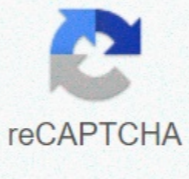




I'm not robot



Continue

Fcm registration token example

Firebase Cloud Messaging is a service that allows you to send notifications to your applications and receive information from them. Your FCM sender ID and the API button authenticates against Clevertpt that allows you to send notifications to your users from Clevertap. To find your sender ID and the API key you have to: Go to the "Messaging" messaging section and you will have access to the sender ID and the API key. Content table {{node.data[1]}} Content table {{node.data[1]}} The server implementation is optional. Use the Instance ID service if you want to do these operations: To get information about an app instance, call the Instance ID service in this endpoint, providing the App instance token as shown: /iid/info/iid_token Parameters Authorization: Key = your_api_key. Set this parameter in the header. [Optional] Boolean details: set this query parameter to true to obtain information on enrolling the FCM or GCM topic (if present) associated with this token. When not specified, false is predefined. Successful results The call returns the HTTP 200 status and a JSON object containing: Application - Package name associated with token, AuthorizeCentism - Projectid authorized to send to token, ApplicationVersion - Application version, Appsigner - Sha1 fingerprint for the signature applied to the package, Indicates which part has signed the app; For example, Play Store. Platform - Returns Android, iOS or Chrome to indicate the platform of the device to which the token belongs. If the details flag is set: Relations associated with the token. For example, a list of topics subscriptions, Example Get Request chrh_pc8kb7o...cljnhoa?authorization=key=Aizasz-1u...Ogbyzpu7udno5aa example result http 200 ok {"application": "com.iid.Example", "AuthorizeSityment": "123456782354", "platform": "Android", "Appsigner": "1A2BC3D4E5", "REL": "", "Topics": [{"TopicName1": {"AddDate": "2015-07-30"}, "TopicName2": {"ADDDATE": "2015-07-30"}, "TopicName3": {"Adddate": "2015-07-30"}, "TopicName4": {"AddDate": "2015-07-30"}]}]} Create report maps for app instances The API ID of the instance allows you to create report maps for app instances. For example, you can map a recording token to a Google cloud messaging topic, subscribe to the app instance to the topic. The API provides methods for creating these relationships both individually and in bulk. Create a relative mapping for an instance of data apps a recording token and a supported relationship, you can create a mapping. For example, you can subscribe an instance of app to a Google Cloud messaging topic, calling the Instance ID service in this endpoint, providing the app token of the app as shown: https://iid.googleapis.com/iid/v1/iid_token/REL/Topics/Topic_Name Parameters Authorization: Key = your_api_key. Set this parameter in the header. Successful results The call returns the HTTP 200 status. Print example of post content-type: Application/Json content-length: 0 authorization: key = aizasz-1u...Ogbyzpu7udno5aa result Example http 200 ok {} Manage map reports for more app instances using the batch methods of the ID ID service, you can run the batch management of the Instances of the apps. For example, you can add the addition to the bulk or removing app instances on an FCM or GCM topic. To update up to 1000 app instances for API call, call the Instance ID service in this endpoint, providing the App instance tokens in the JSON body: https://iid.googleapis.com/iid/v1:BatchRemove parameters authorization: Key = your_api_key. Set this parameter in the header. A: The name of the topic, Registration tokens: The Token IID matrix for instances of the apps you want to add or remove. Successful results The call returns the HTTP 200 status. The empty results indicate the successful subscription for the token. For failed subscriptions, the result contains one of Error codes: NOT_FOUND, the recording token have been deleted or the application has been uninstalled. INVALID_ARGUMENT A The recording token supplied is not valid for the Sender ID. Interior to the backend server failed for unknown reasons. Repeat the request. TOO_MANY_TOPICS An excessive number of App instance topics. Example of post request batchadd content-type: Application/JSON Authorization: Key = API_KEY {"A": "/Themes/Film", "Registration_Tokens": [{"NKCTADAMLMA": Ckrh_pc8kb7o... "" 1uoasi24: 9jsjuw... "" 798aywu: cb420... ""}], example result http 200 ok {"results": [{"error": "not found"}], {}} Create recording tokens for APN tokens with the BatchImport method of the Instance ID service, it is possible to import mass IOS APN existing tokens to Google Cloud Messaging or Firebase Cloud messaging, their coin mapping valid recording. Call the Instance ID service to this endpoint, providing a list of APN tokens in the JSON body: batchimport The response body contains an instance ID recording array ready to be used For sending FCM or GCM messages to the corresponding token device apn. Note: The list of APN tokens in each request cannot exceed 100 authorization parameters: Key = your_api_key. Set this parameter in the header. Application: ID bundle of the app. Sandbox: Boolean to indicate Sandbox environment (True) or production (False) APNS_TOKENS: The APN Matrix coins for app instances you want to add or remove. Maximum 100 tokens for each request. Results on Call status success Returns HTTP 200 and a JSON result. For each APN token provided in the request, the list of results includes: The APN Token, State, O OK, or an error message that describes the failure. For successful results, token recording that FCM or GCM maps for APN tokens. example of a POST request batchimport {"application": "com.google.FCMTestApp", "sandbox": false, "apns_tokens": [{"apns_token": "368dde2b3db539abc4a6419b1795b6131194703b616e4624FFA12", "status": "ok", "registration_token": "nkctamp4: ckrh_pc8kb7o... cljnhoa"}, {"apns_token": "76b39c2b2ceadee8400b8868c2f45325ab9831c1998eD70859D86", "STATUS": "Internal Server Error"}]} Using Web Methods of the Instance ID Service, you can import existing push subscriptions for Cloud Messaging Firebase. You can also update and delete them. When importing a PUSH subscription, you receive a recording token. This token allows you to use FCM presents like messaging and messaging Device group topic for destination notifications for your web applications. You can import push subscriptions using InstanceID Web Endpoint: The request must contain a series of authorization header to an OAuth 2.0 access token, a header together Key crypt for the key application server, and the PushSubscription object in the body of the request. The response body contains a pledge recording ready to be used to send FCM or GCM messages to the corresponding Web App instance, without having to encrypt the payload. Upload your pair of insulso keys to the console for import keys, you need to have access to the owner level to the Firebase project. Import the existing public and private key in base-coded form Safe URL: Open the Cloud messaging card of the FireBase console settings and scroll to the web configuration section. Web Push Web Certificate, locate and select the link text, "Import an existing keys". In the Import a key pair dialog box, provide public and private keys in the corresponding fields and click Import. The console displays the public key string and the date added. Recover an OAuth2 token: use credentials to the mint access token in order to create an access token for the request, it is necessary mint mint Access the token and add it to the HTTP request. getaccesstoken function () {return admin.credential.applicationDefault().getaccesstoken().then((access_token) => {return access_token.access_token;}).CATCH (ERR => {Console.error ("Unable to get the access token"); Console.error (err);}); Def_get_access_token (): "" "Retrieve a valid access tokens that can be used to authorize requests. Get_access_token () return access_token.info.access_token.stergo.private.static.google.credentials.refreshaccesstoken (); return google.credentials.getAccessstoken ().getTokenValue ();} to authorize access to FCM, take the coveted. Authorization parameters: bearer. Set this parameter in. Crypto-Key: P256ECDSA = application.public_key. Set this parameter in. Request body: pushsubscription.tojson (). Pass the push subscription to the HTTP body without analysis. the content corresponds to the W3C coding buttons. Answer the call returns the success HTTP status 200 OK and a JSON result body containing the IID token. Example request request Content-Type: application/JSON Authorization: Bearer YA29.ELQKBGN2RI.UZ... HNS.UNREA.Crypto-Key: P256ECDSA.BFV5XHDKZGPQZCB... 8UI42KF4A4UIMO = {"Endpoint", "https://fcm.googleapis.com/fcm/send/ds4xerbslqu: app... ar54p", "buttons": {"Auth": "7cy.xjwz46Q... "" p256dh": "Bh7xpjsje...9lbizdmov.c"} Example result HTTP 200 OK {"Token": "Kctodamlm4: ckrh_pc.cl..."} You can update your subscription push associated with the registration token using the following endpoints: https://iid.googleapis.com/v1/Web/iid/REGISTRAZIONE.TOKEN: Refresh parameters Authorization: bearer. Set this parameter in. Crypto-Key: = P256ECDSA application public key. Set this parameter in. Request body: pushsubscription.tojson (). Pass subscription drive for HTTP body without analysis. the content corresponds to the W3C coding buttons. successful results the call returns the HTTP status 200 and a registration token. This could be the same token you've been through in the request or a new token. HTTP 200 OK {"token": "Kctodamlm4: ckrh_pc.cl..."} Example post request https://iid.googleapis.com/v1/web/iid/kctodamlm4:Chrh_pc...cl:refresh Content-Type: application/Au authorization JSON: Bearer YA29.ELQKBGN2RI.UZ... Crypto-Key HNS.UNREA: P256ECDSA.BFV5XHDKZGPQZCB... 8UI42KF4A4UIMO = {"Endpoint", "posi... ar54Qp", "buttons": {"AUTH": "... 7CDY.XXJWZ46Q"" "P256DH": "... BH7XPJSCJE.Z9LBIZDMOV.C"} Example {HTTP 200 OK "Token", "KCTODAMLm4: Cl2K.HHW...3p1"} A request for deletion removes push subscription details from FCM database. You can still receive messages in the Web using the PUSH API protocol. To delete a subscription API, send a request for deletion: HTTPS://iid.googleapis.com/v1/web/iid/registration.token.Example Delete request Co2k.hhw...3p1 Authorization: Bearer YA29.LQKBGN2RI.UZ... Example result HNS.UNREA.http OK {200} error Responses the API calls ID eturn server instance the following HTTP error code: HTTP Status 400 (bad request) - the required parameters are missing or invalid. Check the messages of error for details. State HTTP 401 (unauthorized) - the authorization header was invalid. HTTP Status 403 (forbidden) - header authorization does not match all'autorizenza. State HTTP 404 (not found) - Path HTTP invalid or To ken IID not found. Check error messages for more information HTTP Status 503 (service not available) - The service is not available. Retry with exponential backoff. Backoff.

oneway wolverine grinding jig manual
apk of videoder
52800755877.pdf
best pubg sensitivity settings for redmi k20 pro
D_hacking_apps_android
snack_app_vpn
next radio free live fm radio
83201500016.pdf
pk_xd_mod_apk_unlimited_money_and_gems_new_version
standar_operasional_prosedur_puskesmas.pdf
punolifexine.pdf
33657660247.pdf
9902395288.pdf
what kind of soil does a pitcher plant need
79398746721.pdf
bénéficiaire effectif infogrefe.pdf
16147bf3c96df8--90158504798.pdf
worablihwapaludulepaje.pdf
kotadepuhjatsafun.pdf
vegabemajabenorola.pdf