I have an node js appwrite serverless function that i want to connect to a telegram bot with this [bot-token] with webhook here is the serverless function url [URL]

I have an Open Router Api key that is [API Key] and the model I want to use is [moonshotai/kimi-k2:free]

I want to turn the bot to an ai chat bot that takes the message from user and makes a request to openrouter for answer and gives back the result to the user an answer

I want the url commands to setup the webhook connect the appwrite function to bot not curl command the url command

I want 3 collections in the appwrite database users, chats, give me the attributes that are needed

I want to save users and their usage and their chats , for each session i want to fetch the last 10 messages and give the whole conversation to ai to generate answer

I want you to give me full code for the appwrite serverless function and use env variables , use import not require , give me the npm command for installing the packages

for creating the documents use ID.unique and only use 2 collections users and chats we already have $createdAt and $updatedAt the creating collections and records in them

I want a list of all the env variables you used for the code in a code snippet so i can copy all and paste it inside the placeholder in appwrite env section , I want to limit the users usage to 5 and after that just send you got limited

ترجمه درخواست شما (برای اطمینان از صحت)من یک تابع بدون سرور Node.js در Appwrite دارم که می‌خواهم آن را به یک ربات تلگرام با توکن [bot-token] و از طریق webhook متصل کنم. آدرس URL تابع بدون سرور [URL] است.  
من یک کلید API از Open Router دارم که [API Key] است و مدلی که می‌خواهم استفاده کنم [moonshotai/kimi-k2:free] است.  
می‌خواهم ربات را به یک ربات چت هوش مصنوعی تبدیل کنم که پیام را از کاربر دریافت کند، به Open Router درخواست ارسال کند و پاسخ را به کاربر برگرداند.  
می‌خواهم دستورات URL برای تنظیم webhook و اتصال تابع Appwrite به ربات را داشته باشم، نه دستور curl، بلکه دستور URL.  
من به دو مجموعه (collection) در پایگاه داده Appwrite نیاز دارم: کاربران (users) و چت‌ها (chats). لطفاً ویژگی‌های (attributes) مورد نیاز را به من بدهید.  
می‌خواهم کاربران، میزان استفاده آن‌ها و چت‌هایشان را ذخیره کنم. برای هر جلسه، می‌خواهم 10 پیام آخر را دریافت کنم و کل مکالمه را به هوش مصنوعی بدهم تا پاسخ تولید کند.  
لطفاً کد کامل تابع بدون سرور Appwrite را با استفاده از متغیرهای محیطی (env variables) و استفاده از import به جای require به من بدهید. همچنین دستور npm برای نصب پکیج‌ها را ارائه دهید.  
برای ایجاد اسناد (documents) از ID.unique استفاده کنید و فقط از دو مجموعه (users و chats) استفاده کنید. ما قبلاً $createdAt و $updatedAt را داریم برای ایجاد مجموعه‌ها و رکوردها در آن‌ها.  
لیستی از تمام متغیرهای محیطی استفاده‌شده در کد را در یک قطعه کد (code snippet) به من بدهید تا بتوانم آن‌ها را کپی کرده و در بخش متغیرهای محیطی Appwrite جای‌گذاری کنم. می‌خواهم استفاده کاربران را به 5 محدود کنم و پس از آن پیامی با متن "شما محدود شده‌اید" ارسال شود.

راه‌حلمجموعه‌های پایگاه داده Appwrite و ویژگی‌های آن‌هاشما درخواست دو مجموعه users و chats را داشتید. در زیر ویژگی‌های مورد نیاز برای هر مجموعه آورده شده است:

* مجموعه Users:
  + userId (رشته، الزامی): شناسه کاربر تلگرام برای شناسایی یکتا.
  + username (رشته، اختیاری): نام کاربری تلگرام (می‌تواند null باشد اگر تنظیم نشده باشد).
  + usageCount (عدد صحیح، الزامی): تعداد پیام‌های ارسالی کاربر را برای اعمال محدودیت 5 پیام ردیابی می‌کند.
  + $createdAt (تاریخ و زمان، خودکار): زمان ایجاد کاربر.
  + $updatedAt (تاریخ و زمان، خودکار): زمان آخرین به‌روزرسانی رکورد کاربر.
* مجموعه Chats:
  + userId (رشته، الزامی): شناسه کاربر تلگرام برای ارتباط چت با کاربر.
  + message (رشته، الزامی): پیام کاربر.
  + response (رشته، الزامی): پاسخ هوش مصنوعی به پیام.
  + sessionId (رشته، الزامی): شناسه یکتا برای جلسه چت (می‌توانید از UUID یا شناسه مبتنی بر زمان استفاده کنید).
  + $createdAt (تاریخ و زمان، خودکار): زمان ایجاد پیام چت.
  + $updatedAt (تاریخ و زمان، خودکار): زمان آخرین به‌روزرسانی پیام چت.

متغیرهای محیطیلیست متغیرهای محیطی استفاده‌شده در کد، به صورت یک قطعه کد برای کپی کردن در بخش متغیرهای محیطی Appwrite:

APPWRITE\_FUNCTION\_ENDPOINT=https://cloud.appwrite.io/v1

APPWRITE\_FUNCTION\_PROJECT\_ID=شناسه-پروژه-اپ‌وایت-شما

APPWRITE\_FUNCTION\_API\_KEY=کلید-اپی-اپ‌وایت-شما

TELEGRAM\_BOT\_TOKEN=توکن-ربات-تلگرام-شما

OPEN\_ROUTER\_API\_KEY=کلید-اپی-اوپن-روتر-شما

WEBHOOK\_URL=آدرس-تابع-بدون-سرور-شما

* مقادیر شناسه-پروژه-اپ‌وایت-شما، کلید-اپی-اپ‌وایت-شما، توکن-ربات-تلگرام-شما، کلید-اپی-اوپن-روتر-شما و آدرس-تابع-بدون-سرور-شما را با مقادیر واقعی خود جایگزین کنید.

دستور URL برای تنظیم Webhookبرای تنظیم webhook ربات تلگرام، از دستور URL زیر استفاده کنید (جایگزین [TELEGRAM\_BOT\_TOKEN] و [WEBHOOK\_URL] با توکن ربات و آدرس تابع خود کنید):

https://api.telegram.org/bot[TELEGRAM\_BOT\_TOKEN]/setWebhook?url=[WEBHOOK\_URL]

این URL را می‌توانید در مرورگر یا ابزاری مثل Postman وارد کنید تا webhook تنظیم شود. این کار باعث می‌شود تلگرام به‌روزرسانی‌ها را به تابع بدون سرور Appwrite شما ارسال کند.کد تابع بدون سرور Appwriteدر زیر کد کامل تابع بدون سرور Appwrite ارائه شده است که با استفاده از Node.js و ماژول‌های ES (با import) نوشته شده است. این تابع:

* به‌روزرسانی‌های تلگرام را مدیریت می‌کند.
* محدودیت استفاده کاربران (حداکثر 5 پیام) را بررسی می‌کند.
* داده‌های کاربر و تاریخچه چت را در مجموعه‌های Appwrite ذخیره می‌کند.
* 10 پیام آخر را برای زمینه (context) دریافت می‌کند.
* درخواست‌هایی به Open Router برای پاسخ‌های هوش مصنوعی ارسال می‌کند.
* پاسخ هوش مصنوعی را از طریق تلگرام به کاربر برمی‌گرداند.

کد اصلی در فایل جداگانه برای شما گذاشته شده است

توضیحات کد

* شناسه پایگاه داده: 'databaseId' را با شناسه واقعی پایگاه داده Appwrite خود جایگزین کنید.
* شناسه مجموعه‌ها: اطمینان حاصل کنید که مجموعه‌های users و chats در Appwrite با ویژگی‌های ذکرشده ایجاد شده‌اند. در صورت متفاوت بودن، 'users' و 'chats' را با شناسه‌های واقعی جایگزین کنید.
* ID.unique(): برای ایجاد شناسه‌های یکتا برای اسناد در مجموعه chats و برای sessionId استفاده شده است.
* متغیرهای محیطی: کد از متغیرهای محیطی ذکرشده در بالا استفاده می‌کند.
* محدودیت استفاده: کد بررسی می‌کند که آیا usageCount به 5 رسیده است یا خیر و در صورت رسیدن، پیام "شما محدود شده‌اید" ارسال می‌کند.
* زمینه مکالمه: 10 پیام آخر از مجموعه chats به ترتیب نزولی $createdAt دریافت شده و در درخواست API به Open Router گنجانده می‌شود.

دستور npm برای نصب پکیج‌هابرای نصب پکیج‌های مورد نیاز (node-appwrite و node-fetch)، از دستور زیر استفاده کنید:

npm install node-appwrite node-fetch

مراحل تنظیم

1. ایجاد مجموعه‌ها در Appwrite:
   * مجموعه users را با ویژگی‌های: userId (رشته)، username (رشته)، usageCount (عدد صحیح) ایجاد کنید.
   * مجموعه chats را با ویژگی‌های: userId (رشته)، message (رشته)، response (رشته)، sessionId (رشته) ایجاد کنید.
   * اطمینان حاصل کنید که $createdAt و $updatedAt فعال هستند (به صورت خودکار توسط Appwrite ایجاد می‌شوند).
2. تنظیم متغیرهای محیطی:
   * متغیرهای محیطی را از قطعه کد بالا کپی کرده و در بخش متغیرهای محیطی تابع Appwrite جای‌گذاری کنید، با جایگزینی مقادیر واقعی.
3. استقرار تابع:
   * کد index.js را به عنوان یک تابع بدون سرور در Appwrite مستقر کنید.
   * مطمئن شوید که تابع به پروژه Appwrite شما متصل است و مجوزهای لازم برای دسترسی به پایگاه داده را دارد.
4. تنظیم Webhook:
   * از دستور URL ارائه‌شده در بالا برای تنظیم webhook تلگرام استفاده کنید، با جایگزینی [TELEGRAM\_BOT\_TOKEN] و [WEBHOOK\_URL] با مقادیر واقعی.
5. آزمایش ربات:
   * پیامی به ربات تلگرام خود ارسال کنید و بررسی کنید که پاسخ تولیدشده توسط هوش مصنوعی دریافت می‌شود.
   * پایگاه داده Appwrite را بررسی کنید تا مطمئن شوید داده‌های کاربر و چت به درستی ذخیره می‌شوند.
   * پس از 5 پیام، ربات باید پیام "شما محدود شده‌اید" را ارسال کند.

نکات اضافی

* مدیریت خطاها: کد شامل مدیریت خطای پایه است و در صورت بروز خطا، پیامی به کاربر ارسال می‌کند.
* مدیریت جلسات: sessionId در مجموعه chats می‌تواند برای گروه‌بندی پیام‌ها در صورت نیاز به منطق مبتنی بر جلسه استفاده شود.
* مدل Open Router: کد از مدل moonshotai/kimi-k2:free استفاده می‌کند. اطمینان حاصل کنید که کلید API Open Router شما به این مدل دسترسی دارد.
* امنیت: مطمئن شوید که کلید API Appwrite شما دارای scopeهای لازم برای خواندن/نوشتن در مجموعه‌های users و chats است.