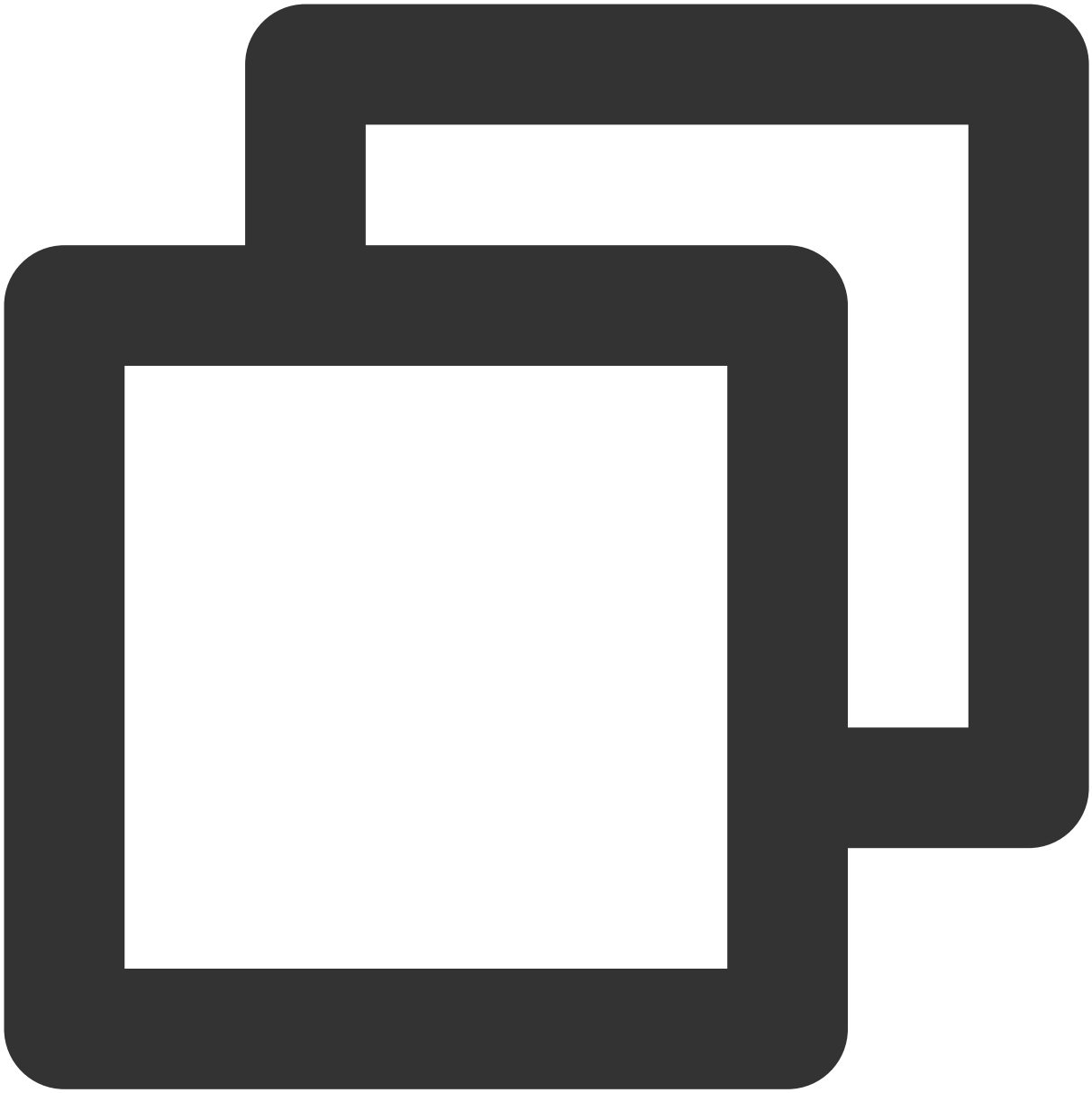
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

Incorrect email address format.



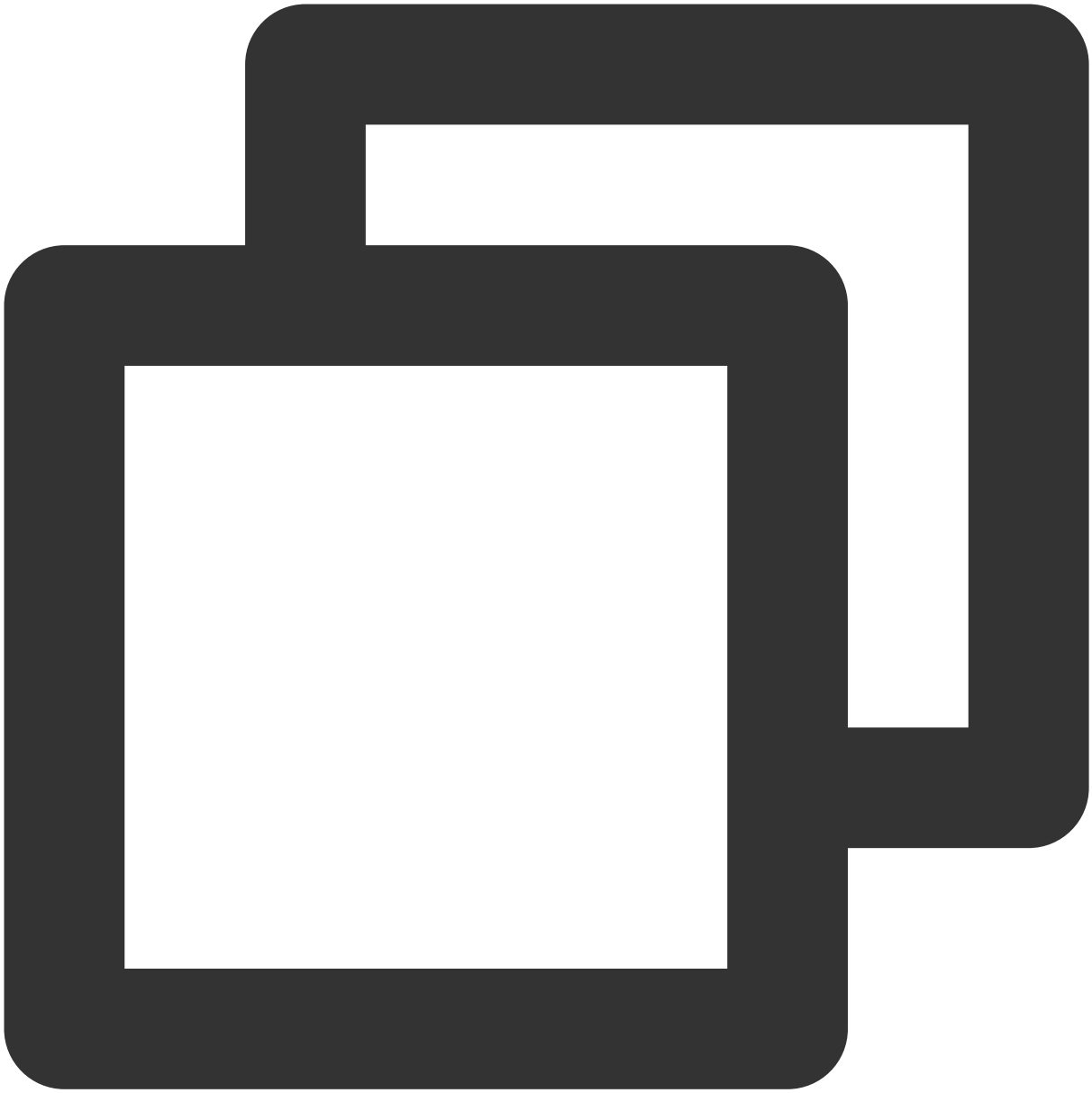
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

SMS message can not be sent due to insufficient SMS quota. This occurs when the free quota has been used up. You need to configure the SMS template on the console.



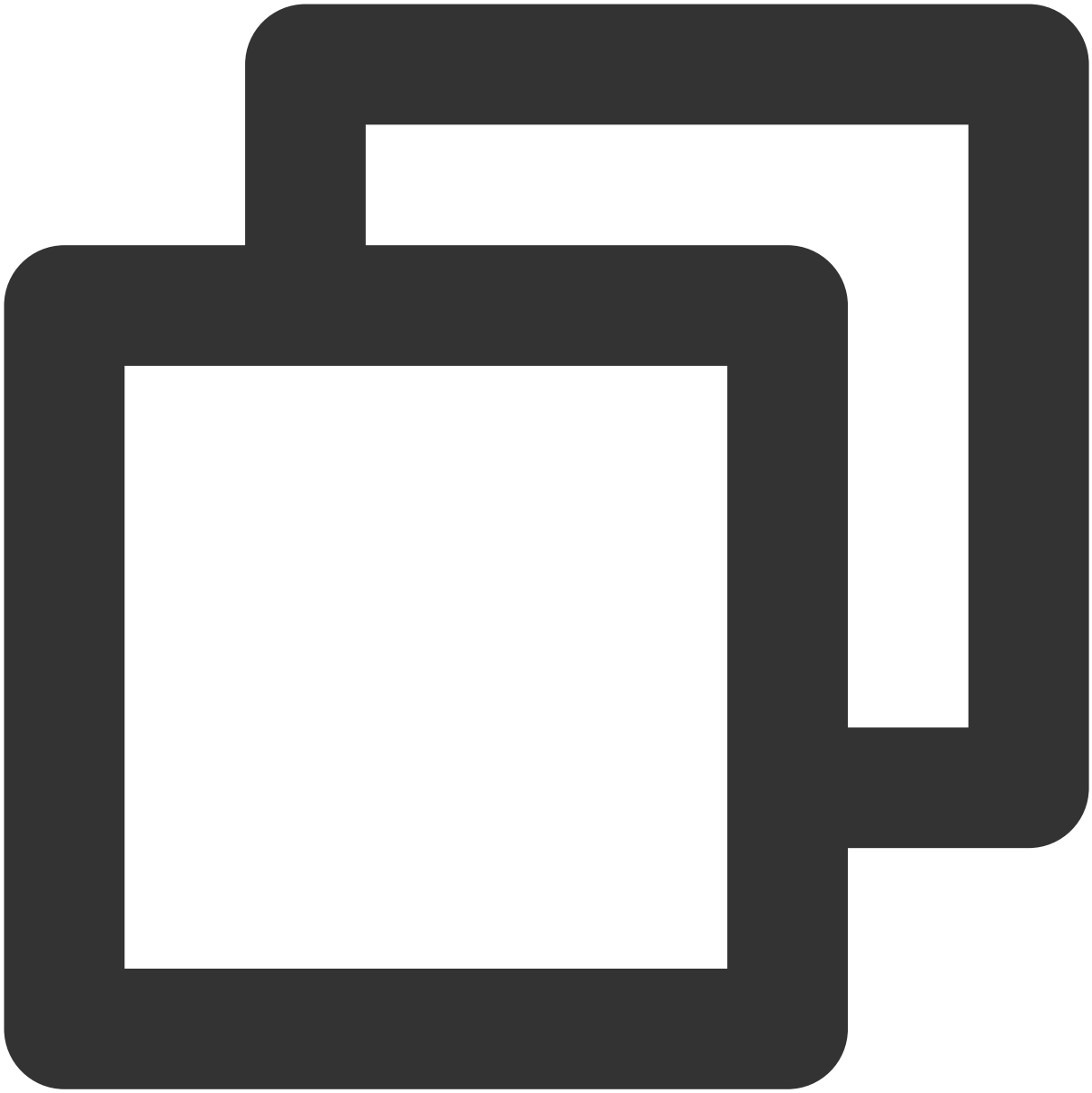
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "insufficient_sms_quota"
}
```

Email can not be sent due to insufficient email quota. This occurs when the free quota has been used up. You need to configure the email template on the console.



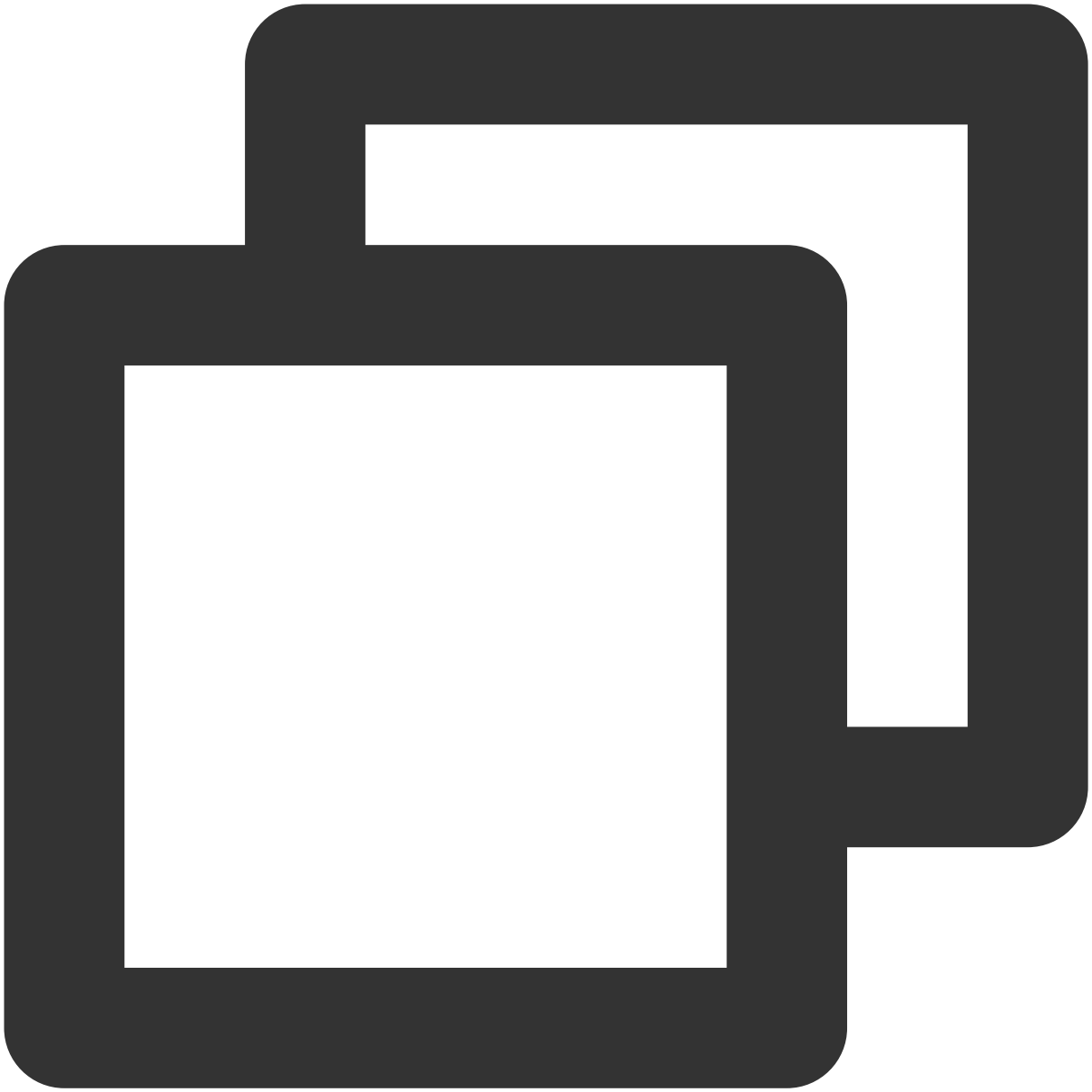
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "insufficient_email_quota"
}
```

The email address does not exist or is on the blocklist.



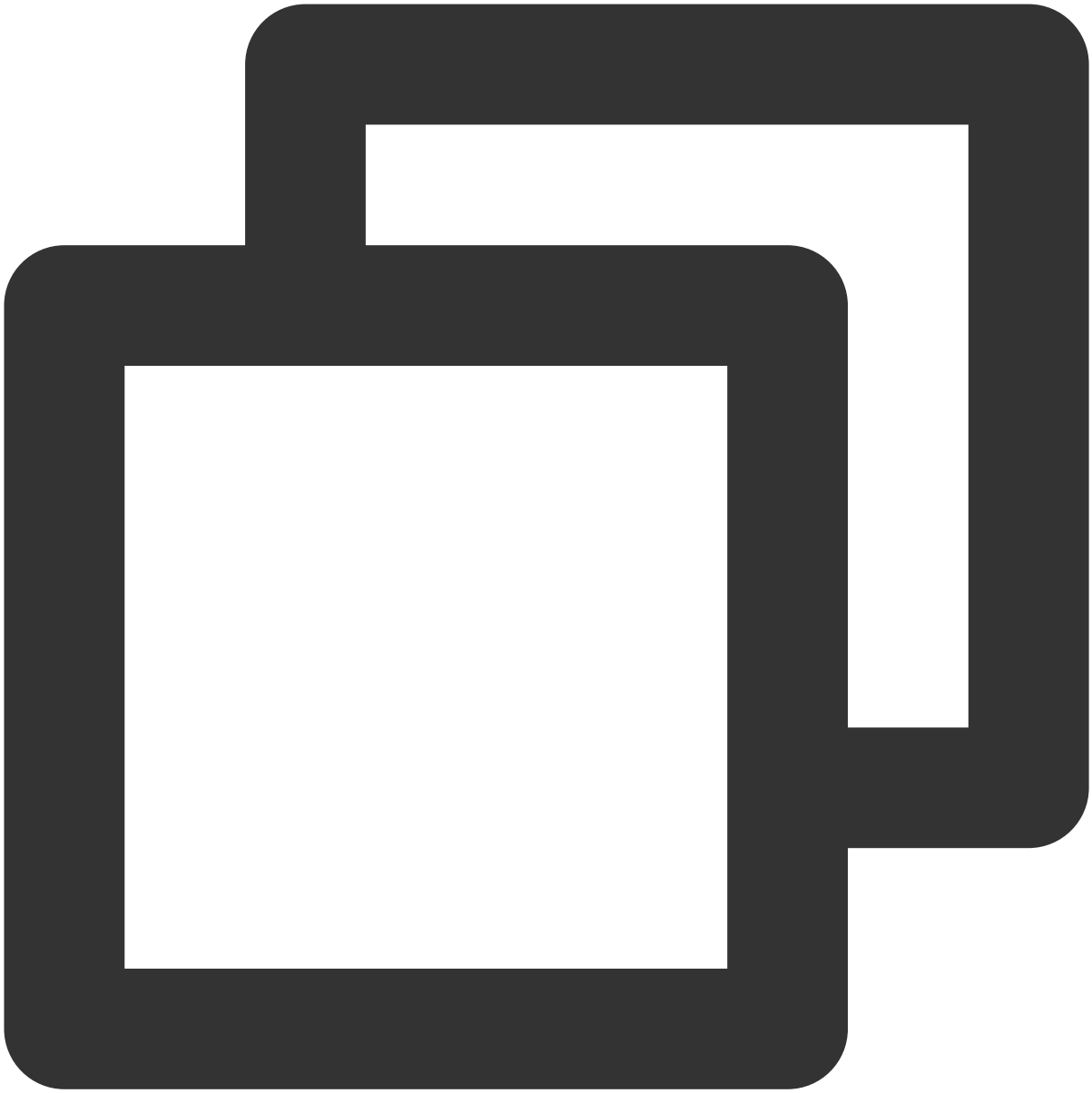
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_email"
}
```

Failed to send the verification code.



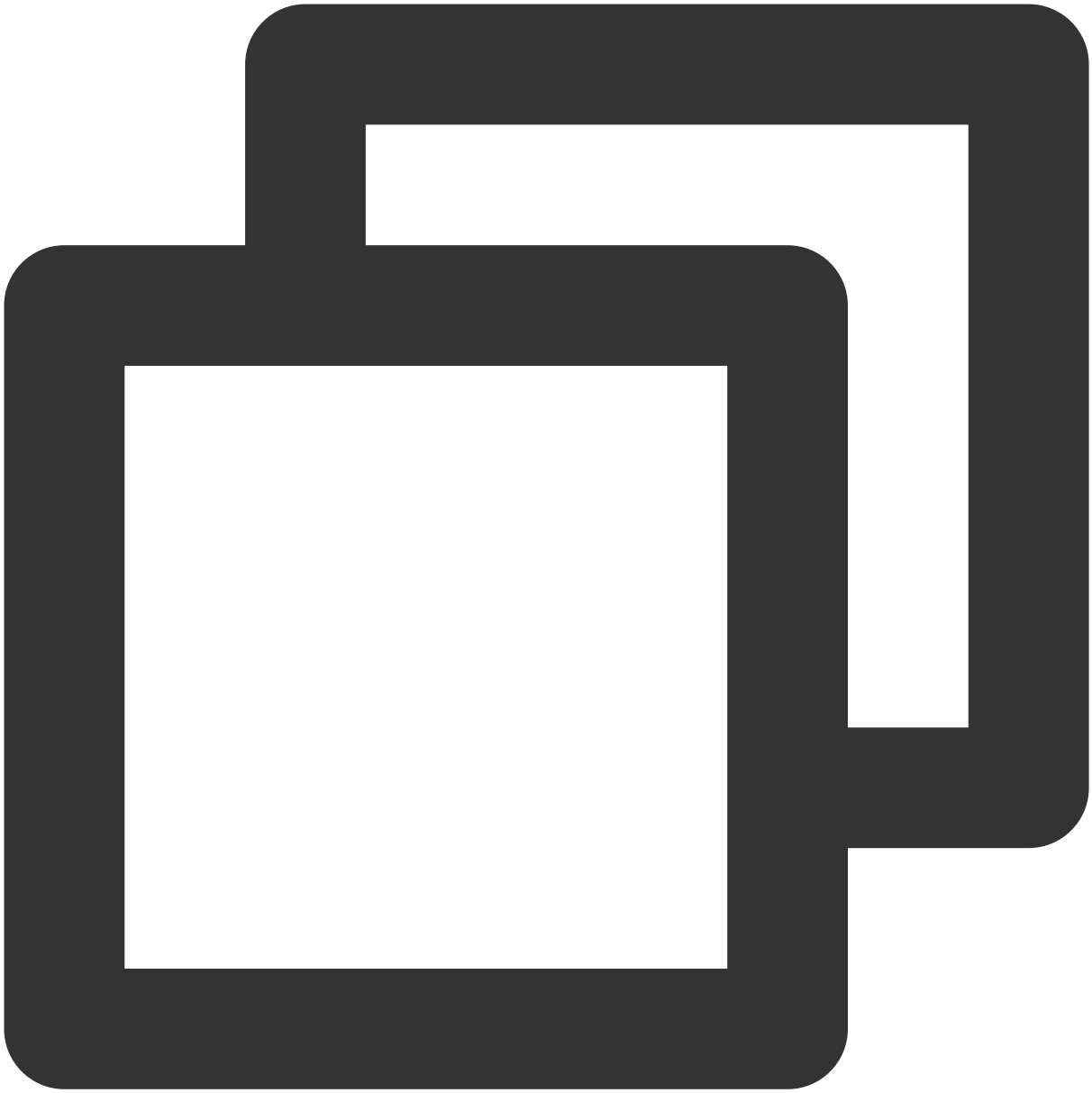
```
HTTP/1.1 503 Service Unavailable
Content-Type: application/json;charset=UTF-8

{
"error" : "temporarily_unavailable",
"error_description" : "Failed to send OTP. Please try again later."
}
```

The email address has already been used for sign-up.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
  "error": "email_is_used"
}
```

The mobile number has already been used for sign-up.

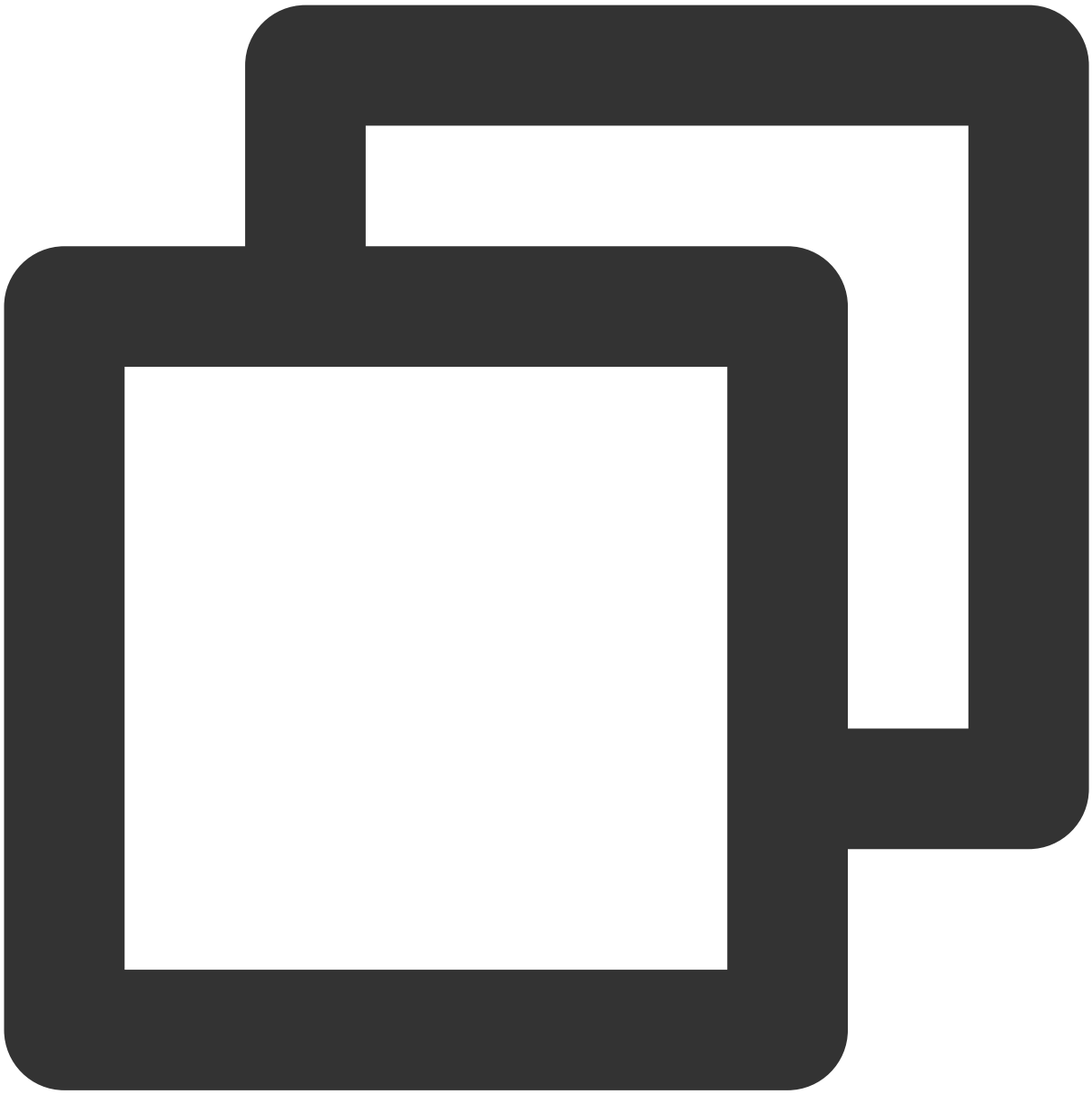


```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8


{
  "error": "phone_number_is_used"
}
```

The limit for sending SMS messages to a mobile number is exceeded.

If you use the purchased SMS quota, you can modify the message sending limit policy in the SMS Console.

If you use the free SMS quota, a maximum of 50 SMS messages can be sent to one mobile number every day, and only one SMS message can be sent to one mobile number within 30 seconds.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "sms_rate_limit_exceeded",
"error_description" : "SMS rate limit exceeded for same phone number"
}
```

Get User Information

Last updated：2023-12-22 11:42:07

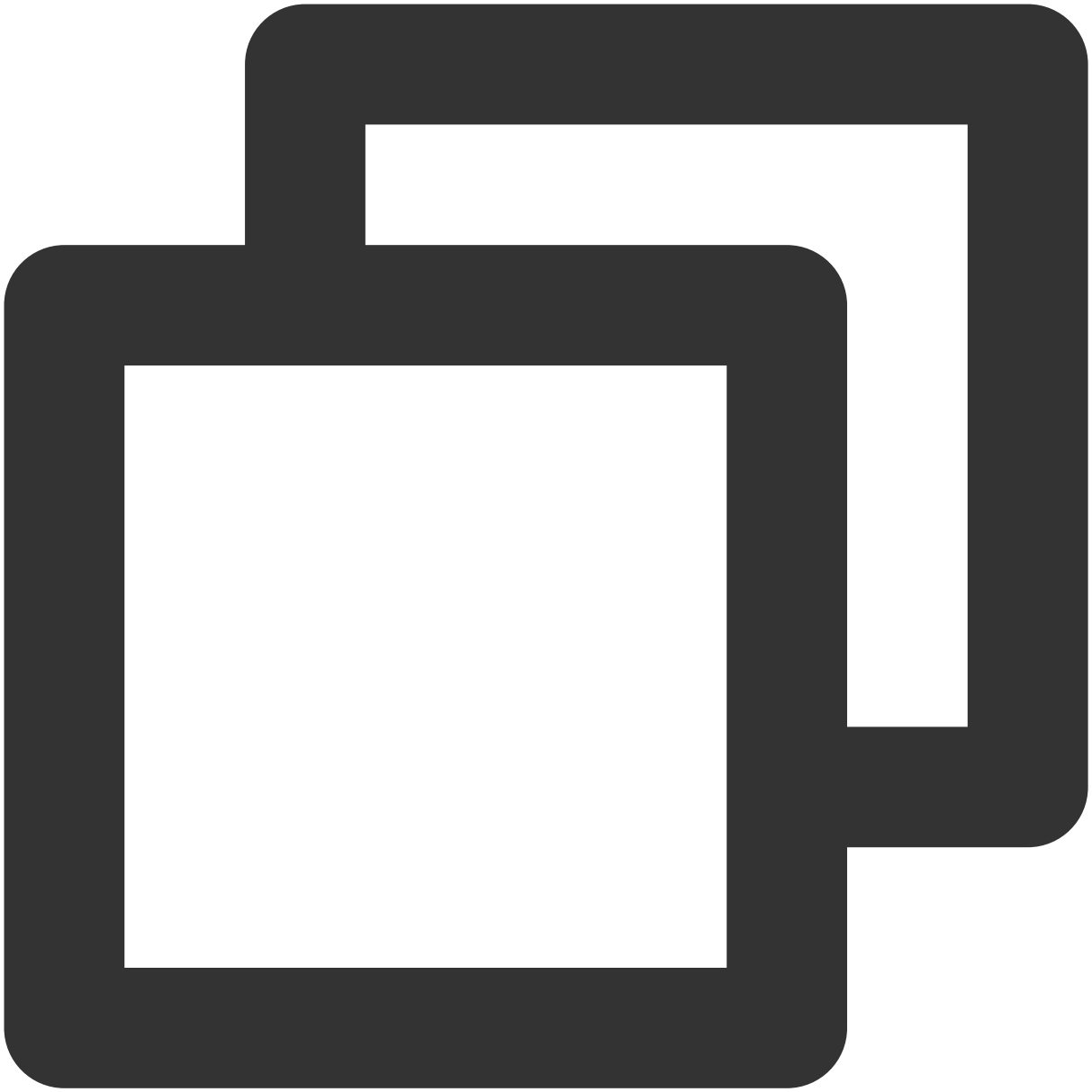
API Description

This API is used to get the user information of a logged-in user. When calling this API, you need to carry the Access Token with `openid scope` returned when the login is successful.

Supported Applications

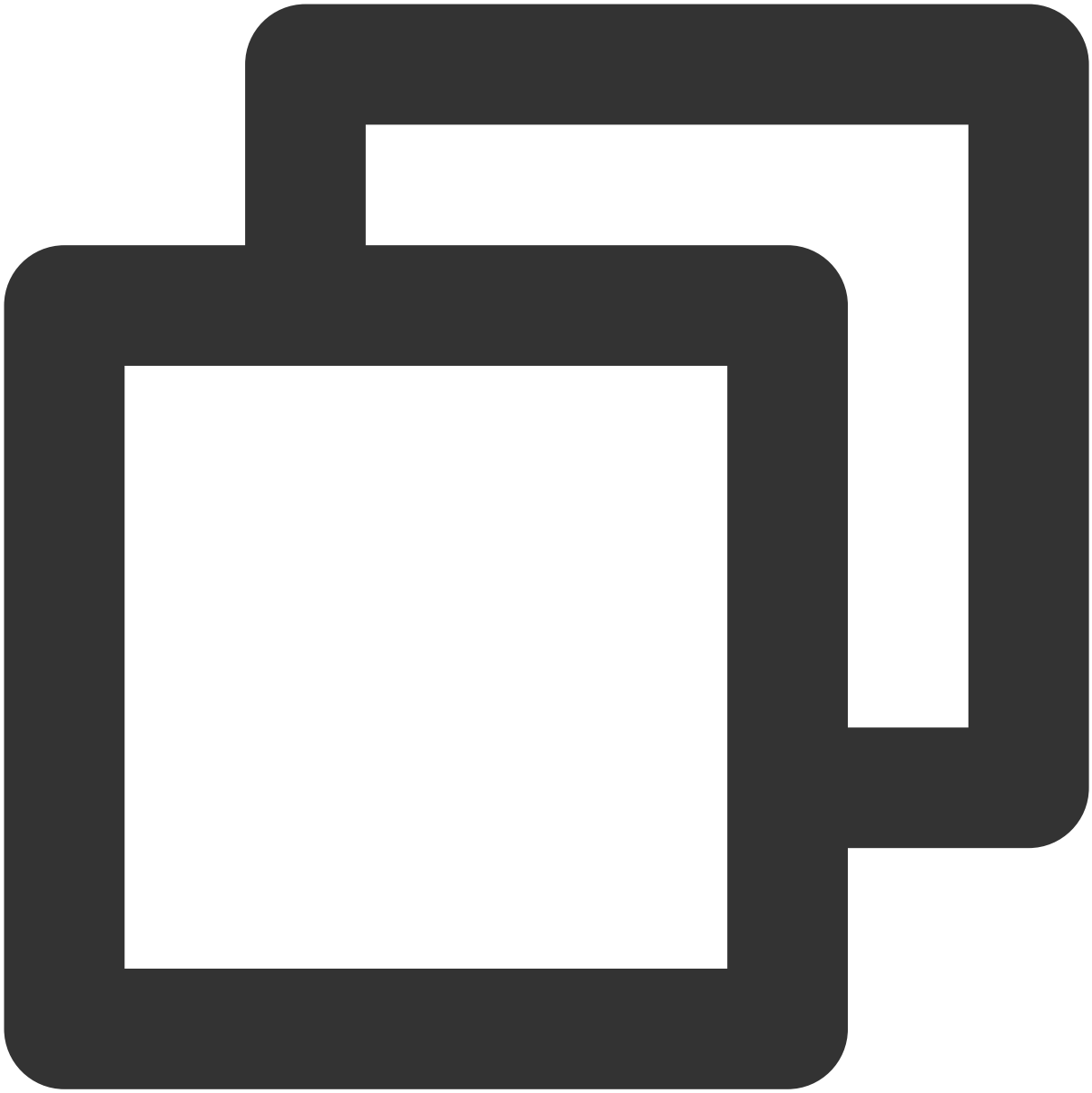
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



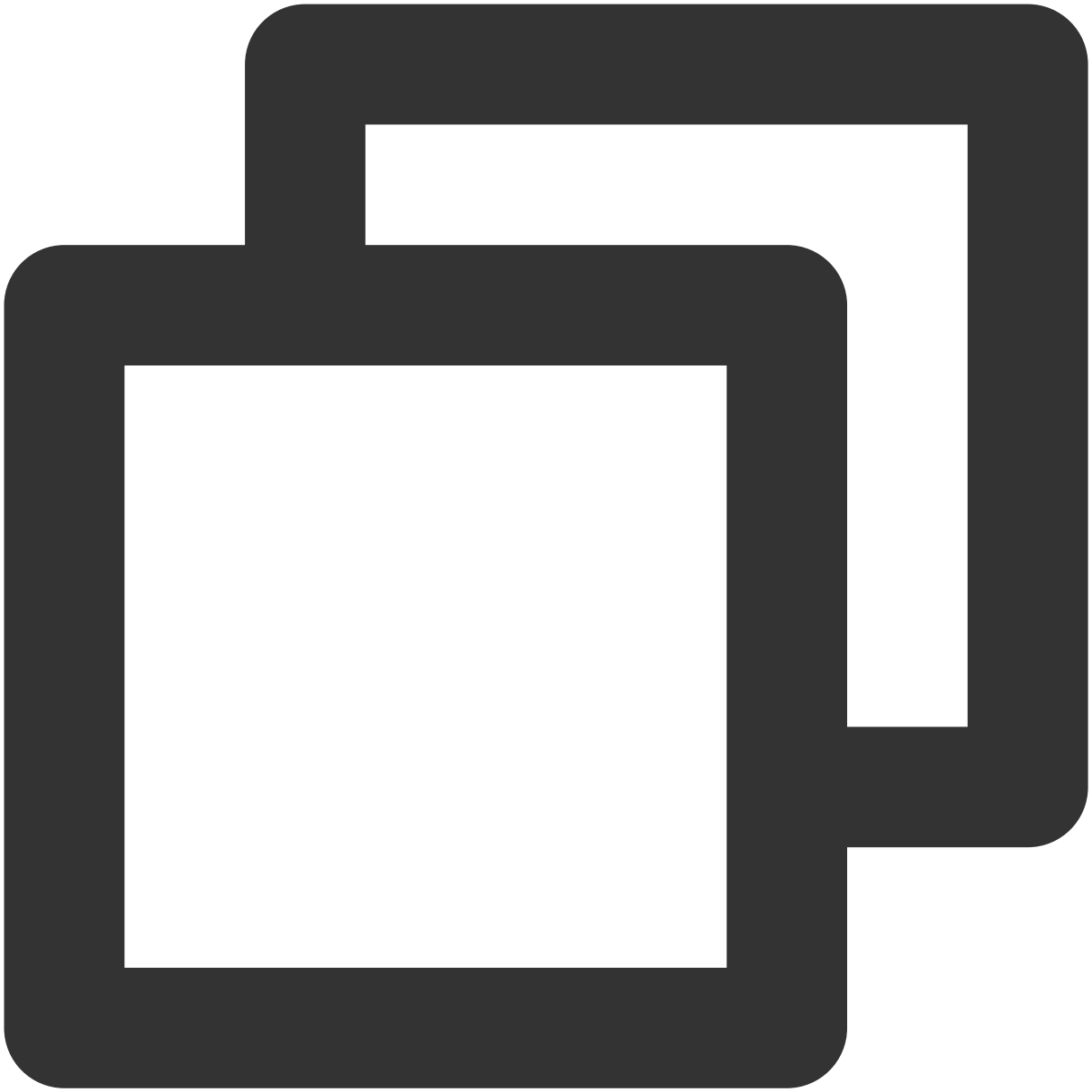
```
GET
```

Request Path



```
/userinfo
```

Sample Requests



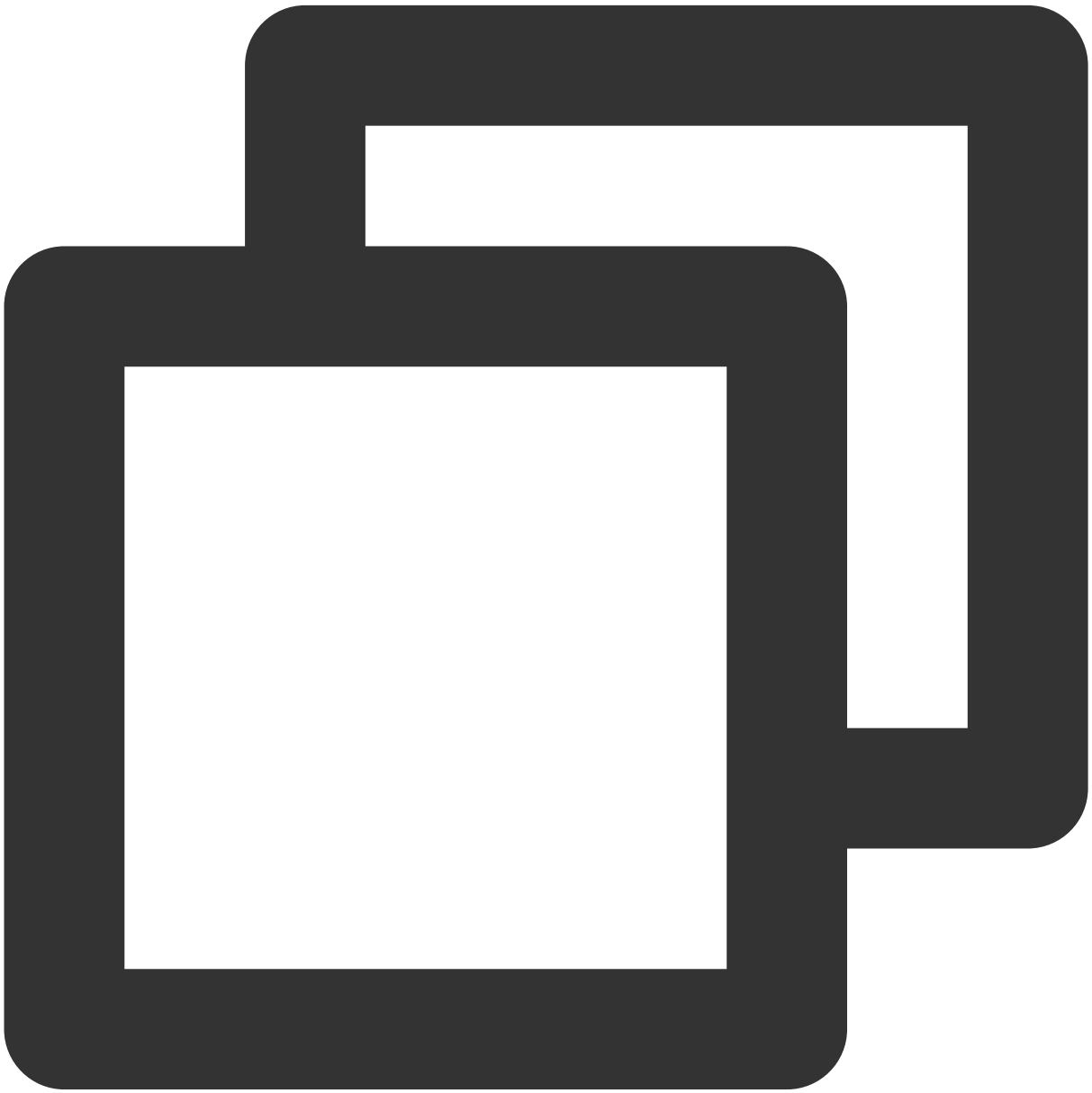
```
GET /userinfo HTTP/1.1
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com
```

Request Headers

Parameter	Description

Authorization	OAuth 2.0 Bearer Token. The format is `Bearer <Token>`, where `Bearer` is a fixed string and `<Token>` is the Access Token with `openid scope` returned when the login is successful. `Bearer` and `<Token>` are separated by a space.

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "sub" : "MOCK_USER_ID",
  "email" : "MOCK_USERNAME@example.com",
  "name" : "MOCK_NAME",
  "nickname" : "MOCK_NICKNAME",
  "zoneinfo" : "Asia/Shanghai",
  "locale" : "zh-CN"
}
```

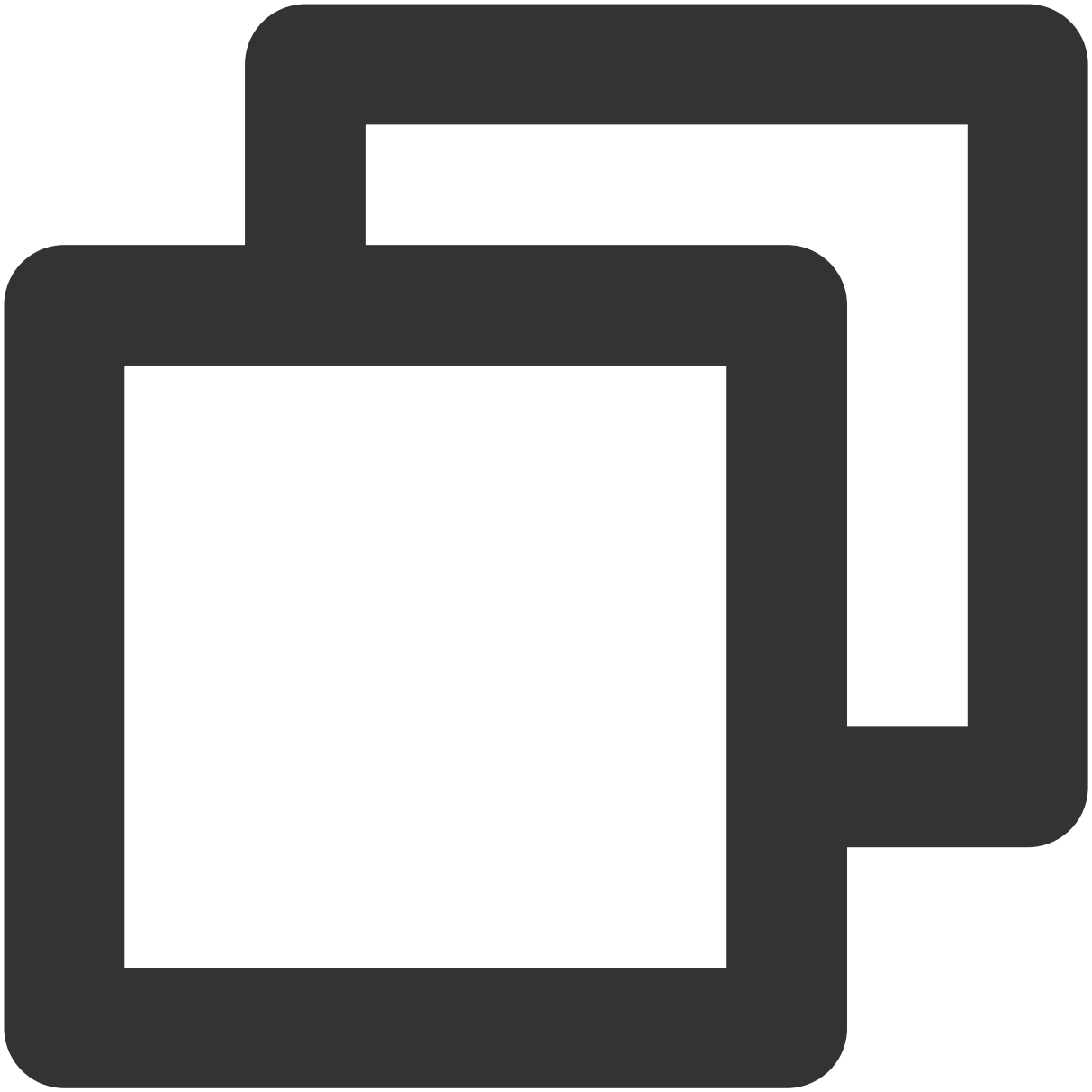
Response Parameters

Parameter	Data Type	Description
sub	String	Unique identifier of the user in the user pool.

Note:
Only the `sub` parameter must be returned, and other parameters returned are determined by `Claims` in the application parameter configuration.

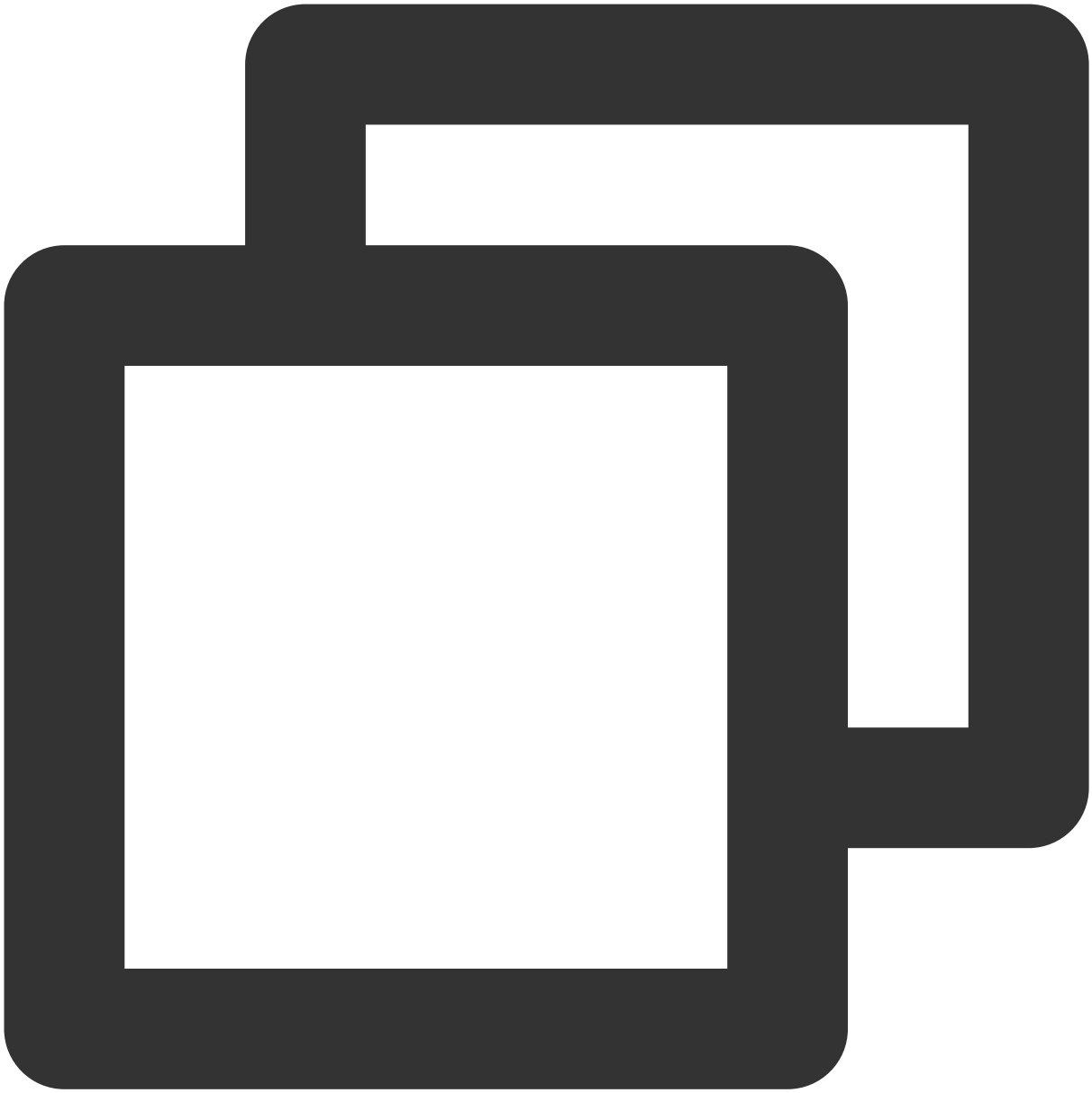
Sample Error Responses

No Access Token.



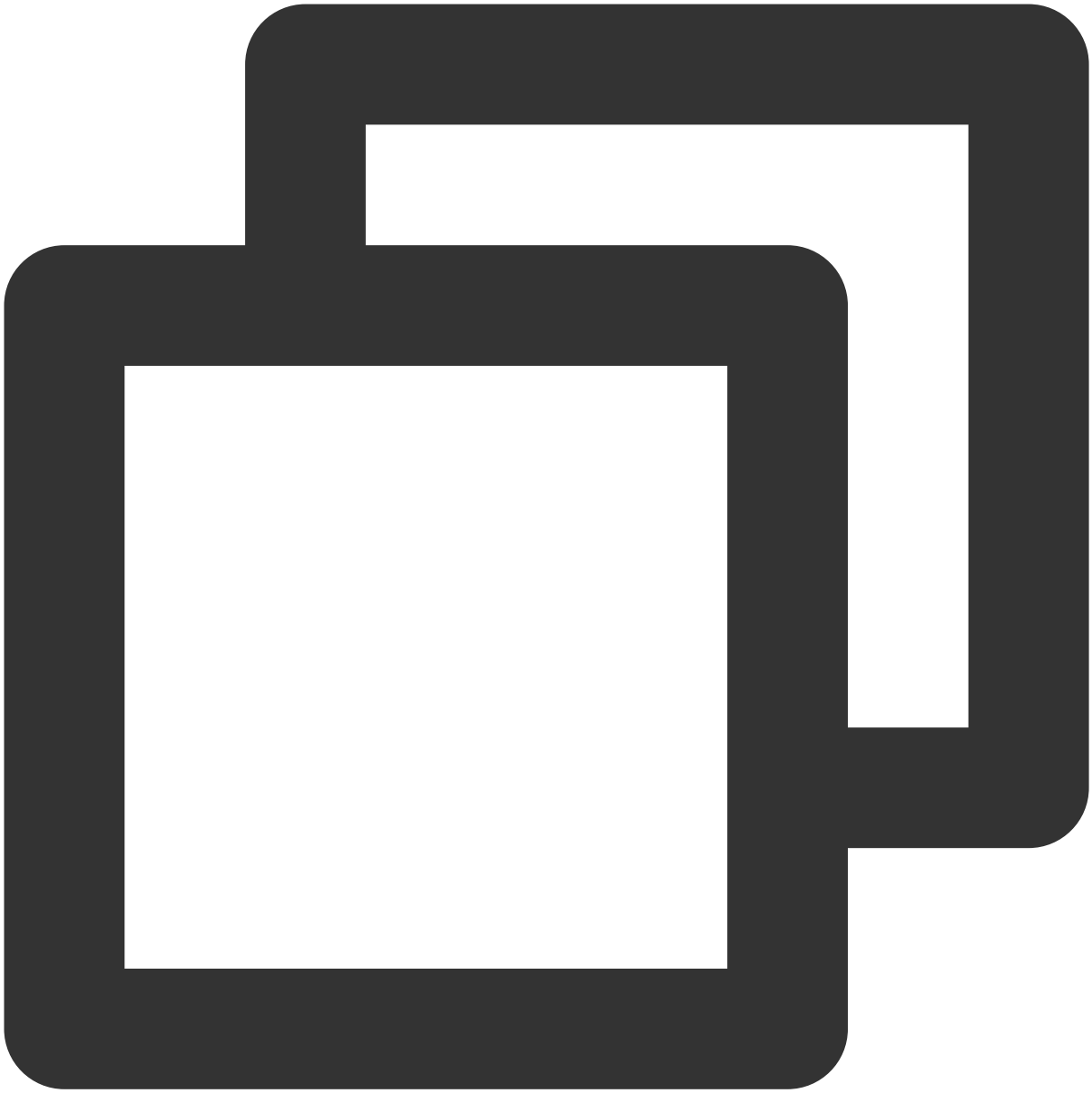
```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token n
```

The Access Token is invalid. For example, the Token format is invalid. The Token has expired or has been revoked.



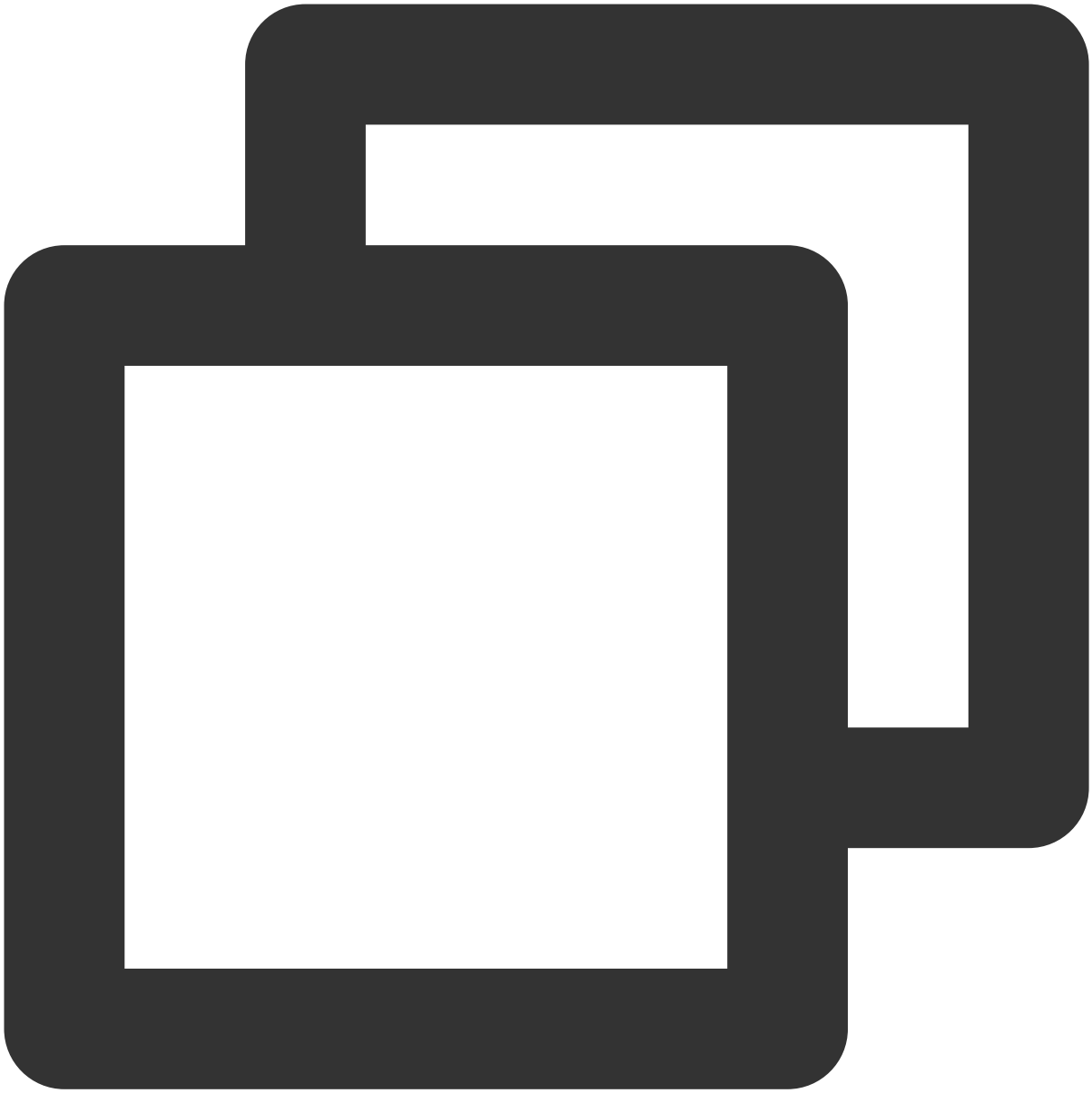
```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding J
```

The Access Token does contain `openid scope` .



```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request
```

No user found for the Access Token.



```
HTTP/1.1 404 Not Found
Content-Type: application/json;charset=UTF-8

{
  "error" : "user_not_found"
}
```

Update User Information

Last updated：2023-12-22 11:42:07

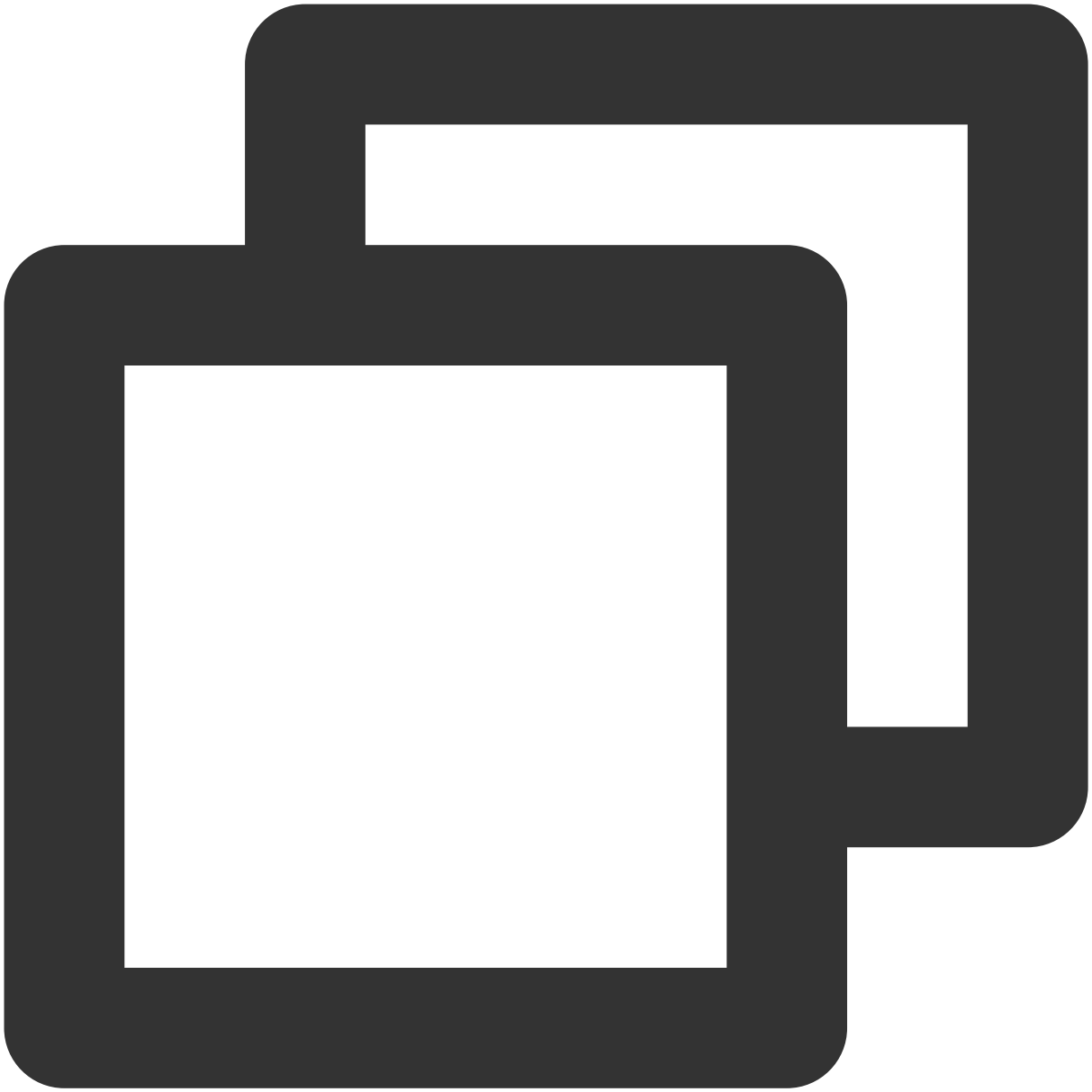
API Description

This API is used to update the personal information of a logged-in user. When calling this API, you need to carry the Access Token with `openid scope` returned when the login is successful. To change the mobile number or email address, call the API for sending one-time password (OTP) verification code to send a verification code to the user.

Supported Applications

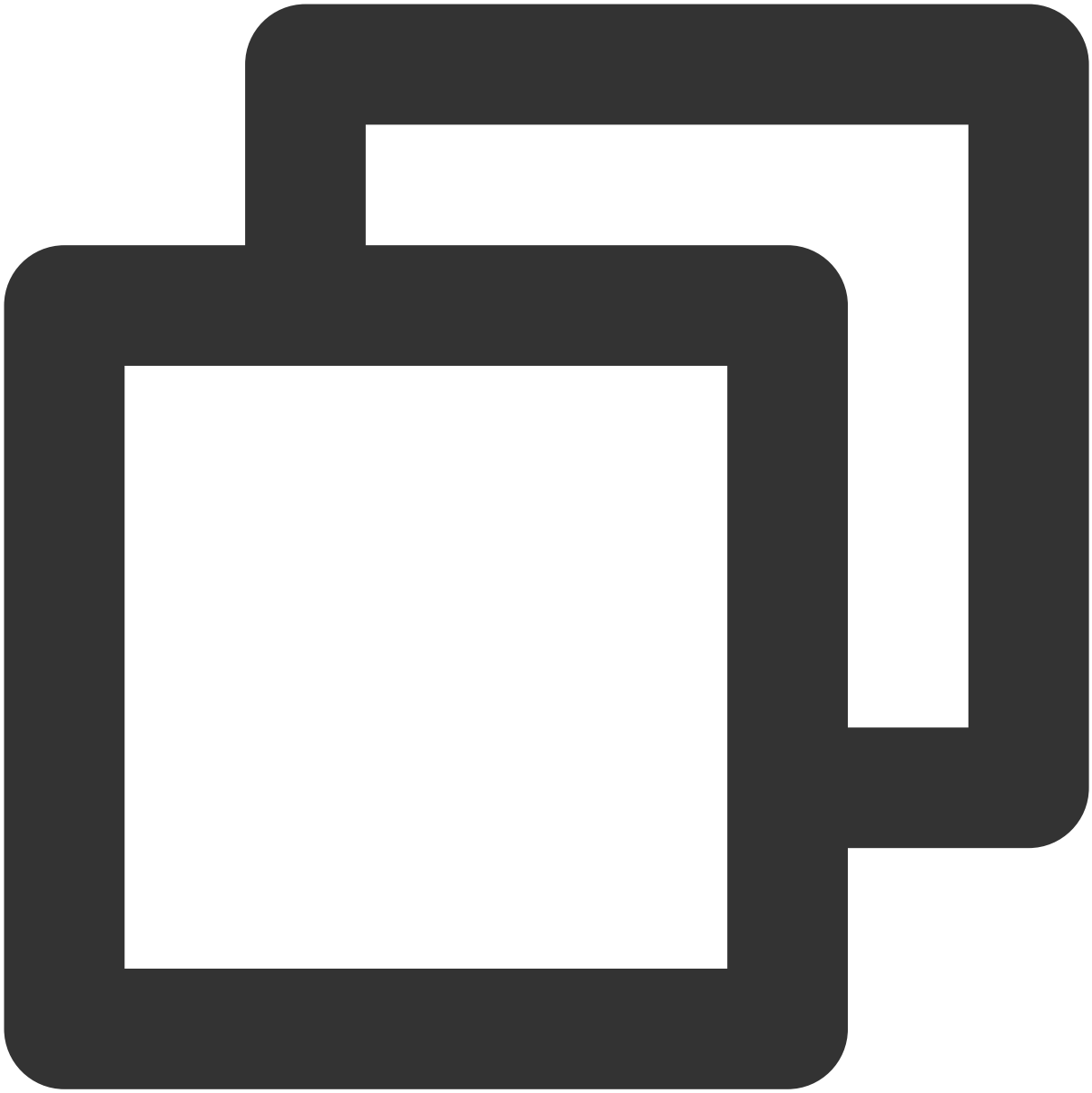
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



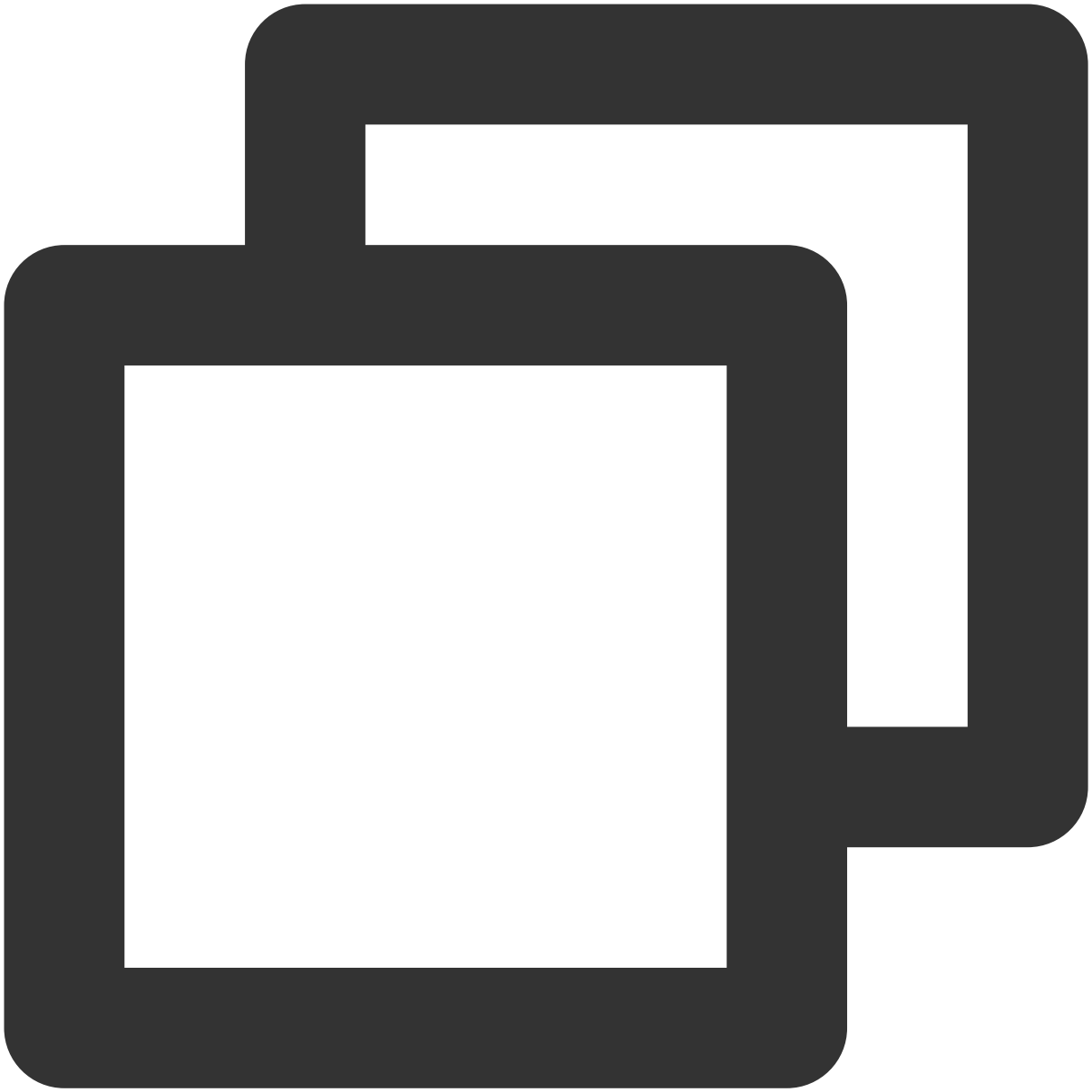
```
PATCH
```

Request Path



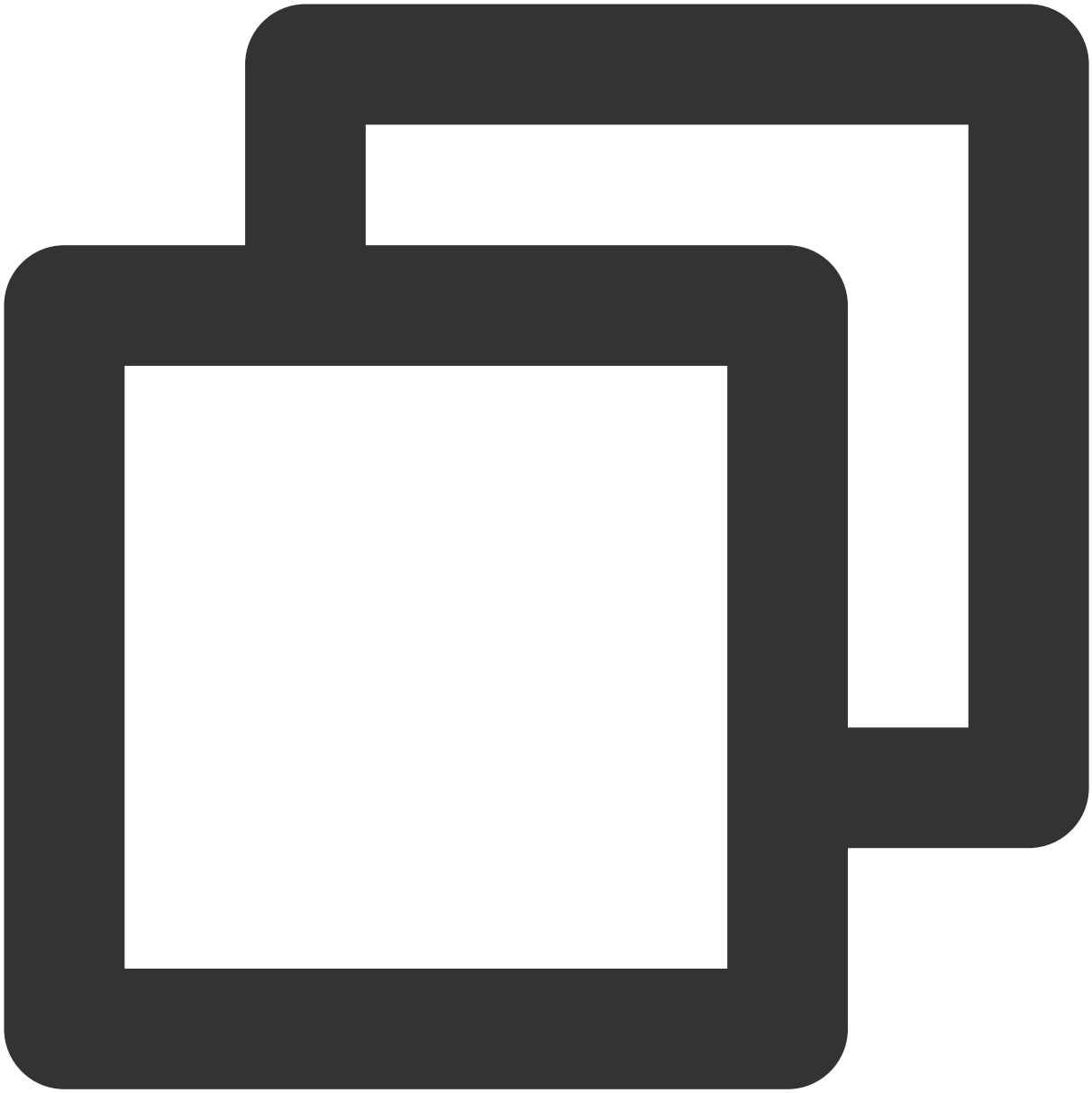
```
/userinfo
```

Request Content-Type



```
application/json
```

Sample Requests



```
PATCH /userinfo HTTP/1.1
Content-Type: application/json
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com

{
  "nickname" : "MOCK_NICKNAME"
}
```

Request Headers

Parameter	Description
Authorization	OAuth 2.0 Bearer Token. The format is `Bearer <Token>>` , where `Bearer` is a fixed string and `<Token>` is the Access Token with `openid scope` returned when the login is successful. `Bearer` and `<Token>` are separated by a space.

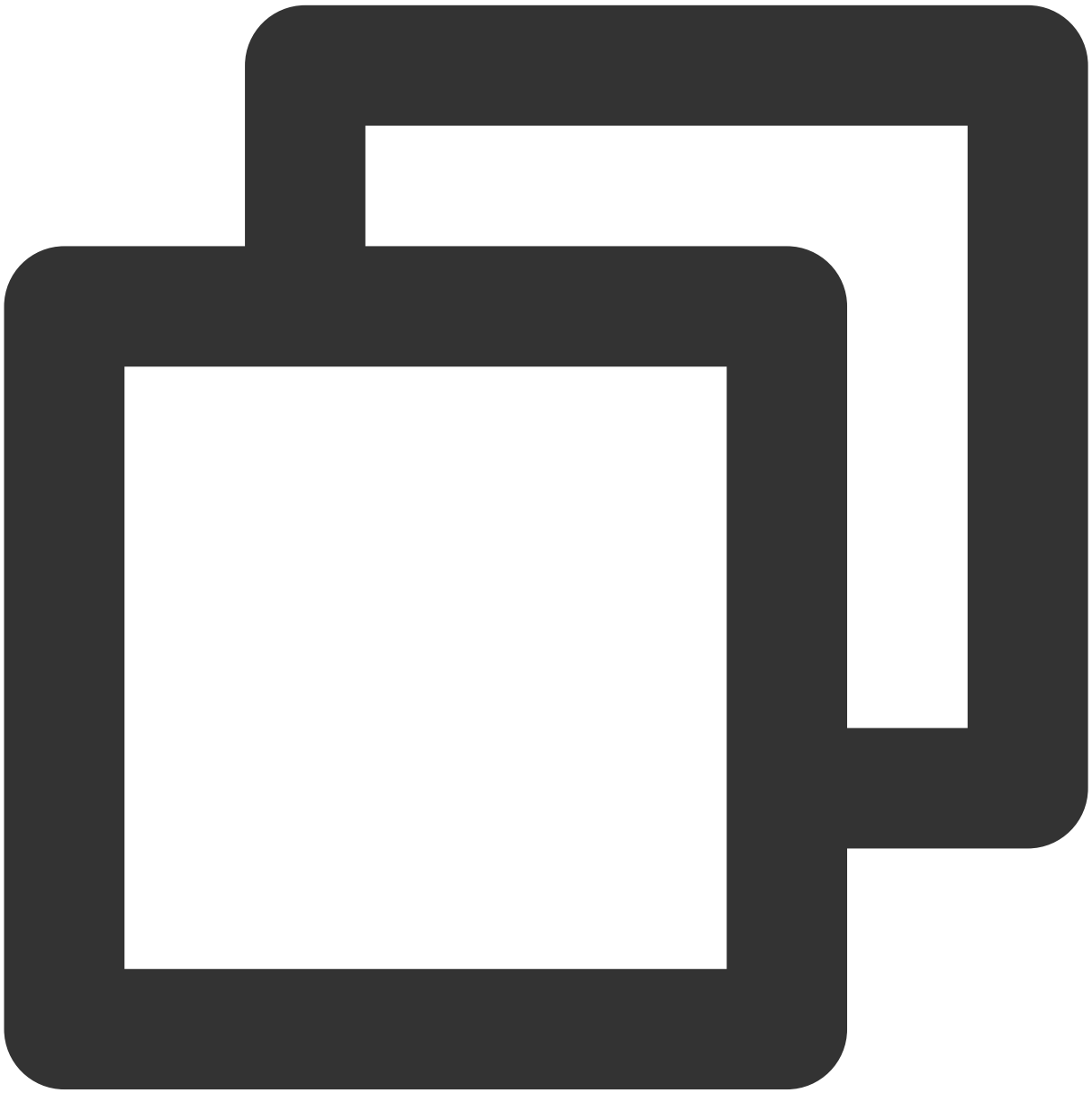
Request Parameters in JSON Format

JSON Path	Data Type	Description
phone_number	String	The user's mobile number, which should be an 11-digit mobile number of the three major carriers in Chinese mainland. When this parameter is specified, both `phone_number_otp_token` and `phone_number_otp` are required.
hone_number_otp_token	String	The `otp_token` returned by the server after the SMS verification code is sent.
phone_number_otp	String	The OTP verification code received by the user's mobile phone.
email	String	The user's email address. When this parameter is specified, both `email_otp_token` and `email_otp` are required.
email_otp_token	String	The `otp_token` returned by the server after the email verification code is sent.
email_otp	String	The OTP verification code received by the user's email.
Name	String	Username
nickname	String	The user's alias.
zoneinfo	String	The user’s time zone, such as `Asia/Shanghai` or `Europe/Paris` .
locale	String	The user `locale` information, such as `zh-CN` or `en-US` .

Note:

The values of other parameters are user attribute identifiers. The attribute identifiers can be viewed on the Custom Attributes page.

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json
```

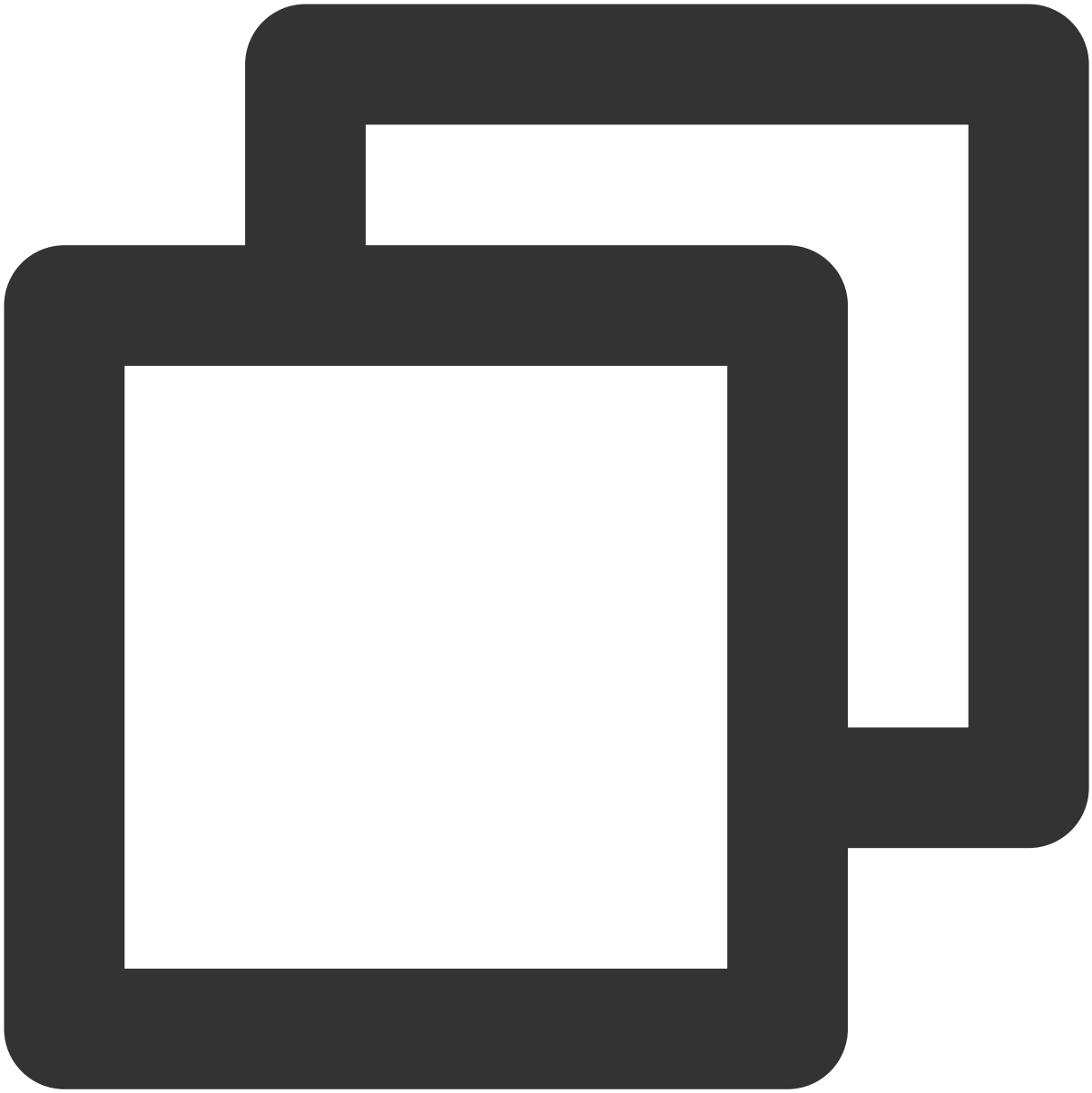
```
{
  "sub" : "MOCK_USER_ID",
  "email" : "MOCK_USERNAME@example.com",
  "name" : "MOCK_NAME",
  "nickname" : "MOCK_NICKNAME",
  "zoneinfo" : "Asia/Shanghai",
  "locale" : "zh-CN"
}
```

Note:

Only the `sub` parameter must be returned, and other parameters returned are determined by `Claims` in the application parameter configuration.

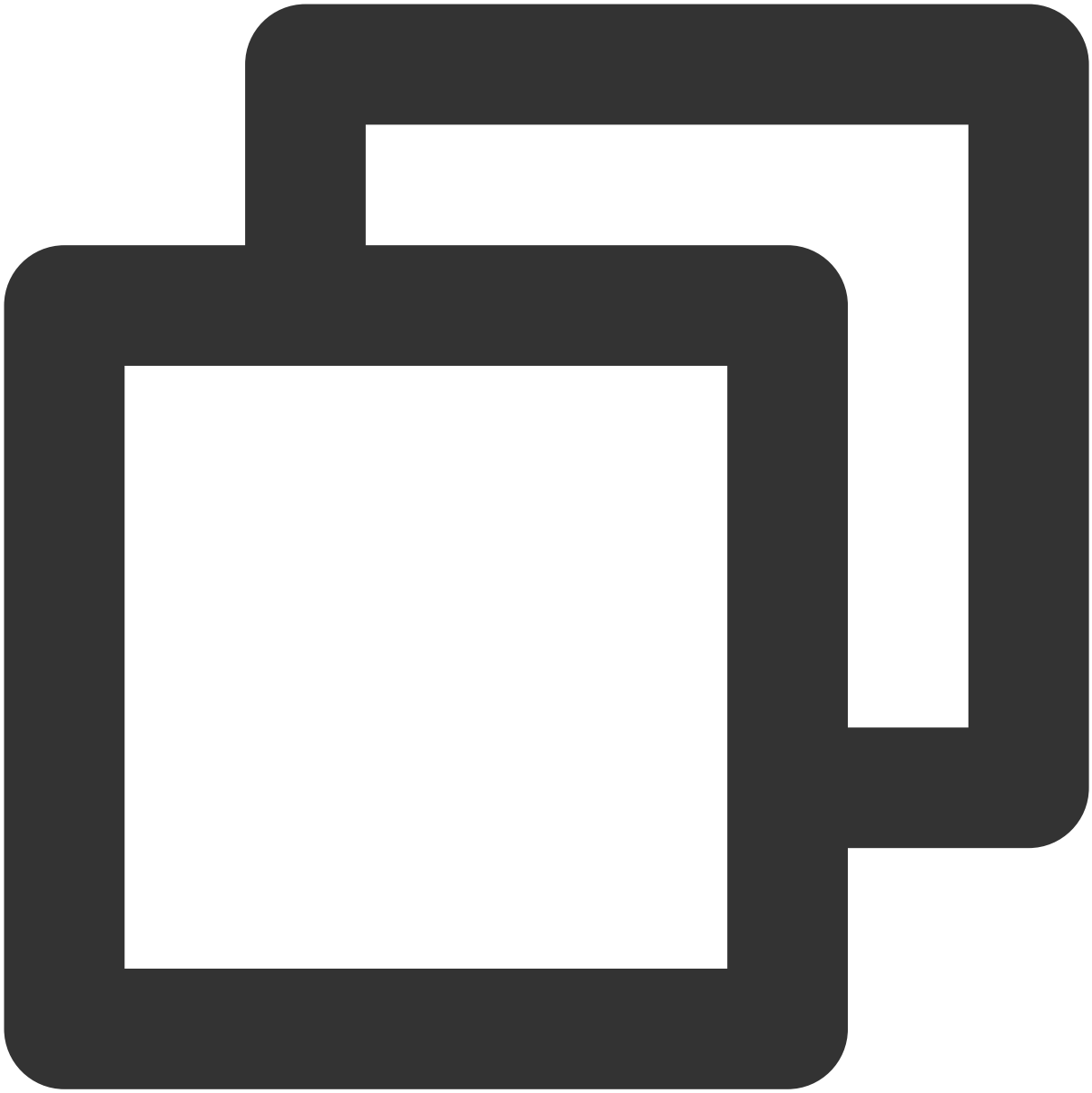
Sample Error Responses

The input parameters contain unknown attributes.



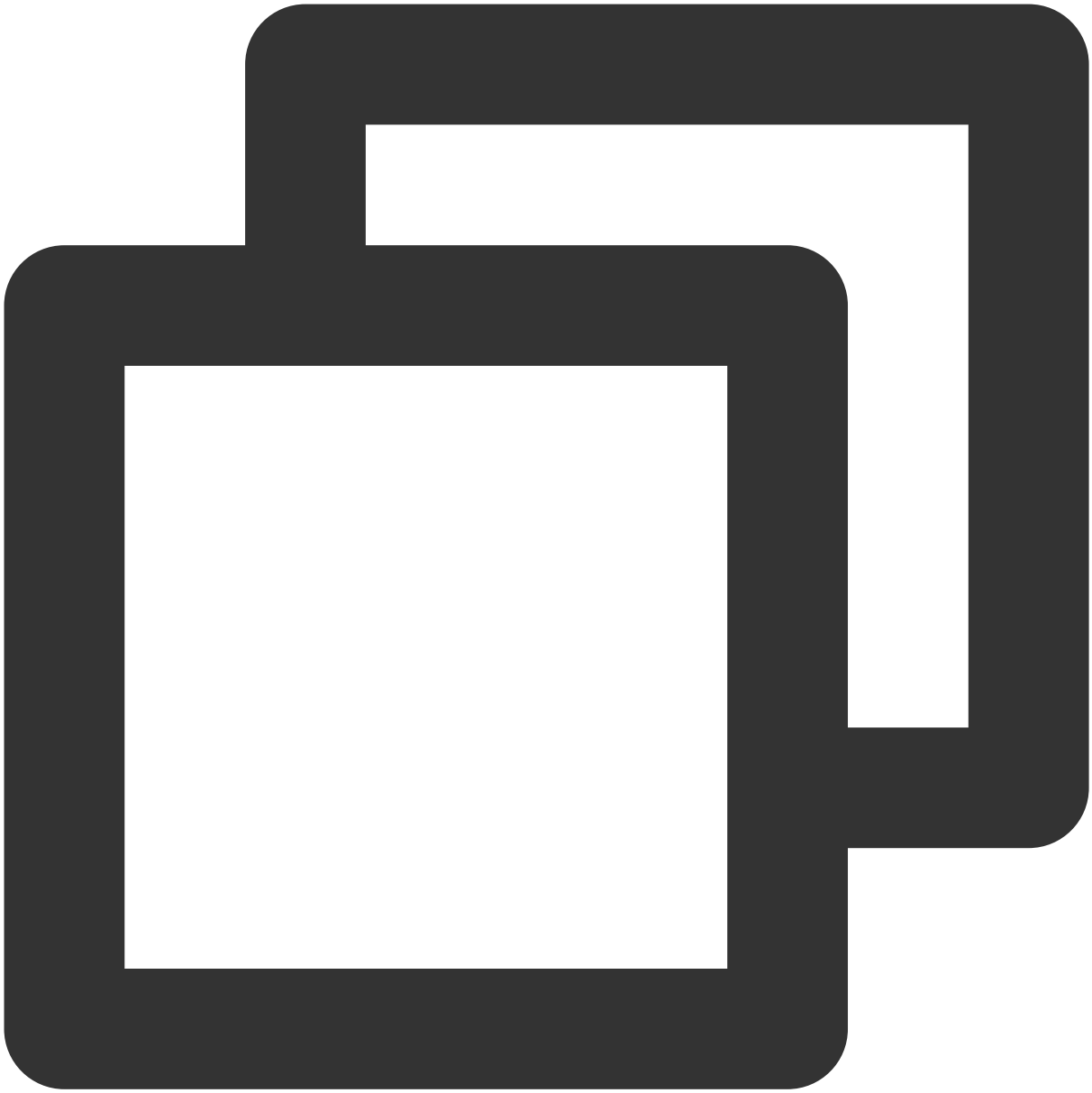
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unknown attribute(s) found."
}
```

The input parameters contain attributes that cannot be modified.



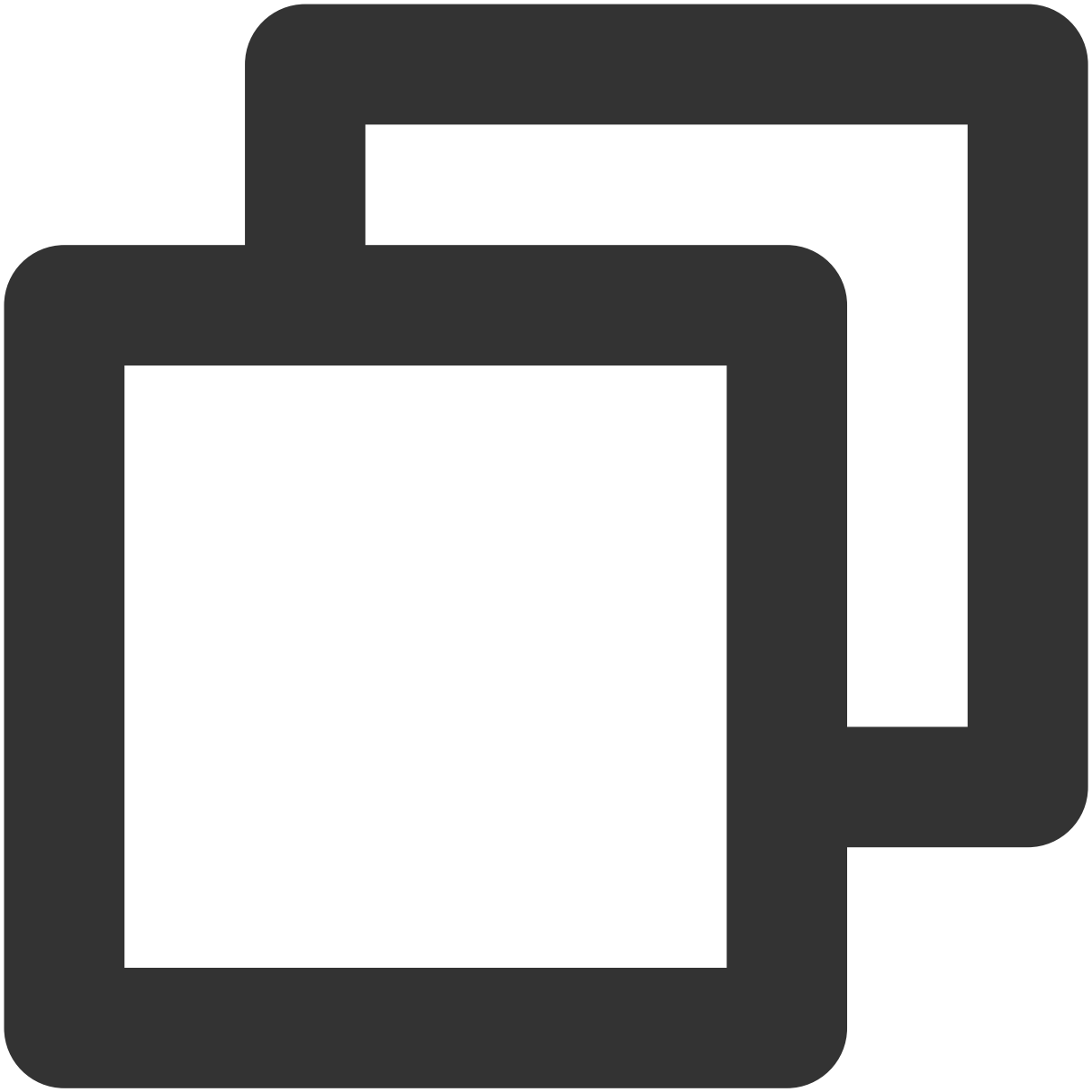
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unsupported user attribute(s) found."
}
```

The mobile number format is invalid.



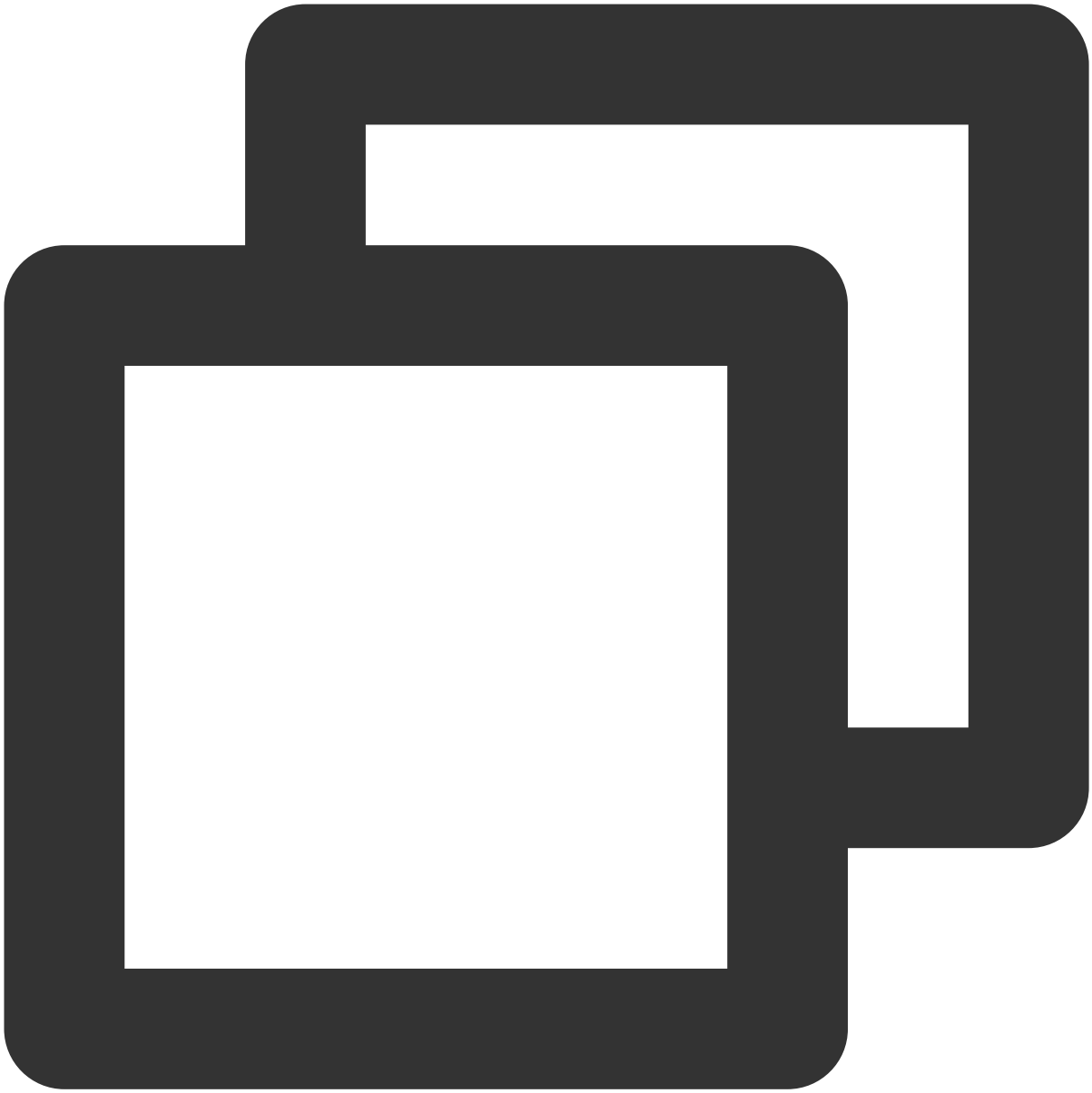
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

The mobile number already exists.



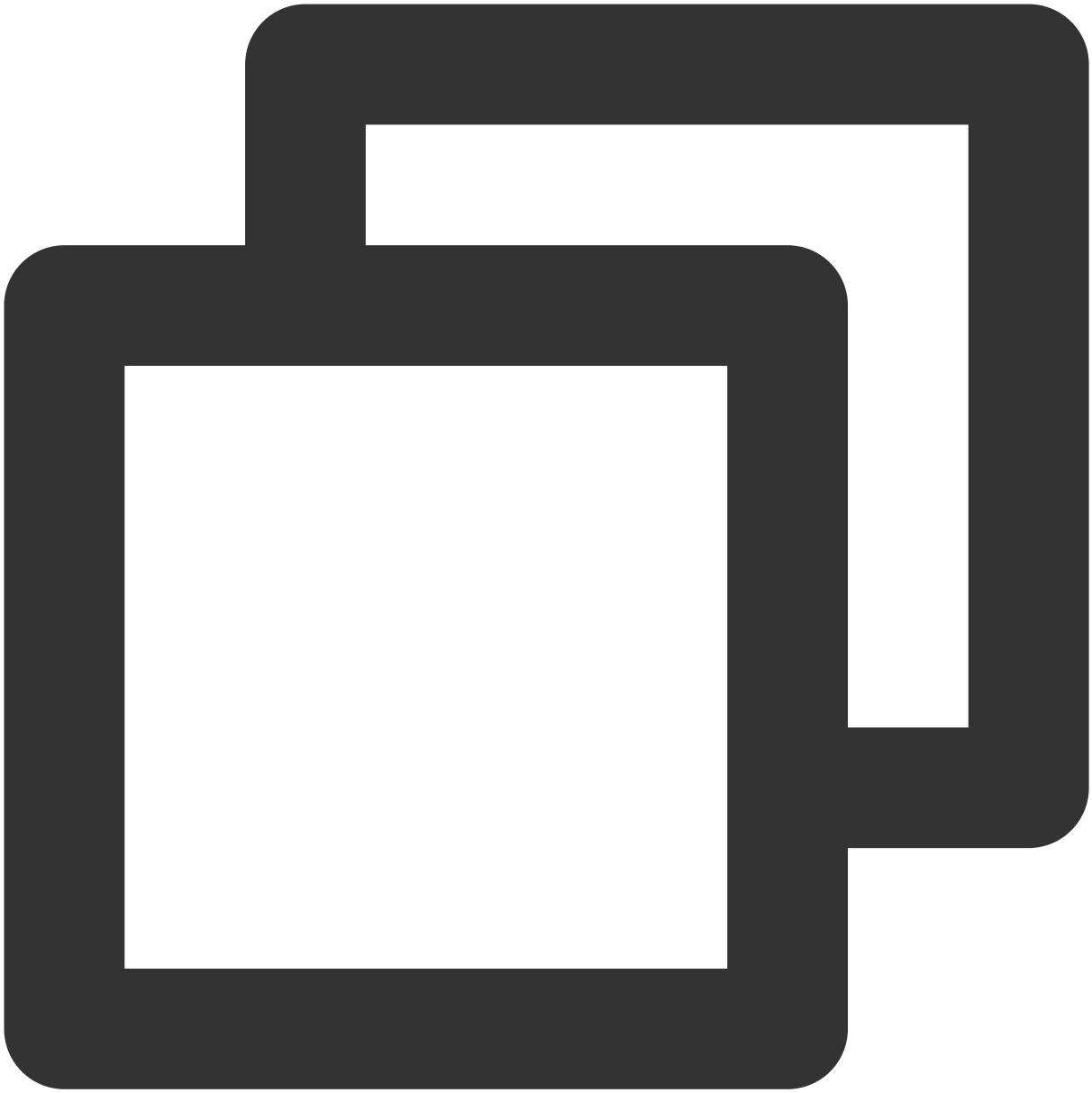
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_phone_number"
}
```

The `phone_number_otp_token` parameter is incorrect or has expired, or the parameter used for sign-up is not the same as the one for sending the verification code. For example, the mobile numbers are different.



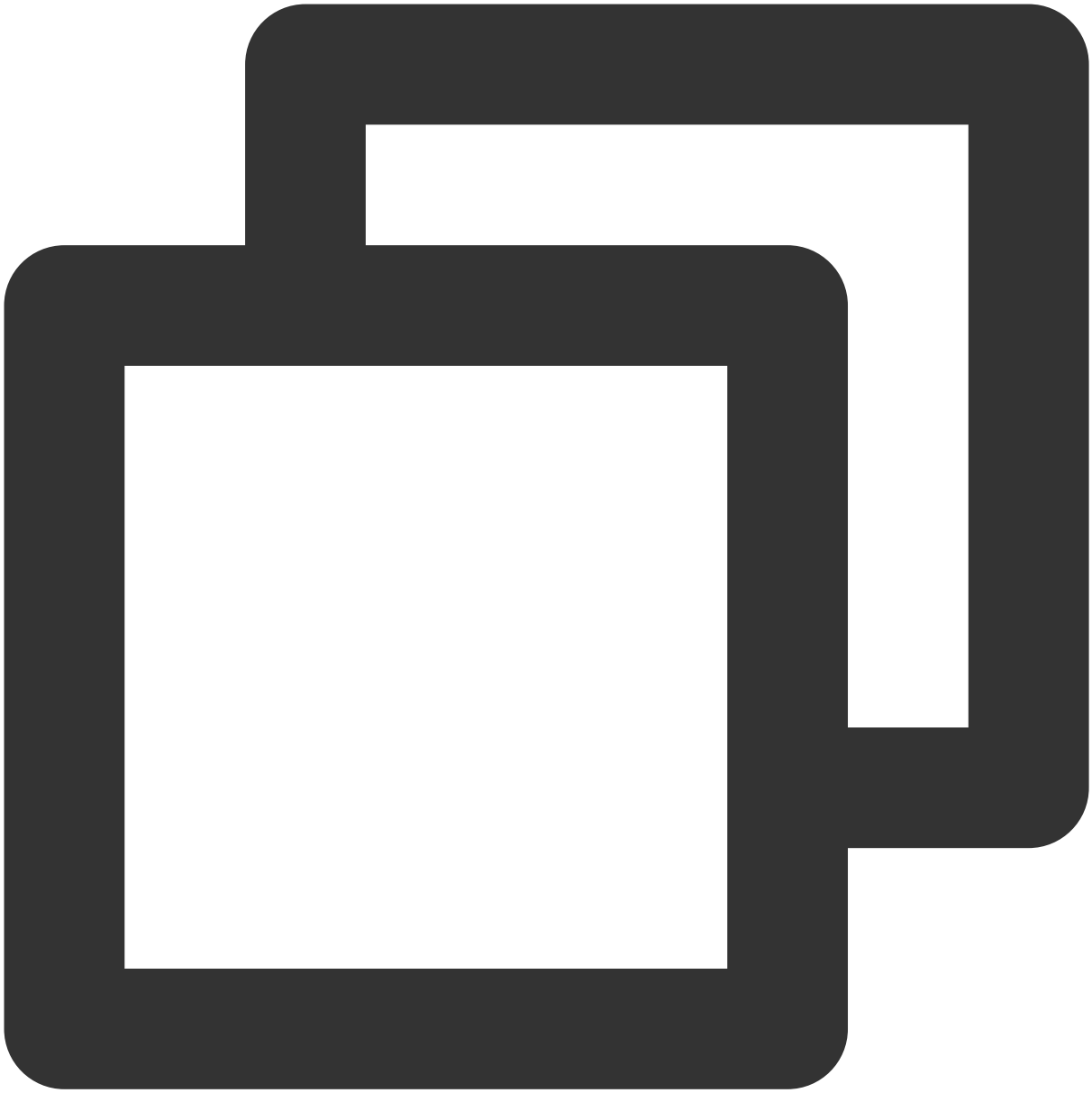
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp_token"
}
```

The `phone_number_otp` parameter is incorrect or has expired.



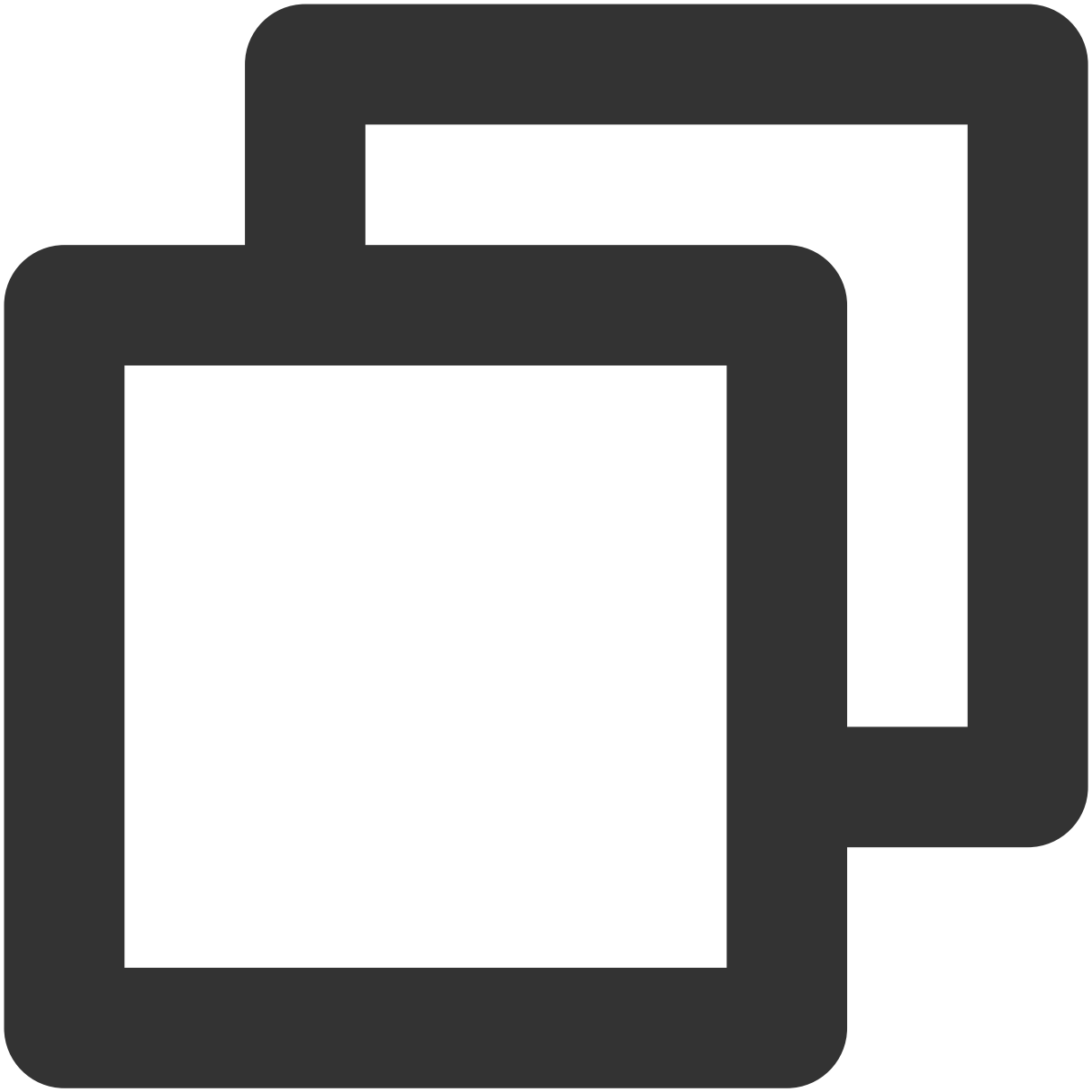
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp"
}
```

The email address format is invalid.



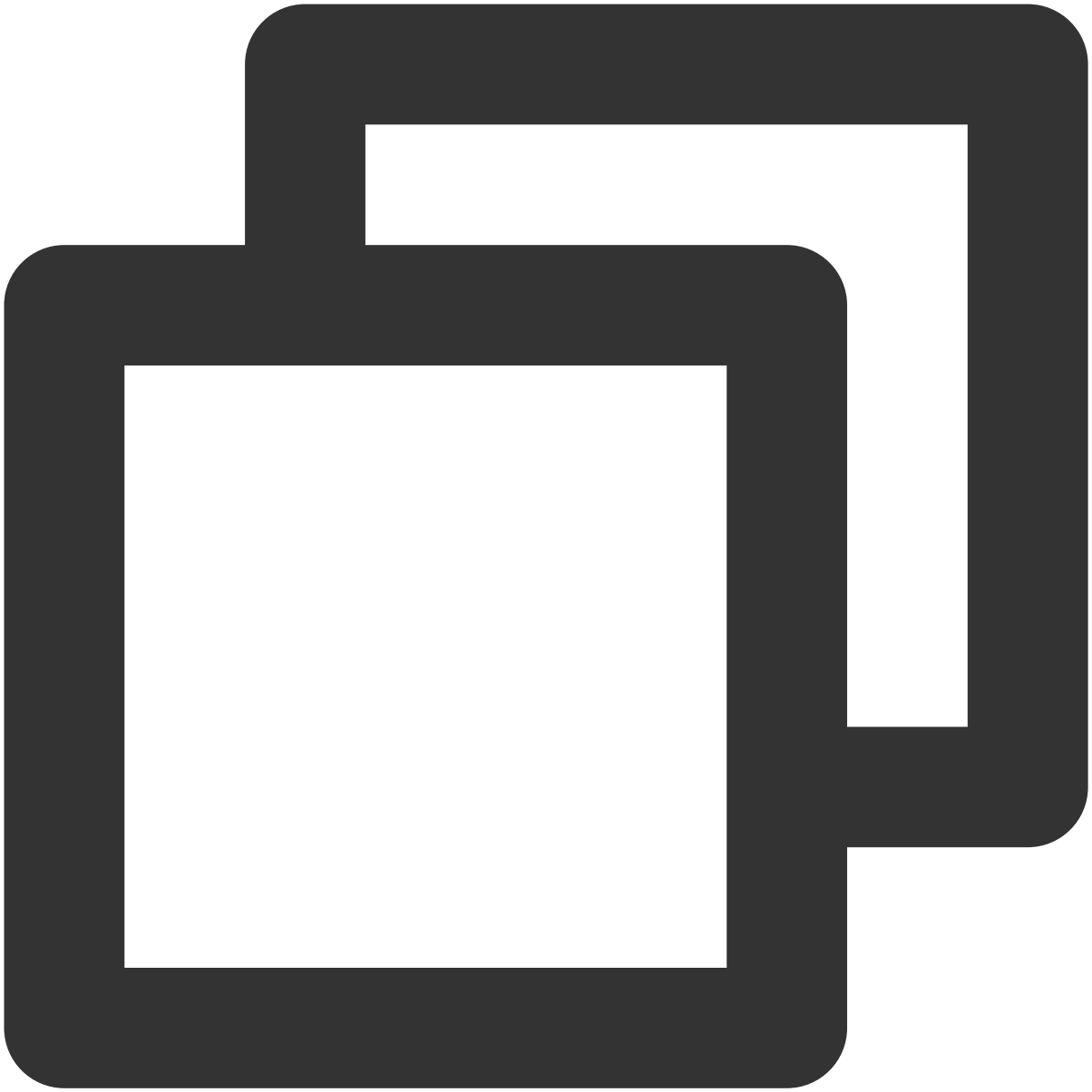
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

The email address already exists.



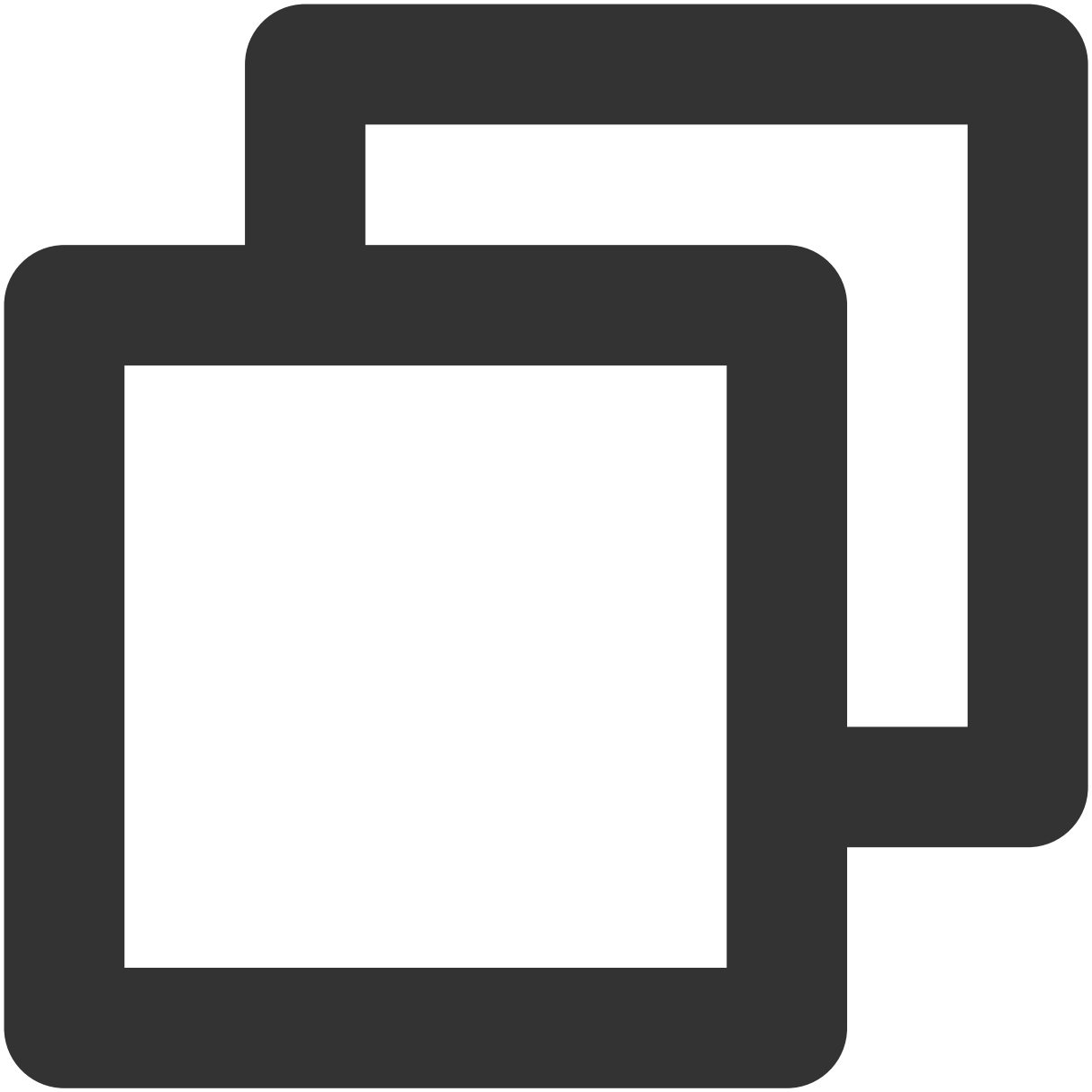
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_email"
}
```

The `email_otp_token` parameter is incorrect or has expired, or the parameter value used for sign-up is not the same as the one used for sending the verification code. For example, the email addresses are different.



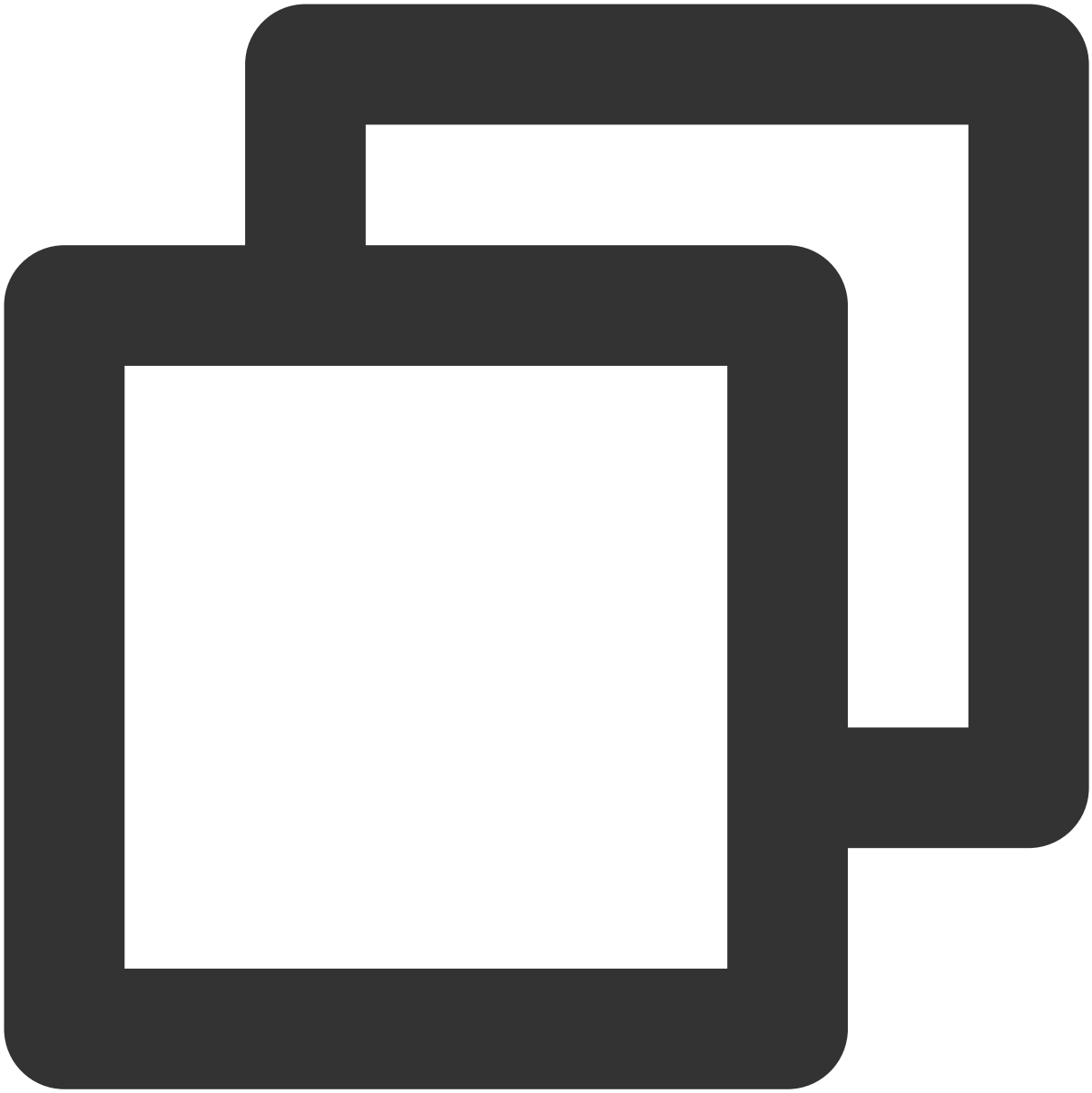
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

The `email_otp` parameter is incorrect or has expired.



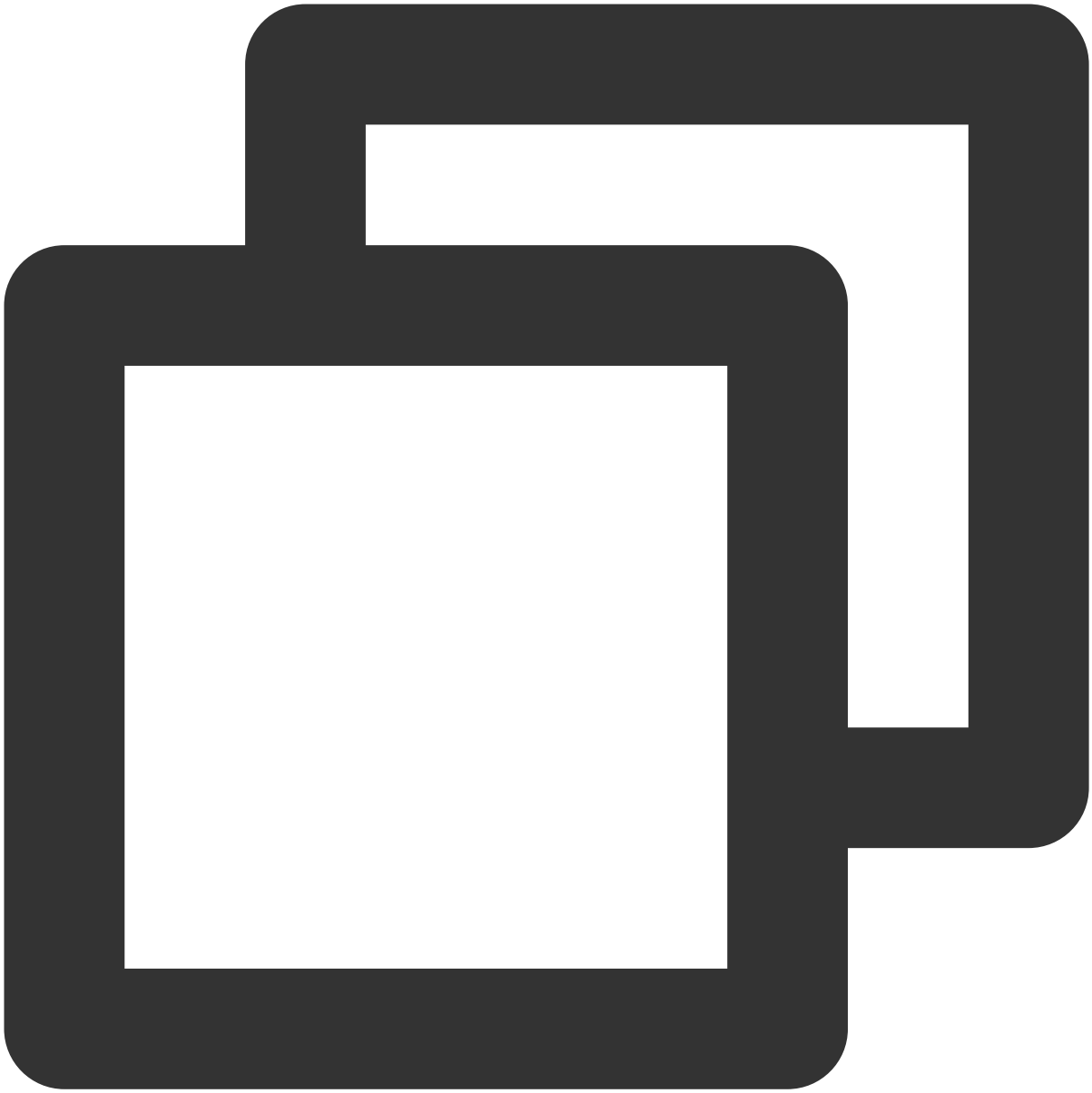
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

The input parameter value is invalid. For example, the value does not conform to the regular expression of user attributes.



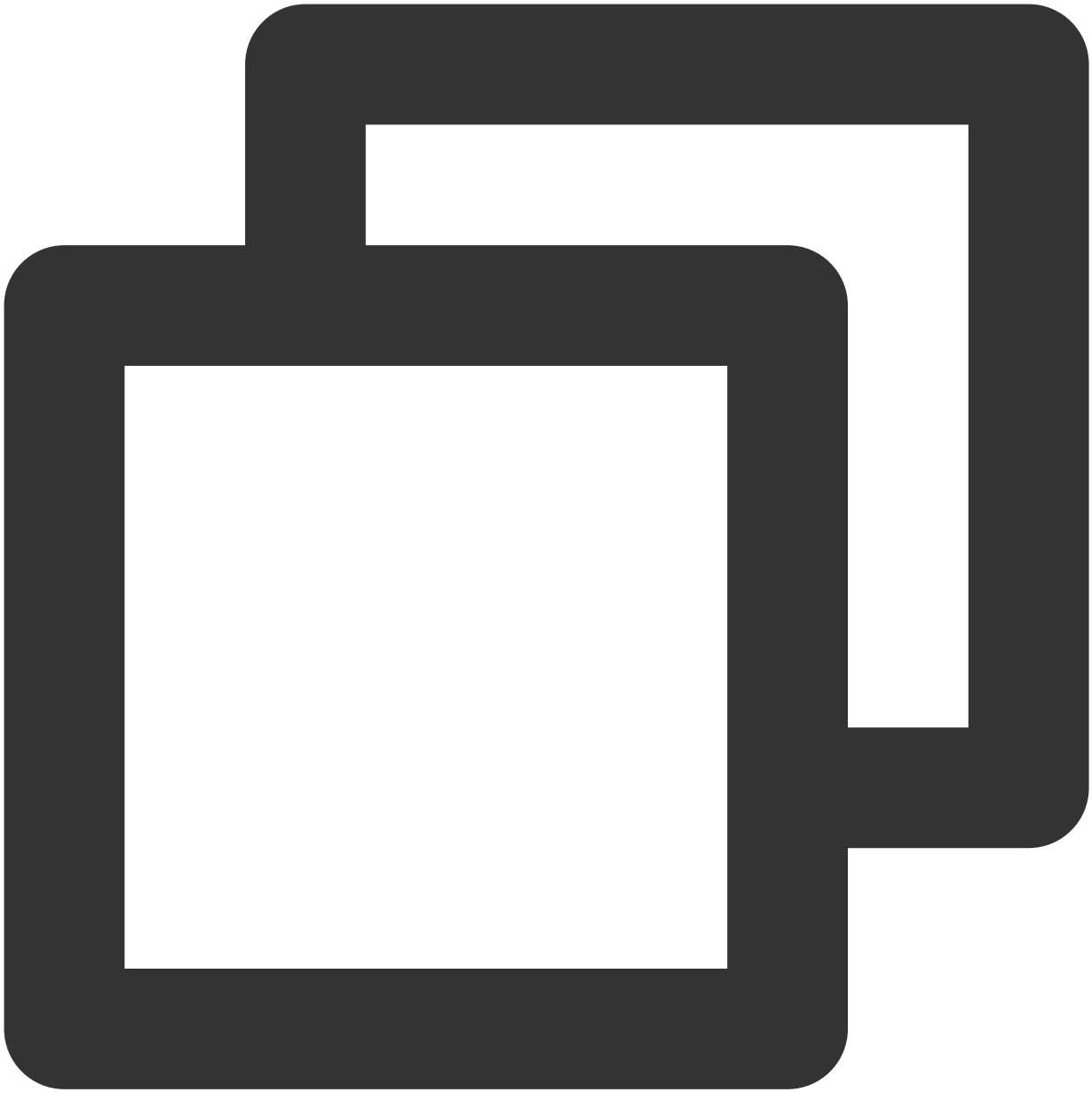
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "illegal_parameter_value"
}
```

The `bearer_token` parameter is missing.



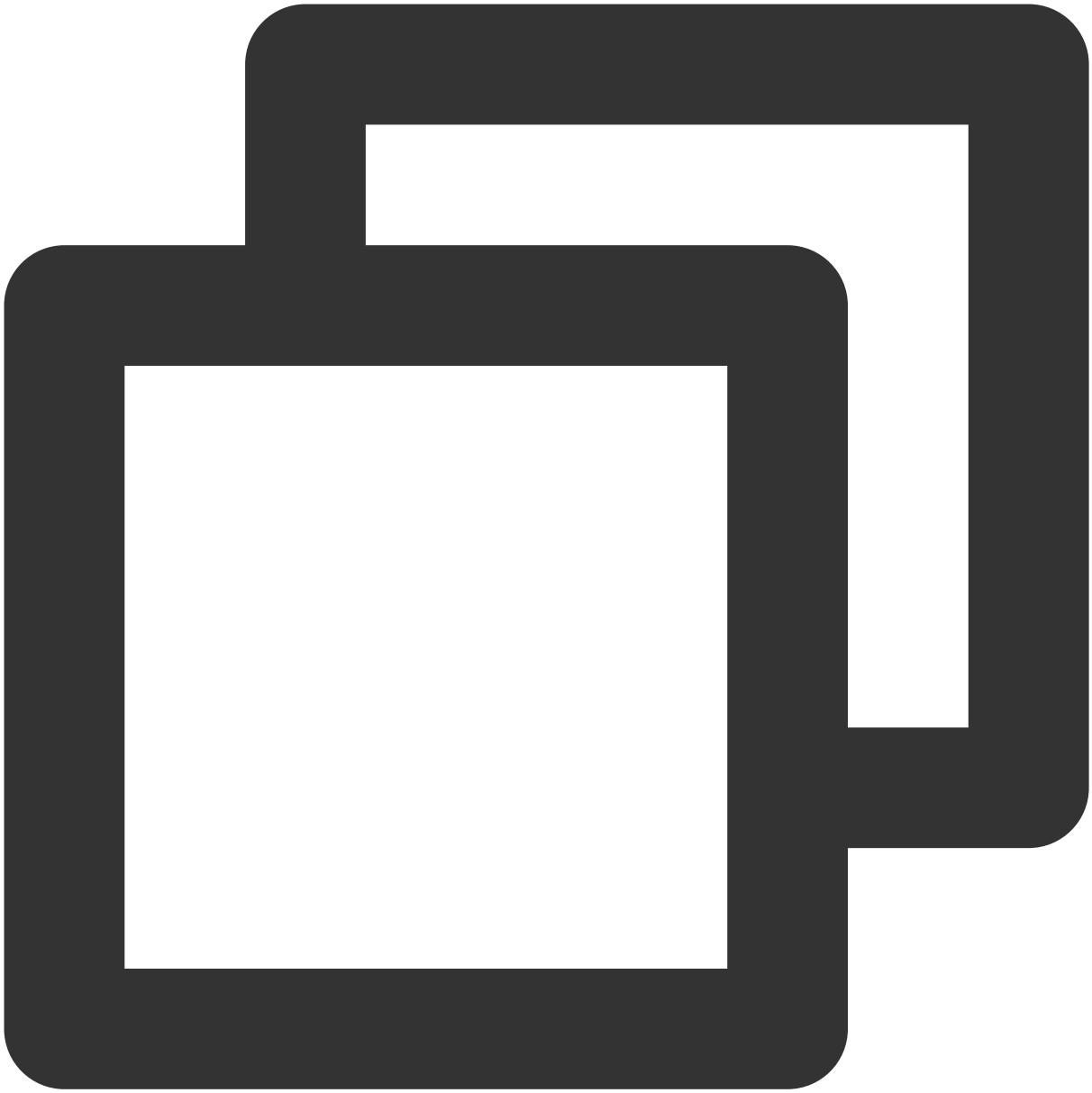
```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token n
```

The `bearer_token` parameter is incorrect.



```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding J
```

The `bearer_token` parameter is invalid.



```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request
```

Modify User Password

Last updated：2023-12-22 11:42:07

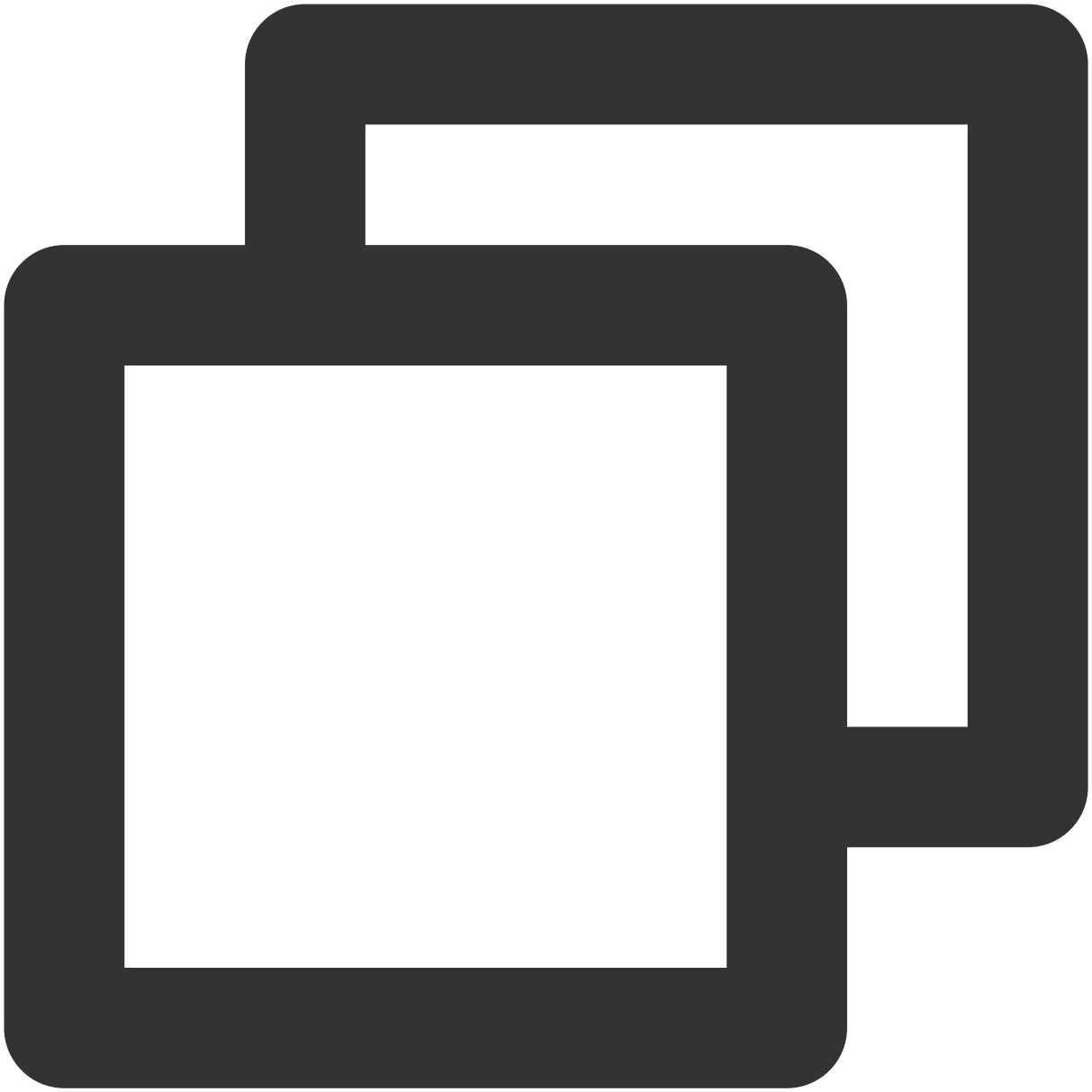
API Description

This API is used to modify the password of a logged-in user. When calling this API, you need to carry the Access Token with `openid scope` returned when the login is successful. The new password must comply with the policies of the account and password authentication source associated with the application. It cannot be the same as the previous N passwords specified in the policy.

Supported Applications

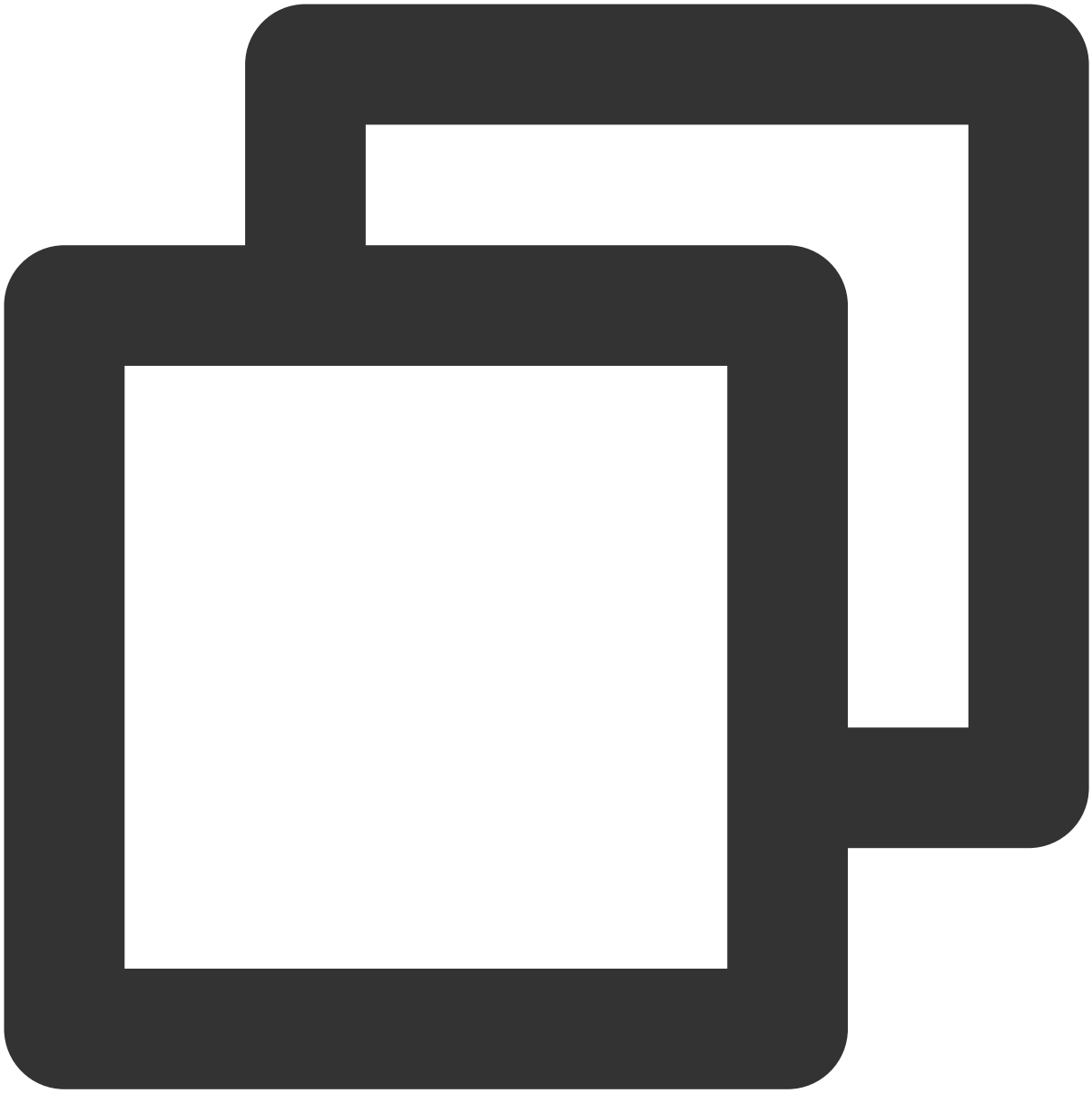
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



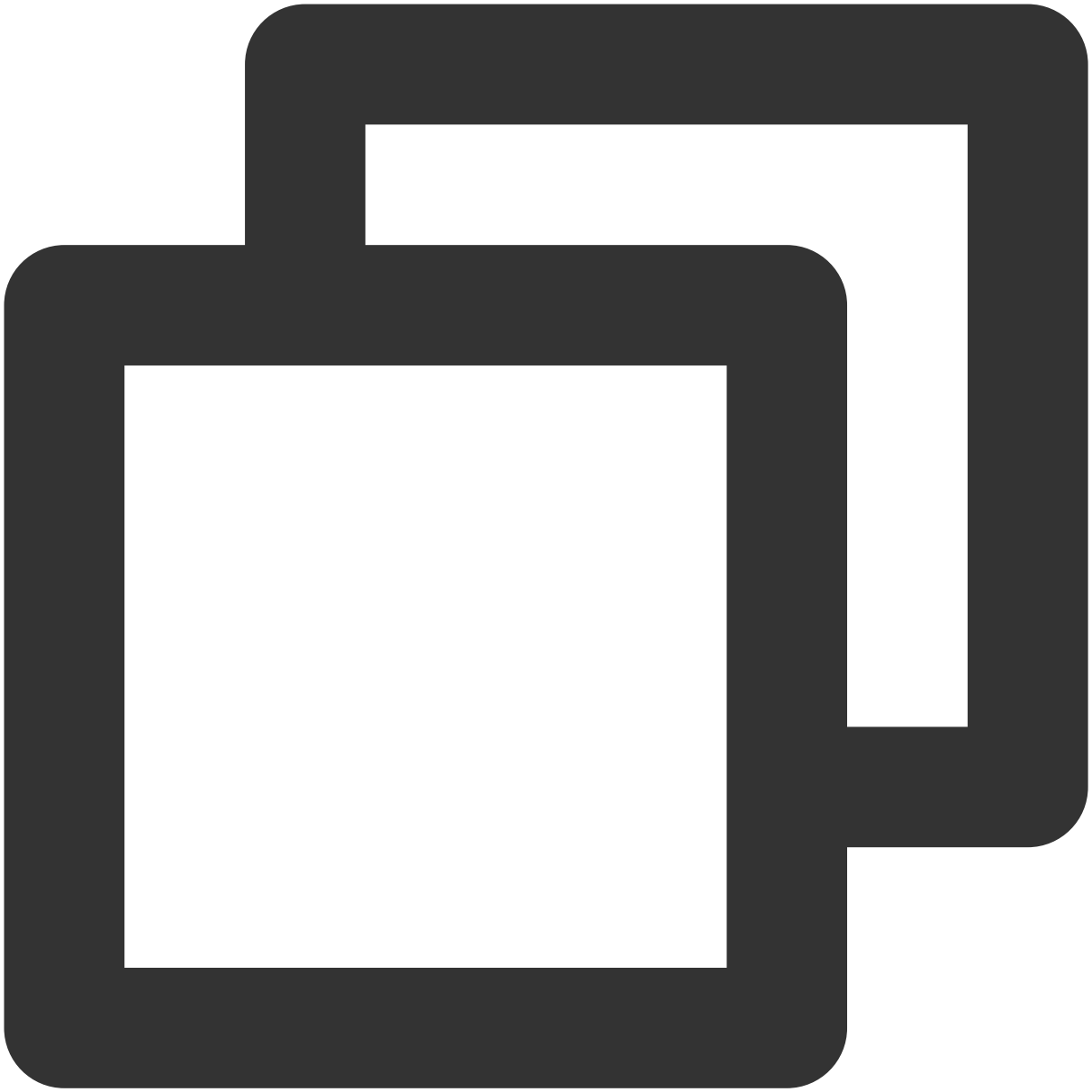
```
POST
```

Request Path



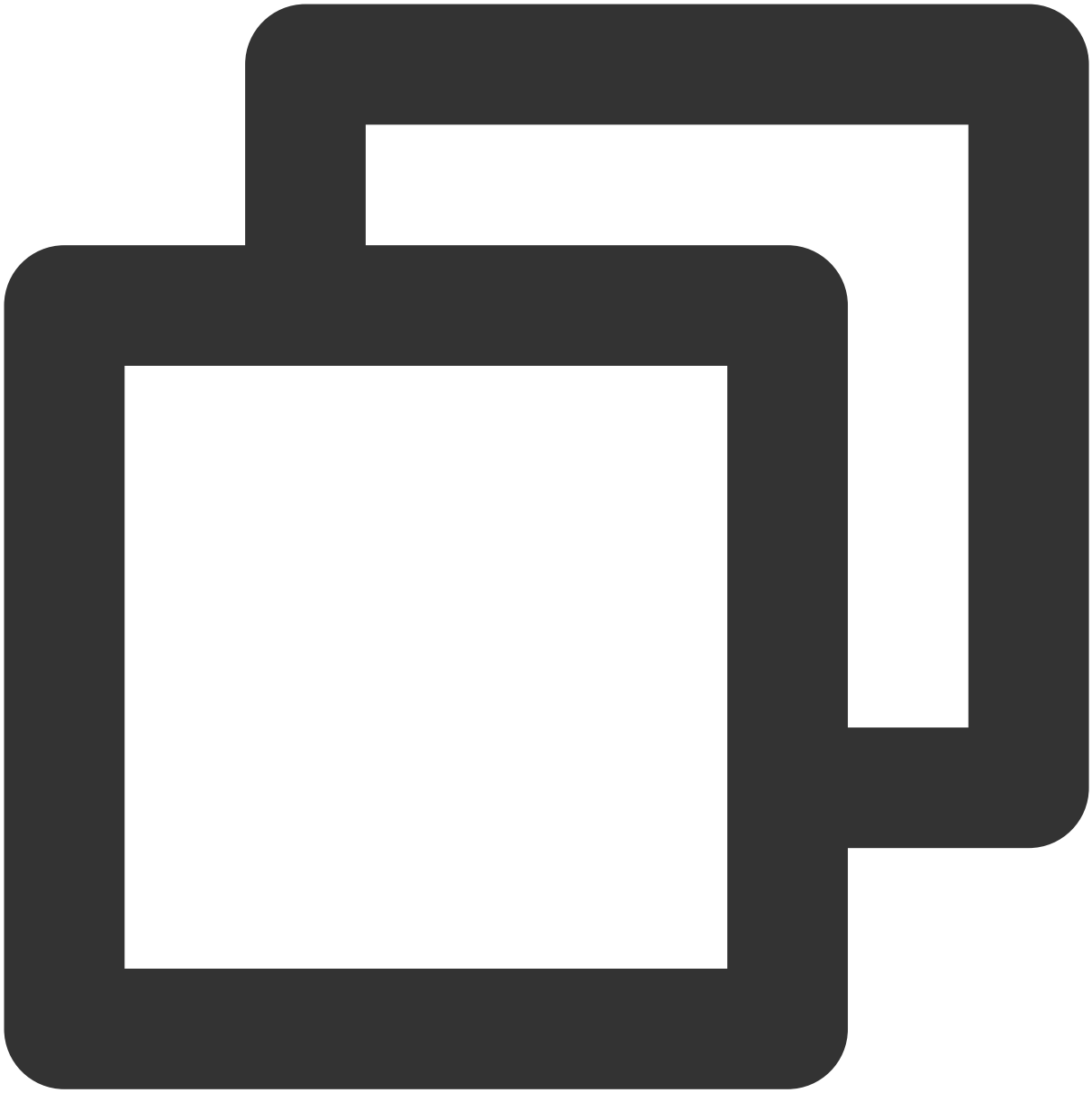
```
/change_user_password
```

Request Content-Type



```
application/json
```

Sample Requests



```
POST /change_user_password HTTP/1.1
Content-Type: application/json
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com

{
  "old_password" : "MOCK_PASSWORD",
  "new_password" : "MOCK_NEW_PASSWORD"
}
```

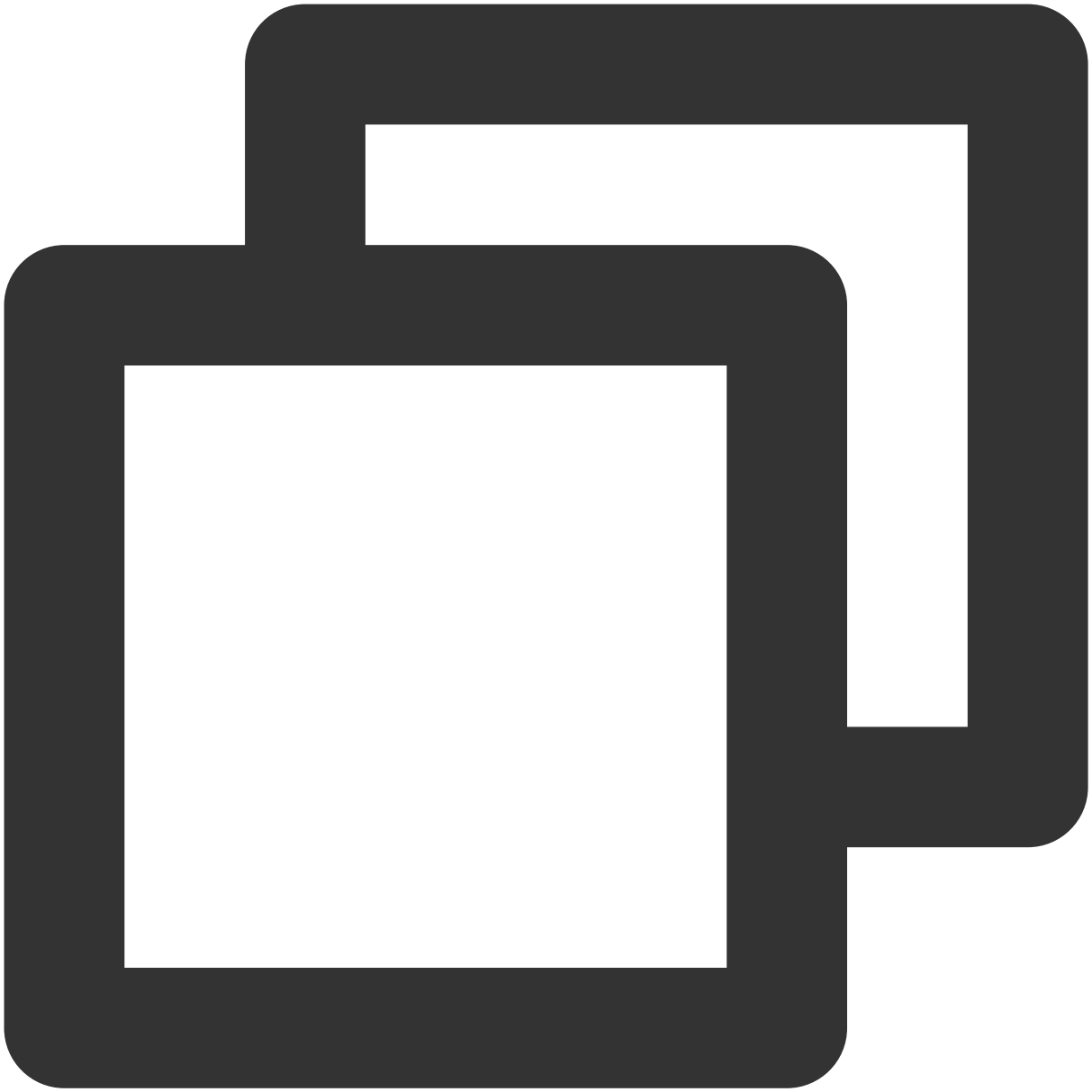
Request Headers

Parameter	Description
Authorization	OAuth 2.0 Bearer Token. The format is `Bearer <Token>>` , where `Bearer` is a fixed string and `<Token>` is the Access Token with `openid scope` returned when the login is successful. `Bearer` and `<Token>` are separated by a space.

Request Parameters in JSON Format

JSON Path	Data Type	Description
old_password	String	Old password.
new_password	String	New password.

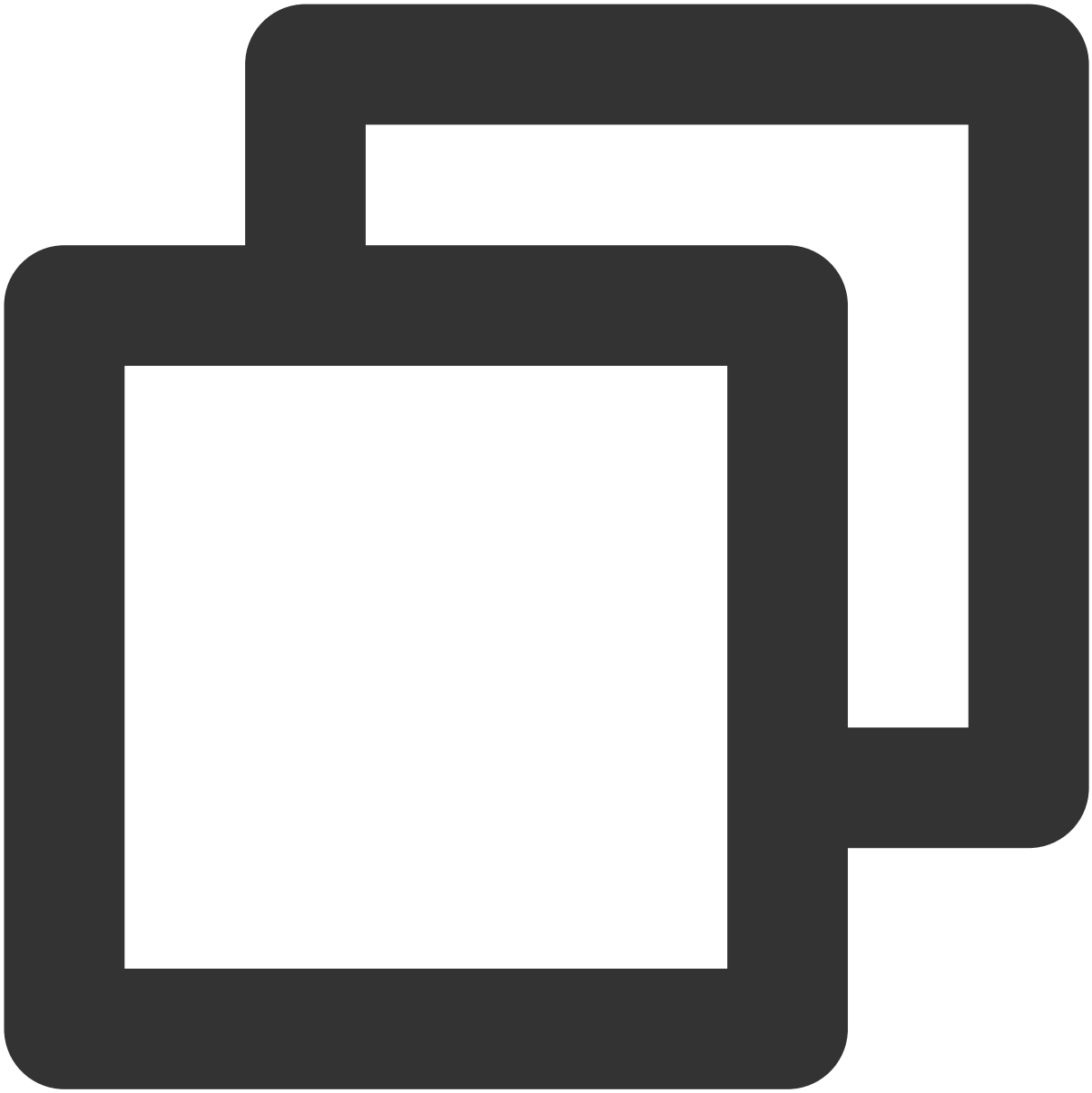
Sample Success Responses



```
HTTP/1.1 200 OK
```

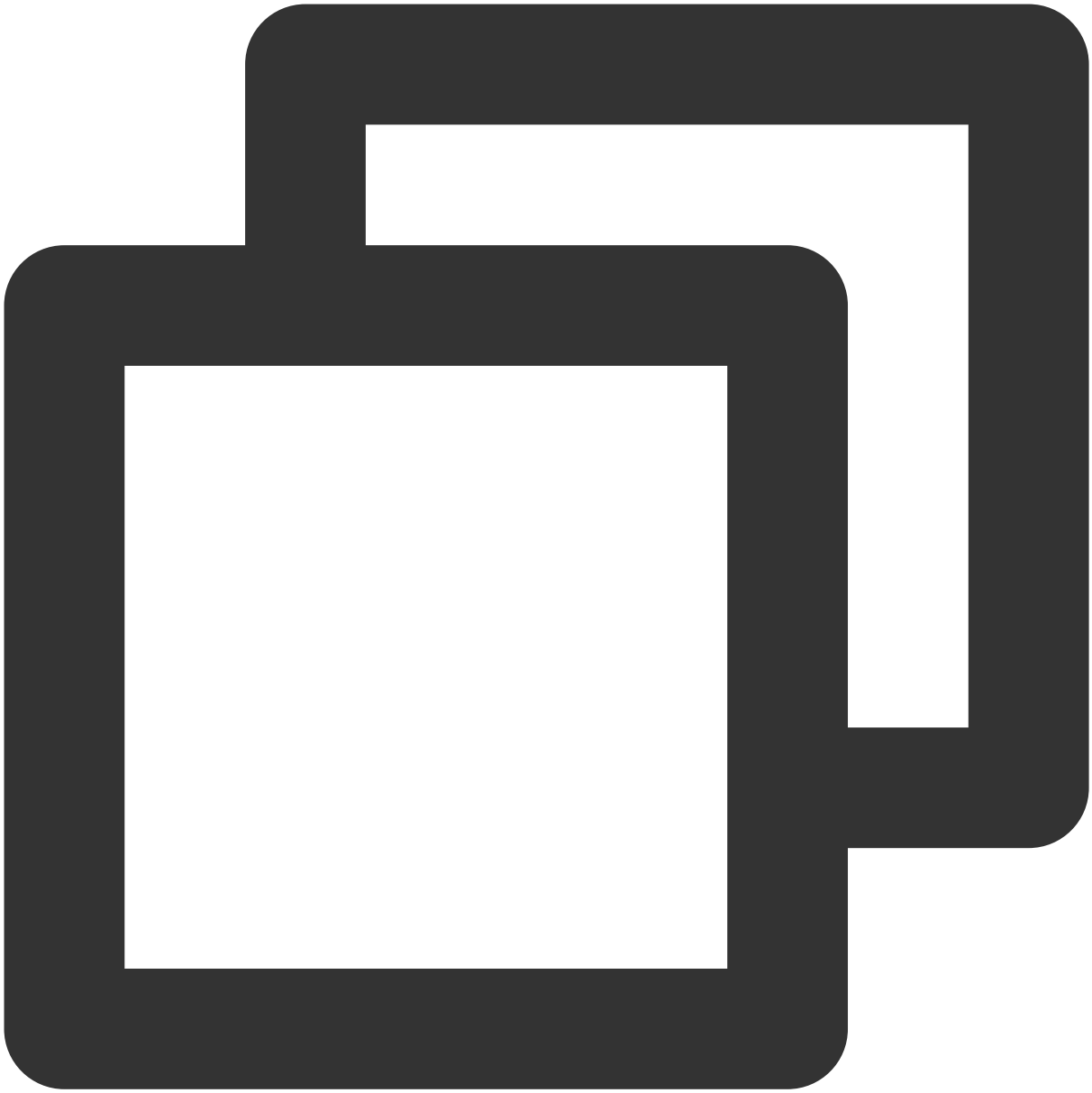
Sample Error Responses

The old password is incorrect.



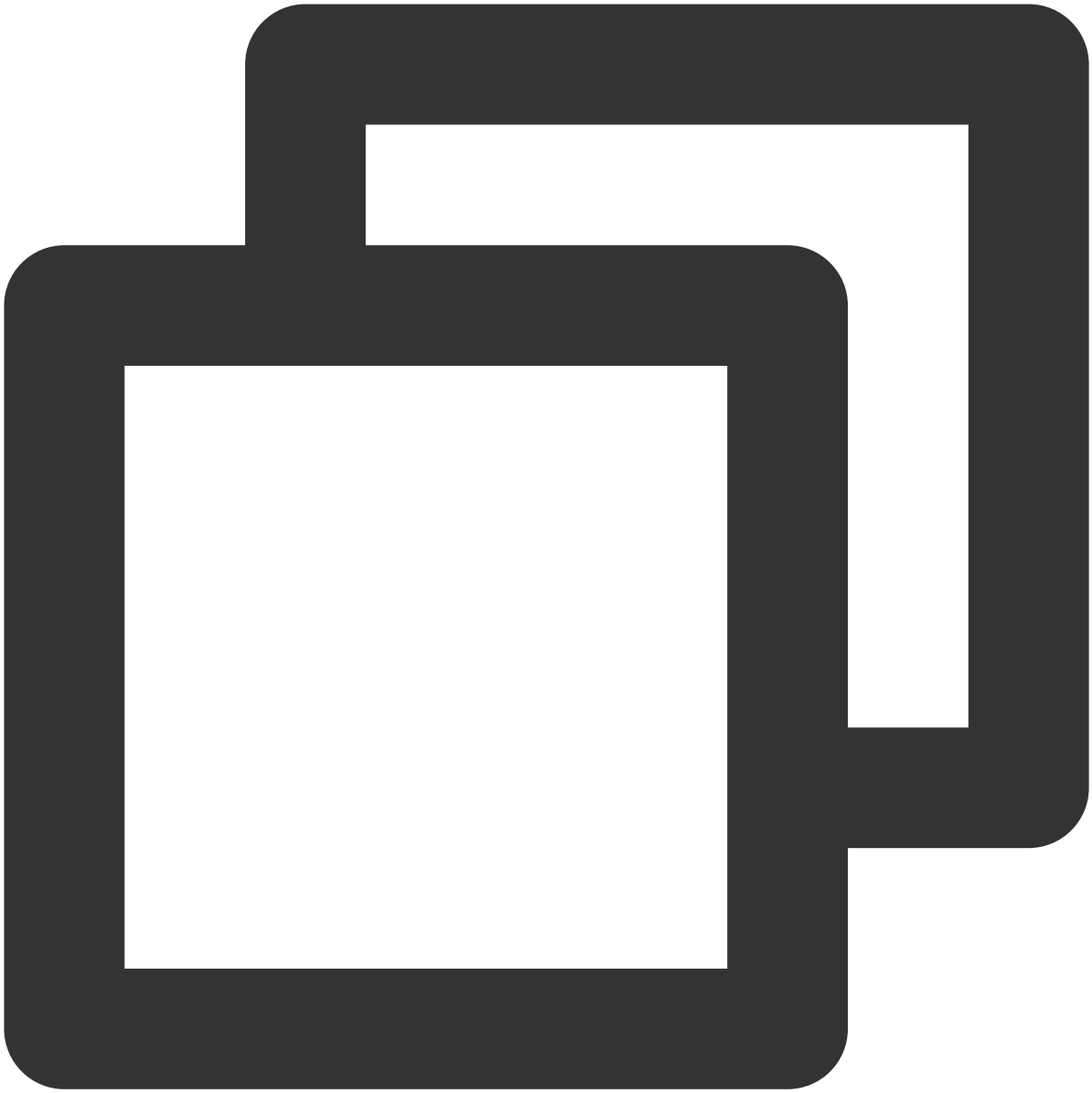
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "wrong_old_password"
}
```

The new password is the same as the old one.



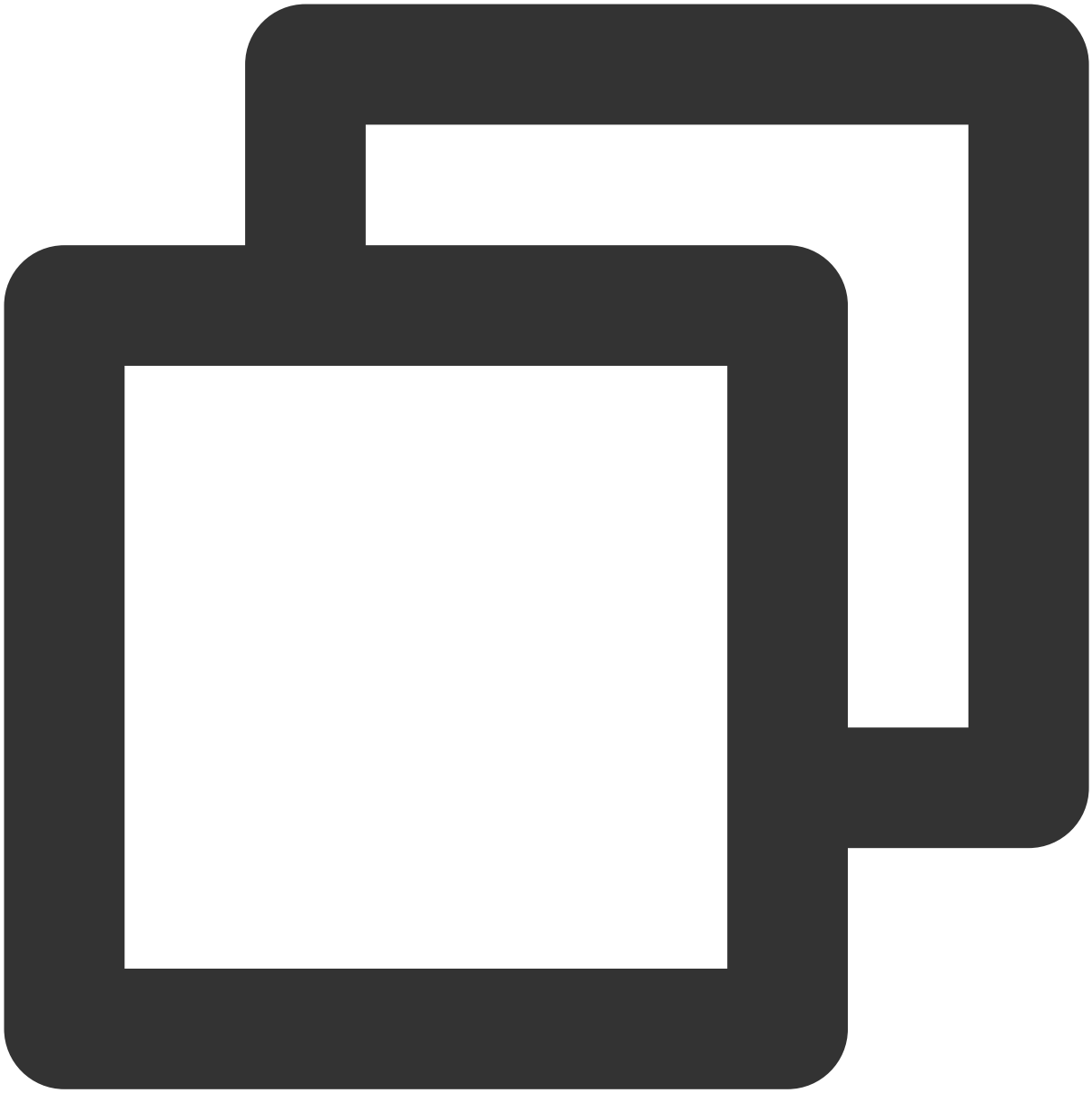
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_password"
}
```

The new password is the same as a previous password.



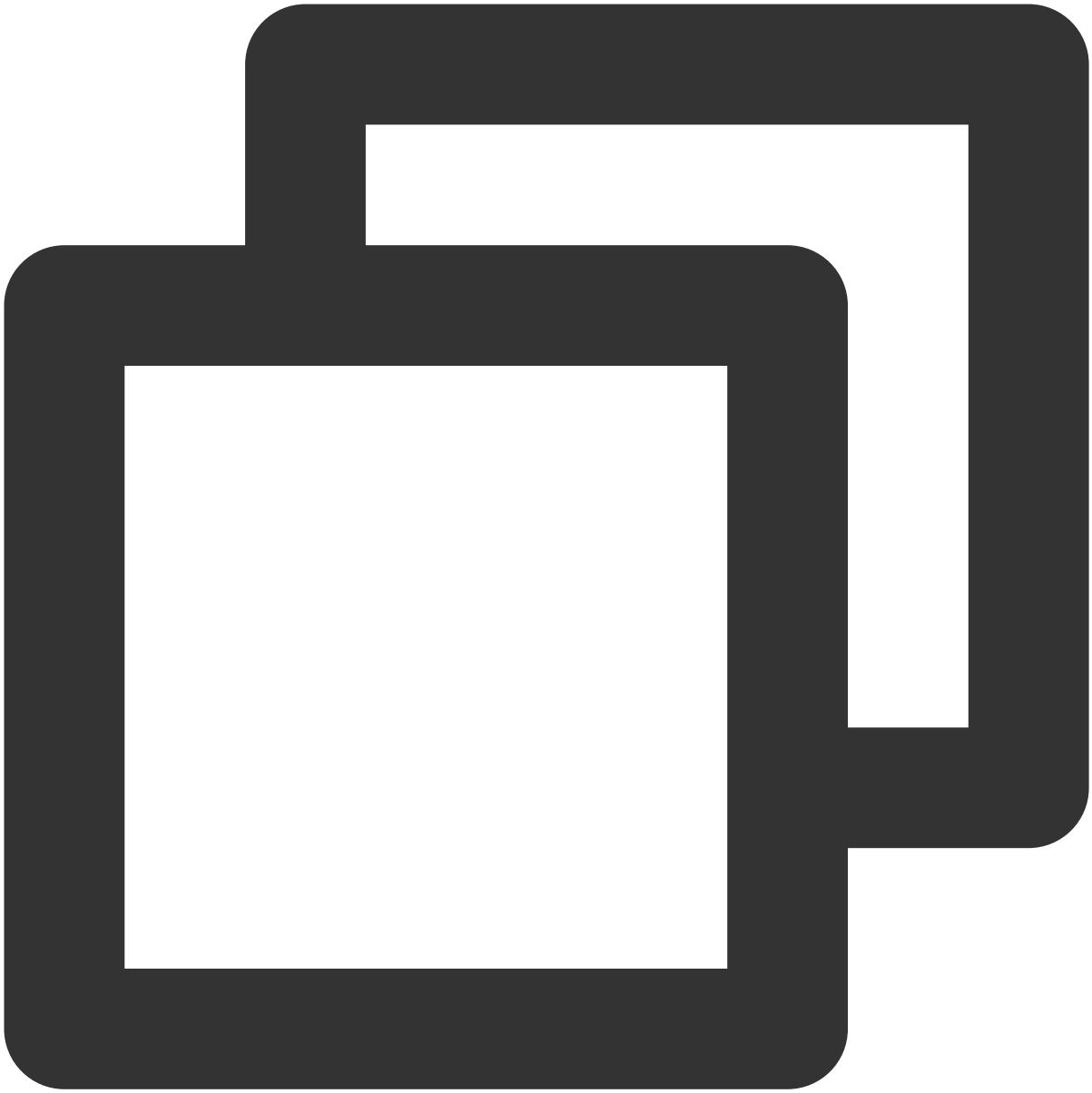
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "recurrent_password"
}
```

The new password does not comply with the password policy.



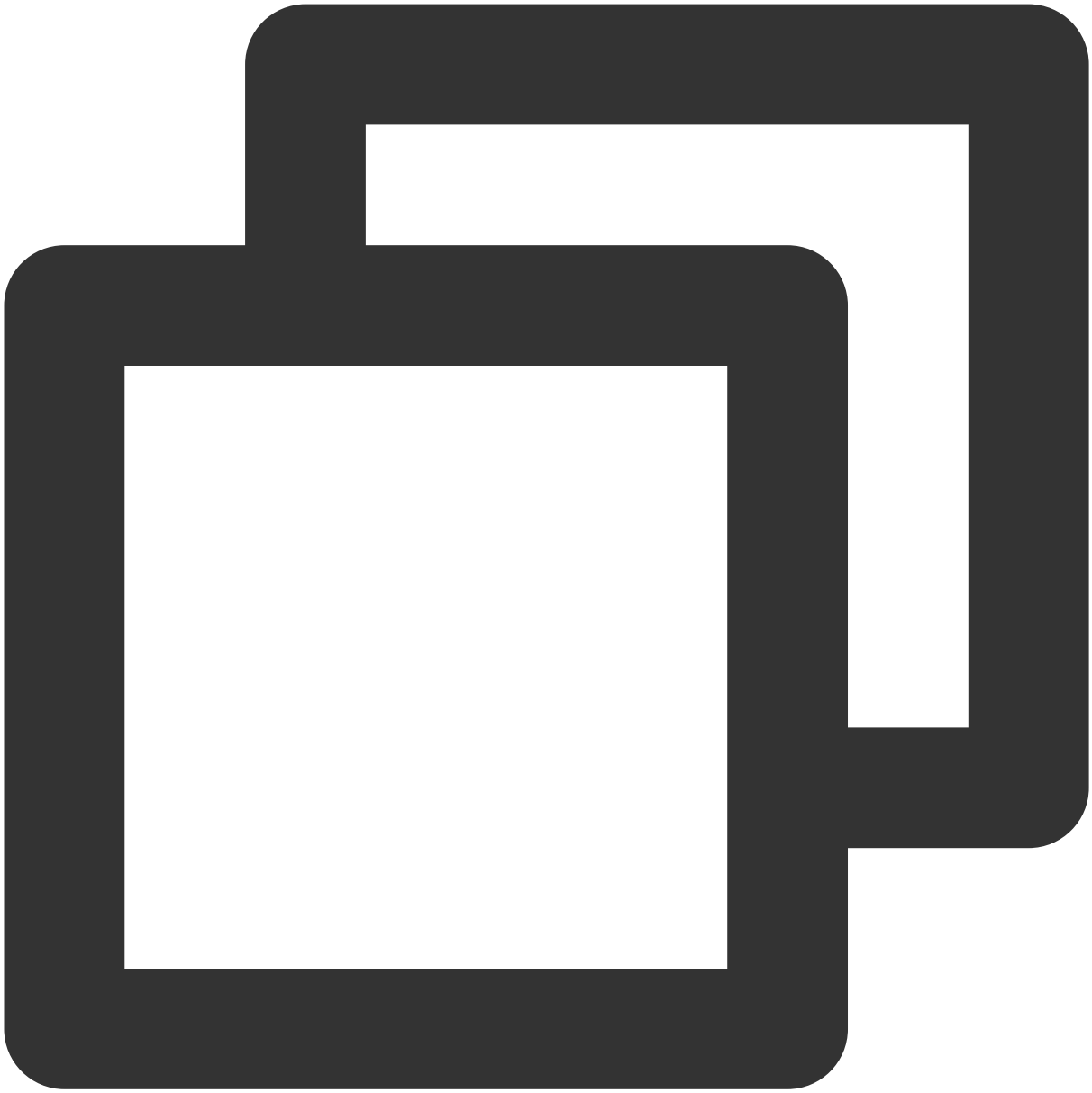
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_new_password"
}
```

User not found.



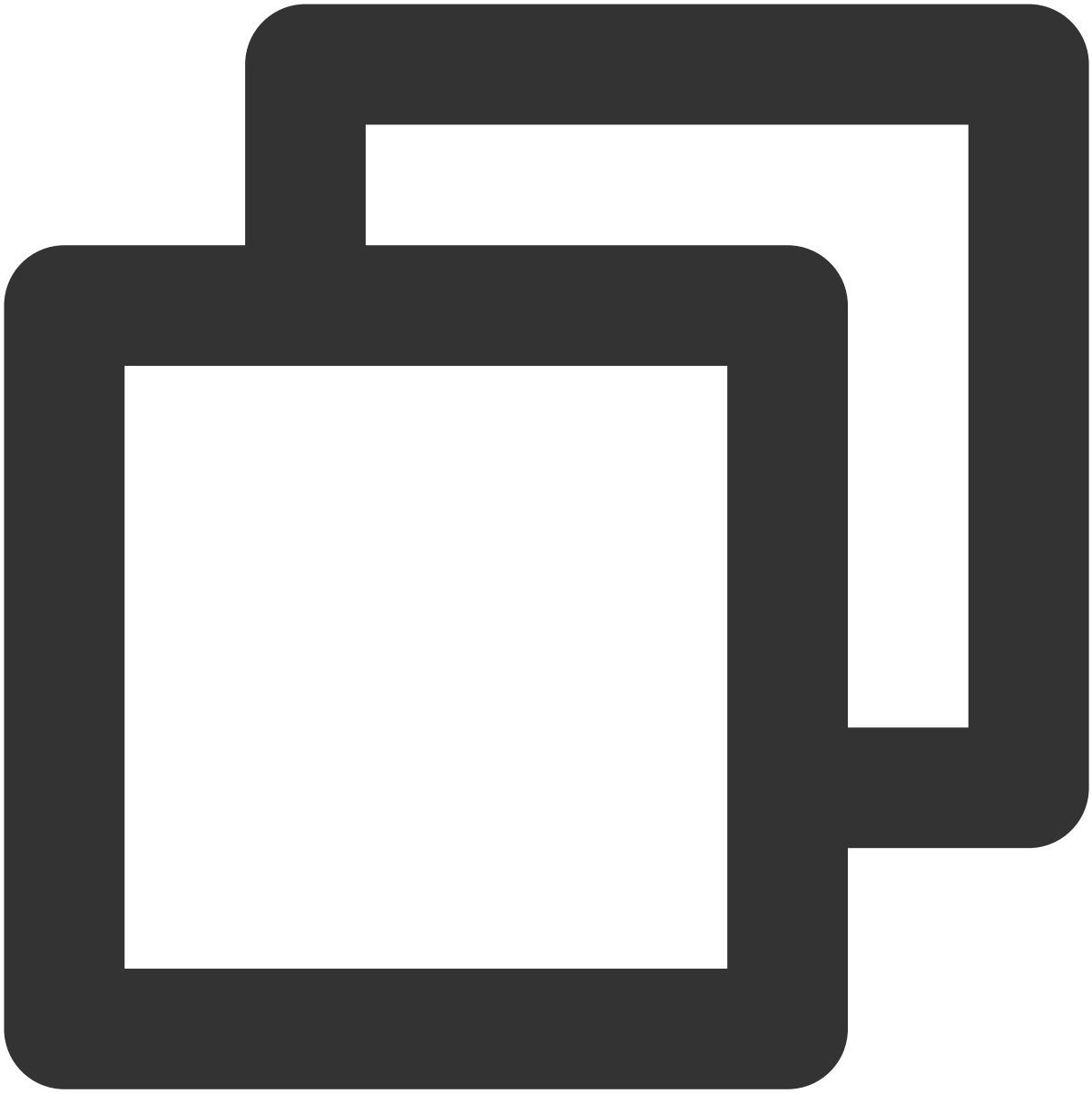
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "user_not_found"
}
```

The user account is frozen.



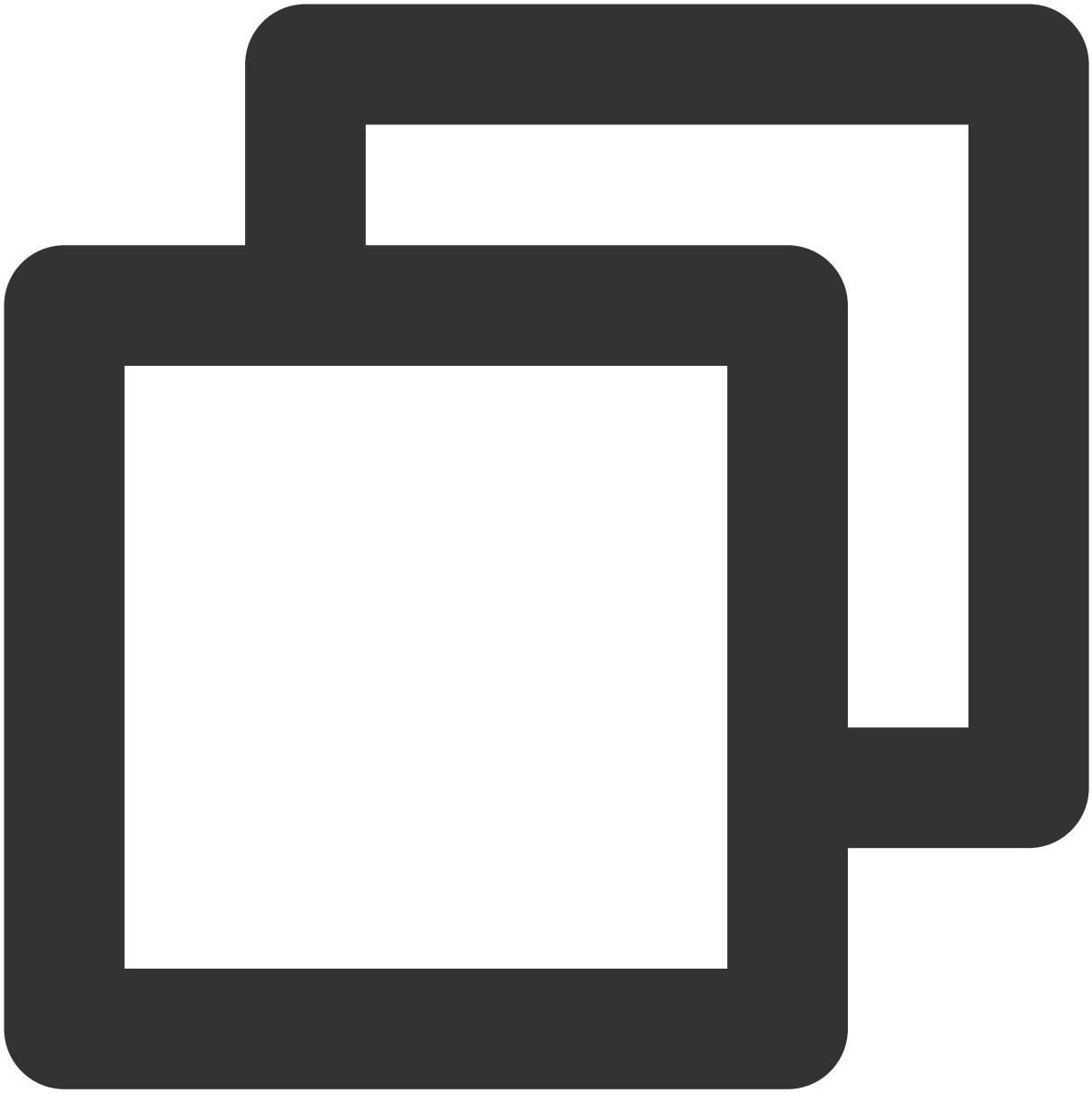
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is frozen."
}
```

The user account is locked.



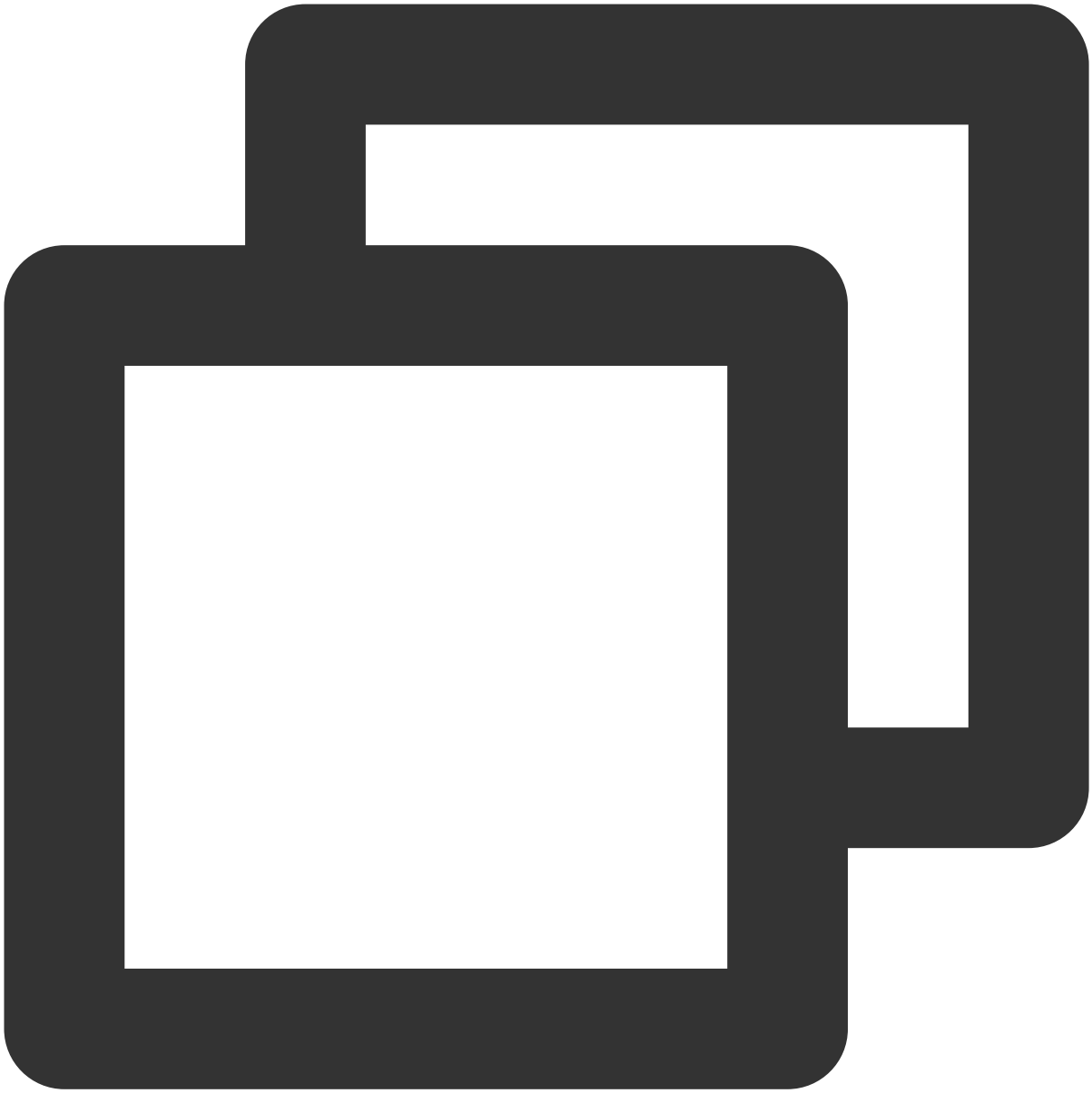
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is locked."
}
```

The `bearer_token` parameter is missing.



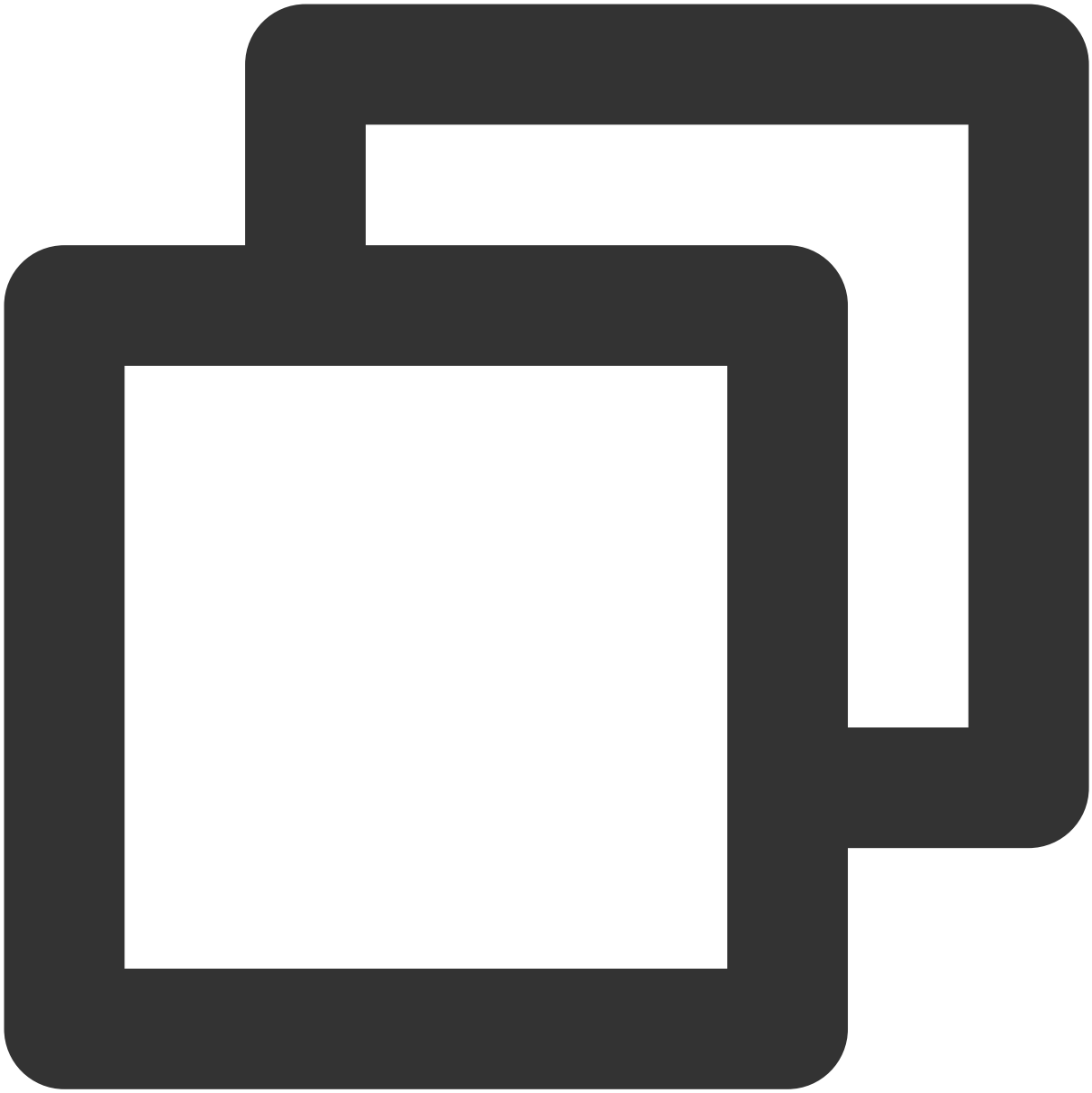
```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token n
```

The `bearer_token` parameter is incorrect.



```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding J
```

The `bearer_token` parameter is invalid.



```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request
```

Reset User Password

Last updated：2023-12-22 11:42:07

API Description

This API is used to reset the user password. Before calling this API, you need to call the API for sending one-time password (OTP) verification code to send a verification code to user.

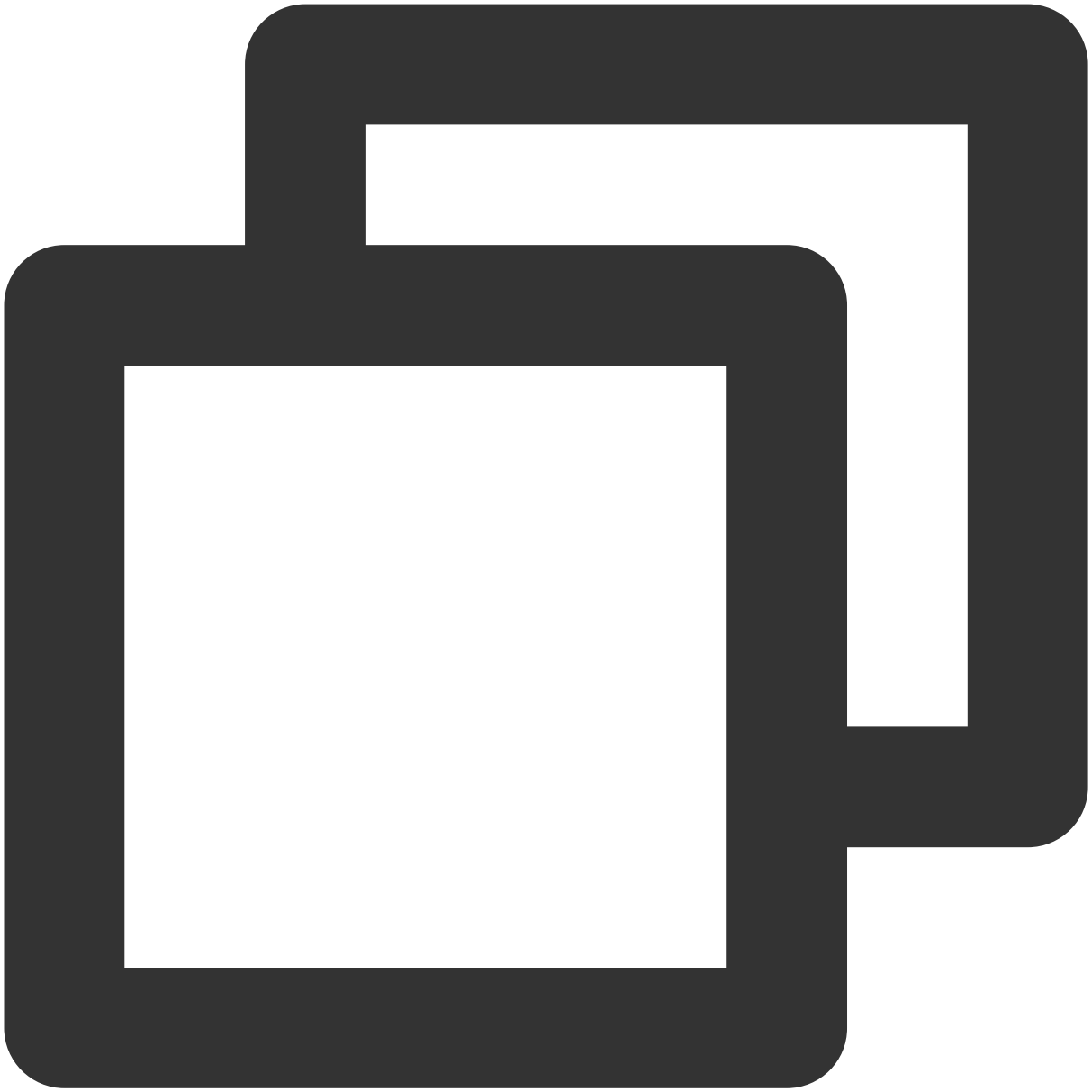
Note:

The new password must comply with the policies of the account and password authentication source associated with the application. It cannot be the same as the previous N passwords specified in the policy.

Supported Applications

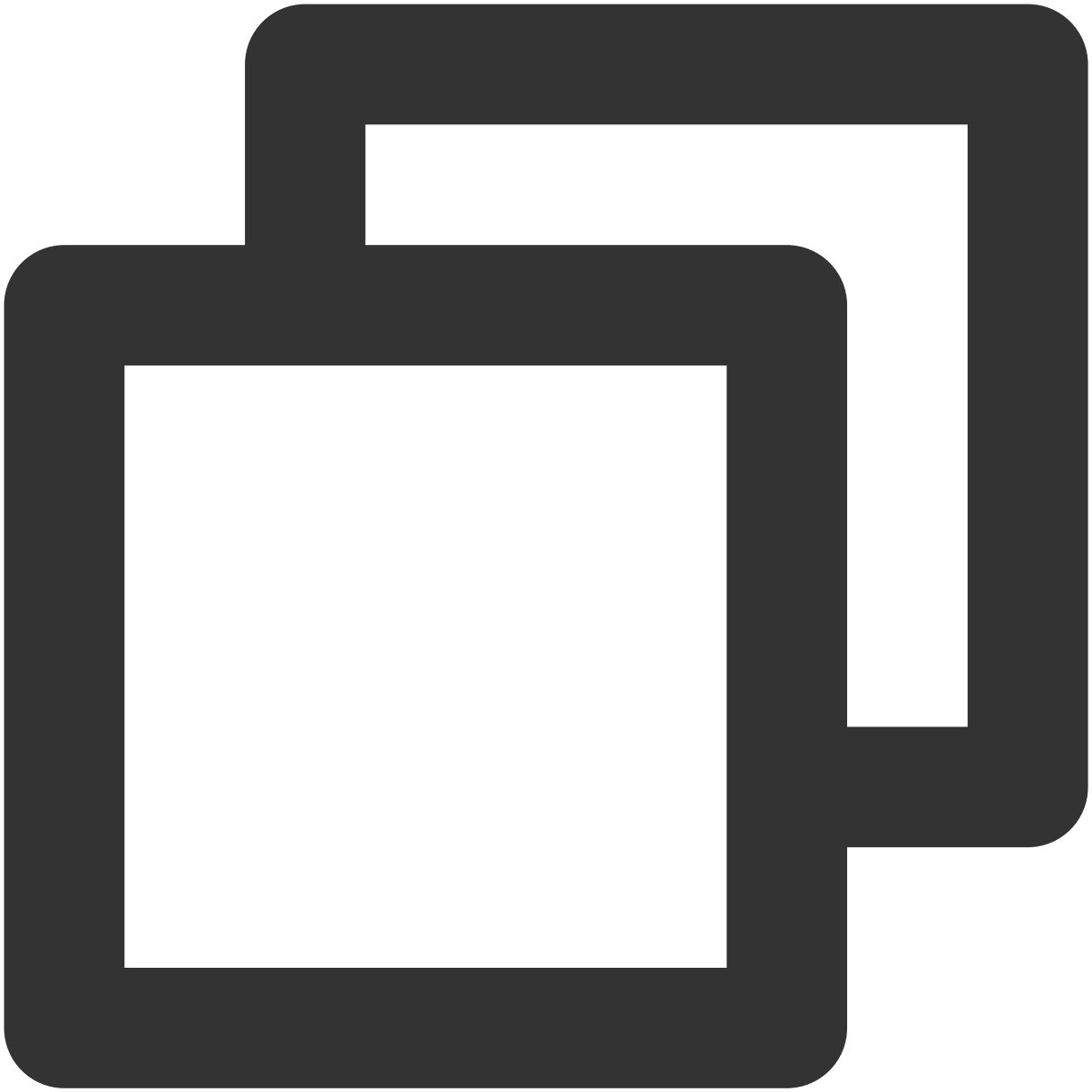
Web applications, single-page applications (SPA), and mobile applications.

Request Method



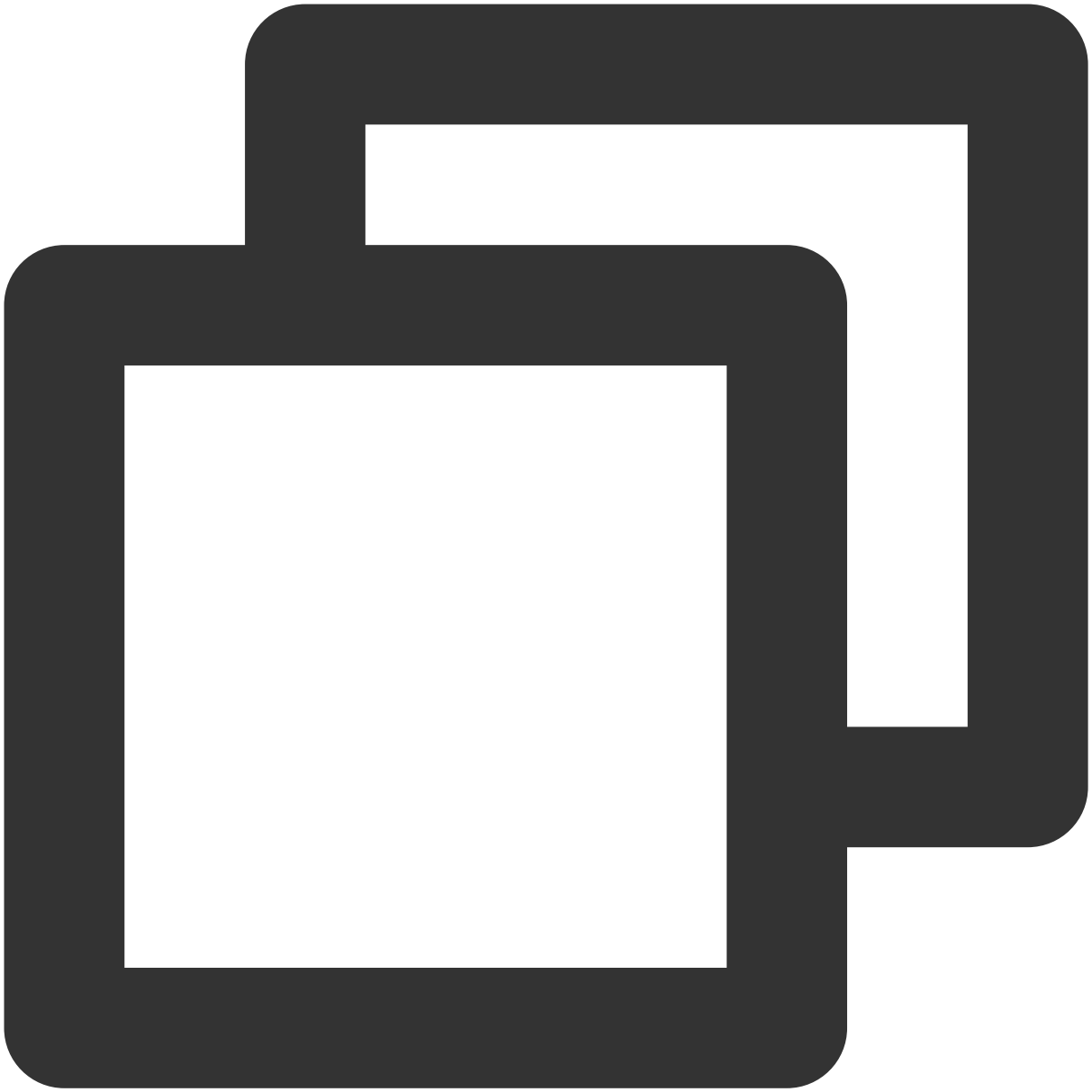
```
POST
```

Request Path



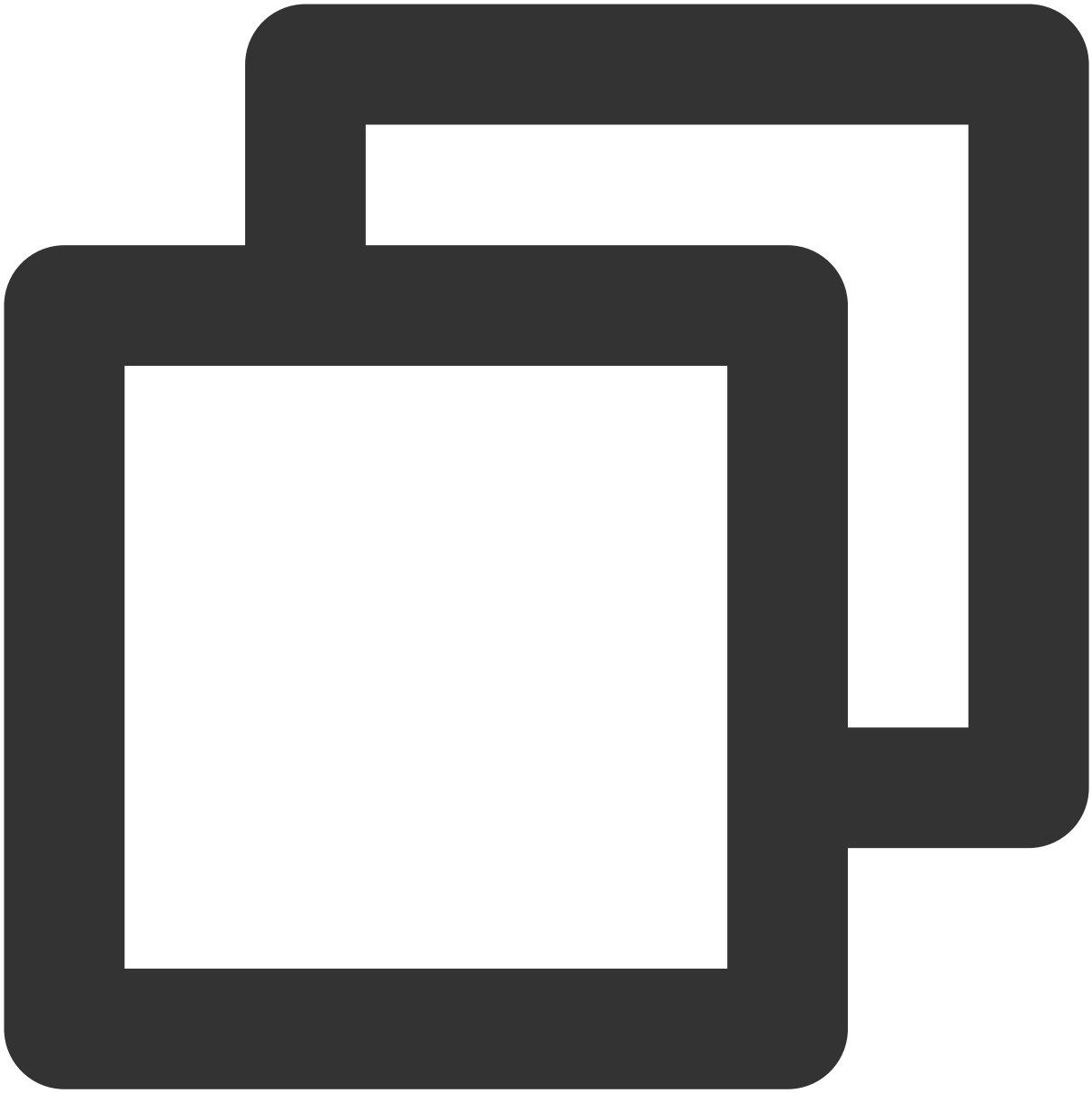
```
/reset_user_password
```

Request Content-Type



```
application/json
```

Sample Requests



```
POST /reset_user_password HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
    "password" : "MOCK_PASSWORD",
    "email" : "MOCK_EMAIL@163.com",
    "email_otp" : "MOCK_EMAIL_OTP",
    "email_otp_token" : "MOCK_EMAIL_OTP_TOKEN"
}
```

Request Headers

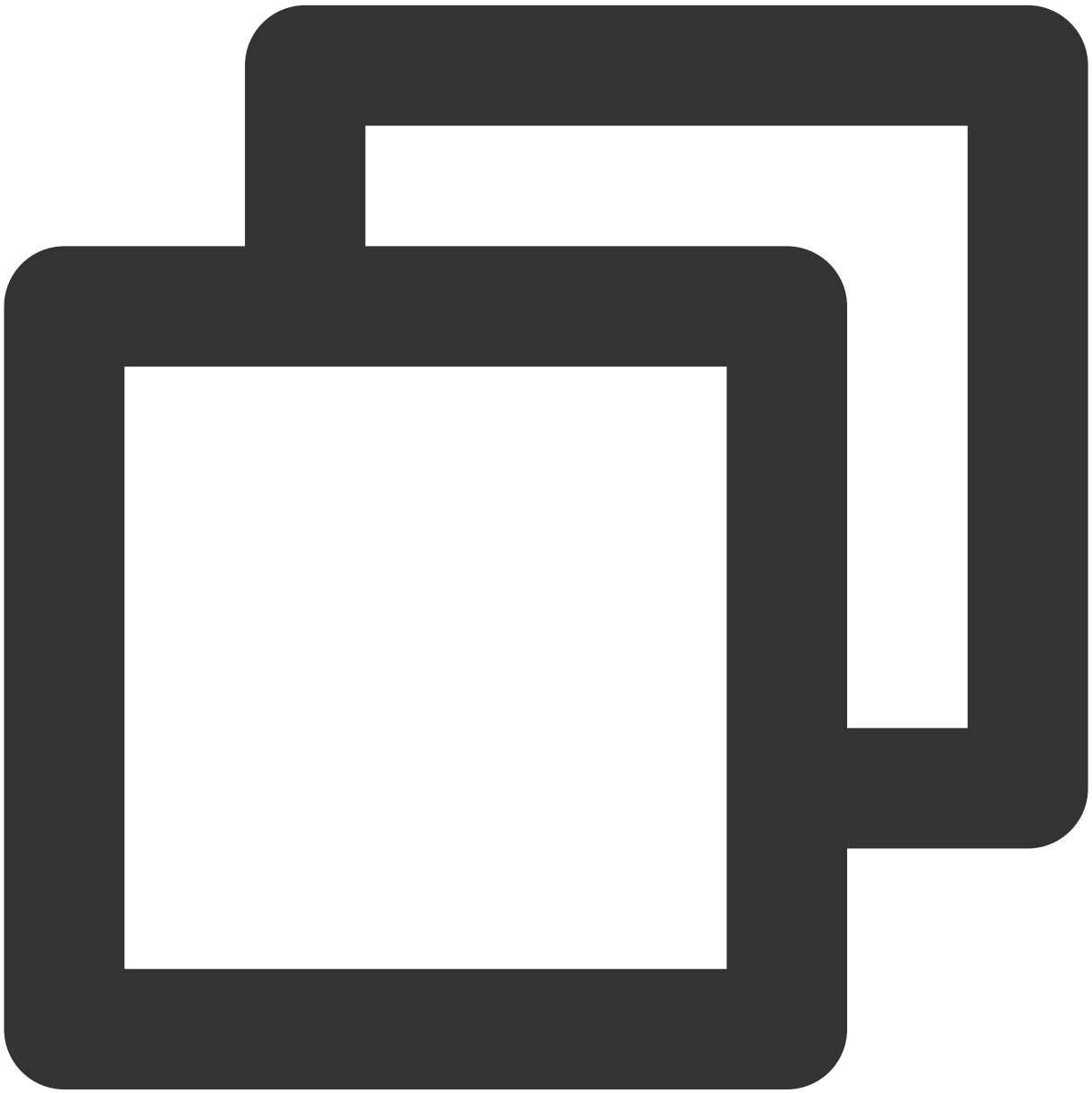
Name	Description
Authorization	HTTP Basic authentication request header. The format is `Basic <credentials>`, where `Basic` is a fixed string and `<credentials>` is calculated by `base64(url_encode(client_id) + ":" + url_encode(client_secret))`. `Basic` and `<credentials>` are separated by a space.

Request Parameters in JSON Format

JSON Path	Data Type	Description
client_id	String	The client_id of the application. This should be the same as that used for sending verification code.
client_secret	String	The client_secret of the application. This parameter is required for web applications, yet it is not needed for SPA and mobile applications.
password	String	New password.
email	String	The user's email address. This parameter is required for sending email OTP verification code.
email_otp_token	String	The otp_token returned by the server after the email verification code is sent.
email_otp	String	The OTP verification code received by the user's email.
phone_number	String	The user's mobile number. This parameter is required for sending SMS OTP verification code.
phone_number_otp_token	String	The otp_token returned by the server after the SMS verification code is sent.
phone_number_otp	String	The OTP verification code received by the user's mobile phone.

Sample Success Responses

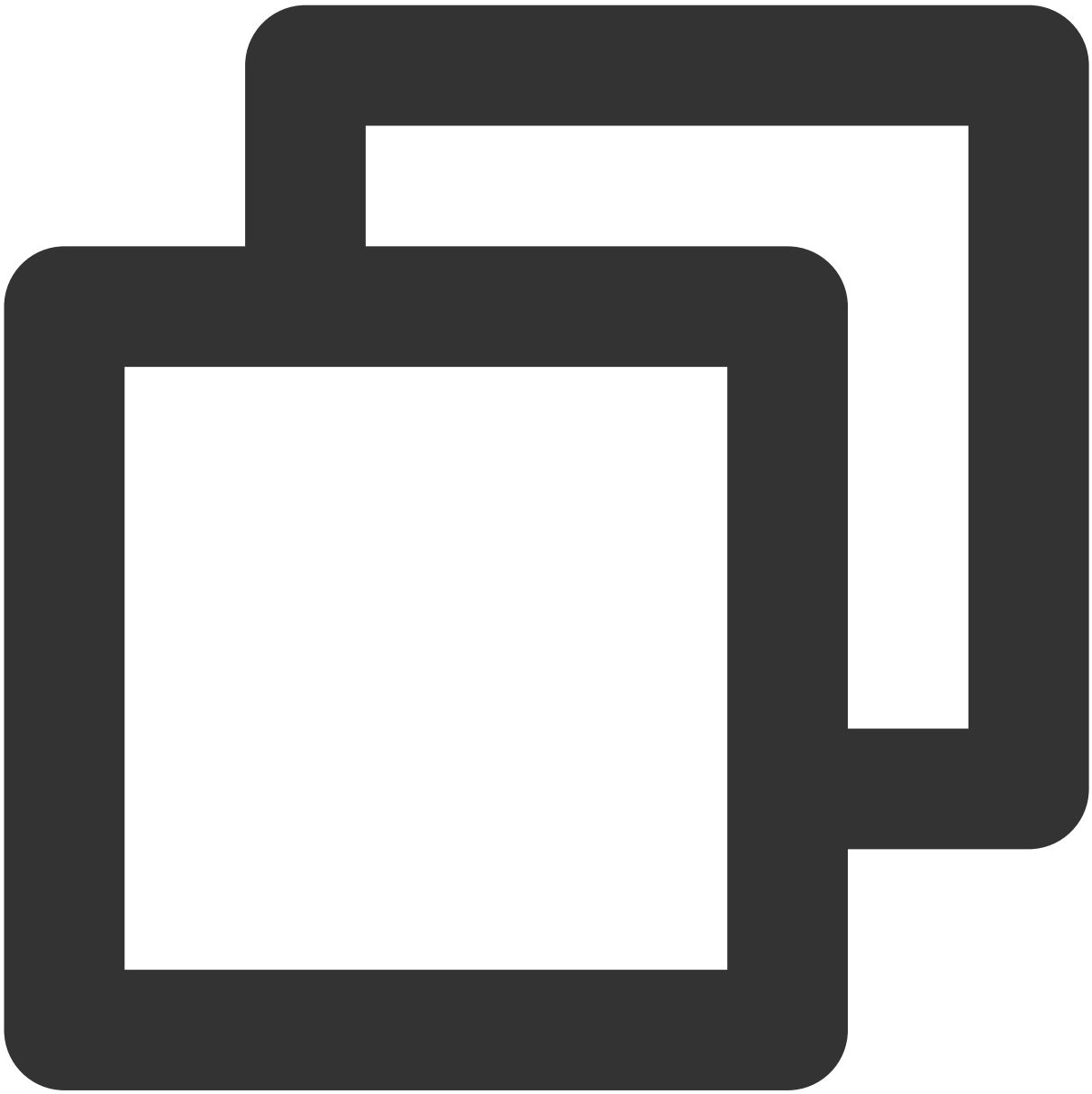
The password is reset.



```
HTTP/1.1 200 OK
```

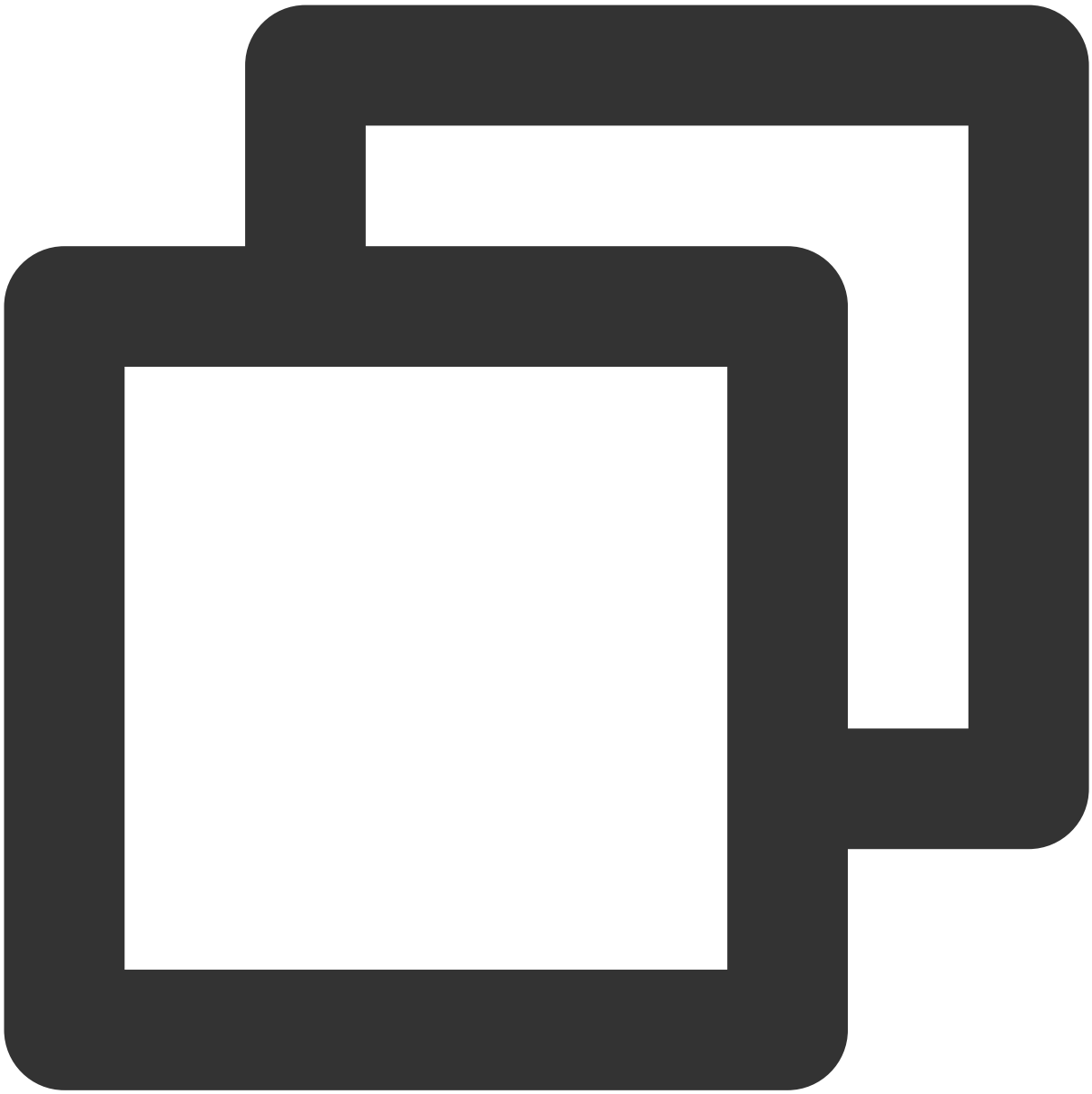
Sample Error Responses

The new password is the same as a previous password.



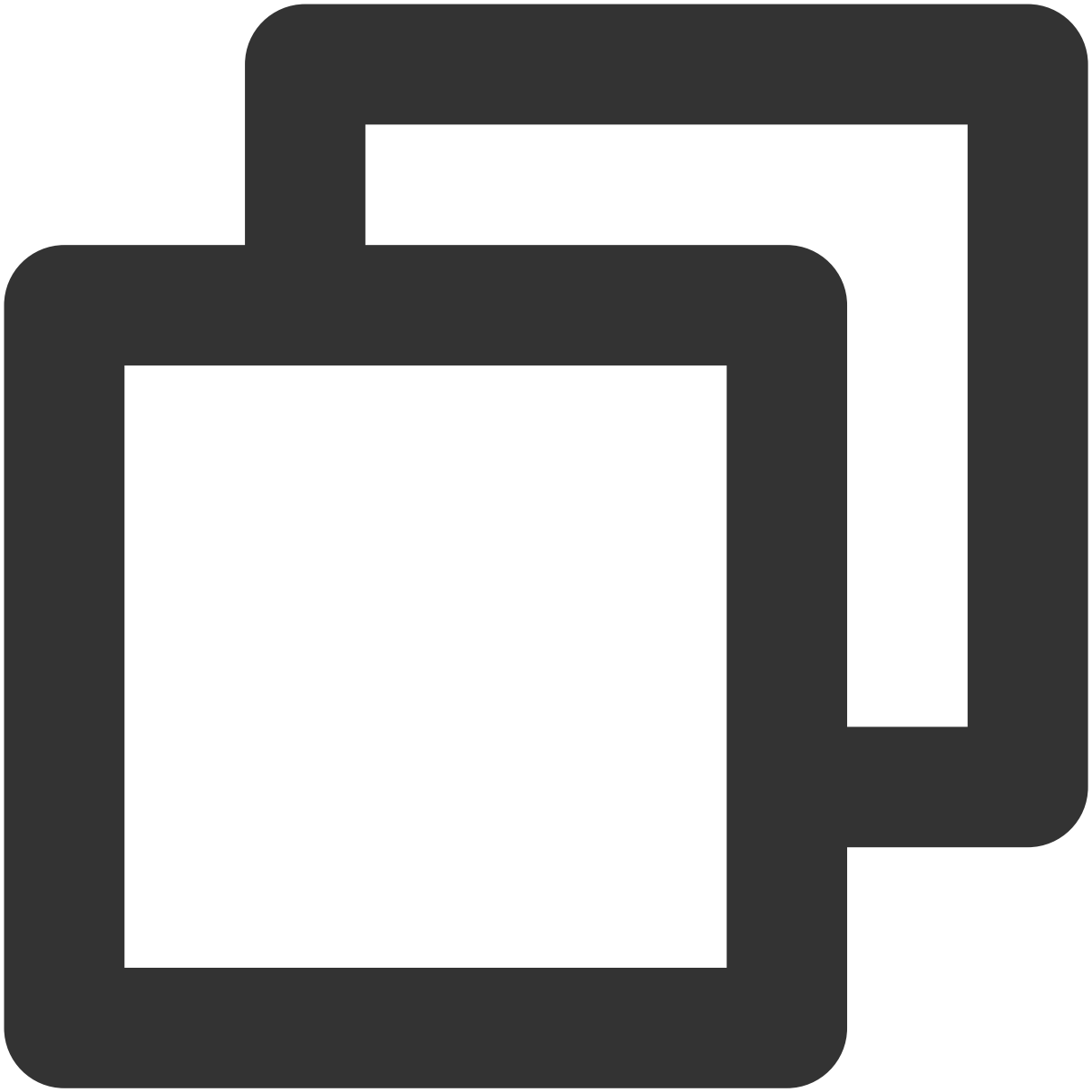
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "recurrent_password"
}
```

The new password does not comply with the password policy.



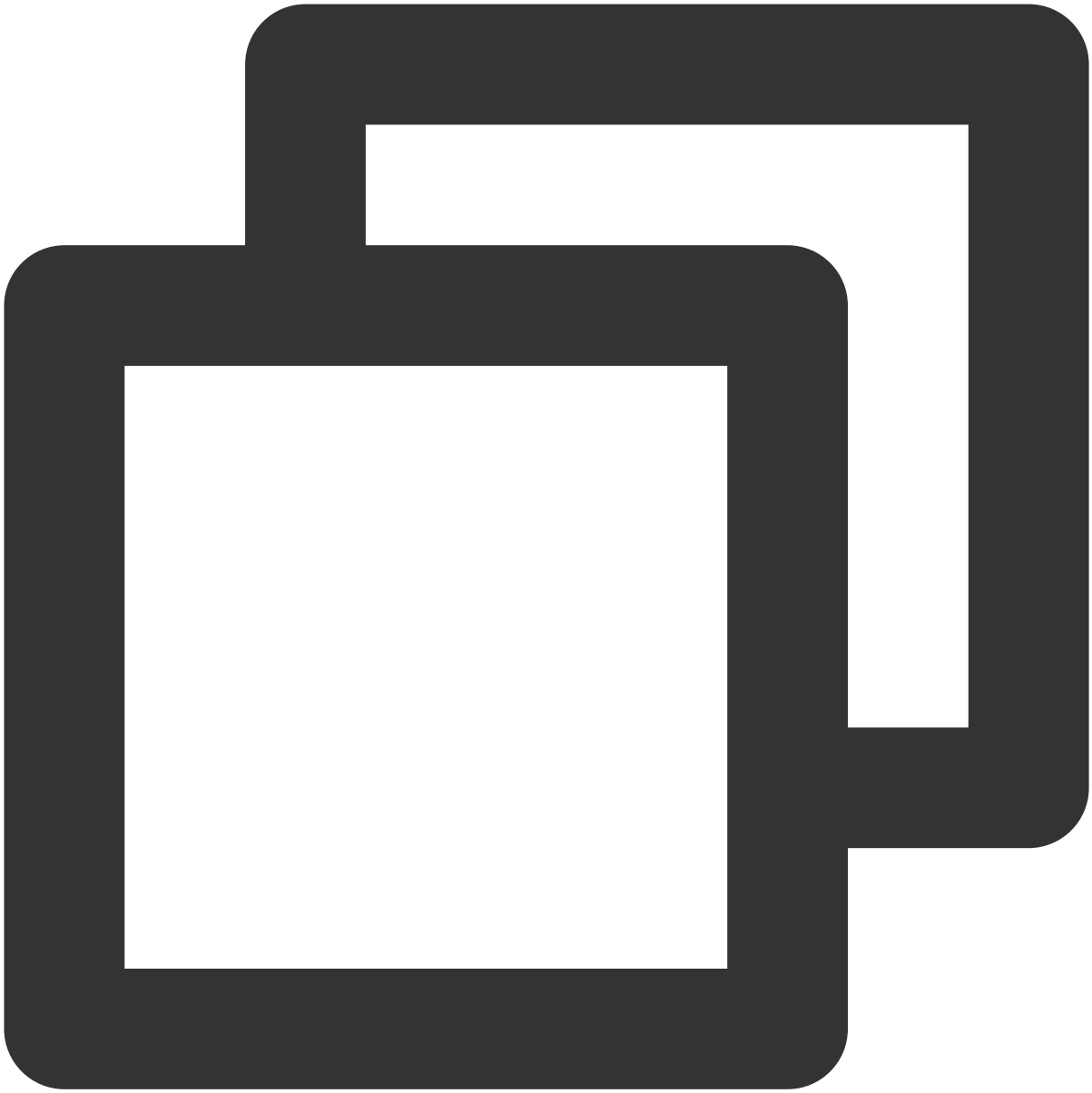
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_new_password"
}
```

No user found for the email address.



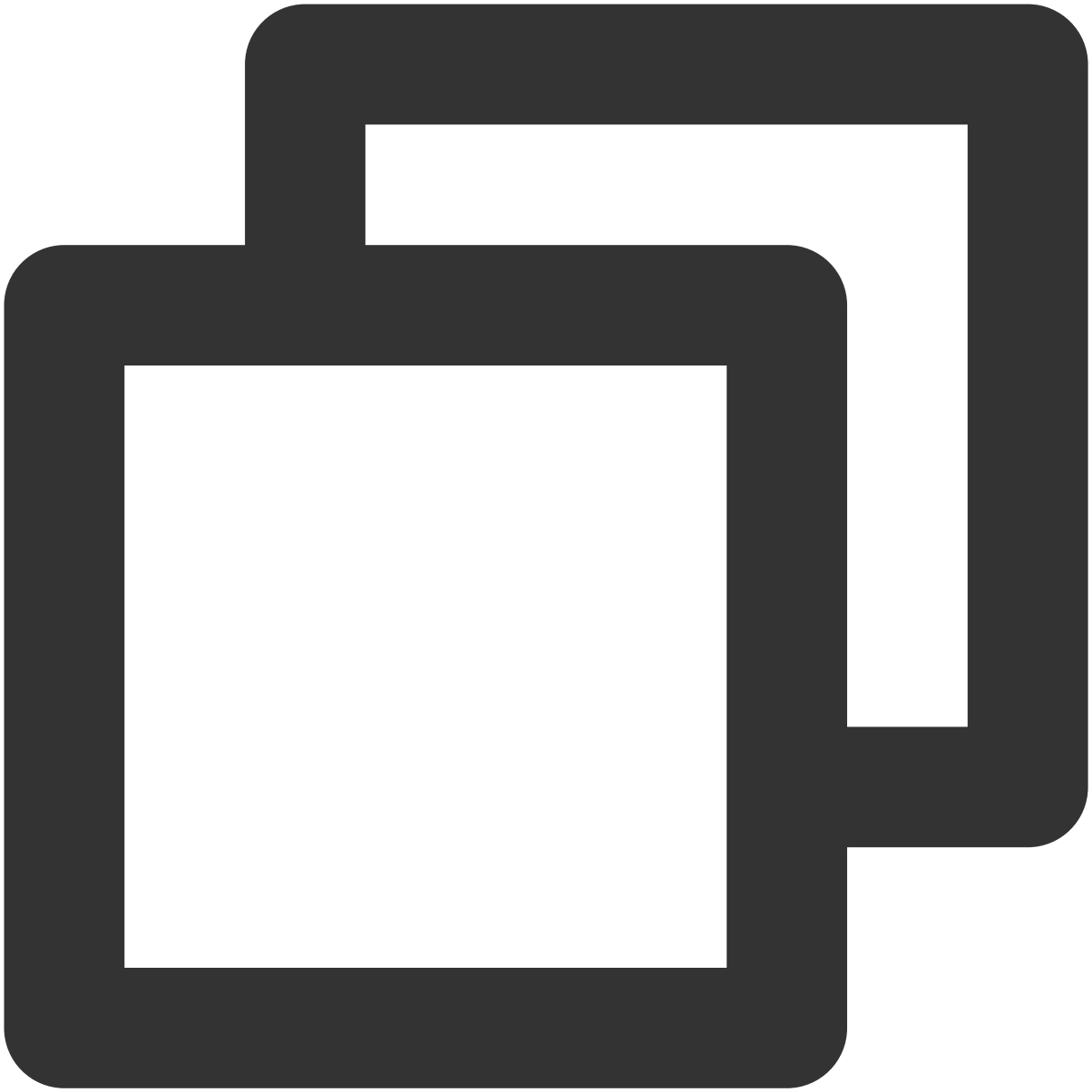
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "user_not_found"
}
```

The password cannot be reset because the user account is frozen.



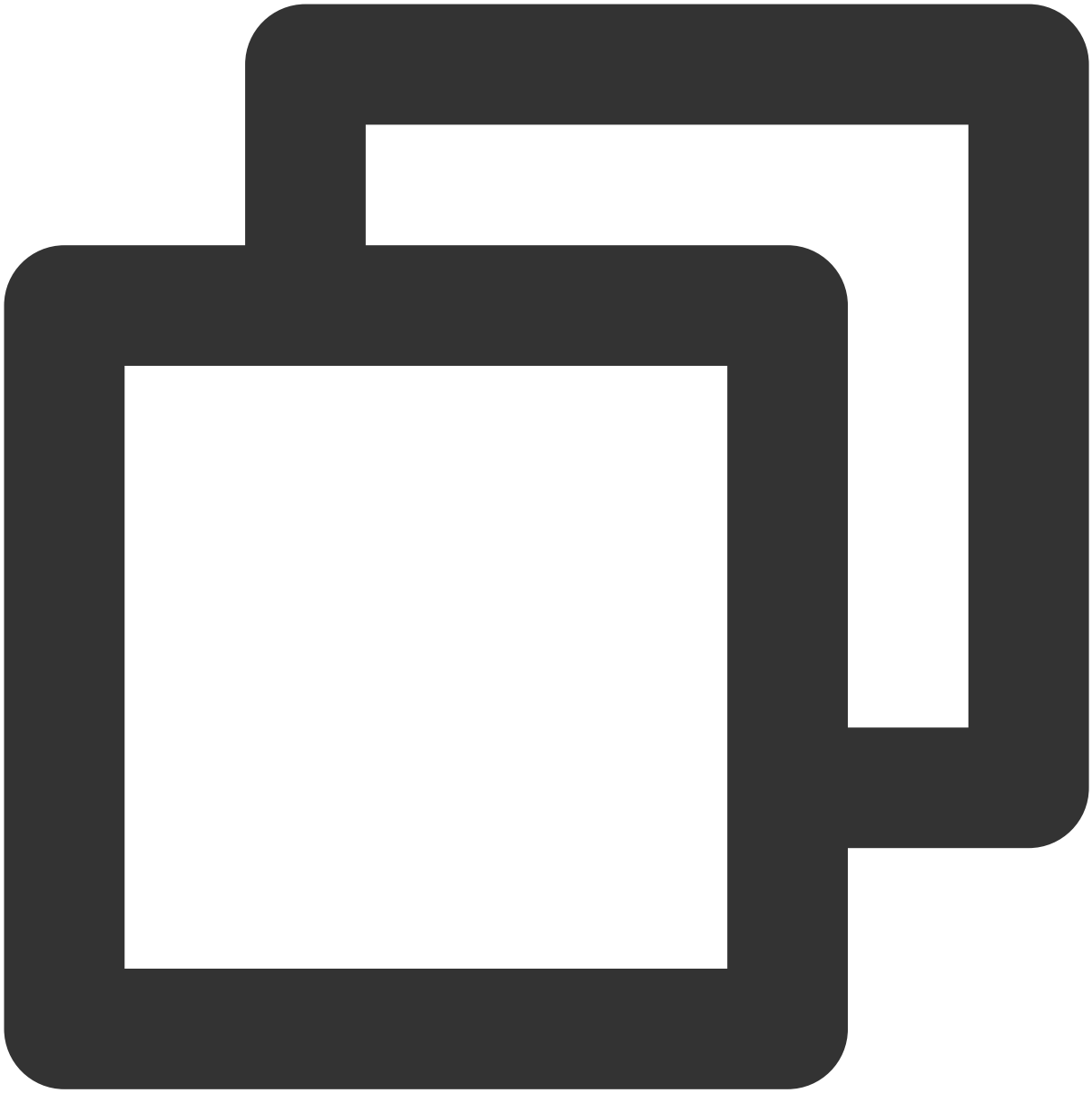
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is frozen."
}
```

The `email_otp_token` parameter is incorrect or has expired, or Reset parameter value used for resetting is not the same as the one used for sending the verification code. For example, the email addresses are different.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

The `email_otp` parameter is incorrect or has expired.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

Get Token
PKCE Authorization Code Mode

Last updated：2023-12-22 11:42:07

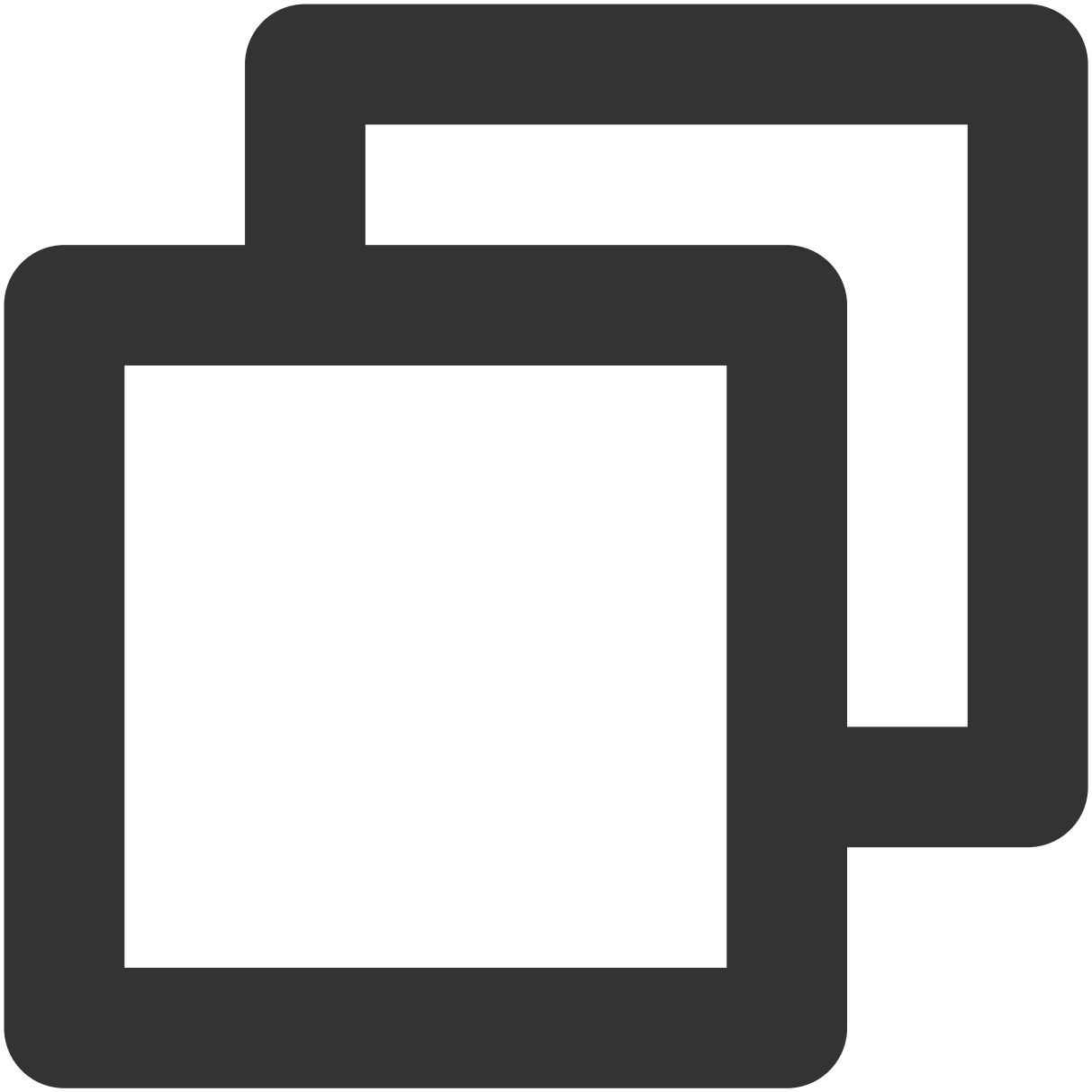
API Description

This API is used to get the Access Token and ID Token for login after the application system gets the `code` returned by the authentication portal through the Proof Key for Code Exchange (PKCE) authorization code mode.

Supported Applications

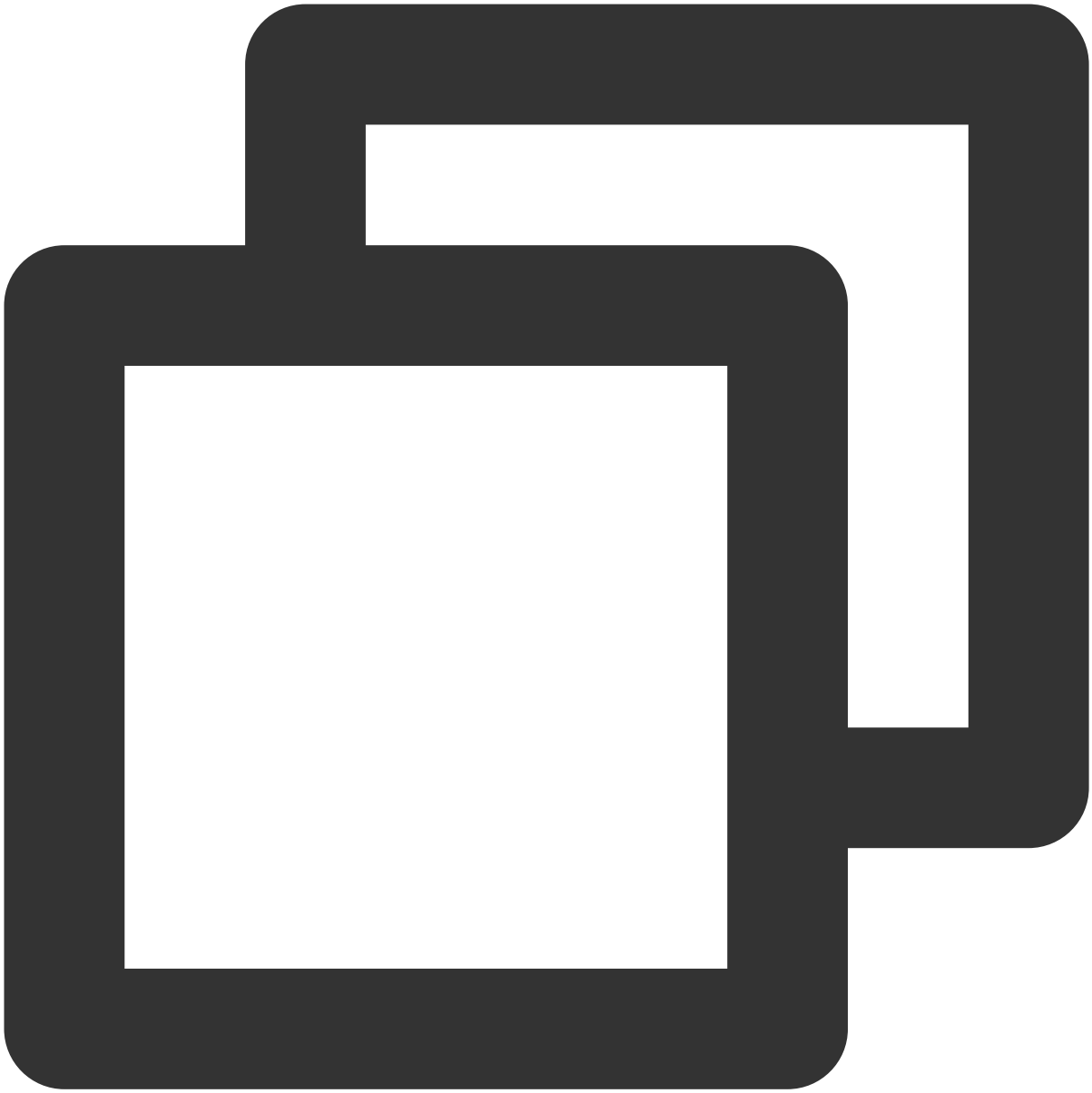
Web applications, single-page applications (SPA), and mobile applications.

Request Method



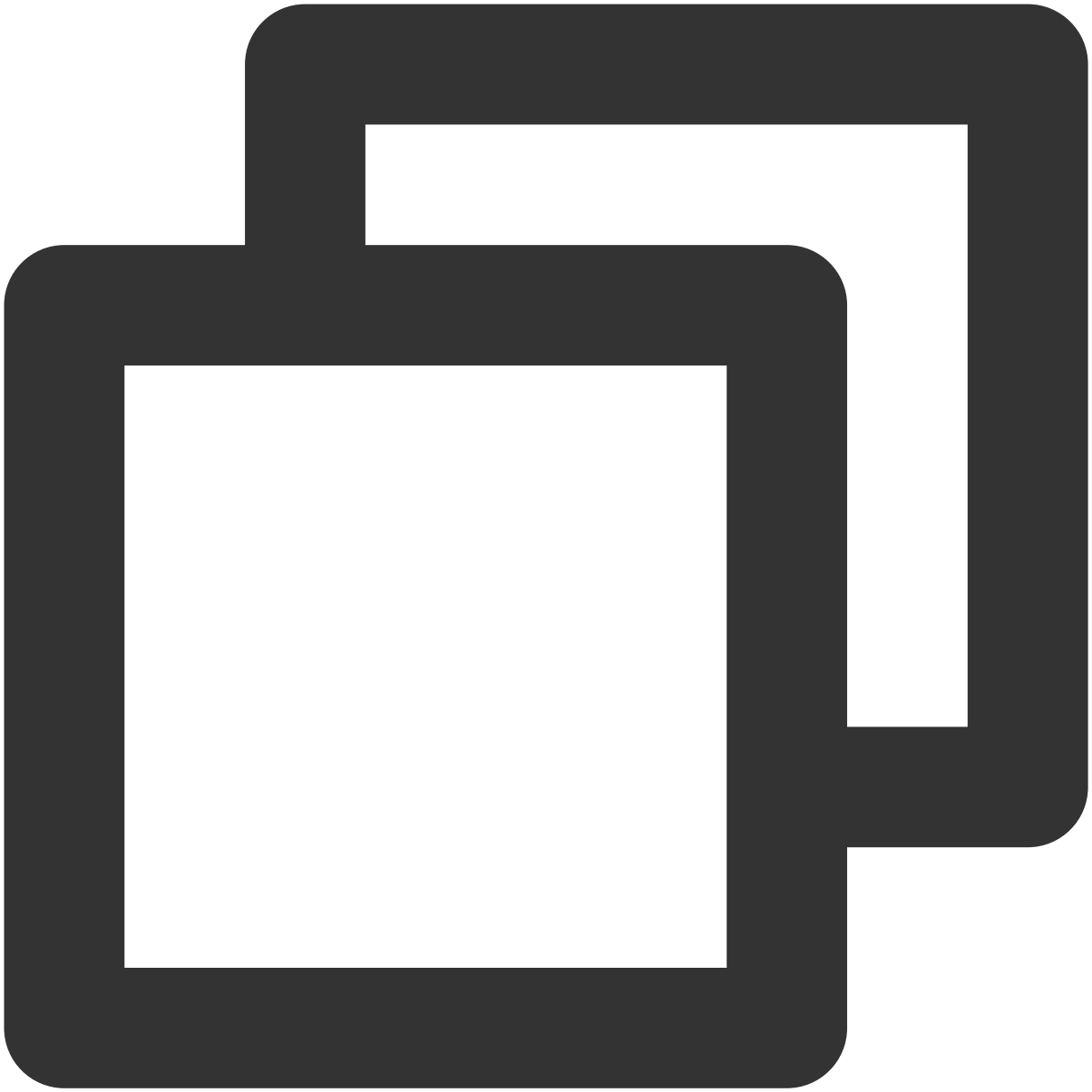
```
POST
```

Request Path



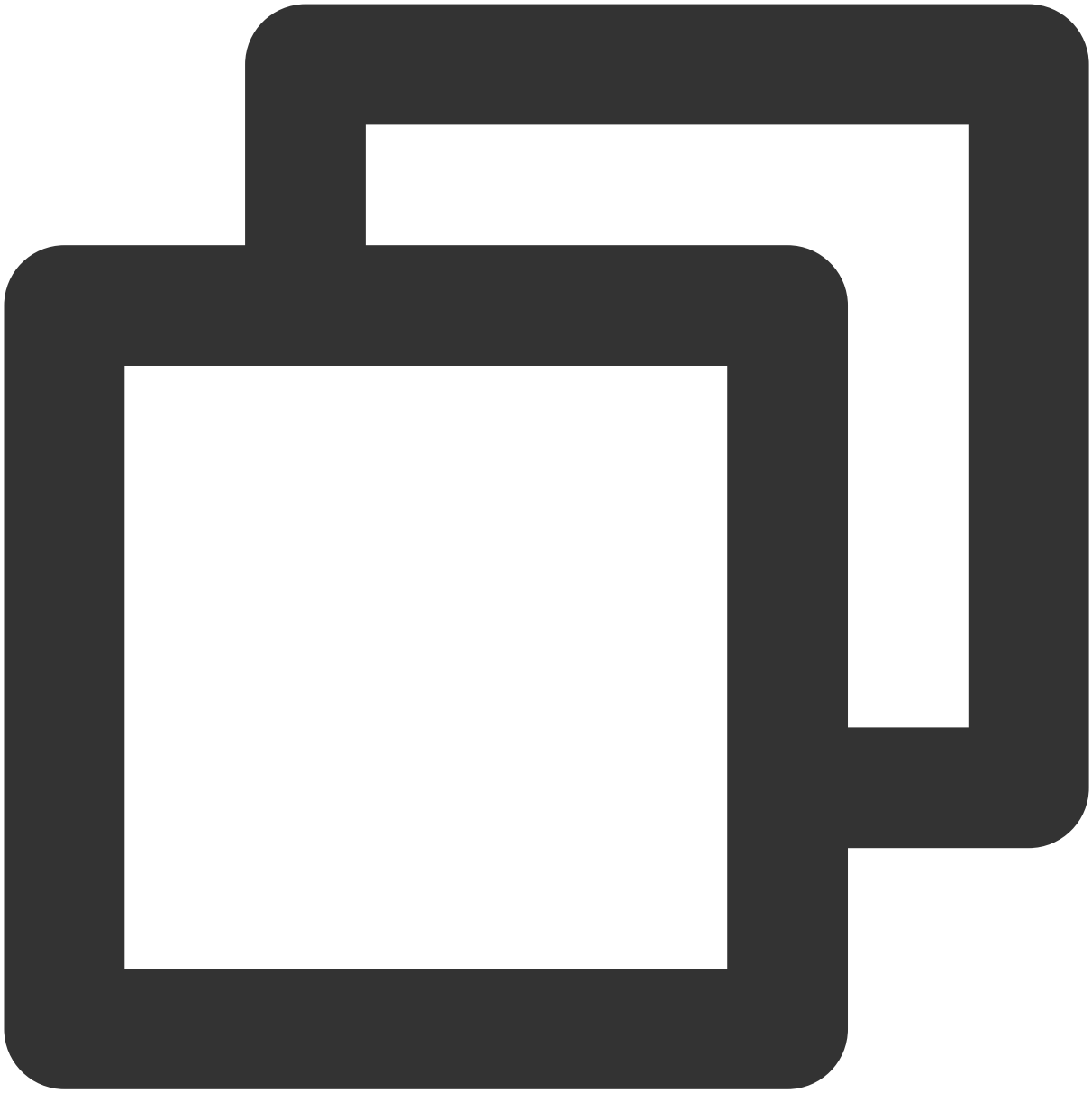
```
/oauth2/token
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests

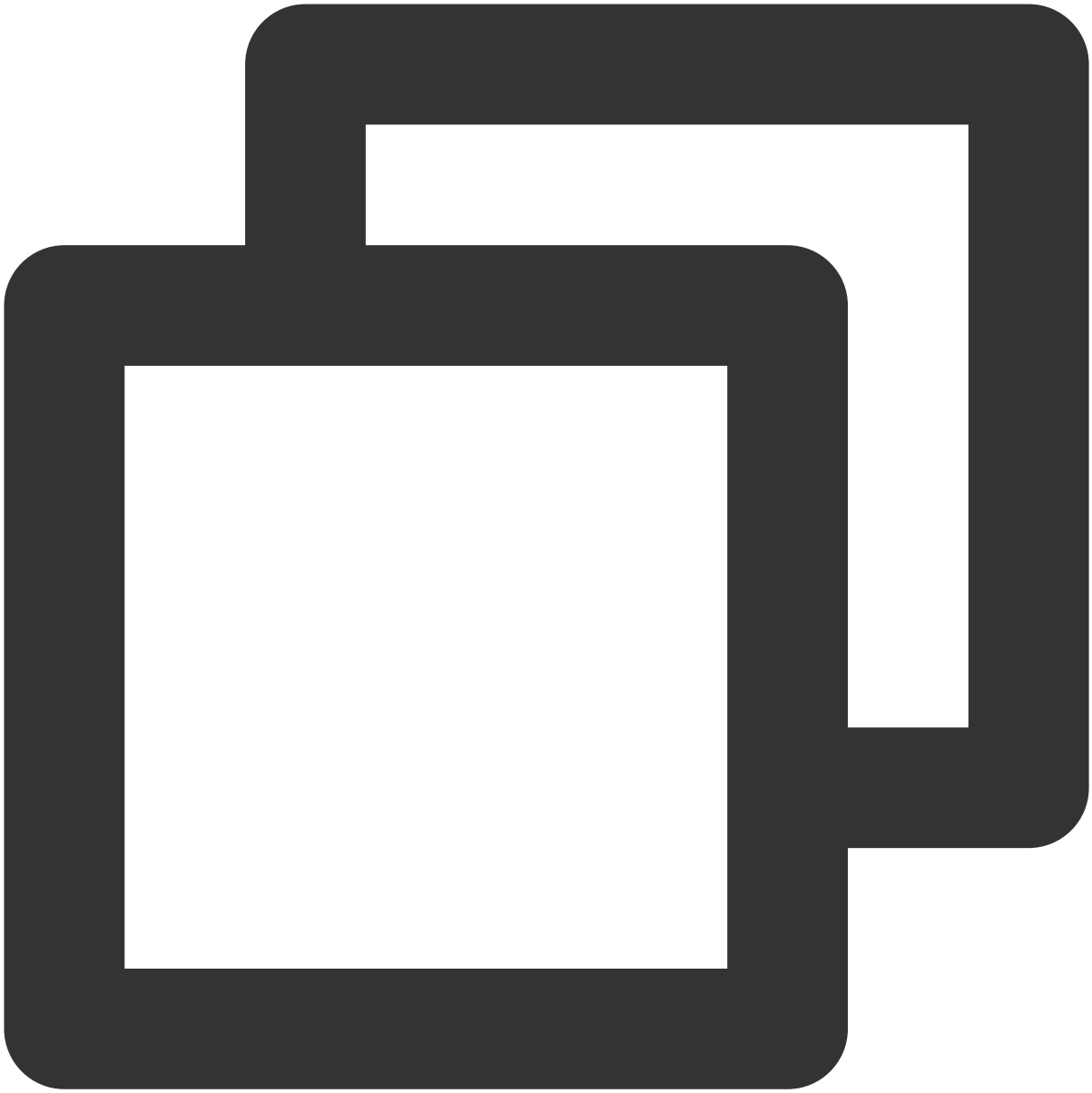


```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&grant_type=authorization_code&code=MOCK_CODE&redirect_ur
```

Request Parameters

Parameter	Optional	Description
client_id	false	The `client_id` of the application. This should be the same as the one used for getting the authorization code.
grant_type	false	Fixed value: `authorization_code` .
code	false	The authorization code returned.
redirect_uri	false	The redirected address after authorization. This should be the same as the one used for getting the authorization code.
code_verifier	false	PKCE code verifier. This should be the same as the `code_verifier` used to generate the `code_challenge` .

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ
  "refresh_token" : "8FuXWpwMZI9oA8ASvCUrqap61N7RvPON6DjWFk-Saiv4dOR8y2tNf9eKf36woA
  "scope" : "openid",
  "id_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXA
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

Response Parameters

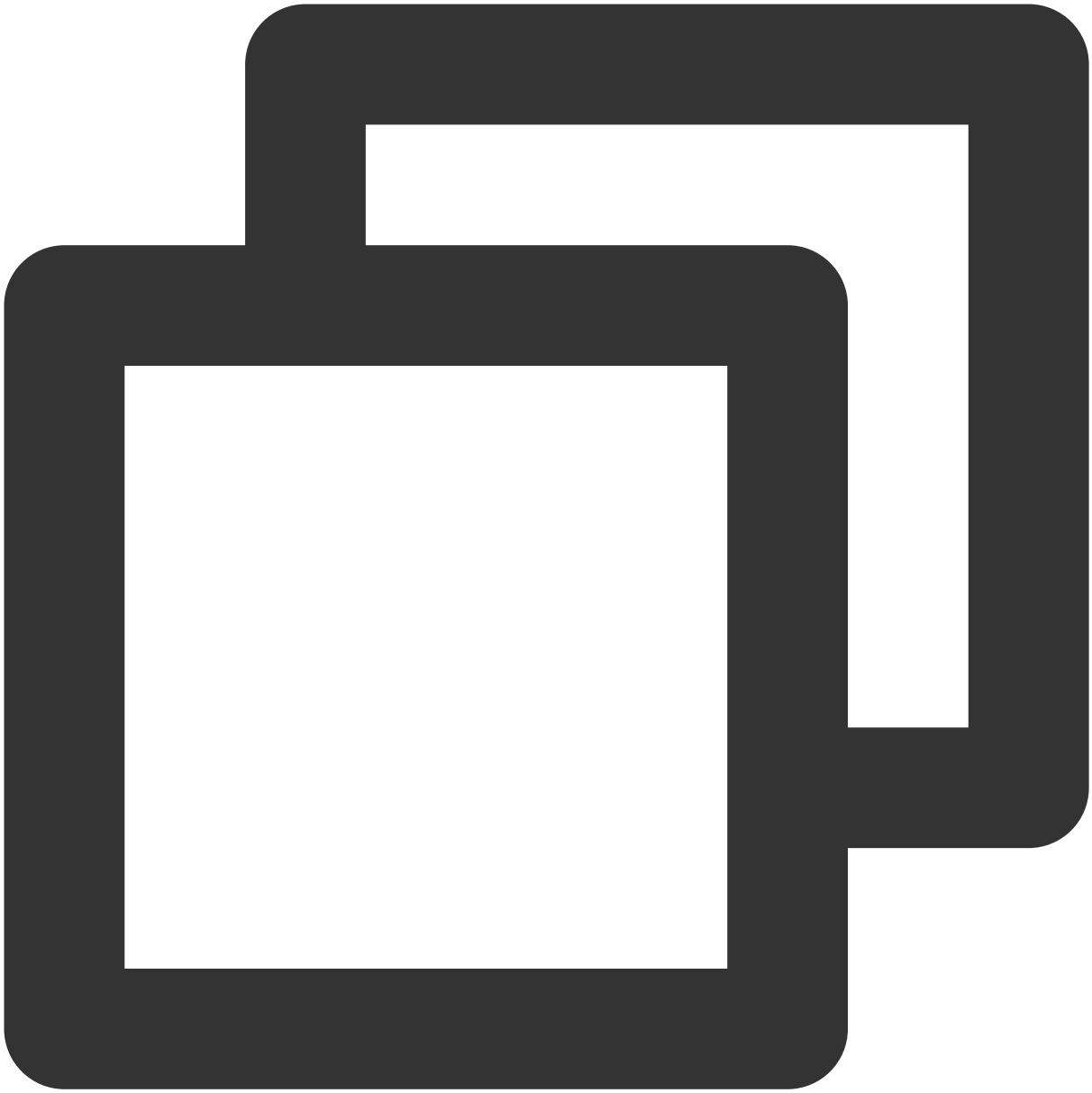
Parameter	Data Type	Description
access_token	String	OAuth 2.0 Access Token (JWT).
refresh_token	String	OAuth 2.0 Refresh Token.
scope	String	Access Token scope.
id_token	String	OpenID Connect (OIDC) ID Token (JSON Web Token, or JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)

Note:

Customer Identity Access Management (CIAM) returns ID Token in JWT format. For more information about how to decrypt and verify the ID Token, see the OpenID Connect document. You can also use the development library for decryption and verification. You can get the public key for verification by calling the API for getting the JWT public key.

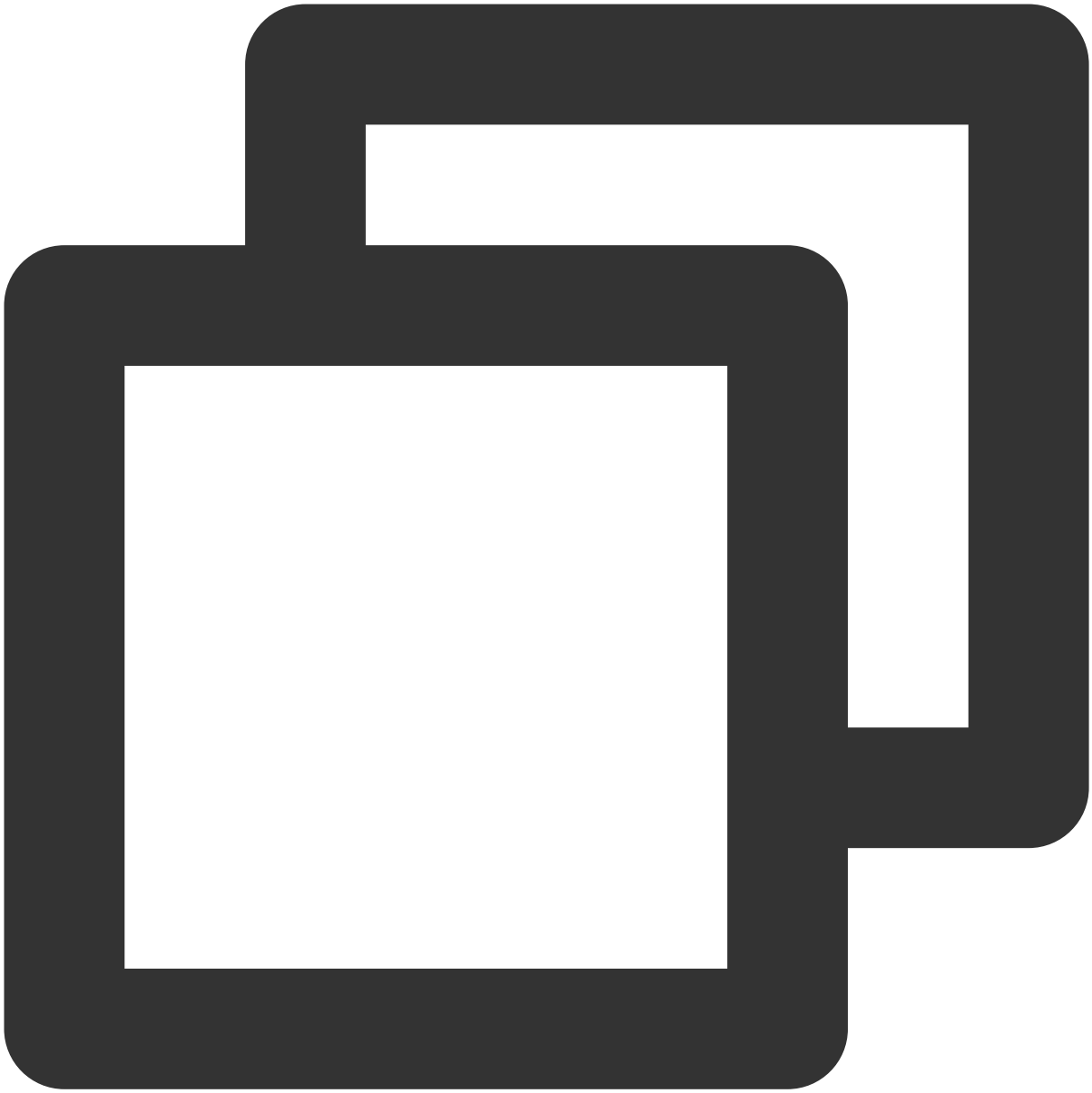
Sample Error Responses

The `client_id` parameter is missing or incorrect.



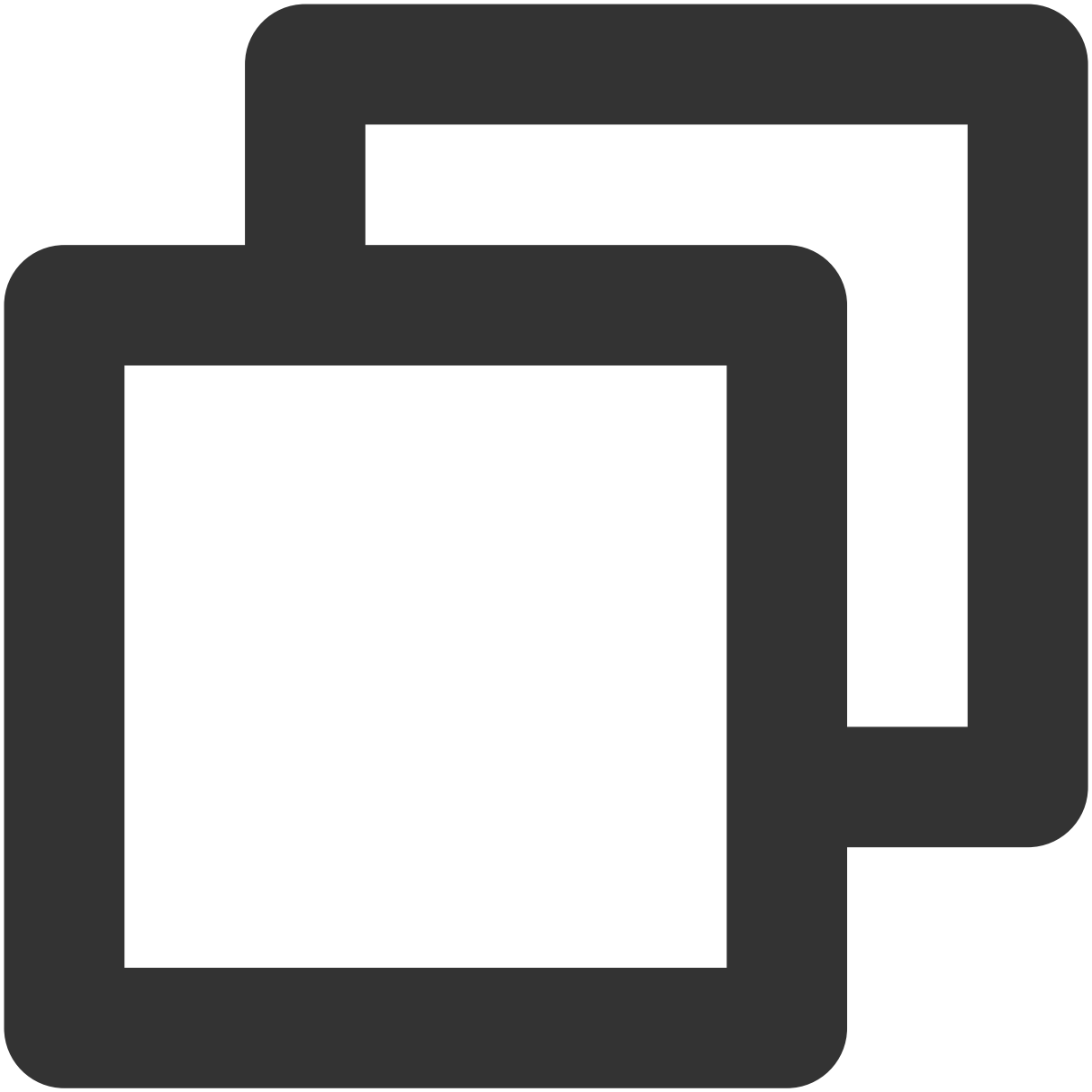
```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_request"
}
```

The `client_id` is not the same as the one used for getting the authorization code and Token.



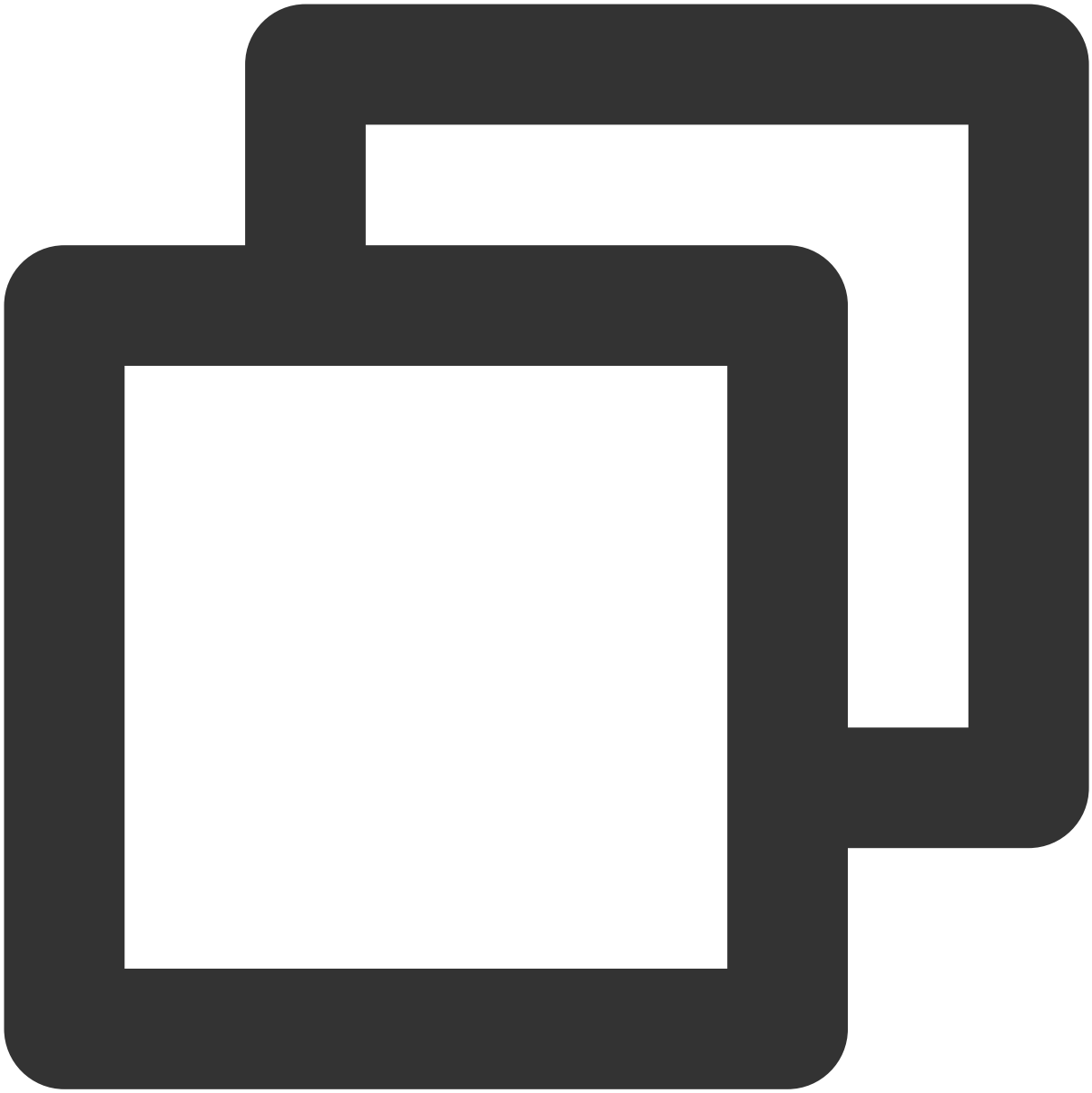
```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_client"
}
```

The `grant_type` parameter is incorrect.



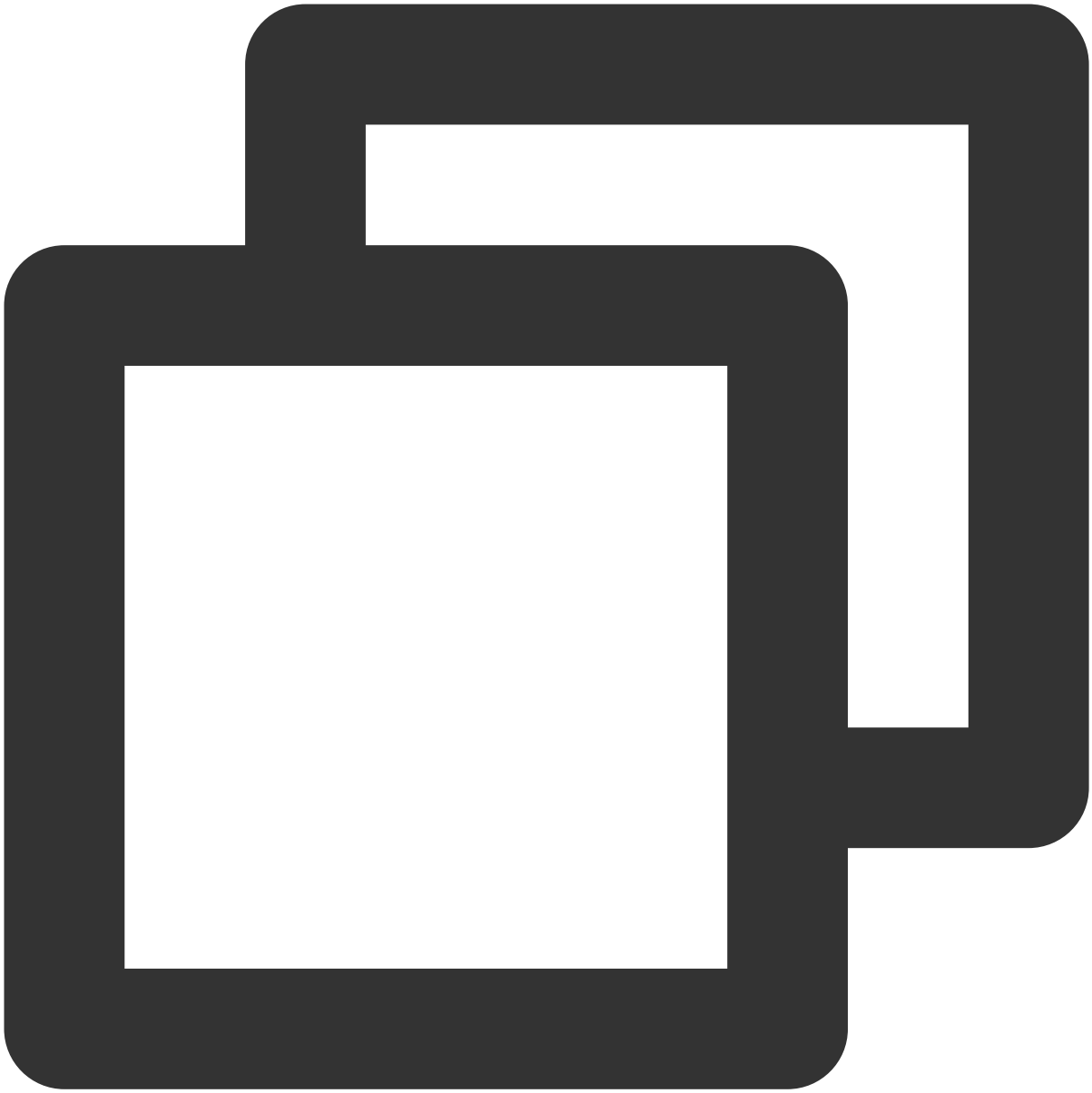
```
HTTP/1.1 401 Unauthorized
```

The `code` parameter is incorrect.



```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_client"
}
```

The `code_verifier` parameter is incorrect.



```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_client"
}
```

General Authorization Code Mode

Last updated：2023-12-22 11:42:07

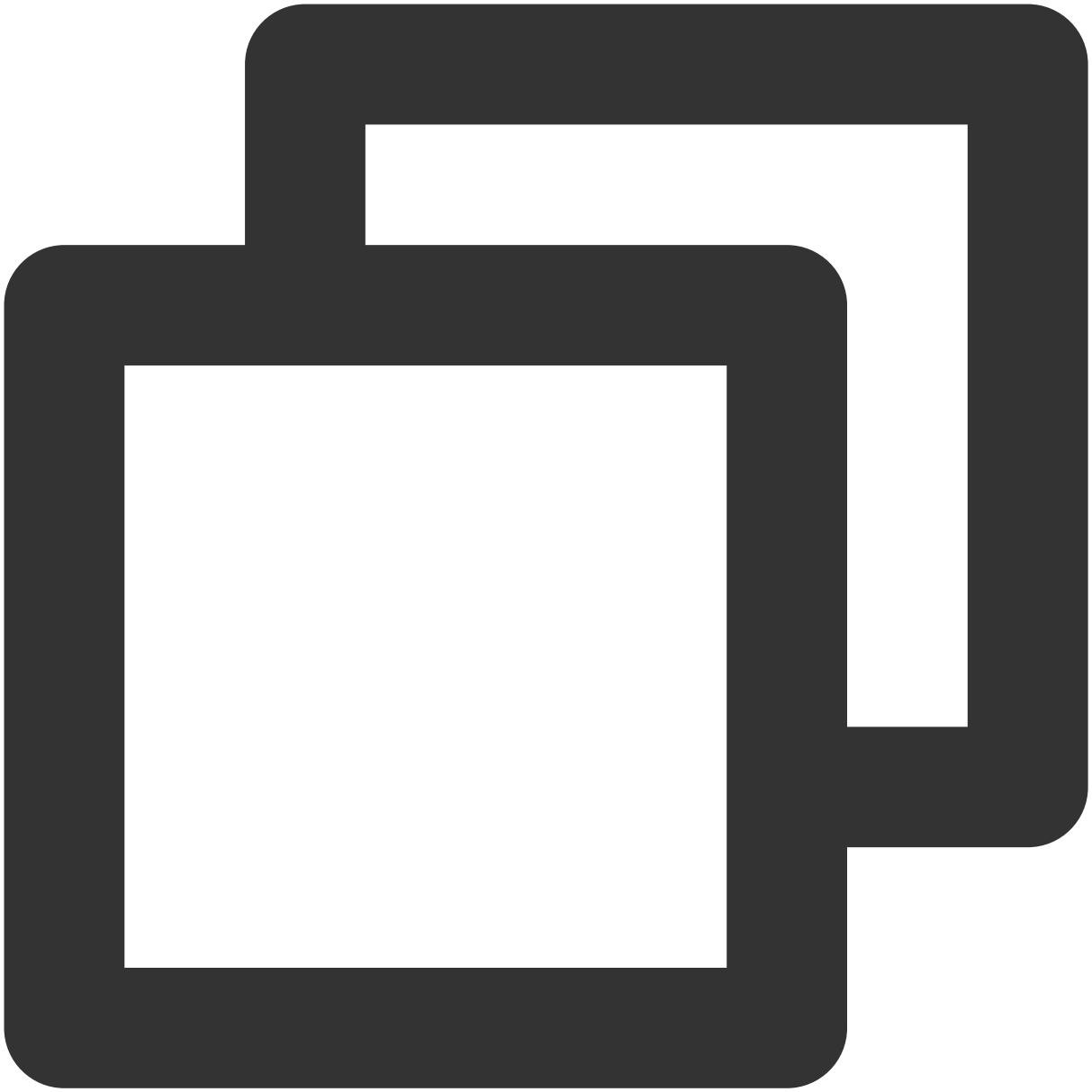
API Description

This API is used to get the Access Token and ID Token for login after the application system gets the `code` returned by the authentication portal through the normal authorization code mode.

Supported Applications

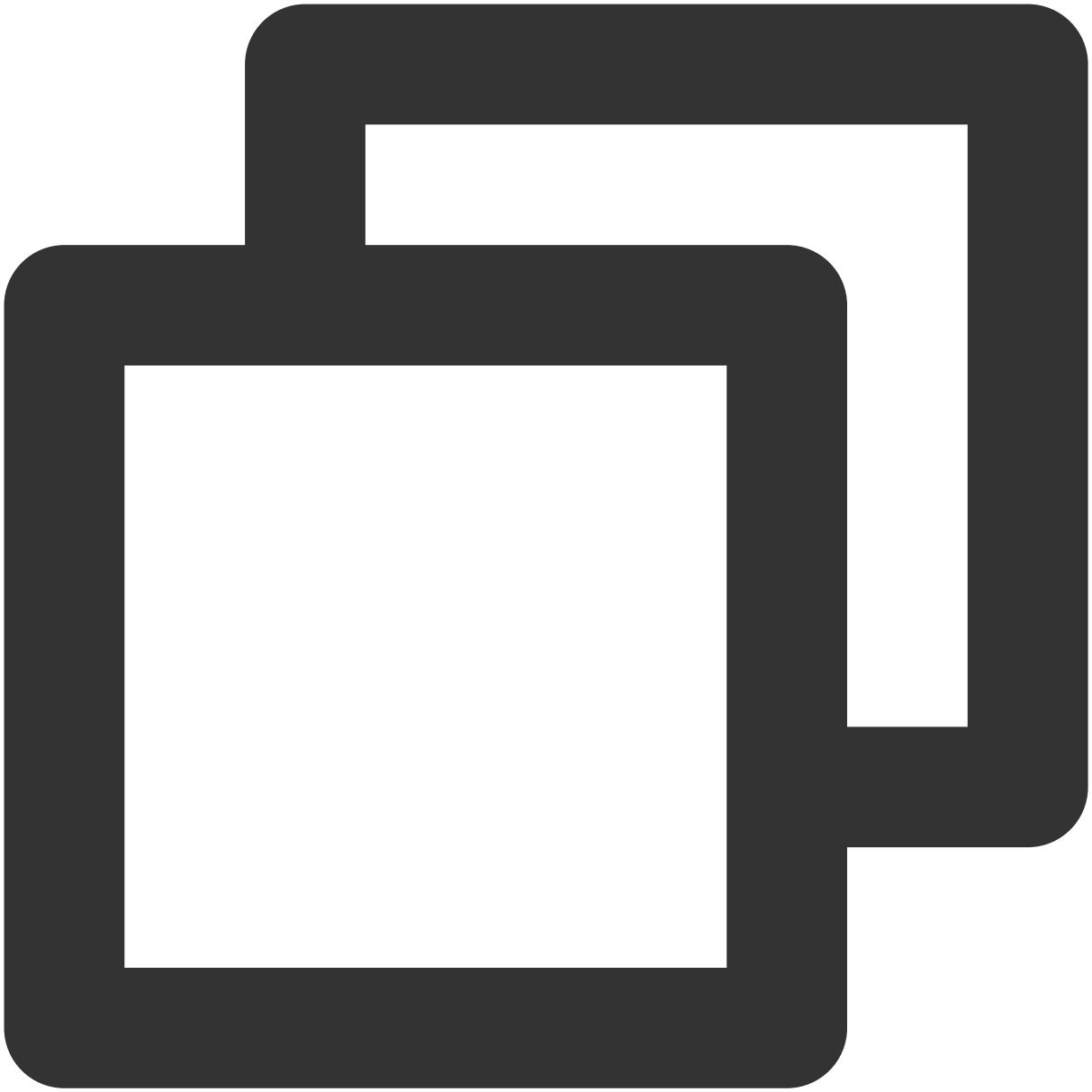
Web applications, single-page applications (SPA), and mobile applications.

Request Method



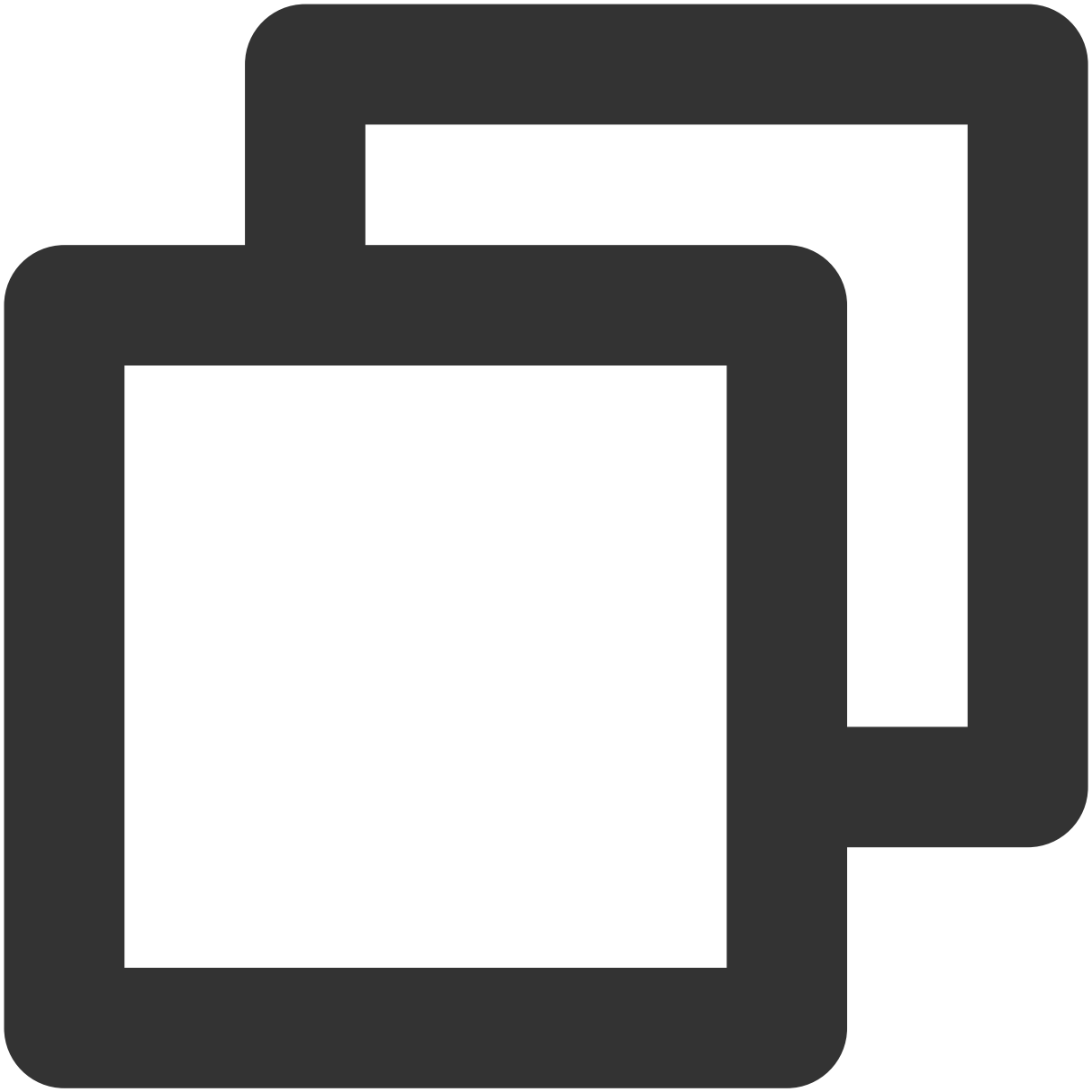
```
POST
```

Request Path



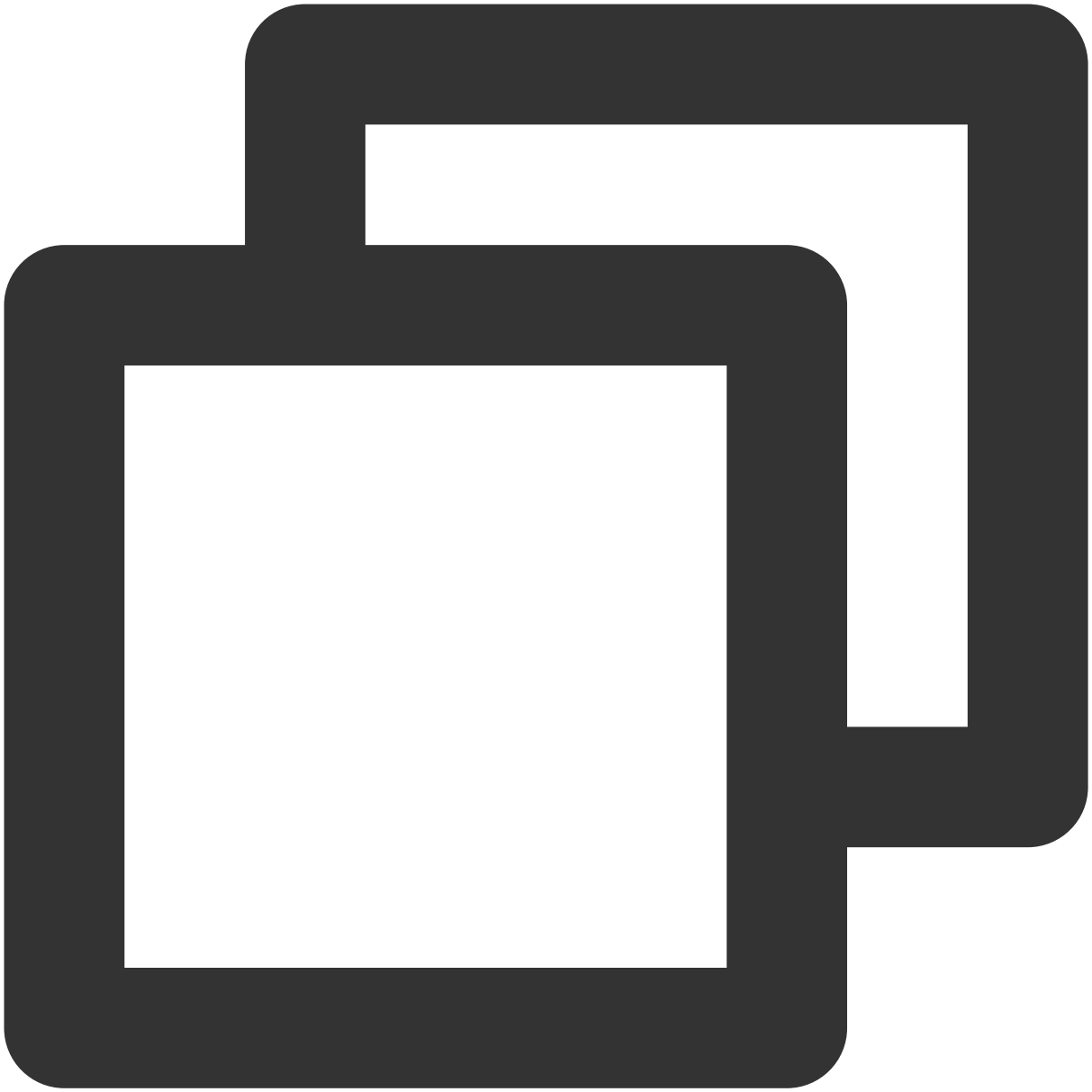
```
/oauth2/token
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests

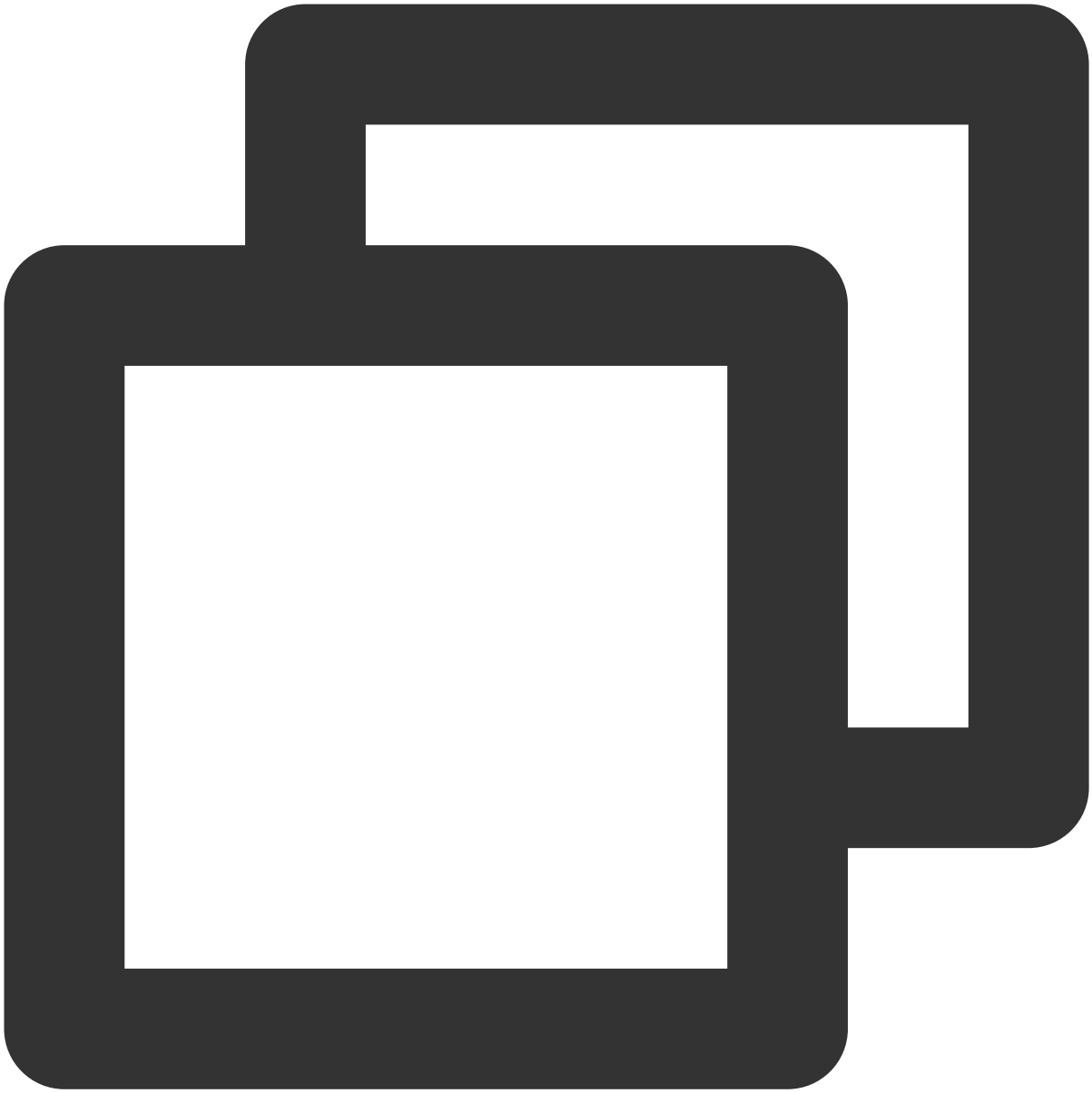


```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&grant_type=authorizat
```

Request Parameters

Parameter	Optional	Description
client_id	false	The `client_id` of the application. This should be the same as the one used for getting the authorization code.
client_secret	false	The `client_secret` of the application. Go to the **application management page** and **select the application**, and then click **Application Configuration** to find the **client_secret**.
grant_type	false	Fixed value: `authorization_code` .
code	false	The authorization code returned.
redirect_uri	false	The redirected address after authorization. This should be the same as the one used for getting the authorization code.

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ
  "refresh_token" : "Ugvo1lO7Se8vvIPrIOwn_eBe0hoi5-5ynR3H-aFYl0e1Gej-SfUAaBDBXkWmoj
  "scope" : "openid",
  "id_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXA
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

Note:

Customer Identity Access Management (CIAM) returns ID Token in JSON Web Token (JWT) format. For more information about how to decrypt and verify the ID Token, see the OpenID Connect (OIDC) document. You can also use the development library for decryption and verification. You can get the public key for verification by calling the API for getting the JWT public key.

Response Parameters

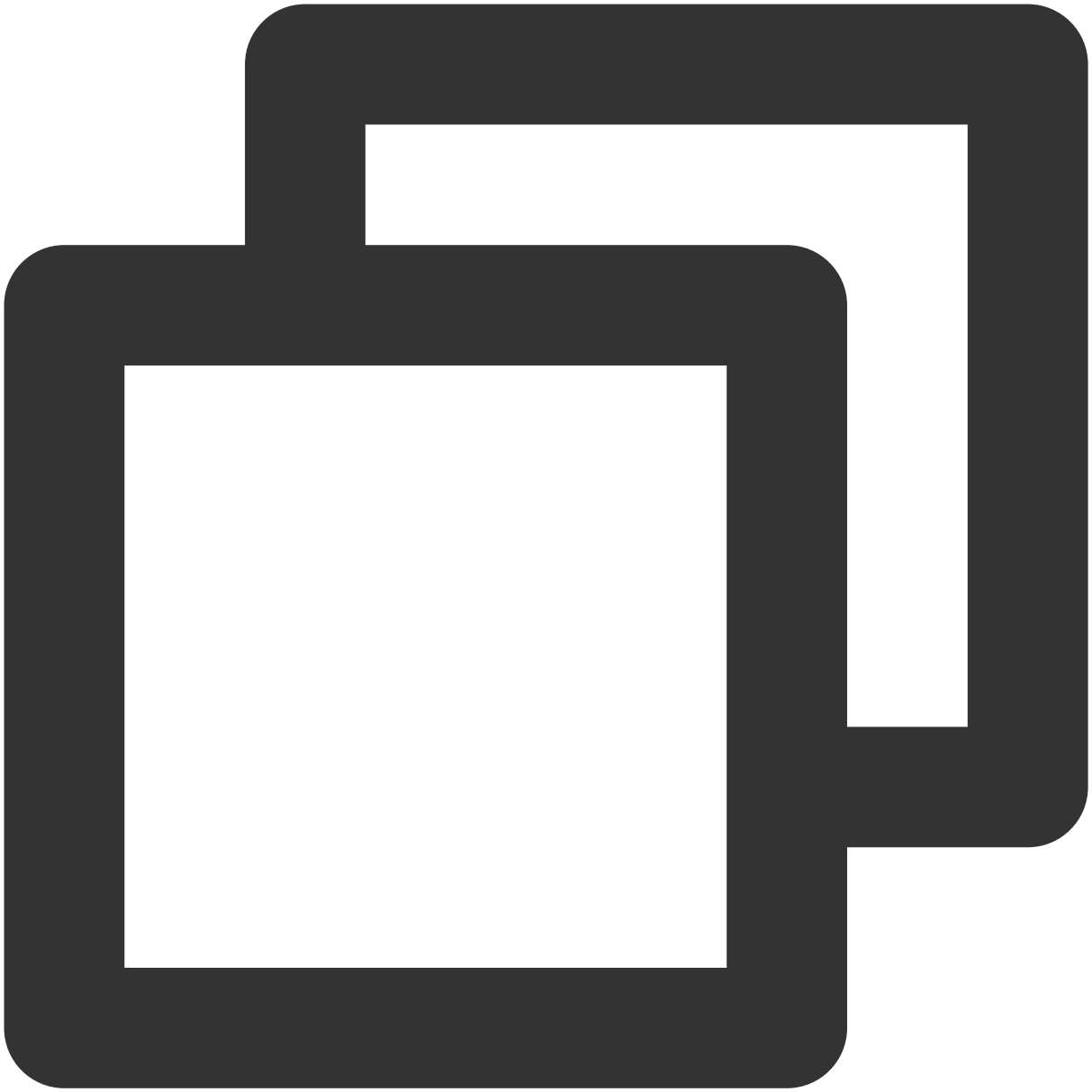
Parameter	Data Type	Description
access_token	String	OAuth 2.0 Access Token (JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)
scope	String	Access Token scope.
refresh_token	String	OAuth 2.0 Refresh Token.
id_token	String	OIDC ID Token (JWT).

Note:

Customer Identity Access Management (CIAM) returns ID Token in JWT format. For more information about how to decrypt and verify the ID Token, see the OpenID Connect document. You can also use the development library for decryption and verification. You can get the public key for verification by calling the API for getting the JWT public key.

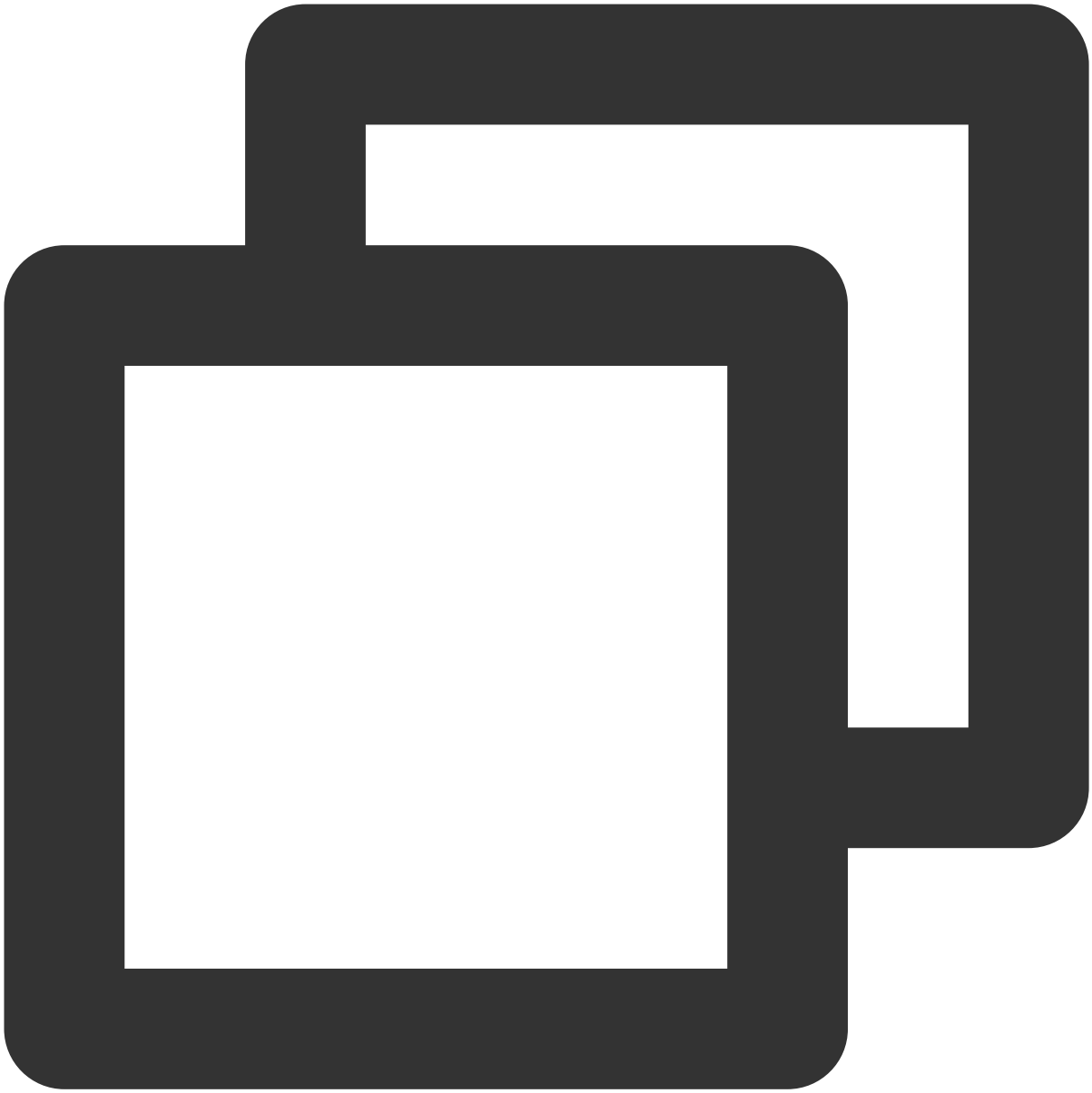
Sample Error Responses

The `client_id` parameter is missing or incorrect.



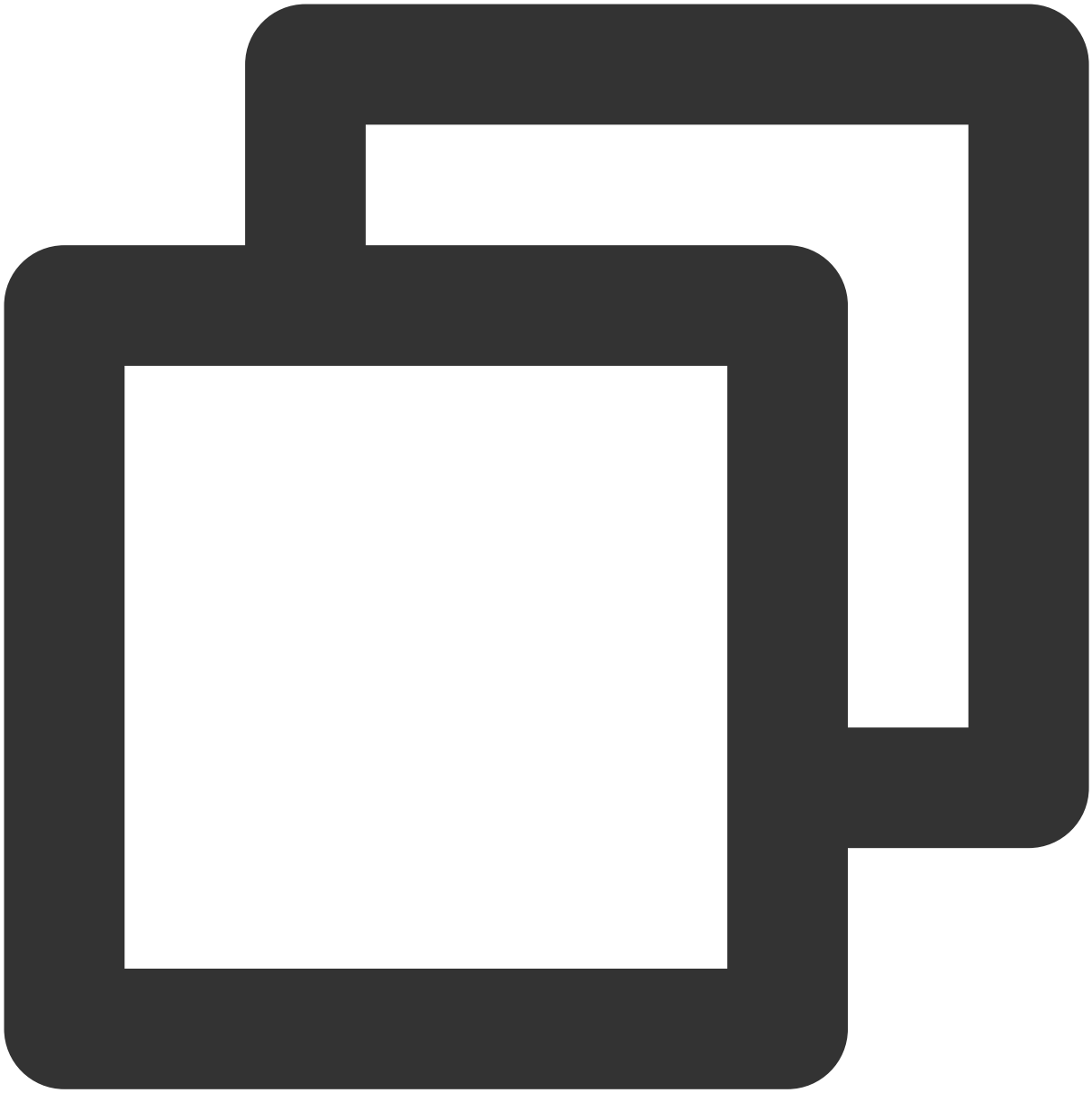
```
HTTP/1.1 401 Unauthorized
```

The `client_id` is not the same as the one used for getting the authorization code and Token.



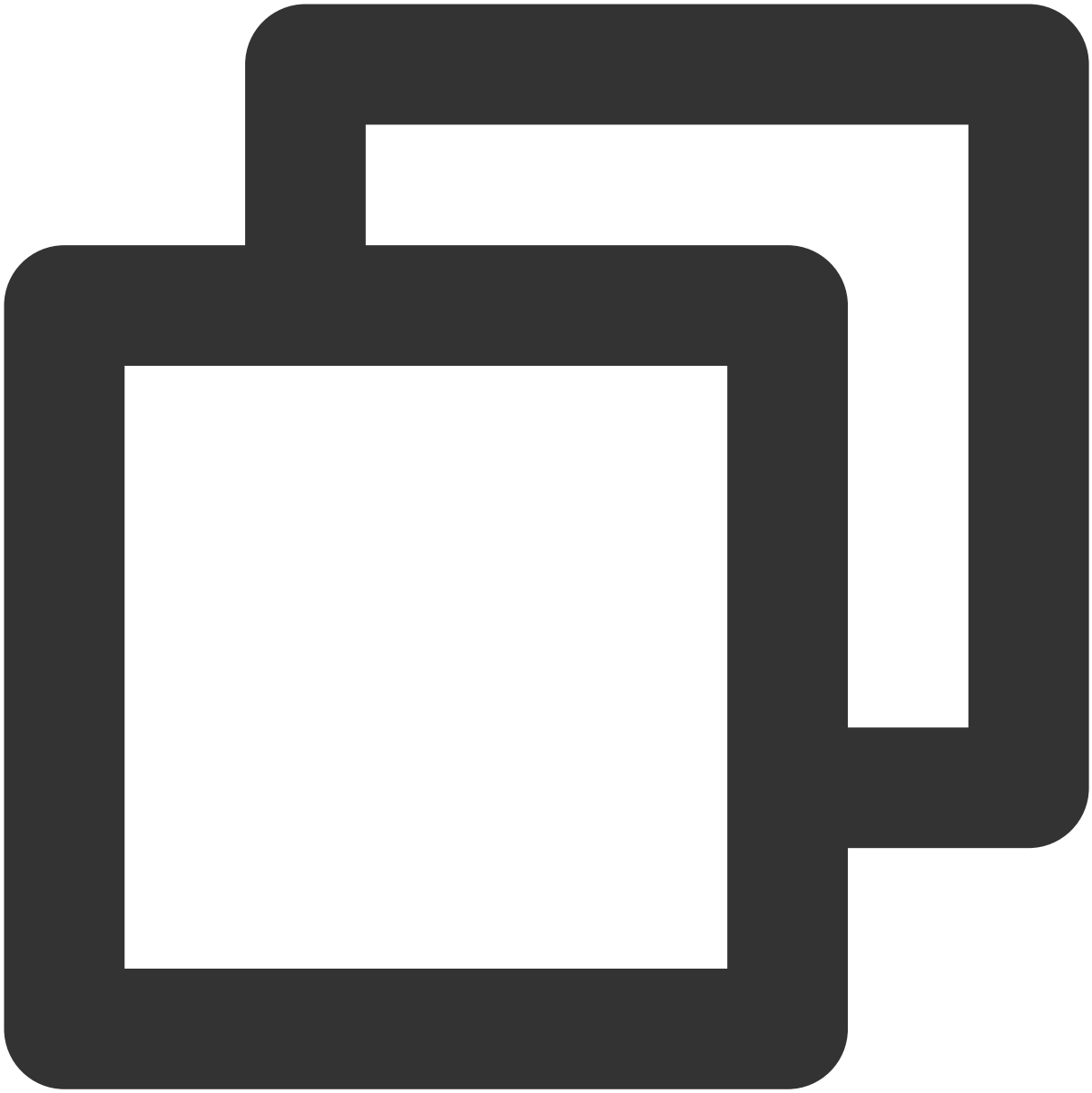
```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_client"
}
```

The `grant_type` parameter is incorrect.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "unsupported_grant_type",
"error_description" : "OAuth 2.0 Parameter: grant_type",
"error_uri" : "https://datatracker.ietf.org/doc/html/rfc6749#section-5.2"
}
```

The `code` parameter is incorrect.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant"
}
```

Client Credentials Mode

Last updated：2023-12-22 11:42:07

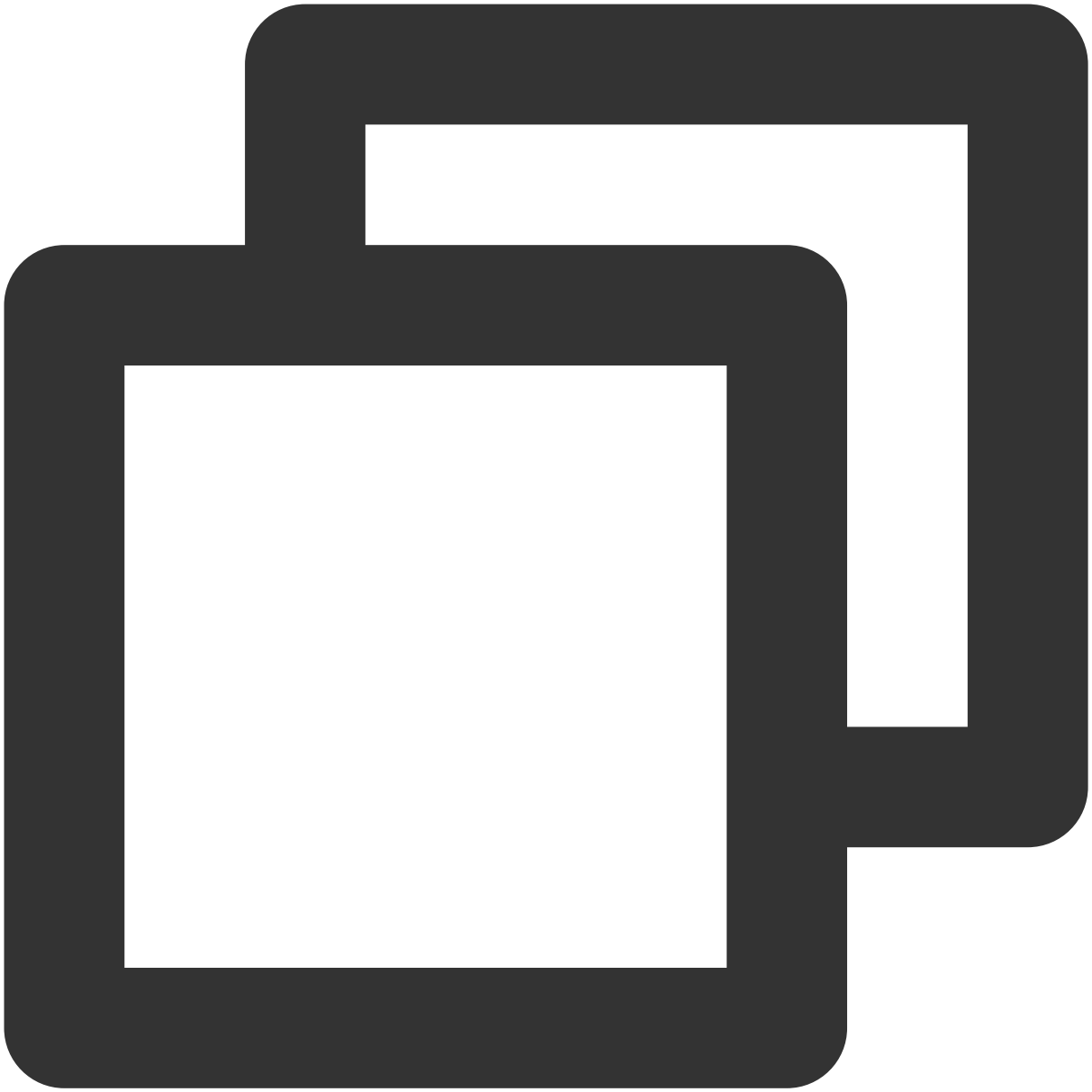
API Description

This API is used to get an Access Token using the OAuth client credentials (`client_credentials`).

Supported Applications

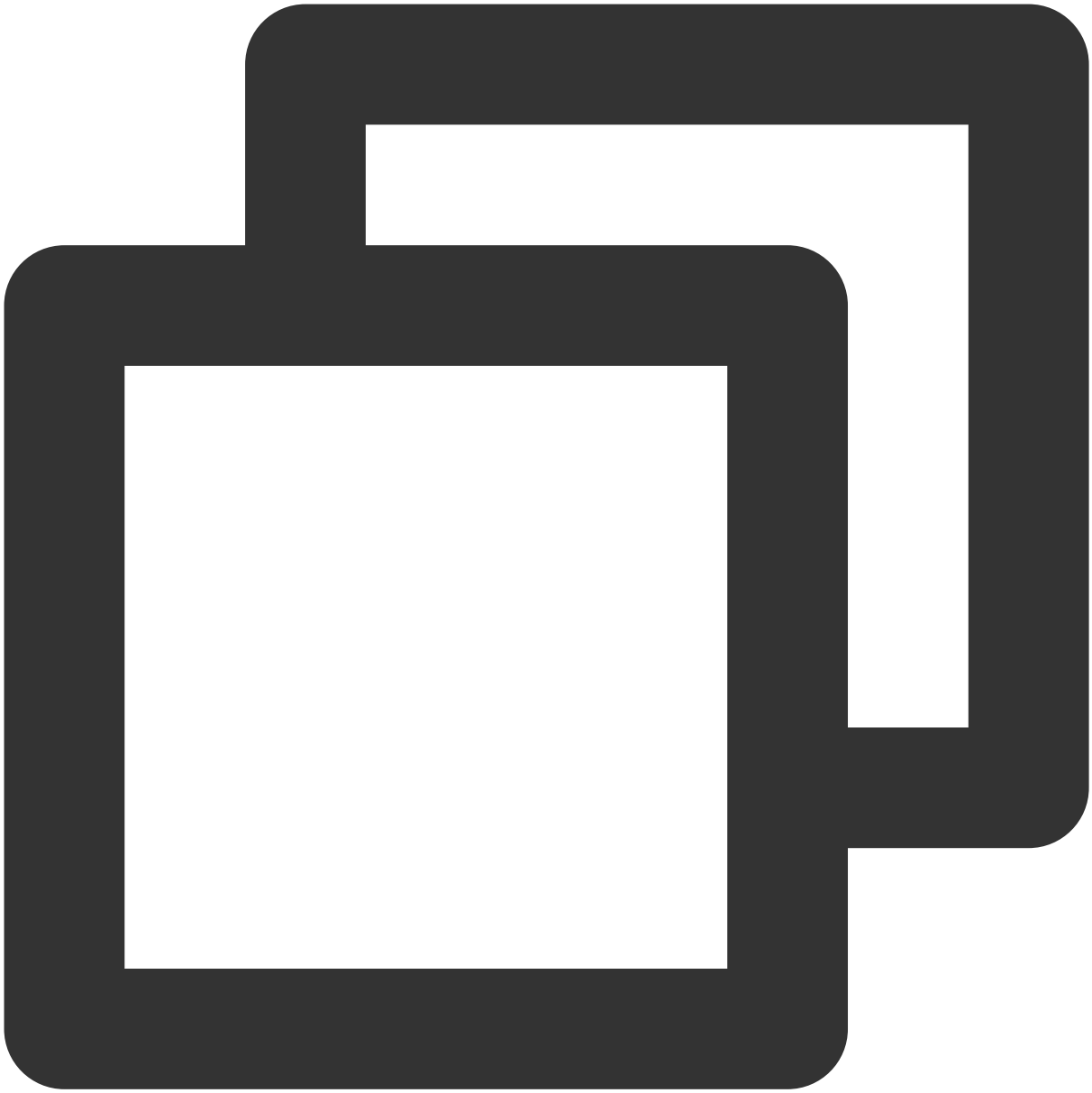
Web applications and machine-to-machine (M2M) applications.

Request Method



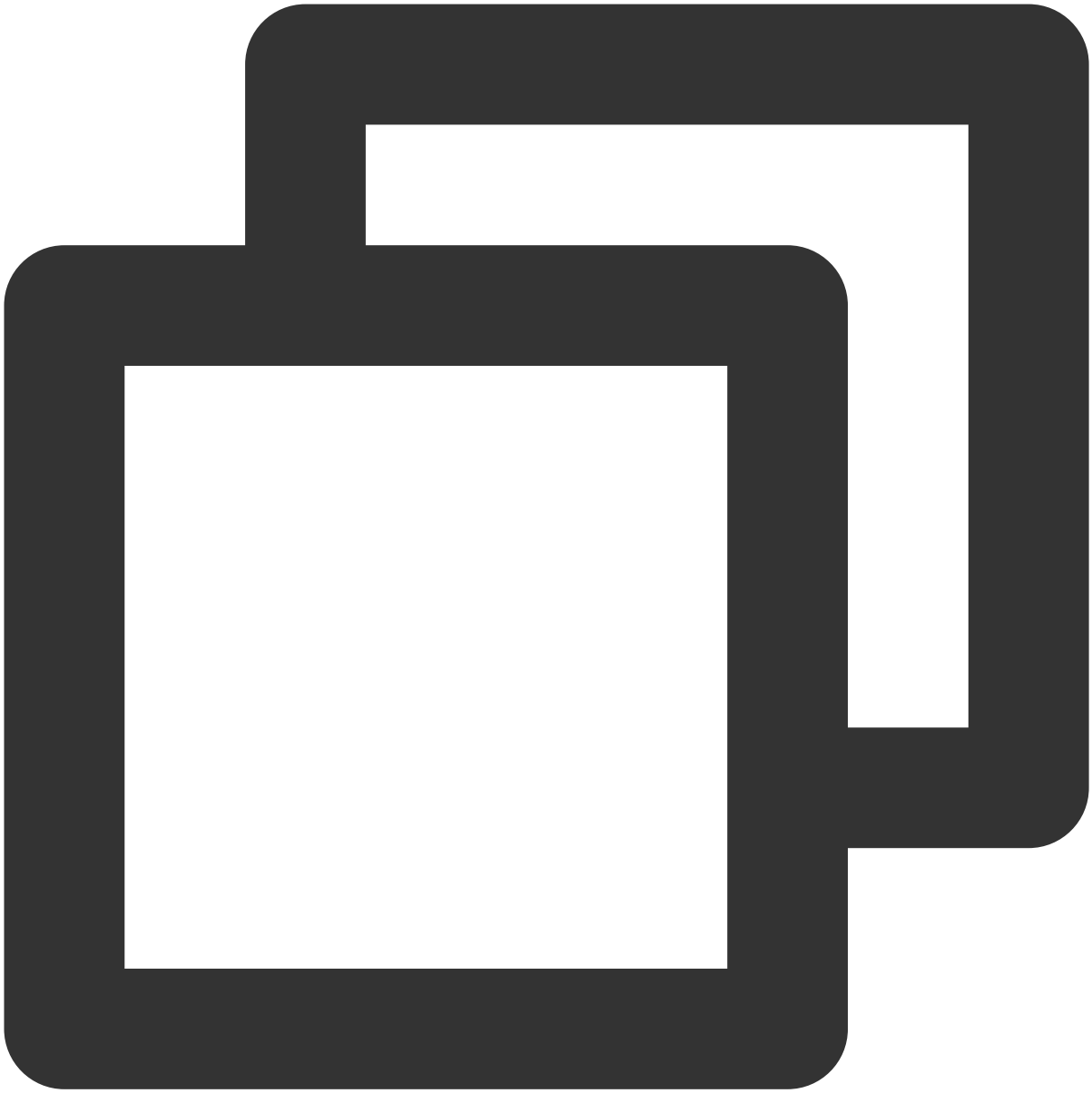
```
POST
```

Request Path



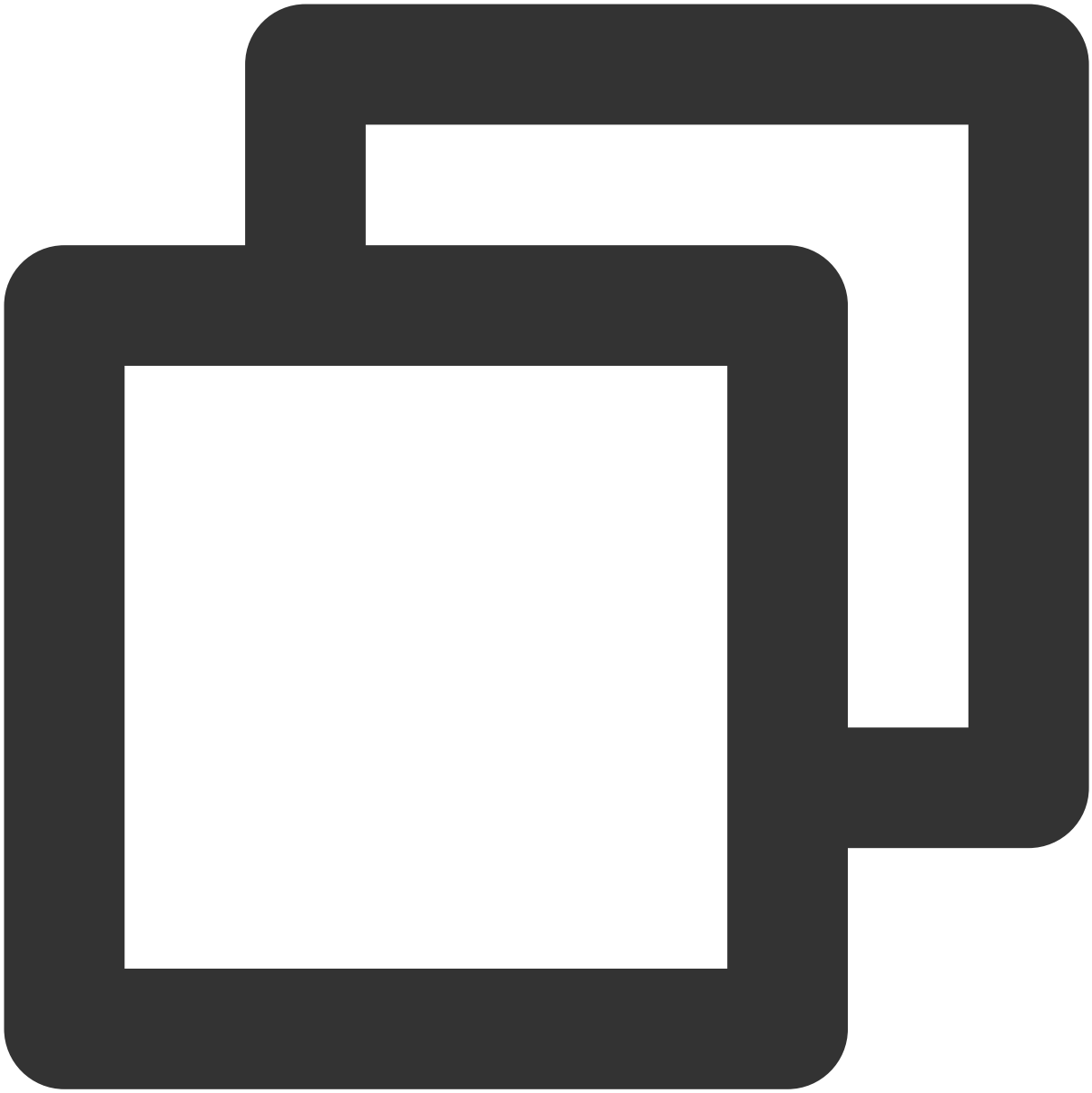
```
/oauth2/token
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests

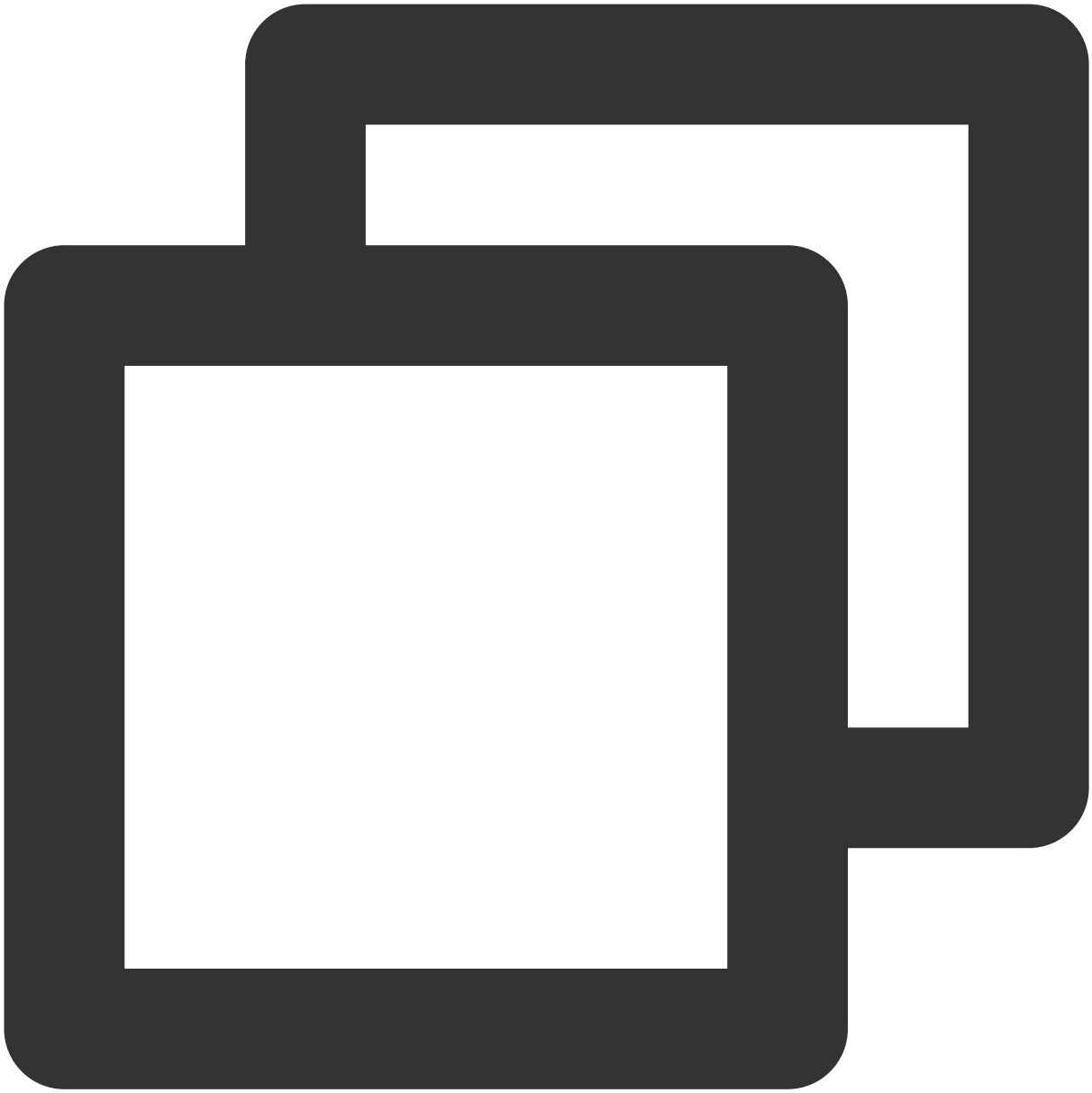


```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIEN
```

Request Parameters

Parameter	Optional	Description
grant_type	false	Fixed value: `client_credentials` .
client_id	false	The `client_id` of the application. Go to the **application management page** and **select the application**, and then click **Application Configuration** to find the **Client ID**.
client_secret	false	The `client_secret` of the application. Go to the **application management page** and **select the application**, and then click **Application Configuration** to find the **client_secret**.
scope	true	The `scope` used to apply for authorization. Multiple `scope` values are separated by spaces.

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiJmOTY5NGQ5My1kNTQxLTQ5ODUtODhkYy00MjIyOTg3MzAwOGUiLCJ
  "scope" : "identity_proofing",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

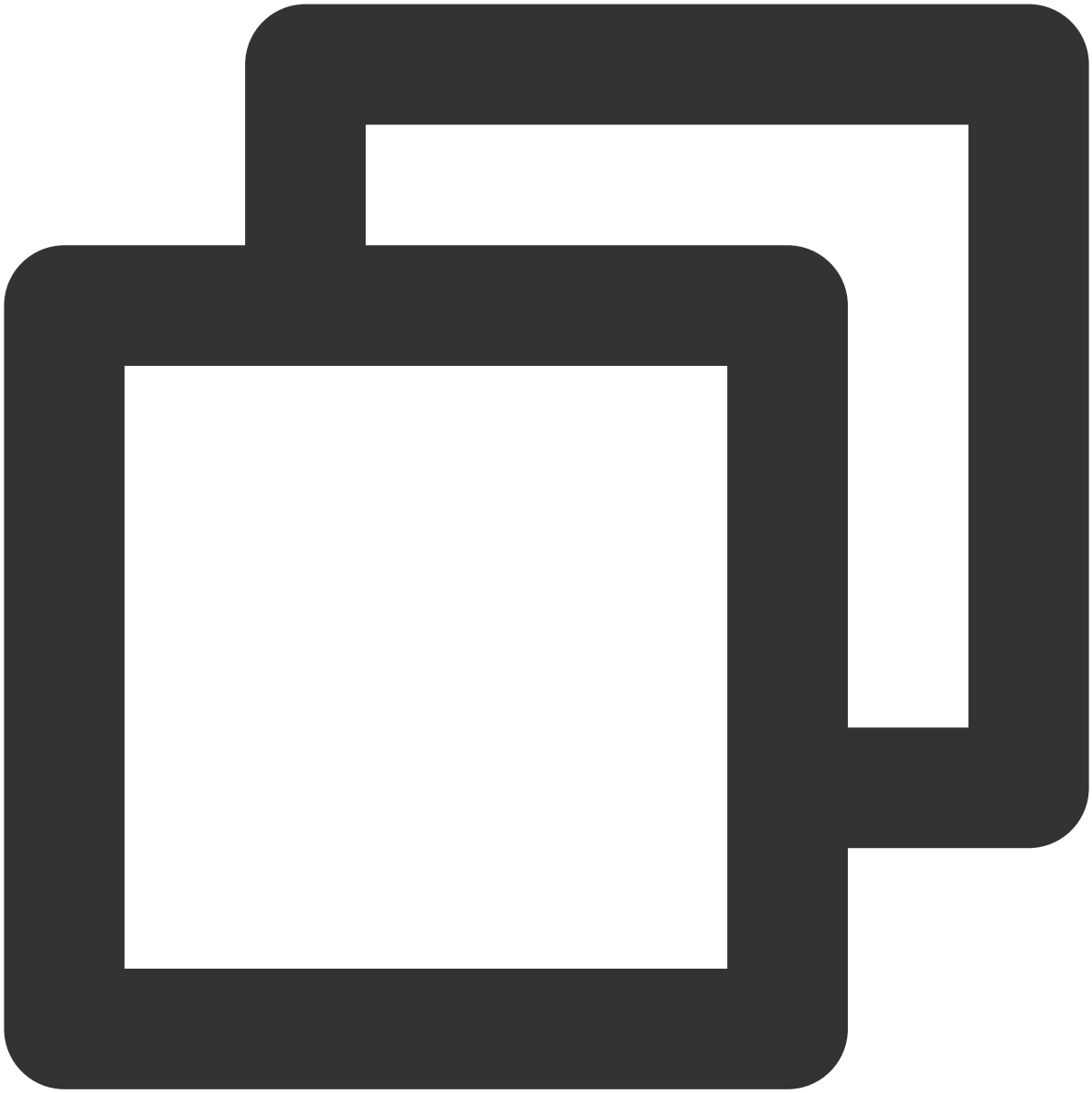
Response Parameters

Parameter	Data Type	Description
access_token	String	Access Token (JSON Web Token, or JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)
scope	String	Access Token scope.

Note:
The Refresh Token is not included in the response for client credentials mode. When the Access Token expires, the application calls this API again to get a new Access Token.

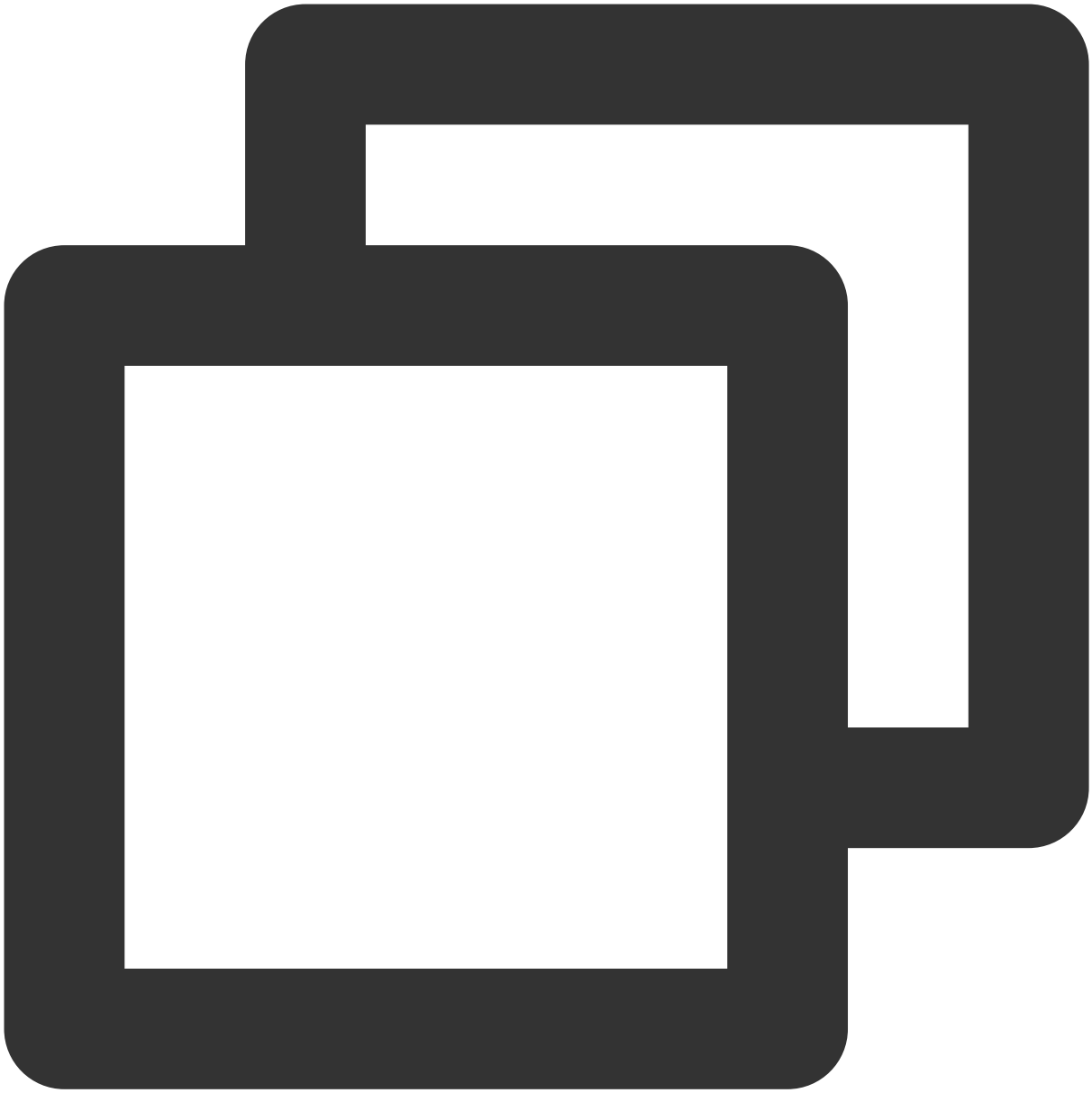
Sample Error Responses

The application does not exist or is not enabled; or the verification of the application key failed.



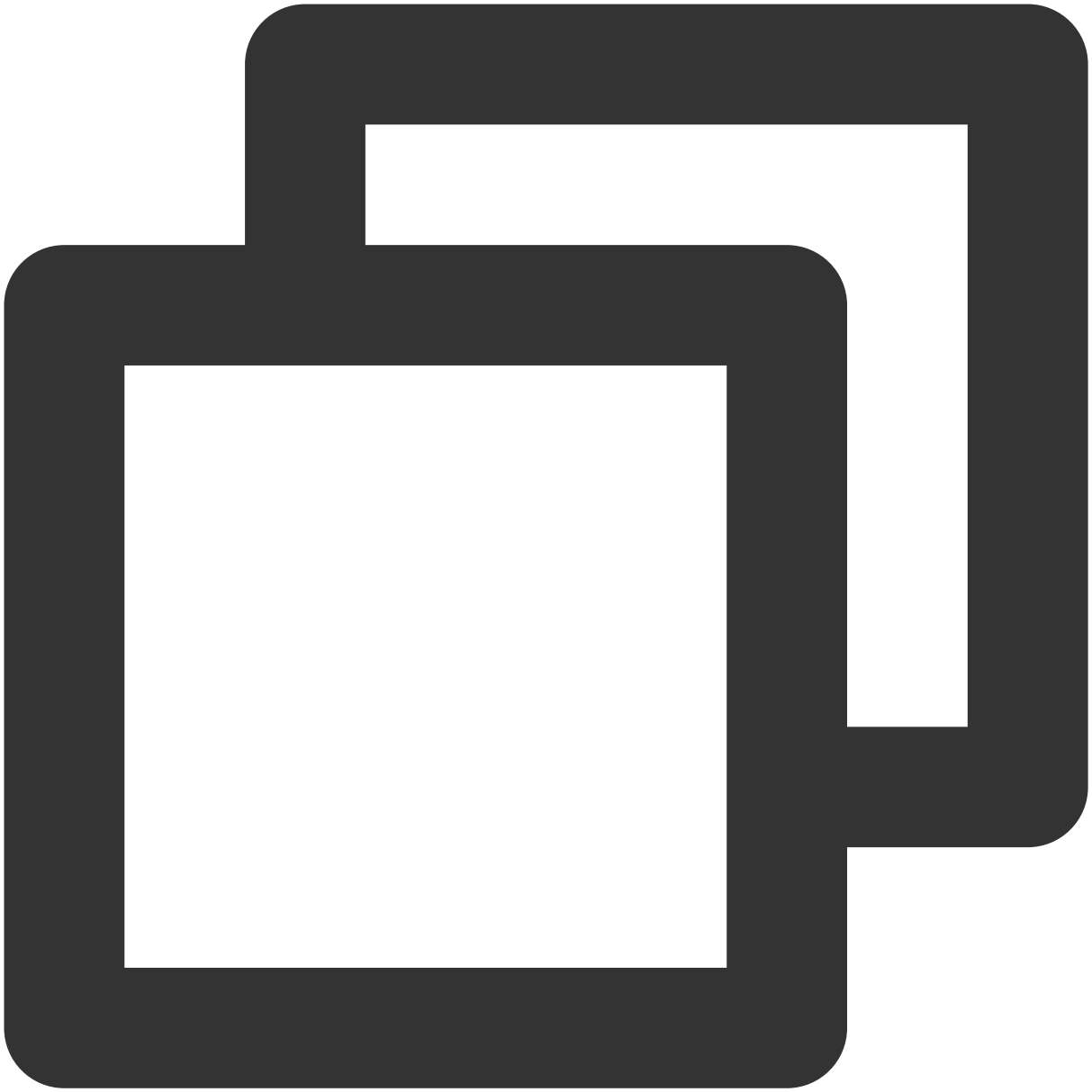
```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_client"
}
```

The application does not have permission to use `client_credentials` mode to get the Token.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "unauthorized_client"
}
```

Incorrect `scope` parameter or no permission.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_scope"
}
```

Get JWT Public Key

Last updated：2023-12-22 11:42:07

API Description

This API is used to use the JSON Web Token (JWT) public key to verify the ID Token and Access Token in JWT format.

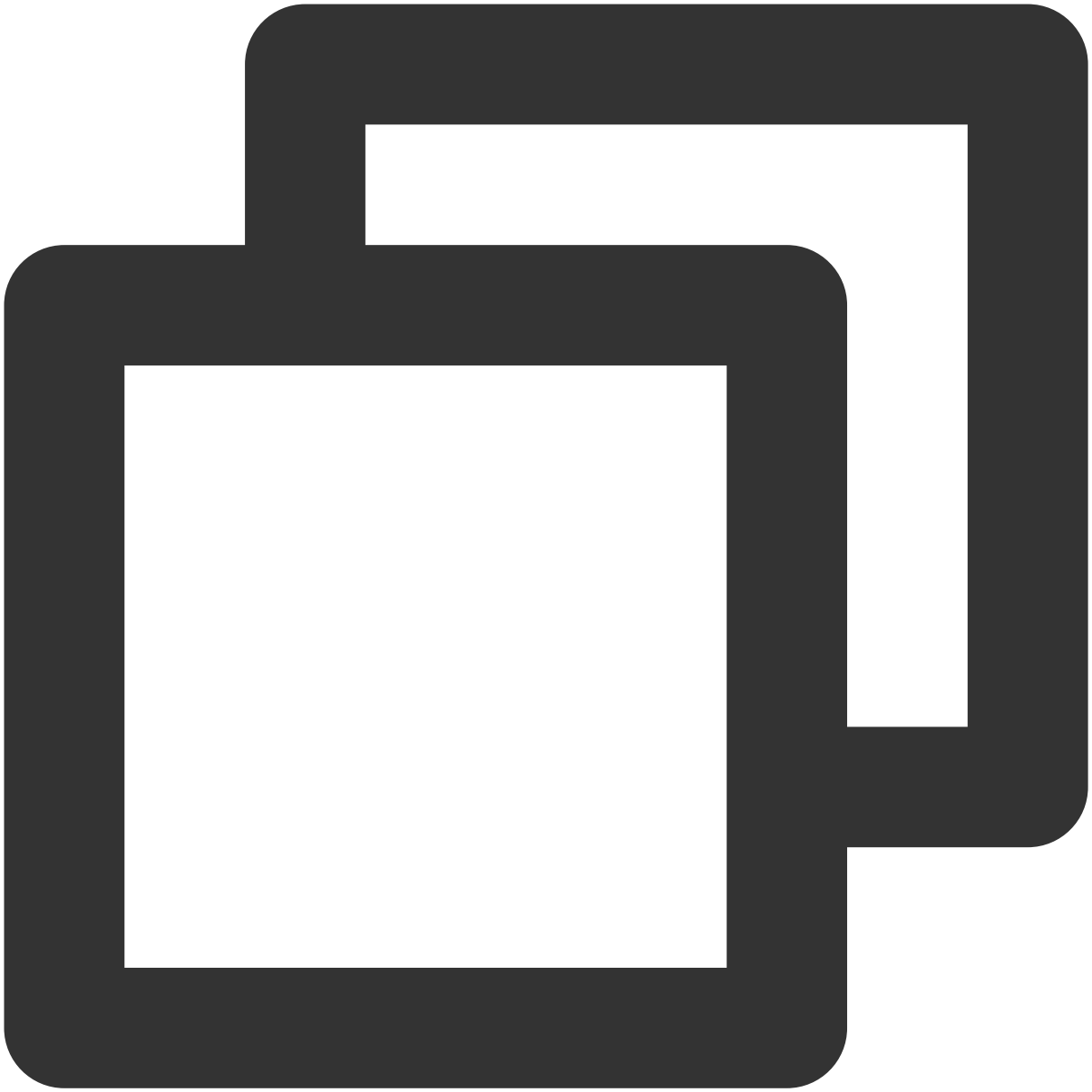
Note:

The JWT key is automatically generated when the user directory is created, and the key is different for different user directories.

Supported Applications

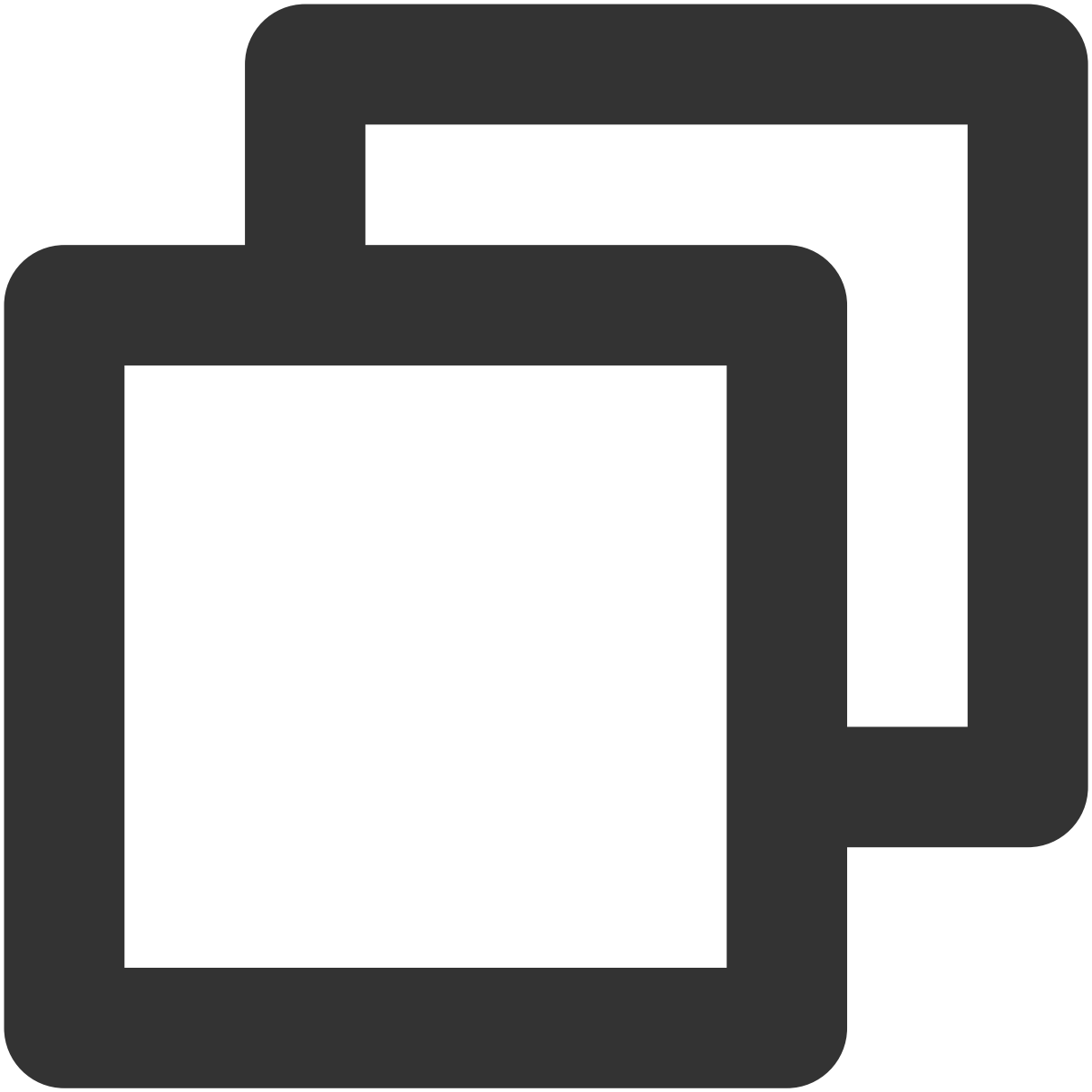
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



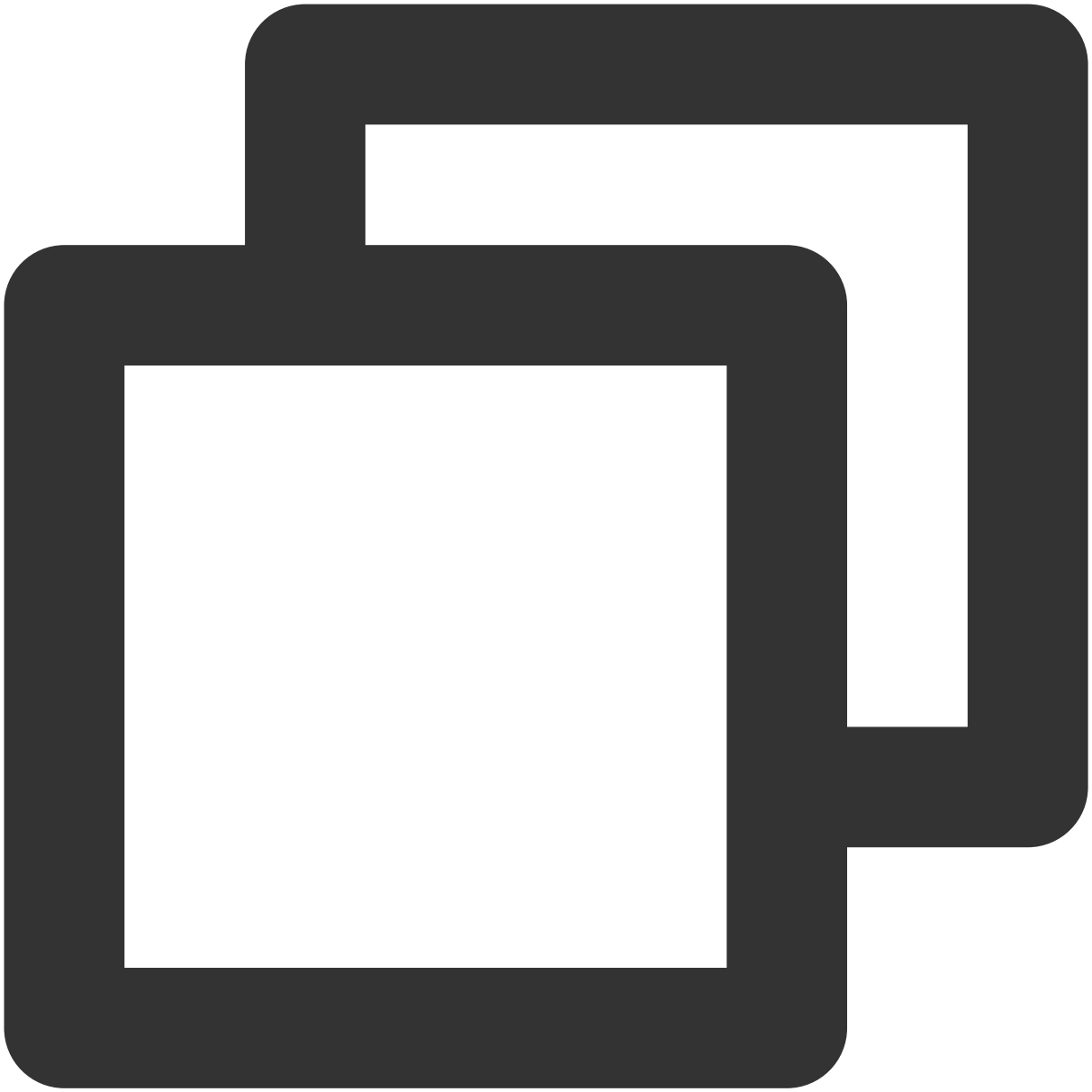
```
GET
```

Request Path



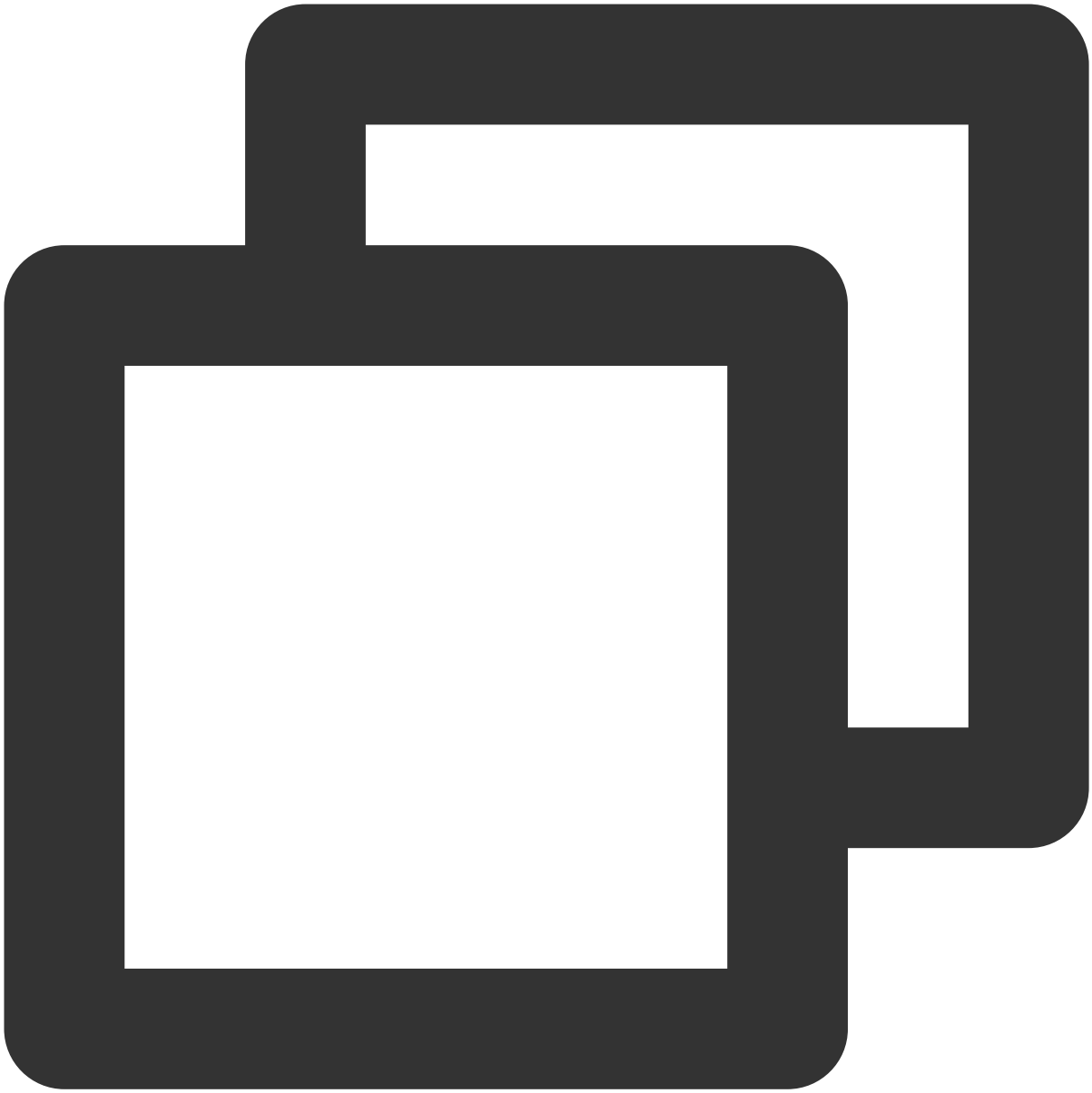
```
/oauth2/jwks
```

Sample Requests



```
GET /oauth2/jwks HTTP/1.1
Host: sample.portal.tencentciam.com
```

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "Keys" : [ {
    "kty" : "RSA",
    "e" : "AQAB",
    "kid" : "f9694d93-d541-4985-88dc-42229873008e",
    "n" : "wYmf-IL7_pXqEjtfHme7KqS06hRQ0BzhTzORjgwnsJD_CPexMHQAd82vZfOQioW9oaMXTiSA
  } ]
}
```

Response Parameters

Parameter	Data Type	Description
keys	Array	An array of public keys.
keys[].kty	String	Key type, such as RSA.
keys[].kid	String	Key identifier.
keys[].e	String	RSA public key.
keys[].n	String	RSA public key.

Refresh Token

Last updated：2023-12-22 11:43:32

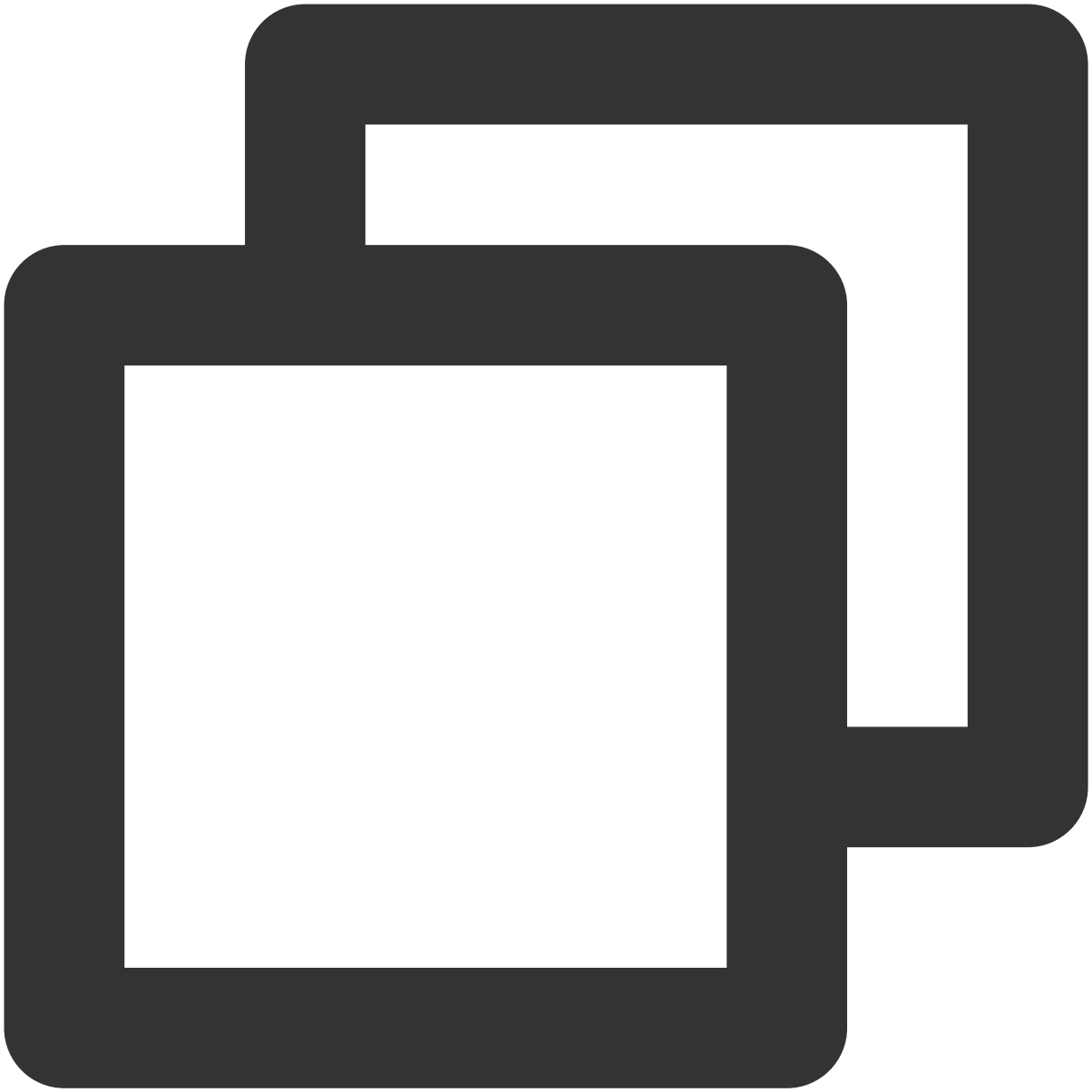
API Description

This API is used to get a new `access_token` using `refresh_token`.

Supported Applications

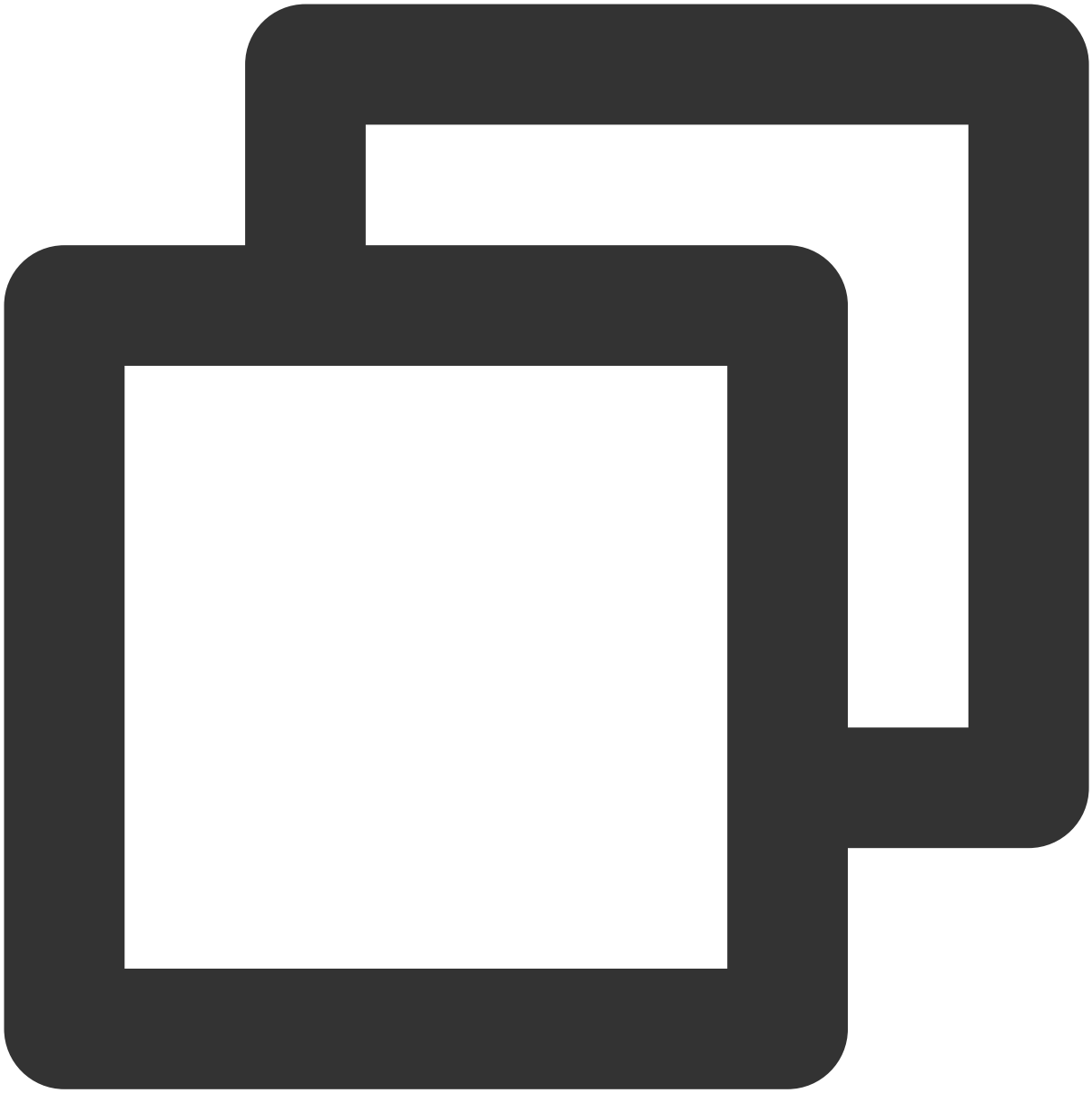
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



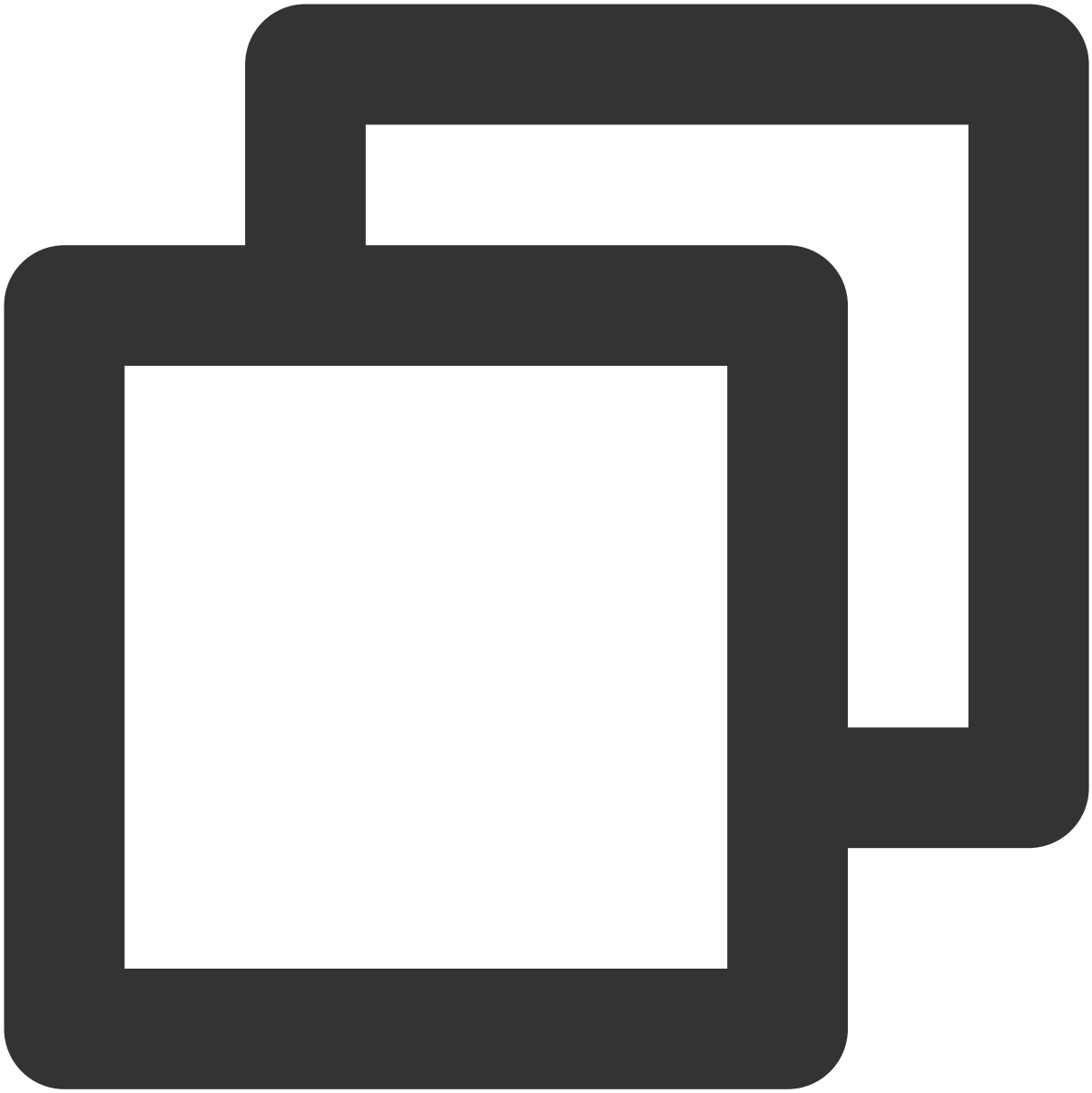
```
POST
```

Request Path



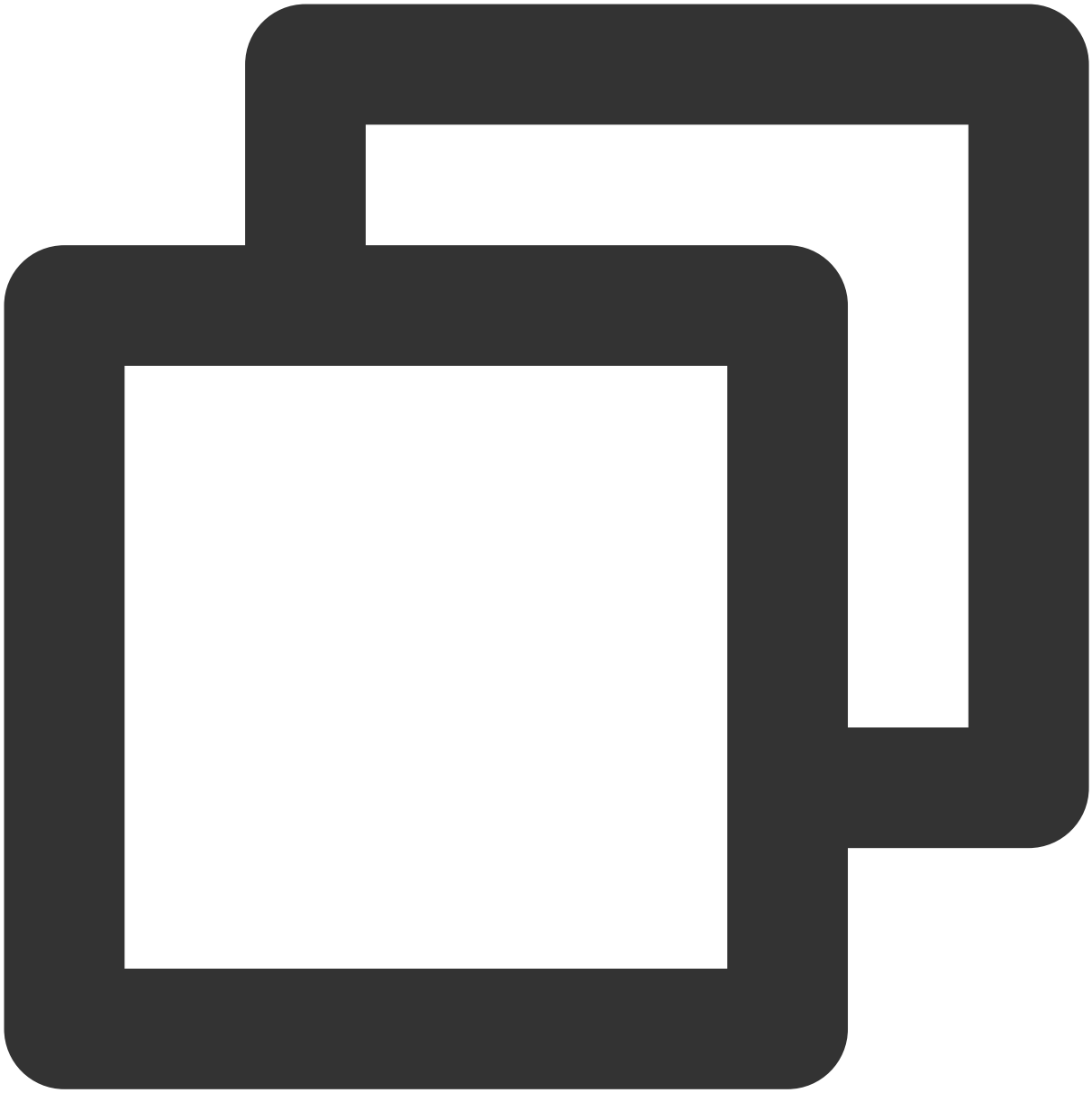
```
/oauth2/token
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests

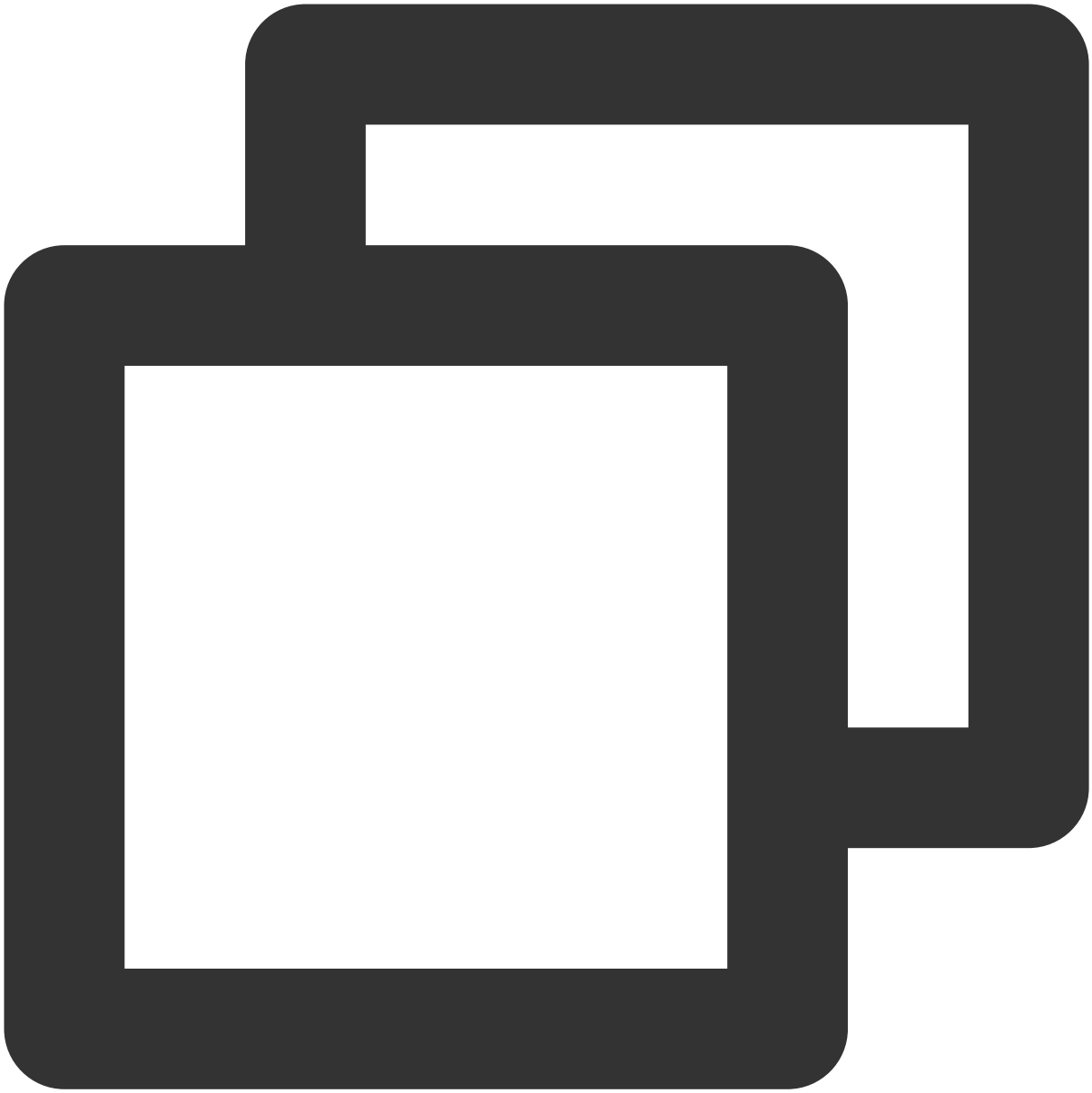


```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&grant_type=refresh_to
```

Request Parameters

Parameter	Optional	Description
client_id	false	The `client_id` of the application. This should be the same as the one used for getting the authorization code.
client_secret	false	The `client_secret` of the application. It can be viewed on the basic information page of the application on the Customer Identity Access Management (CIAM) console.
grant_type	false	Fixed value: `refresh_token` .
refresh_token	false	The `refresh_token` returned when the Token is requested.

Sample Success Responses



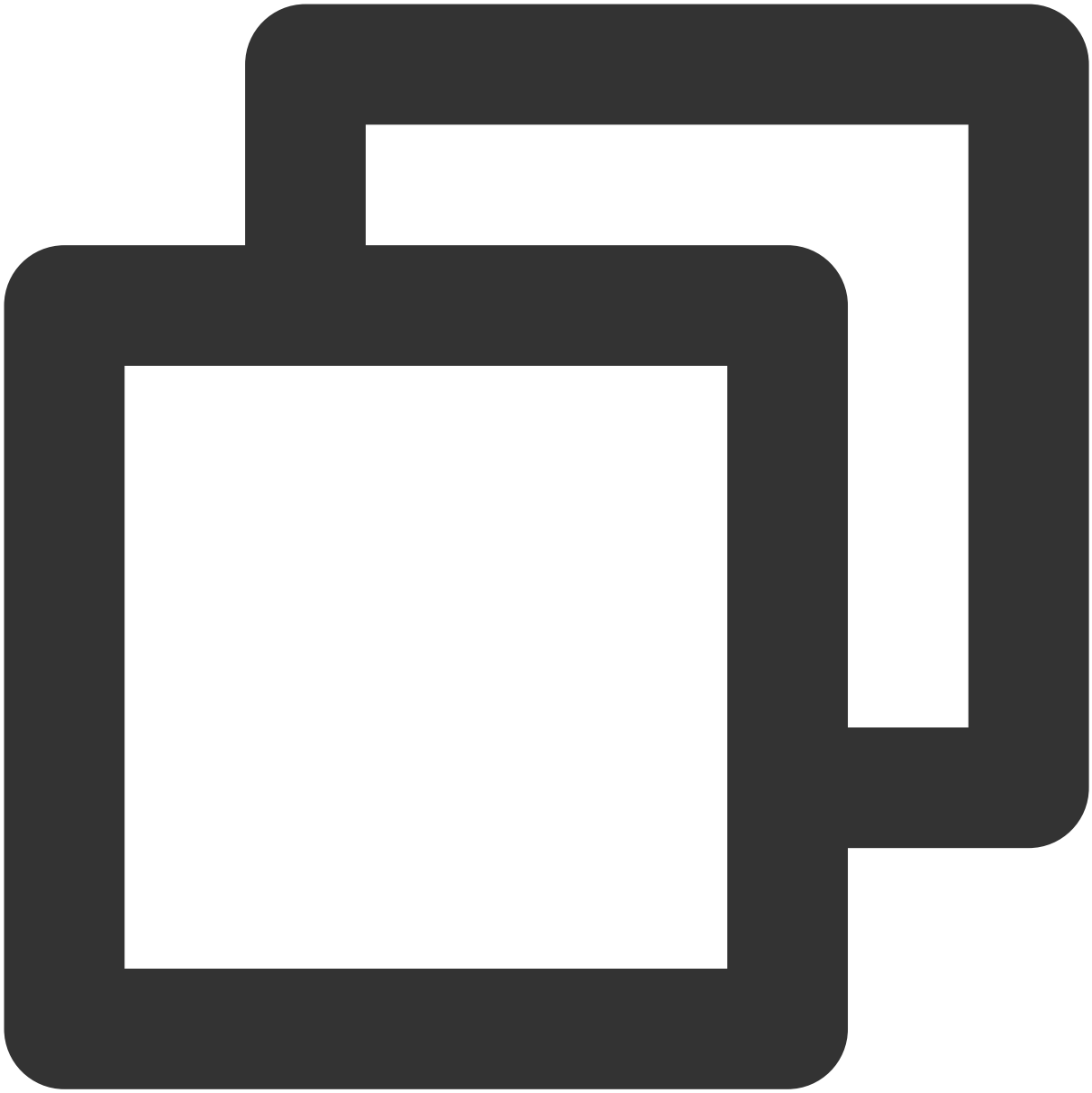
```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ
  "refresh_token" : "MOCK_REFRESH_TOKEN",
  "scope" : "openid",
  "id_token" : "eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXA
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

Response Parameters

Parameter	Data Type	Description
access_token	String	The refreshed OAuth 2.0 Access Token (JSON Web Token, or JWT).
refresh_token	String	The refreshed OAuth 2.0 Refresh Token.
scope	String	Access Token scope.
id_token	String	The refreshed OpenID Connect (OIDC) ID Token (JWT).
token_type	String	Token type. Fixed value: `Bearer` .
expires_in	Number	Validity period of Access Token (unit: sec)

Sample Error Responses

The `refresh_token` parameter is incorrect.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
  "error" : "invalid_grant"
}
```

Revoke Token

Last updated：2023-12-22 11:43:32

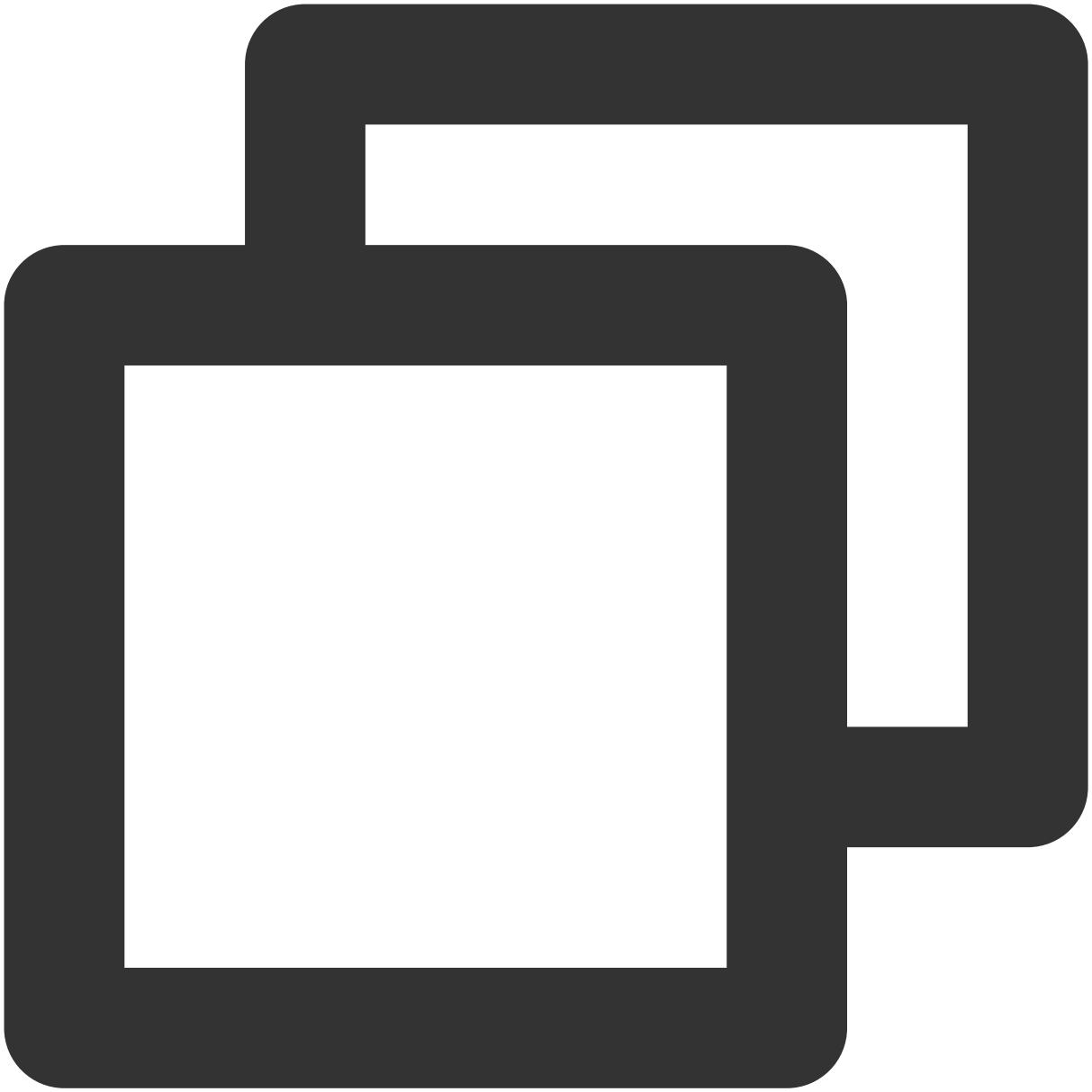
API Description

This API is used to revoke the OAuth 2.0 Token. If an `access_token` is passed, only the `access_token` will be revoked. If a `refresh_token` is passed, both the `refresh_token` and its associated `access_token` will be revoked.

Supported Applications

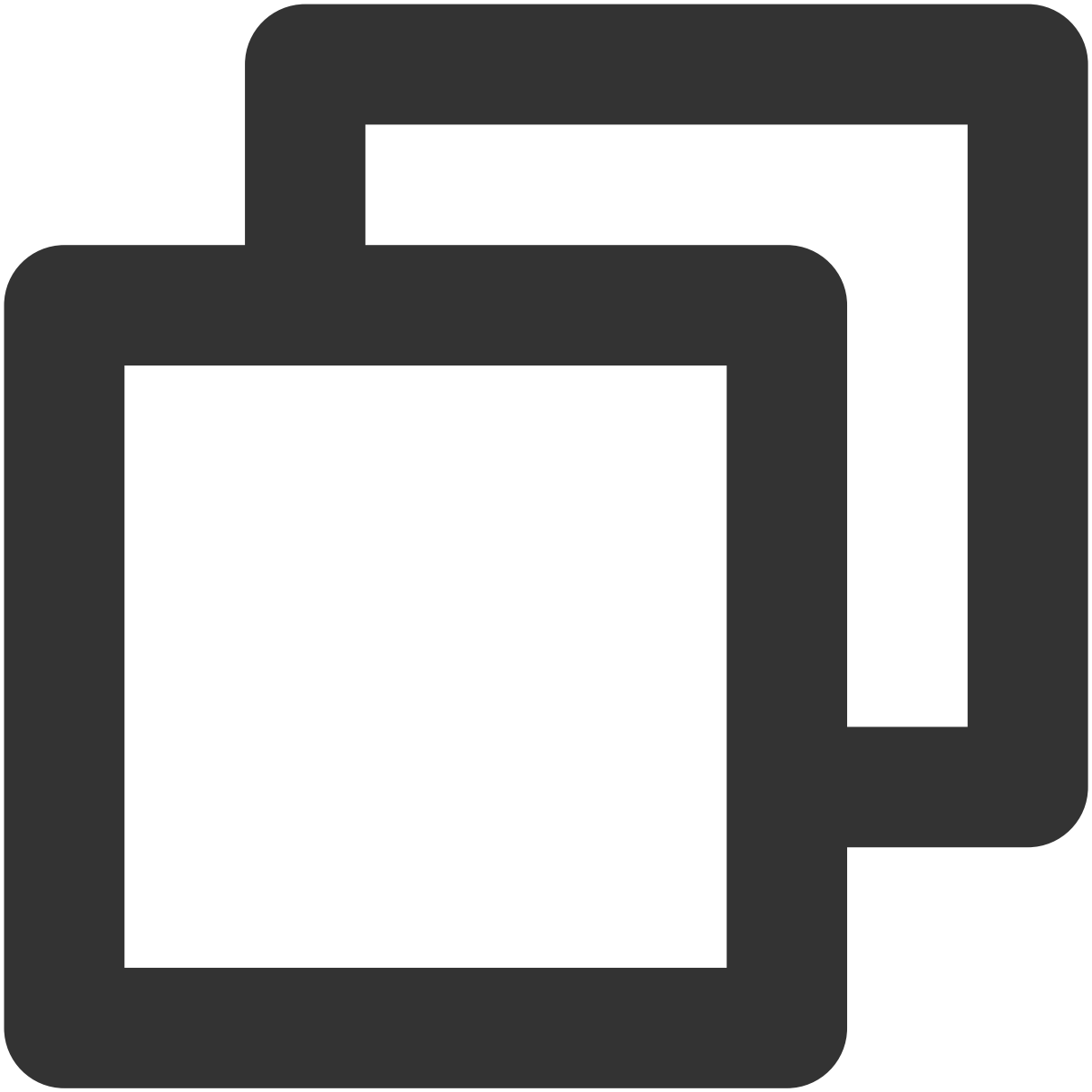
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



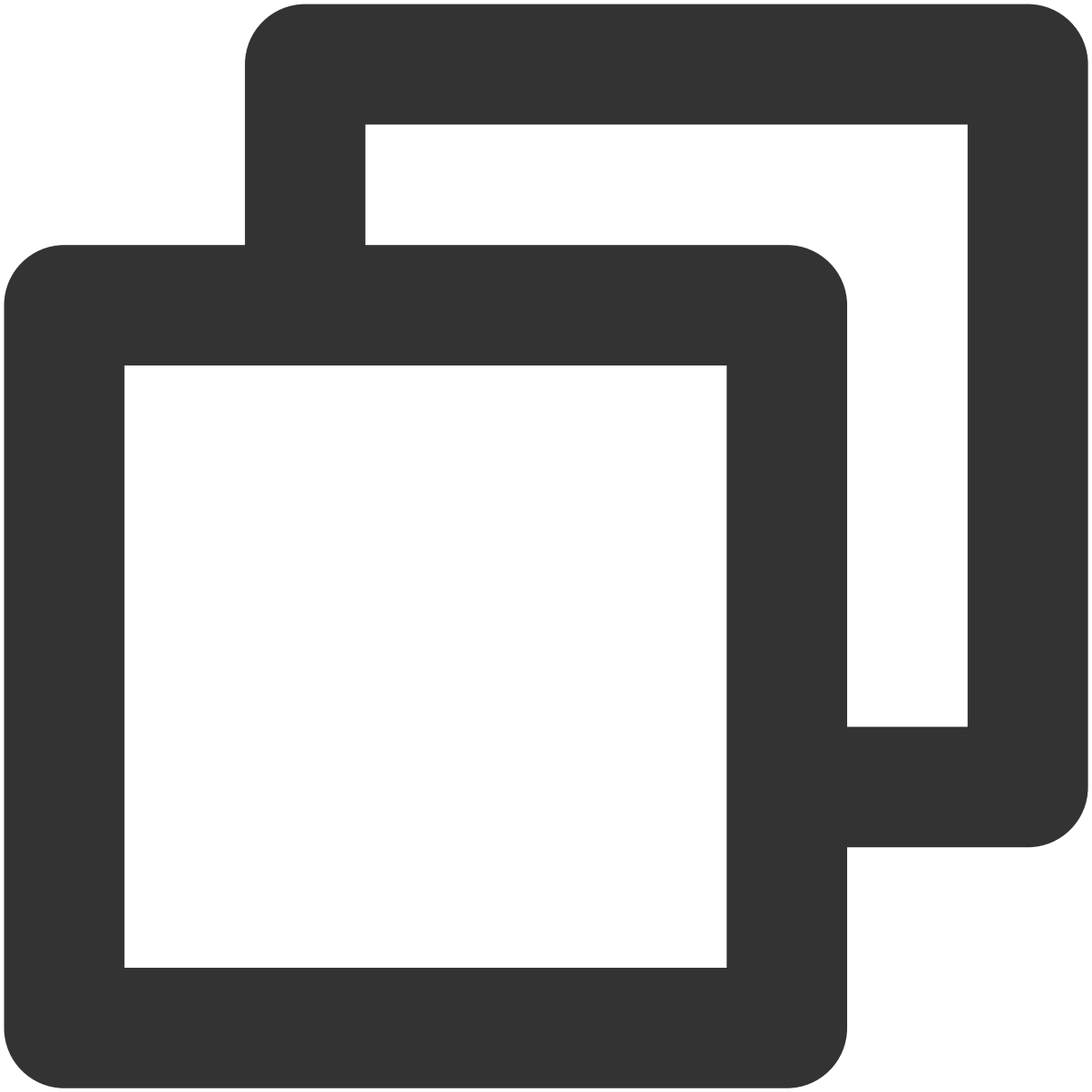
```
POST
```

Request Path



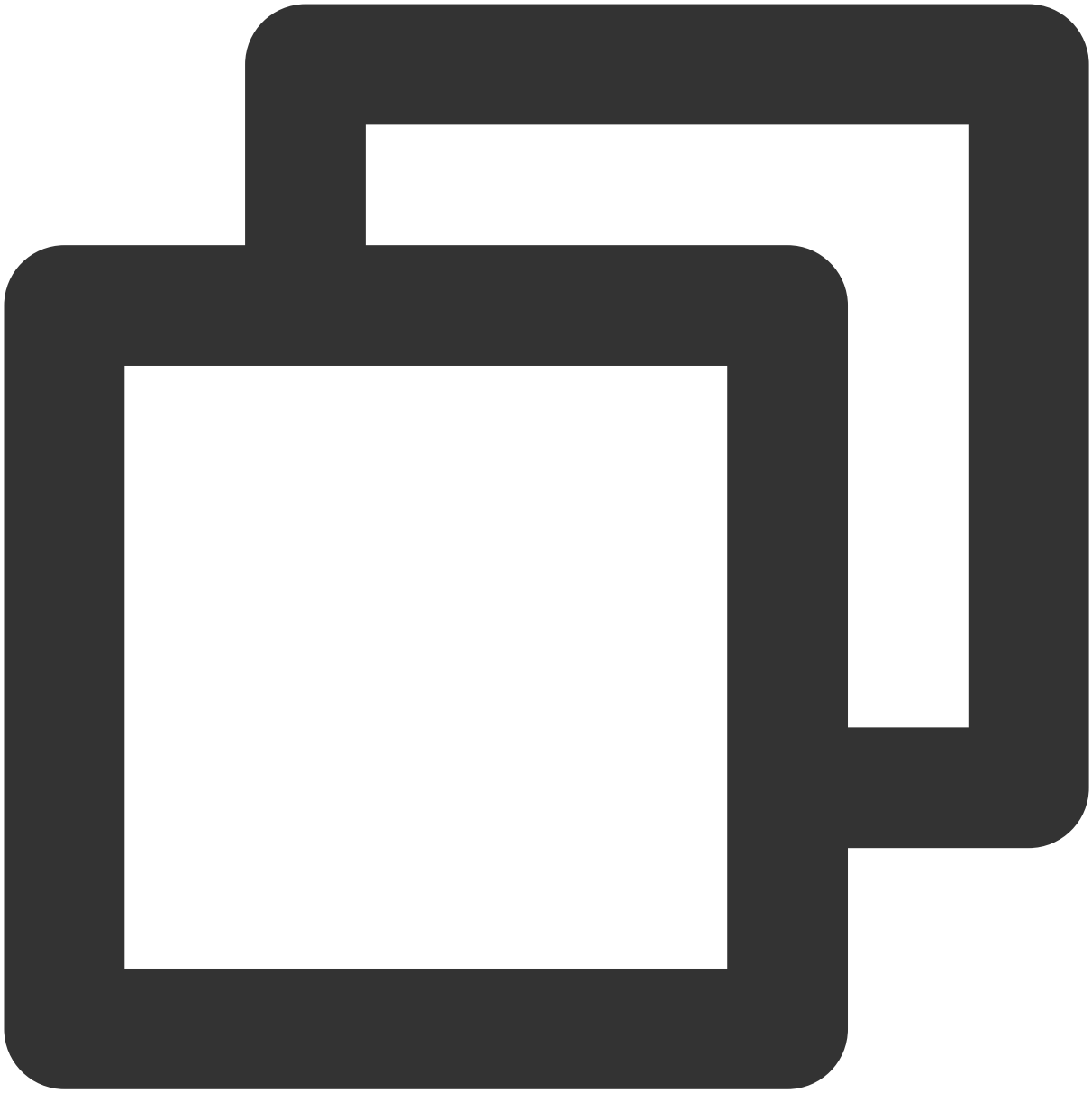
```
/oauth2/revoke
```

Request Content-Type



```
application/x-www-form-urlencoded
```

Sample Requests

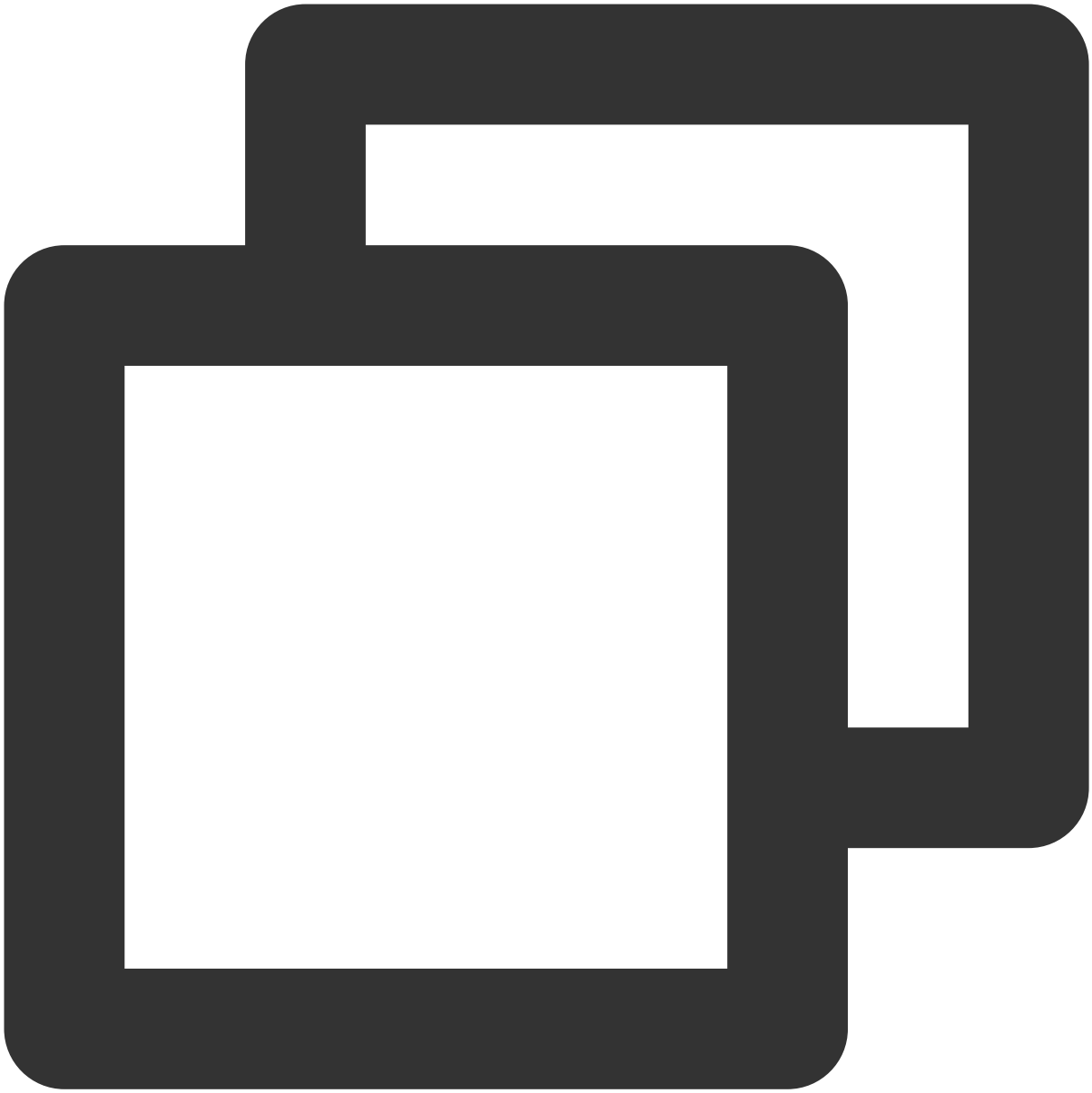


```
POST /oauth2/revoke HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&token=MOCK_ACCESS_TOK
```

Request Parameters

Parameter	Optional	Description
client_id	false	The `client_id` of the application. This should be the same as the one used for getting the authorization code and Token.
client_secret	false	The `client_secret` of the application. It can be viewed on the basic information page of the application on the Customer Identity Access Management (CIAM) console.
token	false	The value of `access_token` or `refresh_token`.

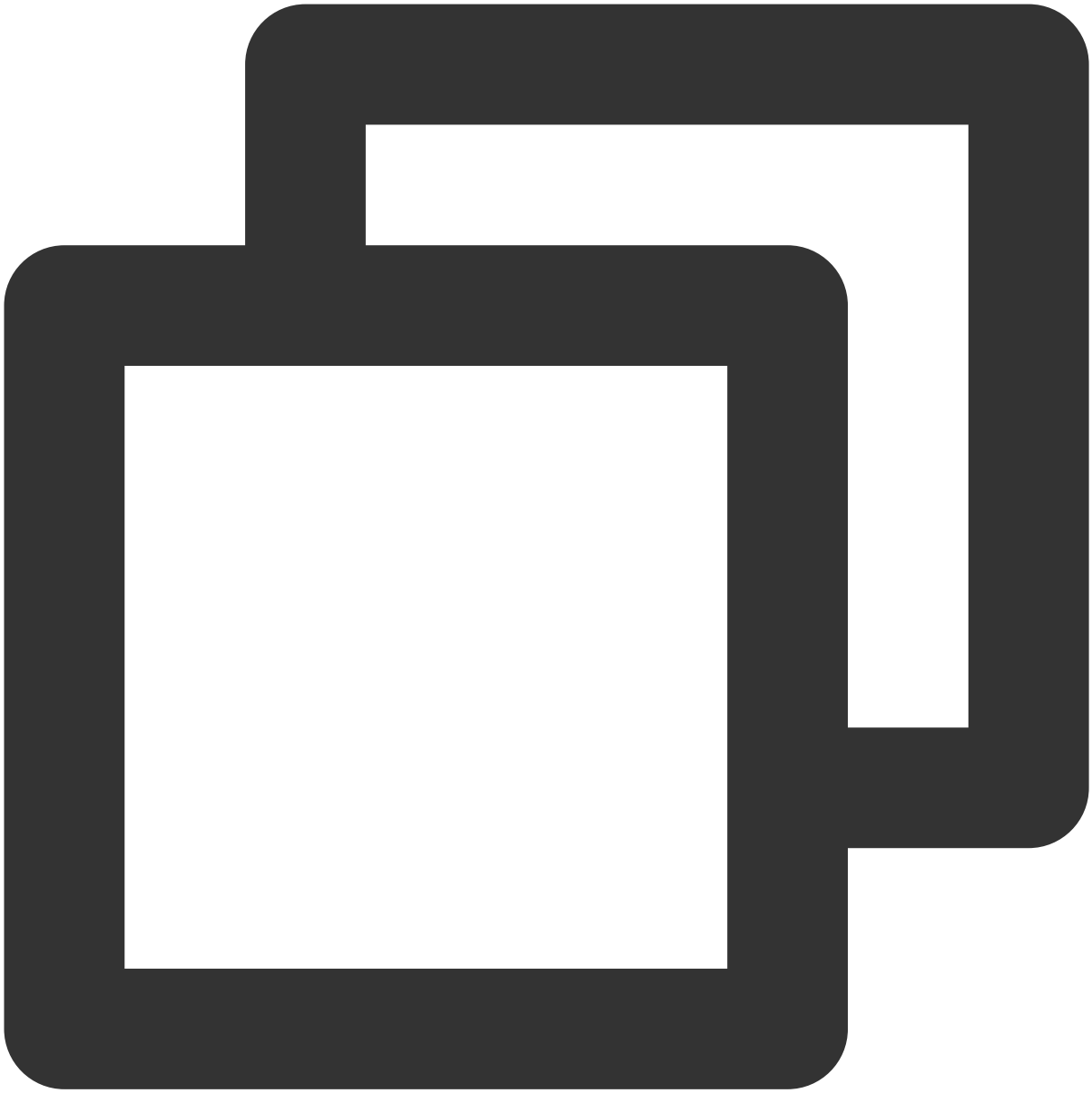
Sample Success Responses



```
HTTP/1.1 200 OK
```

Sample Error Responses

The `client_id` is not the same as the one used for initiating login and getting the Token.



```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
  "error" : "invalid_client"
}
```

Get OpenID Provider Configuration

Last updated：2023-12-22 11:43:32

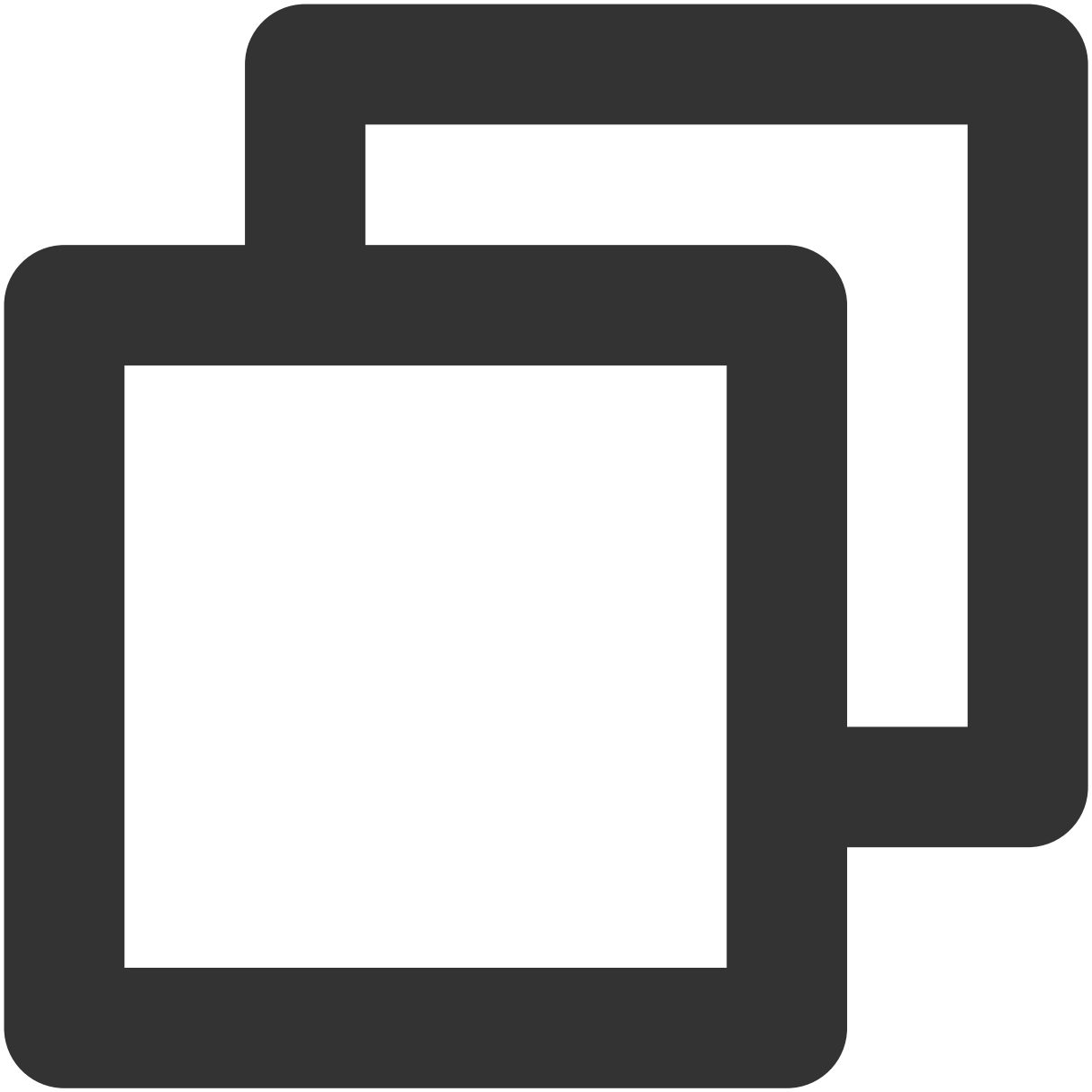
API Description

This API is used to get the configurations of the OpenID Connect (OIDC) authorization server for simplified local configurations. For more information about the configurations, see OpenID Connect Discovery.

Supported Applications

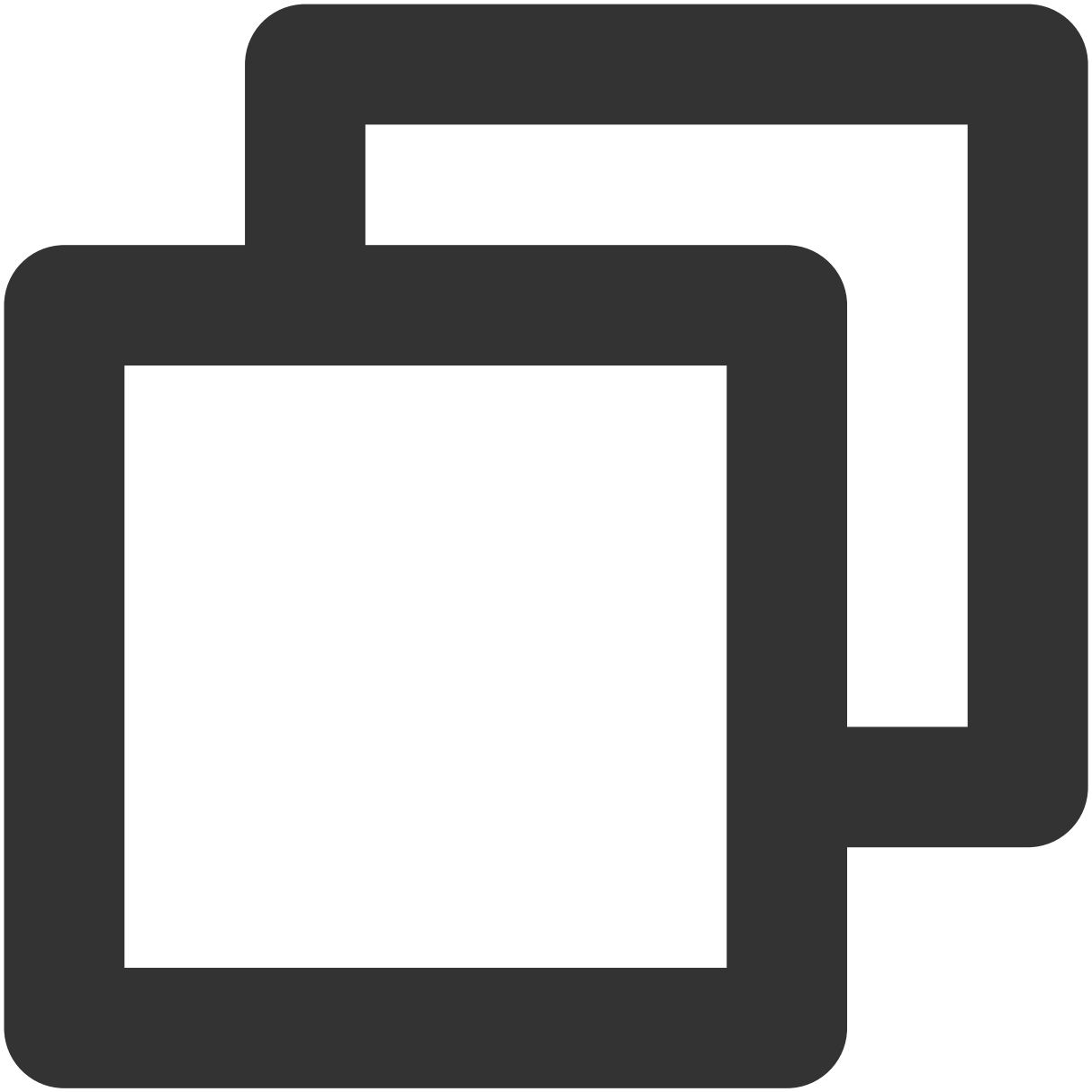
Web applications, single-page applications (SPA), mobile applications, and machine-to-machine (M2M) applications.

Request Method



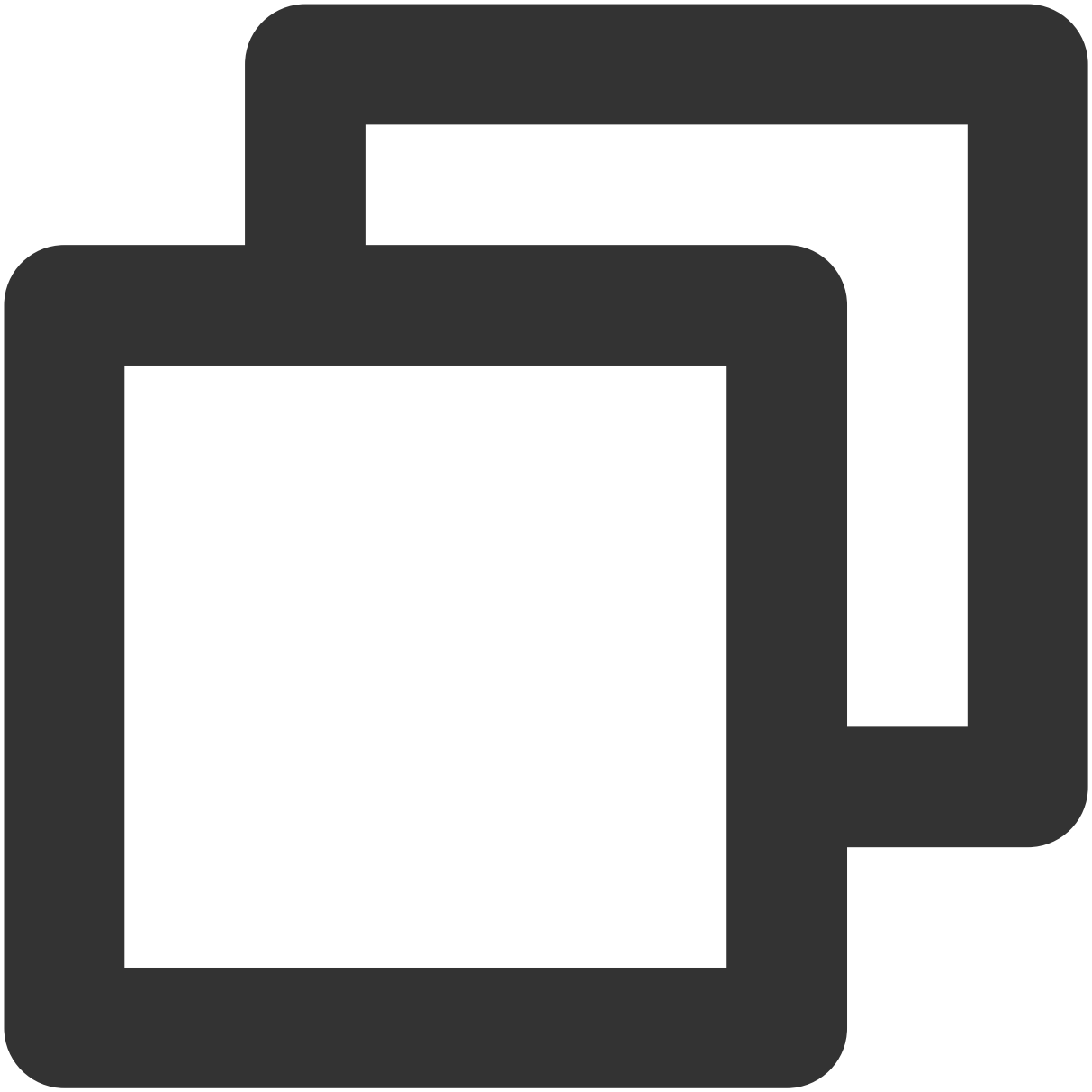
```
GET
```

Request Path



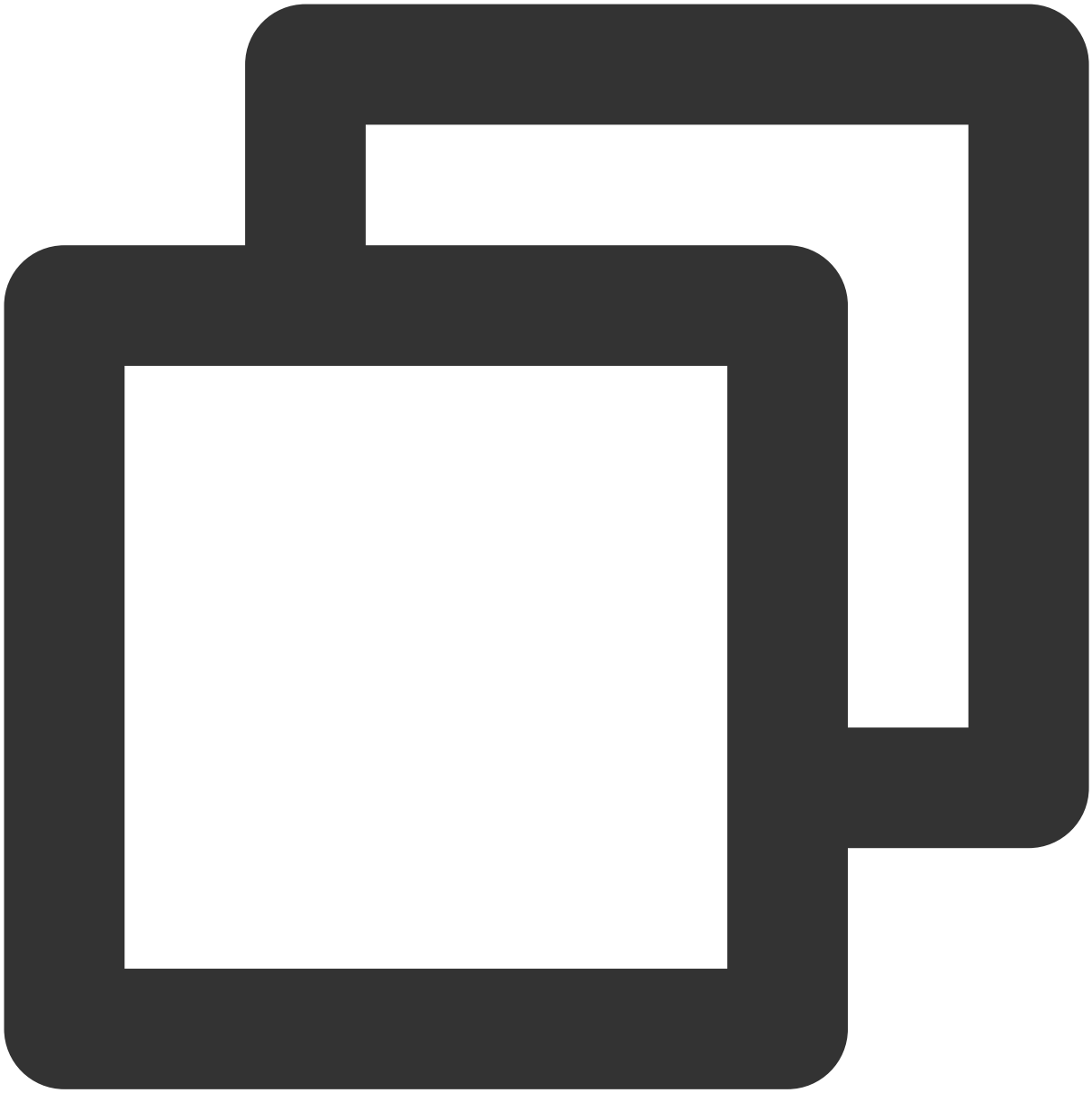
```
/.well-known/openid-configuration
```

Sample Requests



```
GET /.well-known/openid-configuration HTTP/1.1
Host: sample.portal.tencentciam.com
```

Sample Success Responses



```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "issuer" : "https://sample.portal.tencentciam.com",
  "authorization_endpoint" : "https://sample.portal.tencentciam.com/oauth2/authoriz
  "token_endpoint" : "https://sample.portal.tencentciam.com/oauth2/token",
  "token_endpoint_auth_methods_supported" : [ "client_secret_basic", "client_secret
  "jwks_uri" : "https://sample.portal.tencentciam.com/oauth2/jwks",
  "response_types_supported" : [ "code" ],
  "grant_types_supported" : [ "authorization_code", "client_credentials", "refresh_
```

```
  "subject_types_supported" : [ "public" ],
  "id_token_signing_alg_values_supported" : [ "RS256" ],
  "scopes_supported" : [ "openid" ]
}
```

Response Parameters

Parameter	Data Type	Description
issuer	String	OIDC issuer.
authorization_endpoint	String	The URL of the OAuth 2.0 authorization endpoint.
token_endpoint	String	The URL of the OAuth 2.0 token endpoint.
jwks_uri	String	The URL of the JSON Web Token (JWT) public key.
grant_types_supported	Array	Supported OAuth 2.0 Grant Type values.
response_types_supported	Array	Supported OAuth 2.0 Response Type values.
token_endpoint_auth_methods_supported	Array	The authentication methods supported by the OAuth 2.0 token endpoint.
subject_types_supported	Array	Supported OIDC Subject Identifier Type values.
id_token_signing_alg_values_supported	Array	Supported JSON Web Signature (JWS) algorithms.
scopes_supported	Array	Supported OAuth 2.0 scope values.