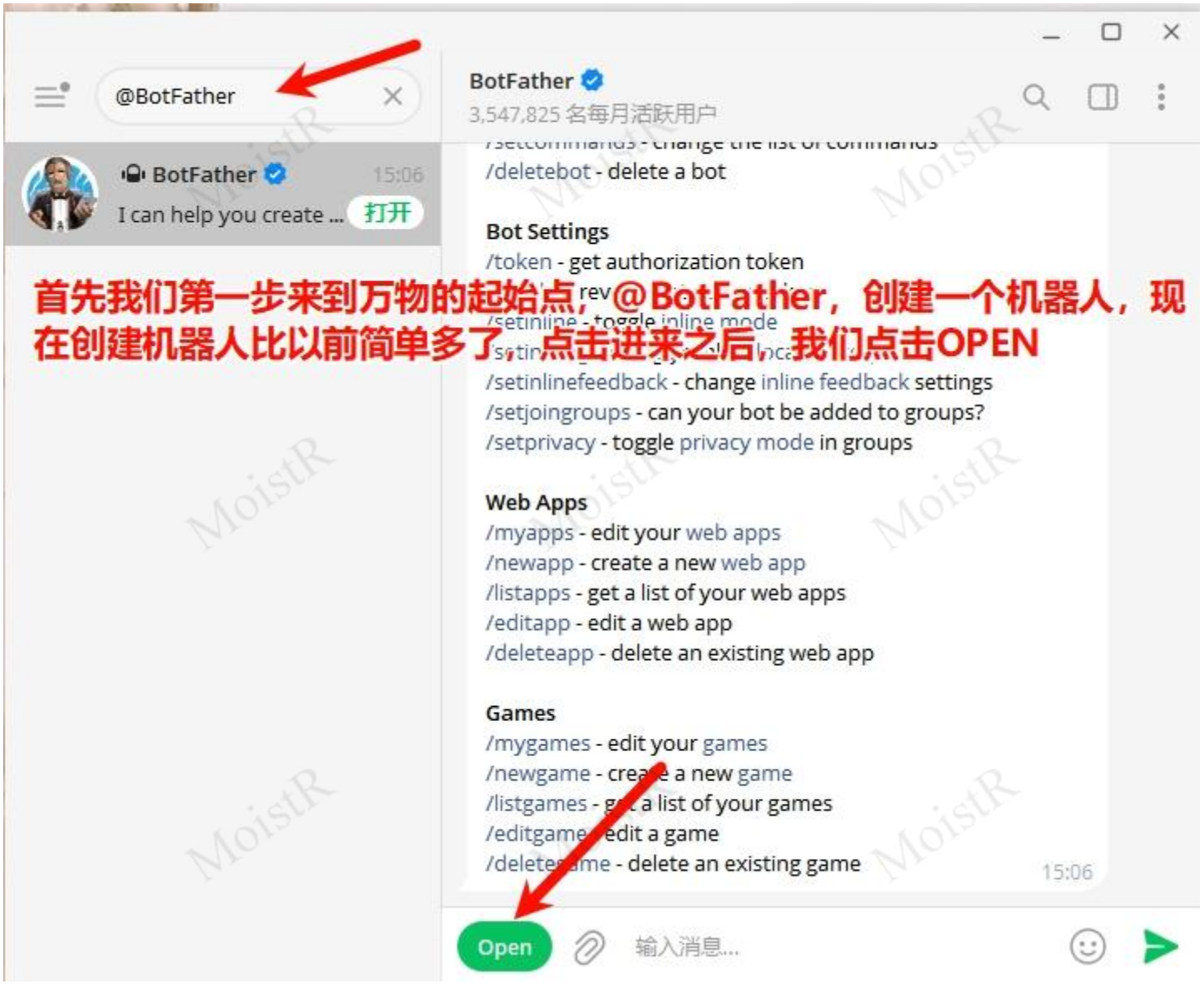
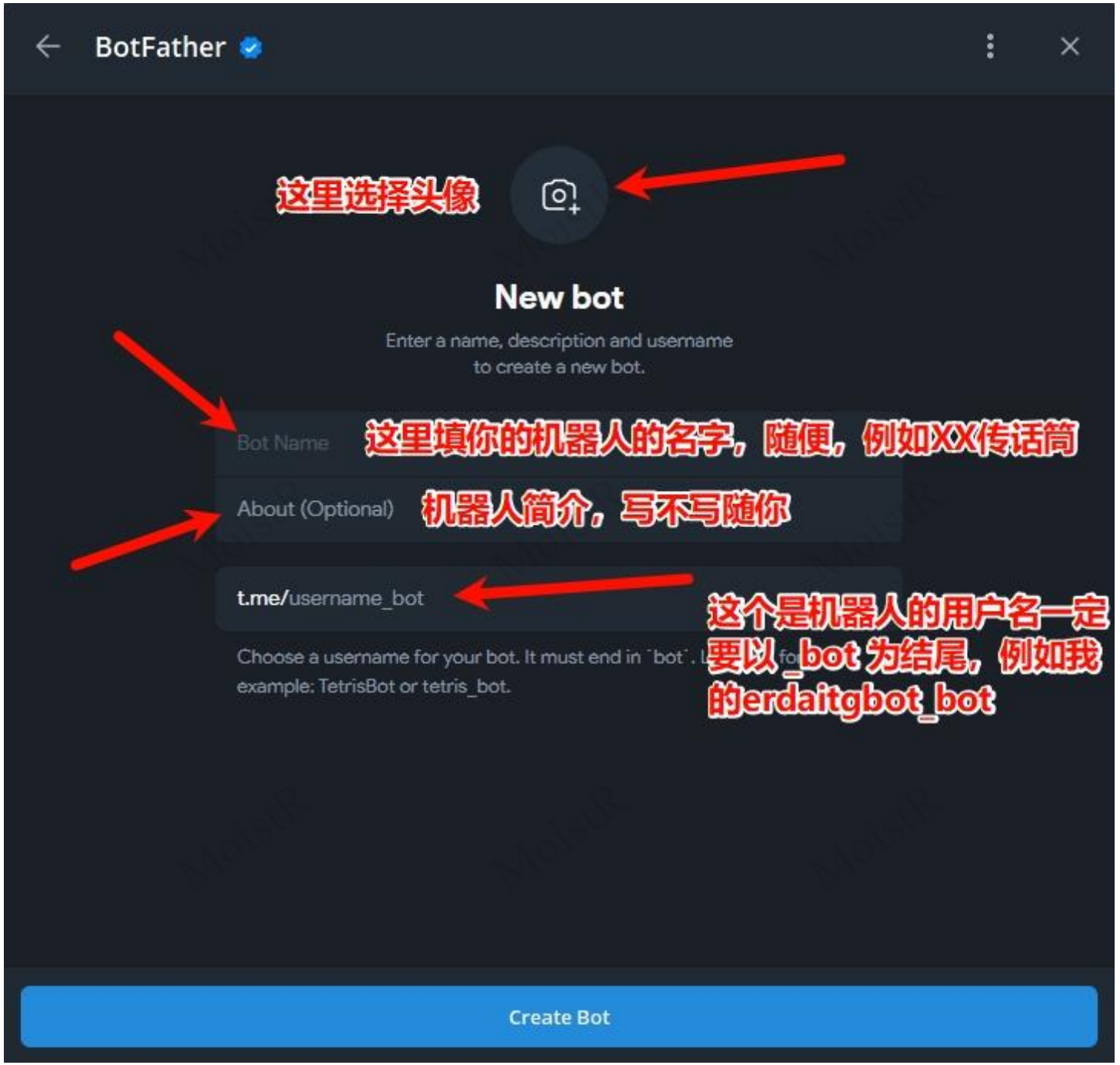
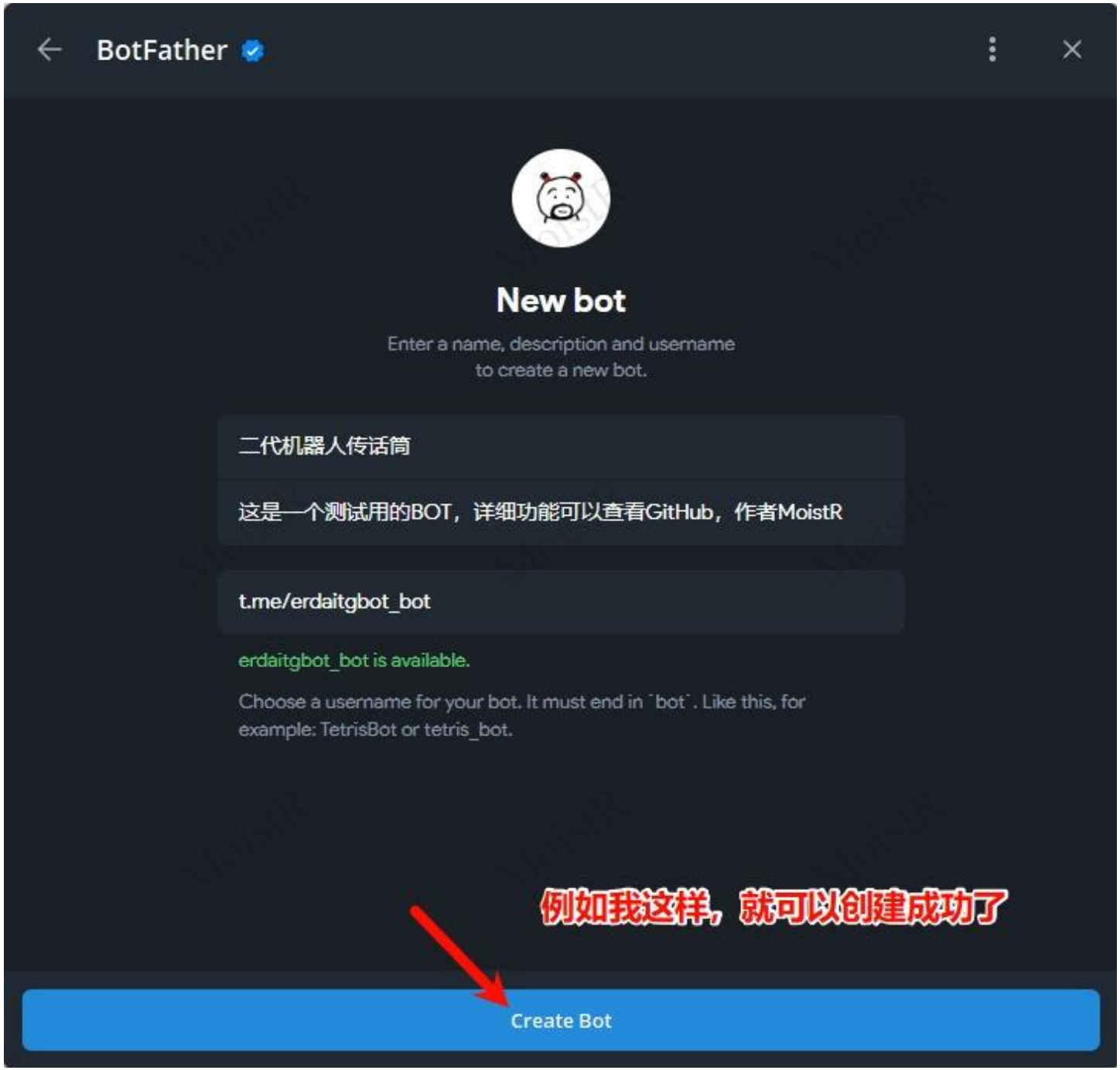
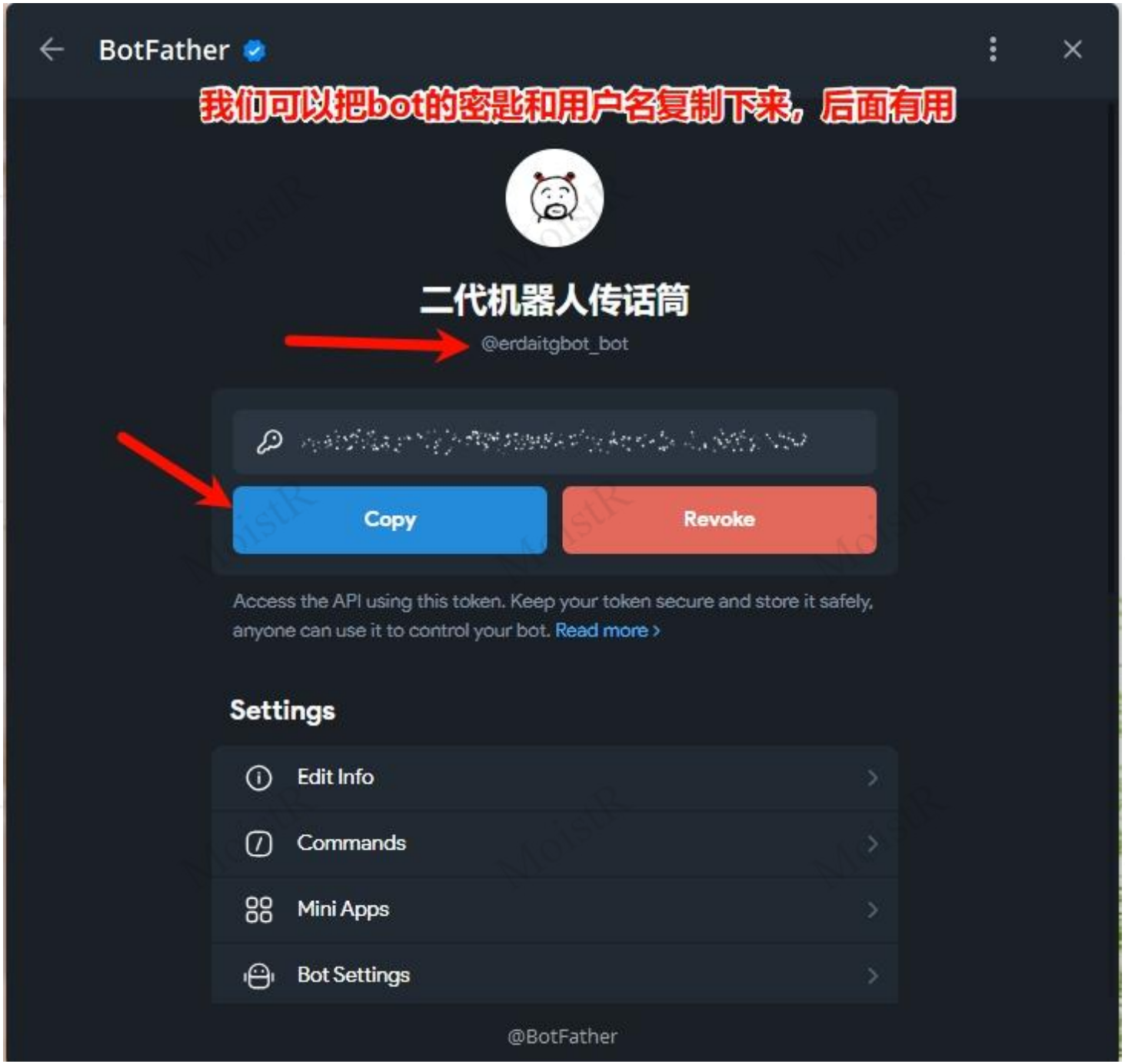
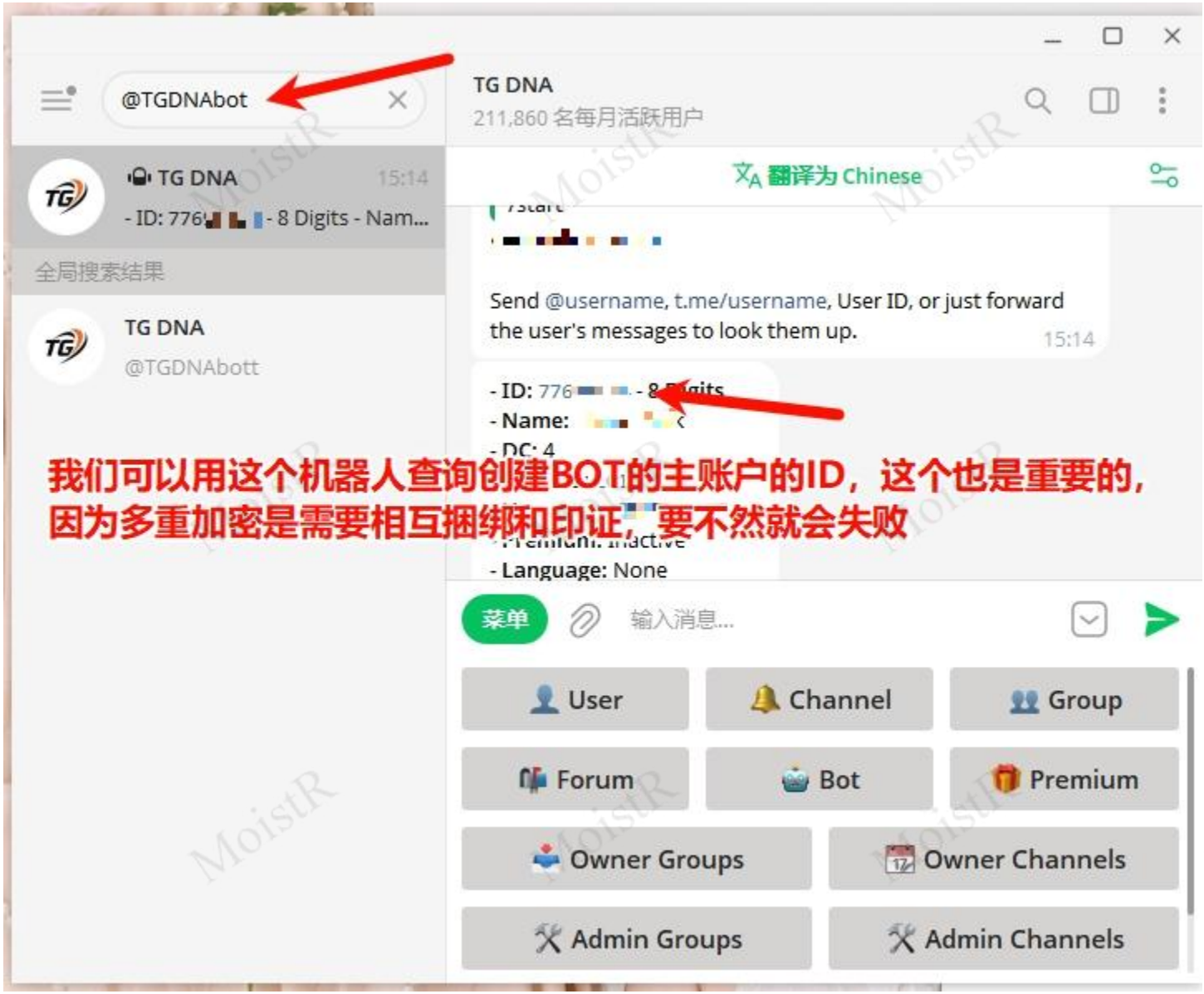

@BotFather
BotFather
3,547,825 名每月活跃用户
BotFather 15:06
I can help you create ... 打开
首先我们第一步来到万物的起始点，@BotFather，创建一个机器人，现在创建机器人比以前简单多了，点击进来之后，我们点击OPEN
/deletebot - delete a bot
Bot Settings
/token - get authorization token
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups
Web Apps
/myapps - edit your web apps
/newapp - create a new web app
/listapps - get a list of your web apps
/editapp - edit a web app
/deleteapp - delete an existing web app
Games
/mygames - edit your games
/listgames - get a list of your games
/editgame - edit a game
15:06
Open
输入消息...


BotFather
这里选择头像
New bot
Enter a name, description and username
to create a new bot.
Bot Name
这里填你的机器人的名字，随便，例如XX传话筒
About (Optional)
机器人简介，写不写随你
t.me/username_bot
这个是机器人的用户名一定要以_bot 为结尾，例如我的erdaitgbot_bot
Choose a username for your bot. It must end in `bot`.
example: TetrisBot or tetris_bot.
Create Bot

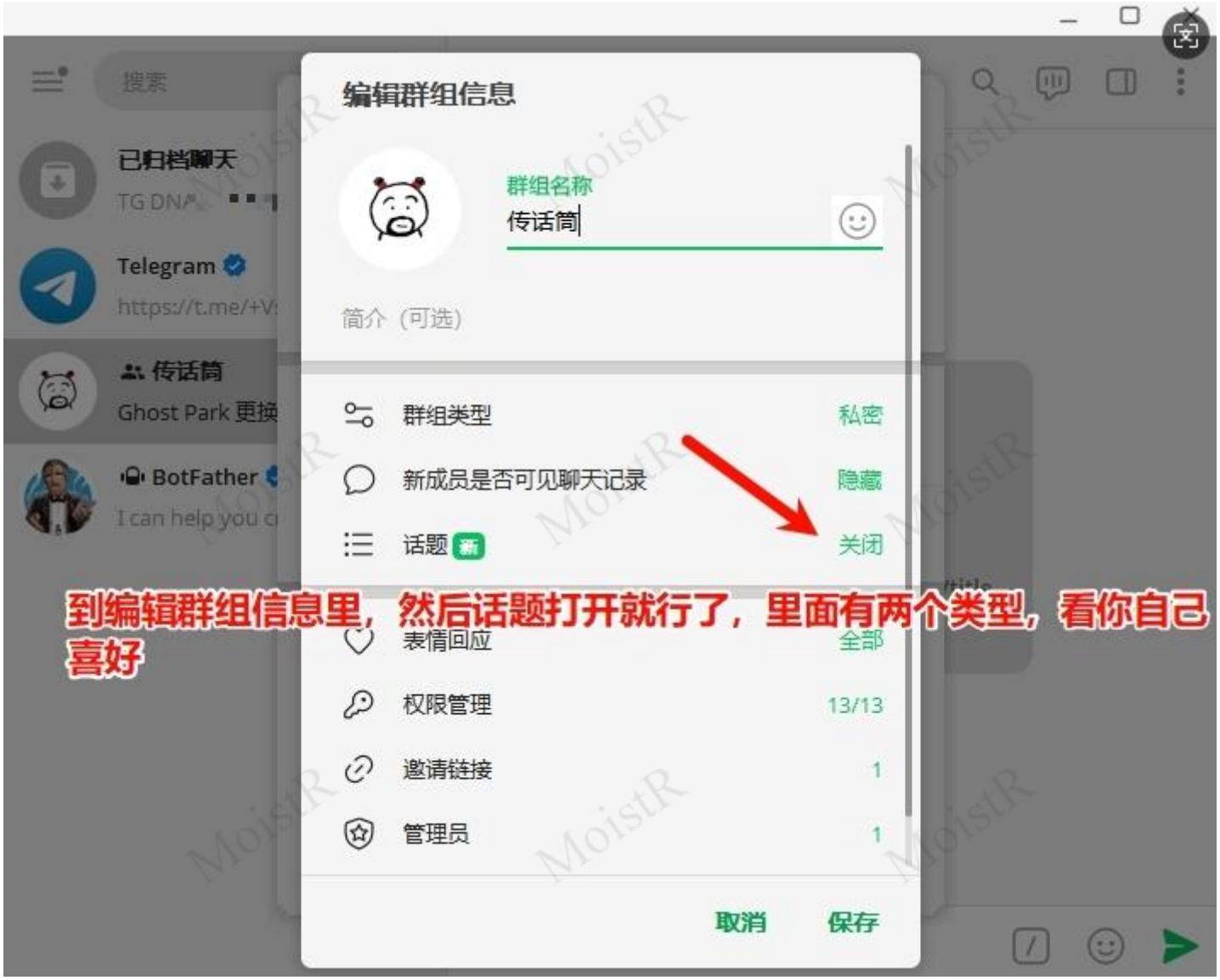

BotFather
New bot
Enter a name, description and username
to create a new bot.
二代机器人传话筒
这是一个测试用的BOT，详细功能可以查看GitHub，作者MoistR
t.me/erdaitgbot_bot
erdaitgbot_bot is available.
Choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
例如我这样，就可以创建成功了
Create Bot


BotFather
我们可以把bot的密匙和用户名复制下来，后面有用
二代机器人传话筒
@erdaitgbot_bot
Copy
Revoke
Access the API using this token. Keep your token secure and store it safely, anyone can use it to control your bot. Read more ›
Settings
Edit Info
Commands
Mini Apps
Bot Settings
@BotFather

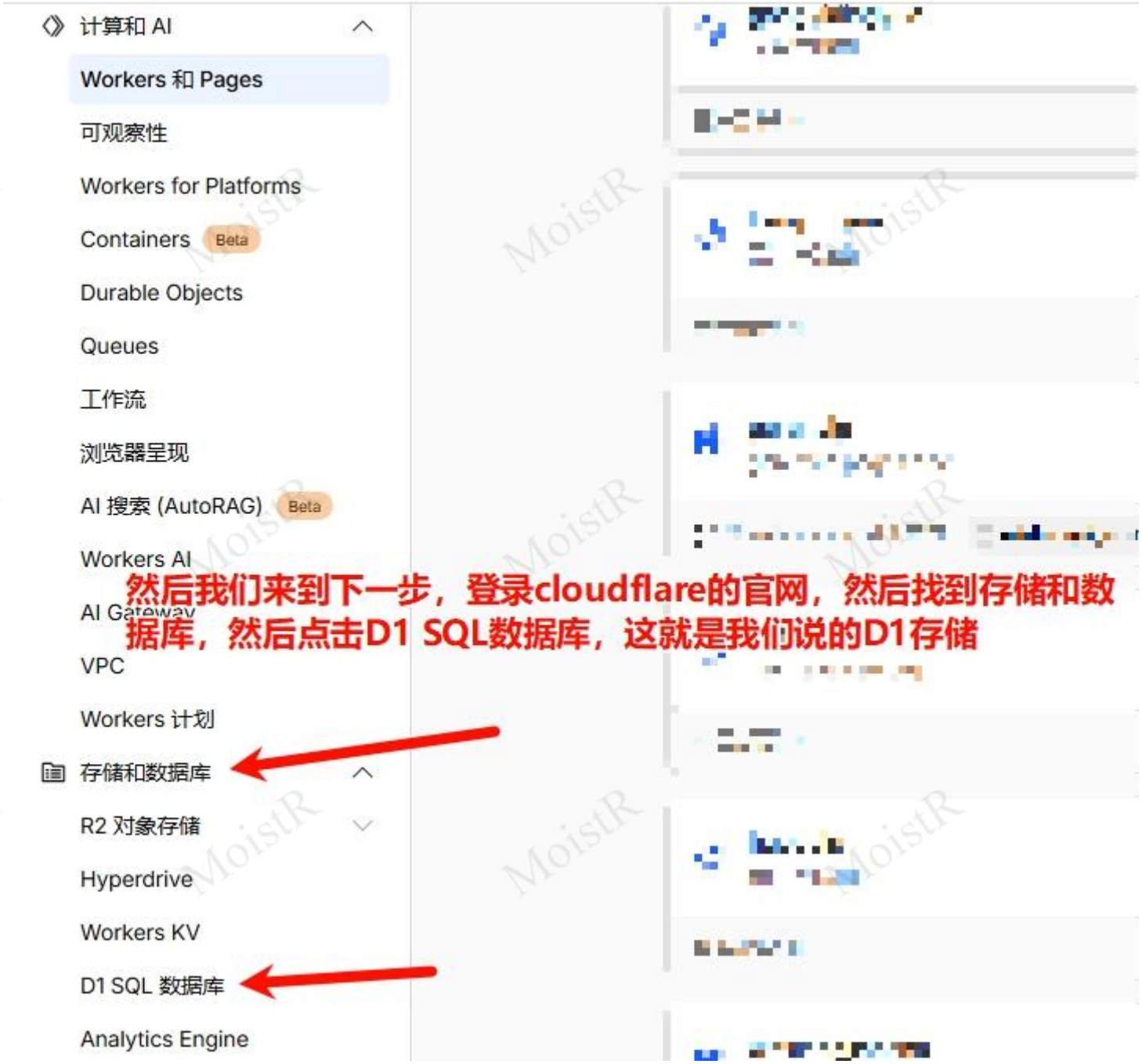

@TGDNAbot
TG DNA
211,860 名每月活跃用户
TG DNA
15:14
- ID: 776 - 8 Digits - Nam...
全局搜索结果
TG DNA
@TGDNAbott
翻译为 Chinese
Send @username, t.me/username, User ID, or just forward the user's messages to look them up.
15:14
- ID: 776
- Name:
- DC: 4
- Language: None
我们可以用这个机器人查询创建BOT的主账户的ID，这个也是重要的，
因为多重加密是需要相互捆绑和印证，要不然就会失败
菜单
输入消息...
User
Channel
Group
Forum
Bot
Premium
Owner Groups
Owner Channels
Admin Groups
Admin Channels


搜索
已归档聊天
TG DNA
Telegram
https://t.me/+Vsc
添加成员 1 / 200000
@erdaitgbot_bot
二代机器人传话筒
@erdaitgbot_bot
这里我们创建一个群组！群组！群组！，不是频道，刚刚我们复制下来的用户名就可以在这里直接邀请进来
groups?
15:06
取消
创建


传话筒
搜索
群组信息
第三步就是点击这里三个点，我们要开启话题模式
传话筒
2 位成员
-4991176645
DC4, Amsterdam, NL
通知
2 位成员
Telegram
传话筒
Ghost Park 更换
BotFather
第一步我们点击群组信息，来到这个界面，然后对着BOT右键进行提权，注意一定是这里，不能是管理员，否则会提权失败
二代机器人传话筒
无权读取消息
注：这里说的管理员是管理员列表设置管理，不是不给BOT管理员


搜索
已归档聊天
TG DNA
Telegram
https://t.me/+V
传话筒
Ghost Park 更换
BotFather
I can help you c
编辑群组信息
群组名称
传话筒
简介（可选）
群组类型
私密
新成员是否可见聊天记录
隐藏
话题 新
关闭
表情回应
全部
权限管理
13/13
邀请链接
1
管理员
1
取消
保存
到编辑群组信息里，然后话题打开就行了，里面有两个类型，看你自己喜好


传话筒
2 位成员
群组信息
如果有细心的朋友一定会发现这个群组的ID变成-100开头的，这就是话题模式的群组ID，记住记住记住，一定是要创建话题模式之后，再保存这个ID代码，如果找不到，那就拉一个用第三方TG的朋友 让他帮你看一下，我们要把这个代码复制下来
-1003244254636
DC4, Amsterdam, NL
通知
2 位成员
所有者
二代机器人传话筒
可读取消息
管理员
我们前一步把机器人从这里进行提权了，可以看到不可读取变成可读取了


计算和 AI
Workers 和 Pages
可观察性
Workers for Platforms
Containers Beta
Durable Objects
Queues
工作流
浏览器呈现
AI 搜索 (AutoRAG) Beta
Workers AI
AI Gateway
VPC
Workers 计划
存储和数据库
R2 对象存储
Hyperdrive
Workers KV
D1 SQL 数据库
Analytics Engine
然后我们来到下一步，登录cloudflare的官网，然后找到存储和数据库，然后点击D1 SQL数据库，这就是我们说的D1存储

README

部署方式（Cloudflare Dashboard 无指令）

本部署指南适用于通过 Cloudflare 网页界面操作，无需使用 Wrangler CLI 或本地开发环境。

步骤一：创建 D1 数据库

然后我们返回到GitHub的使用说明书里面，跟着步骤，创建D1数据库，然后执行下面三个代码

1. 登录 Cloudflare Dashboard。
2. 导航到 **Workers 和 Pages** -> **D1**。
3. 点击 **创建数据库**，输入数据库名称（例如：`tg-bot-db`）。
4. 进入您创建的 D1 数据库，点击 **浏览数据**。
5. 点击 **D1的控制台界面，有一个执行步骤，将下面三段执行语句复制到执行窗口，点击执行即可，会弹出响应时间即为部署成功**：

表名	字段 (Schema)
`users`	`user_id` (TEXT, PRIMARY KEY), `topic_id` (TEXT), `user_state` (TEXT), `is_blocked` (INTEGER), `block_count` (INTEGER), `user_info_json` (TEXT)
`config`	`key` (TEXT, PRIMARY KEY), `value` (TEXT)
`messages`	`user_id` (TEXT), `message_id` (TEXT), `text` (TEXT), `date` (INTEGER), PRIMARY KEY (`user_id` , `message_id`)

-- ① users 表

CREATE TABLE IF NOT EXISTS users (user_id TEXT PRIMARY KEY, topic_id TEXT, user_state TEXT, is_blocked INTEGER, block_count INTEGER, user_info_json TEXT);

-- ② config 表

CREATE TABLE IF NOT EXISTS config (key TEXT PRIMARY KEY, value TEXT);

-- ③ messages 表

CREATE TABLE IF NOT EXISTS messages (user_id TEXT, message_id TEXT, text TEXT, date INTEGER, PRIMARY KEY (user_id, message_id));

D1 数据库

创建用于查询关系数据的无服务器 SQLite 数据库。

文档 + 创建数据库

先点这个

您没有 D1 数据库。

创建 D1 数据库并立即开始将数据存储到边缘服务器上。您可以在仪表板中或通过 CLI 创建它们。

```
$ wrangler d1 create my-db-name
```

创建 D1 数据库

为您的 D1 数据库选择一个名称。创建后，您可以从仪表板或使用 Wrangler CLI 添加表和数据。

名称

tg-bot-db

Data location

位置:
D1 会根据您的位置自动将您的数据库放在最近的可用区域。

> 提供位置提示（可选）

Specify jurisdiction:
The location to restrict the D1 database to run and store data within to comply with local regulations. Note that if jurisdictions are set, the location hint is ignored.

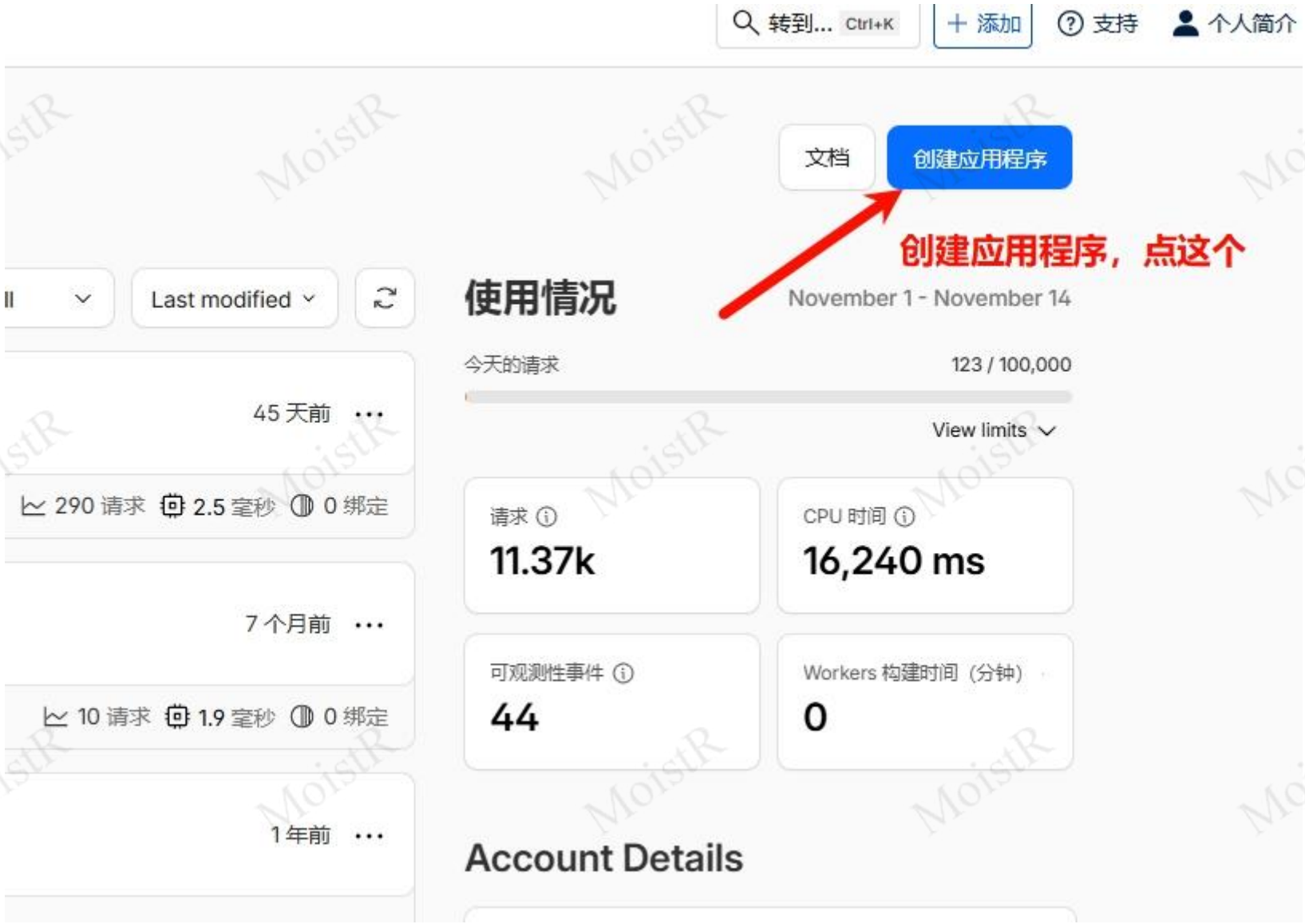
输入数据库名称，然后点击创建

取消 创建


D1 数据库 / tg-bot-db
概述
控制台
按时间顺序查看
设置
1
我们点击控制台，然后到GitHub复制那三个代码在2号位置，然后点3执行，三段代码复制三次，代表创建三个数据库
查询
您可以在此处输入 SQL 查询来查询数据库。
快捷键
up or down: 浏览之前的查询。
斜杠命令
/clear: 清除控制台屏幕。
/help or /?: 再次显示这些提示。
/tables: 显示此数据库中表的列表。
/bookmark: 找回数据库当前状态的书签，以在以后返回。
/bookmark yyyy-mm-ddThh:mm:ss.mmmZ: 获取距所提供的 RFC3339 时间戳最近的书签。
/restore bookmark-id: 将数据库还原到特定时间点。
2
3
执行

看到这里，我只是复制了代码部分，千万别把汉字的那一部分也复制进去了，那报错没得救，看到执行之后，会有一个响应时间，这就说明创建成功了

数据库创建好之后，我们来到workers,我们开始部署代码


转到... Ctrl+K
+ 添加
支持
个人简介
文档
创建应用程序
创建应用程序，点这个
Last modified
使用情况
November 1 - November 14
今天的请求
123 / 100,000
View limits
45 天前
290 请求 2.5 毫秒 0 绑定
7 个月前
10 请求 1.9 毫秒 0 绑定
1 年前
请求
11.37k
CPU 时间
16,240 ms
可观测性事件
44
Workers 构建时间（分钟）
0
Account Details

开始使用

开始使用 Workers。您想要如何开始?

Workers Pages

选择模板
从选择 Cloudflare 的预构建解决方案开始。 开始使用

导入存储库
从导入现有 Git 存储库开始。 开始使用

拖放文件
直接从计算机上传站点的资产，包括 HTML、CSS 和 JS 文件。 开始使用

从 Hello World! 开始
跳过配置，单击两次启动 hello world 应用。 开始使用

选择第四个从hello world开始，空白模板

"Hello World" 脚本

Worker 名称

snowy-meadow-6a24

随便改个名字，不改也行，不重要

您的 Worker 将被部署到：https://**snowy-meadow-6a24**.sankuytju.workers.dev

worker.js

```
/**
 * Welcome to Cloudflare Workers! This is your first worker.
 *
 * - Run "npm run dev" in your terminal to start a development server
 * - Open a browser tab at http://localhost:8787/ to see your worker in action
 * - Run "npm run deploy" to publish your worker
 *
 * Learn more at https://developers.cloudflare.com/workers/
 */

export default {
  async fetch(request, env, ctx) {
    // You can view your logs in the Observability dashboard
    console.info({ message: 'Hello World Worker received a request!' });
    return new Response('Hello World!');
  }
};
```

然后点击部署

部署

TGbot-D1 Public

Pin | Watch 0 | Fork 22 | Star 39

来都来了，点个小星星再走吧

main | 1 Branch | 0 Tags | Go to file t | Add file | Code

moistrr Update README.md 08d68be · 34 minutes ago 13 Commits

README.md	Update README.md	34 minutes ago
worker-D1版本.js	Update worker-D1版本.js	4 hours ago

然后我们点击这个js文件，把内容全选复制

README

🚀 Telegram 双向机器人 Cloudflare Worker

功能简介

这是一个基于 Cloudflare Worker 和 D1 数据库的 Telegram 双向机器人代码。它将用户私聊消息转发到管理员群组的话题（Topic）中，并将管理员在话题中的回复中继回用户私聊。

核心特性与最新增强：

1. **双向中继与话题模式：**
   - 将每个用户私聊会话转发到一个管理员群组的**独立话题**中。
   - 话题名称动态显示用户昵称和 ID，方便管理员区分。

About

实现了：人机验证、私聊到话题模式的转发、管理员回复中继、话题名动态更新、已编辑消息处理、用户屏蔽功能、关键词自动回复,存储已从 Cloudflare KV 切换到 D1 (SQLite) 以获取更高的写入容量。

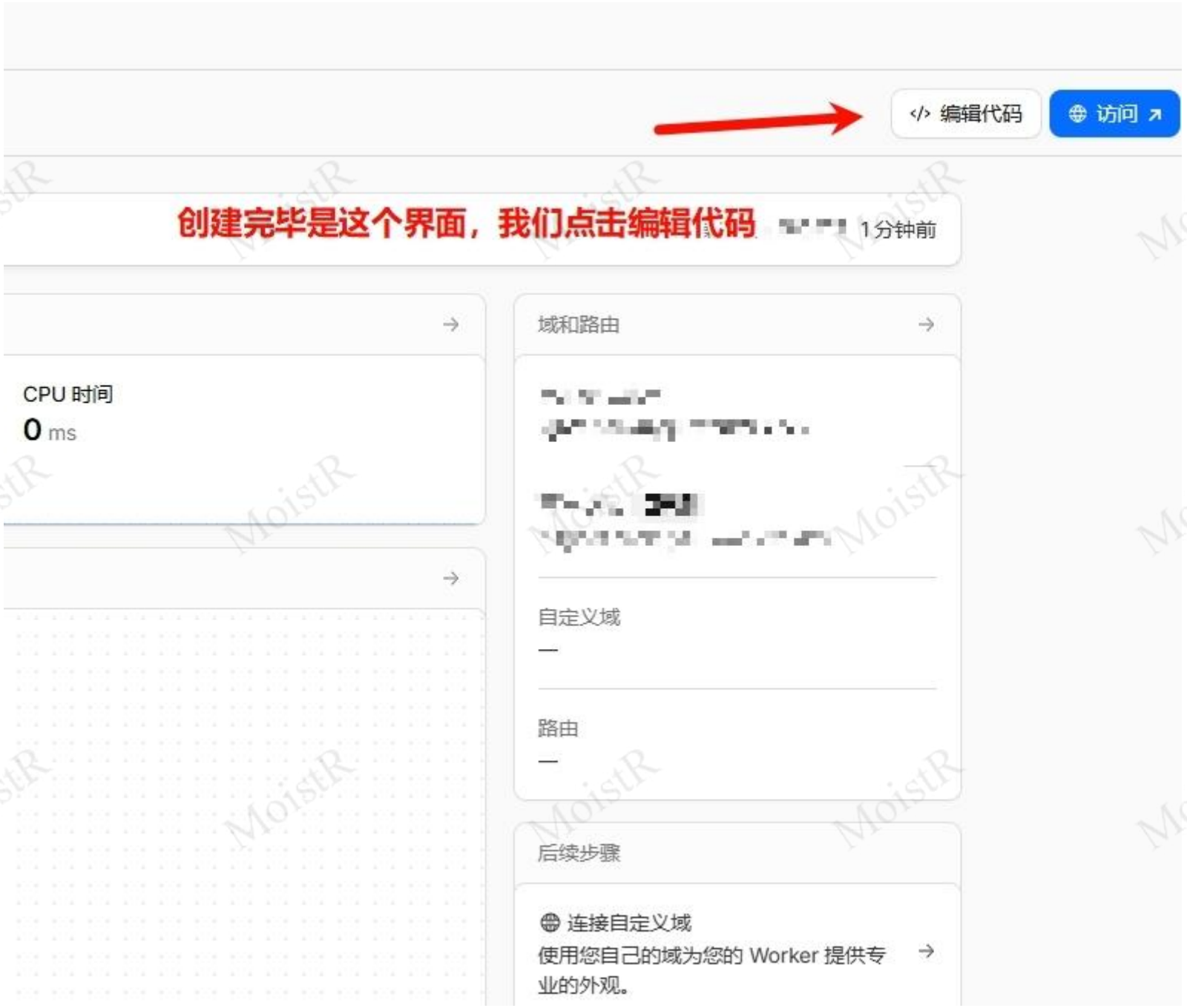
Readme
Activity
39 stars
0 watching
22 forks

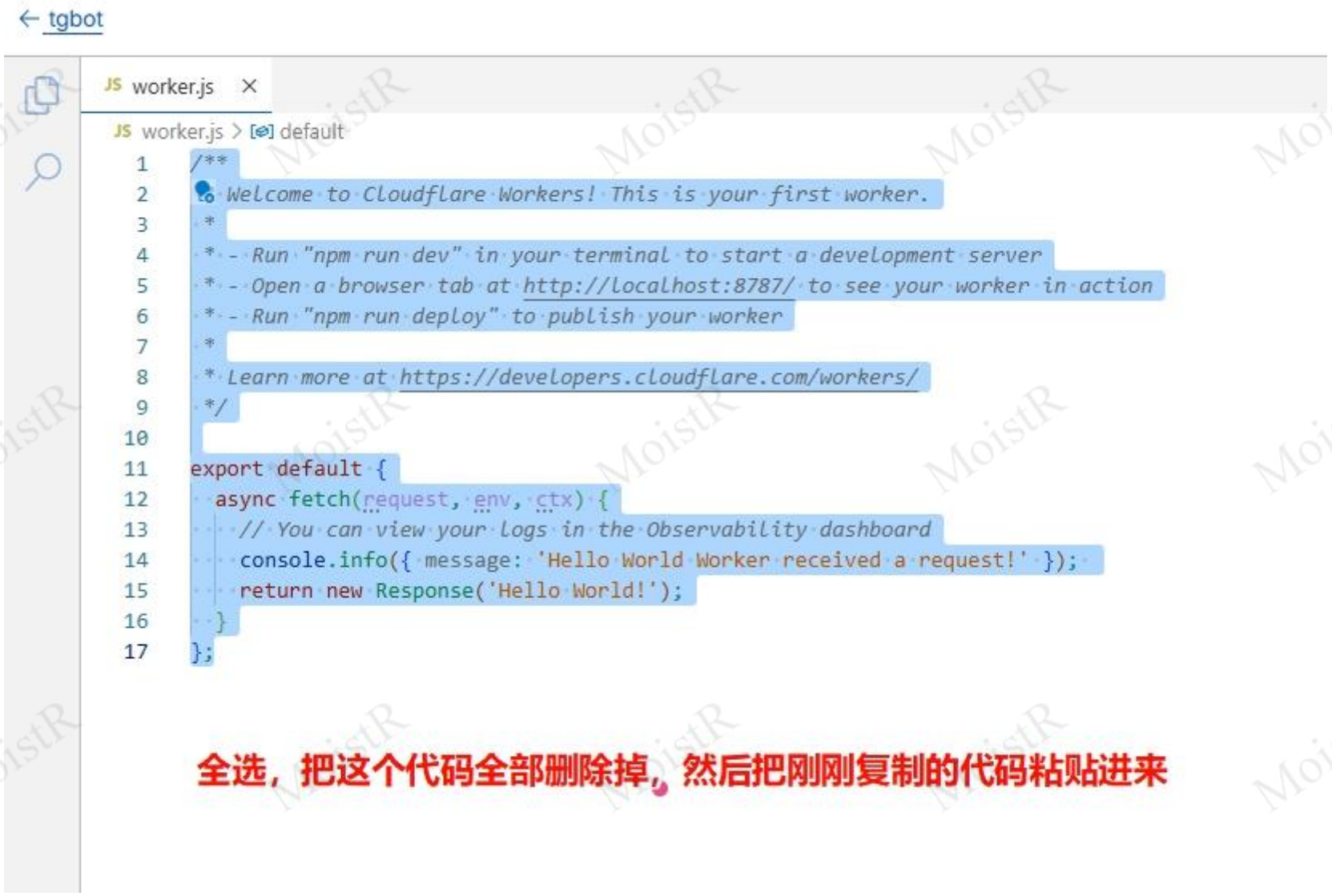
Releases

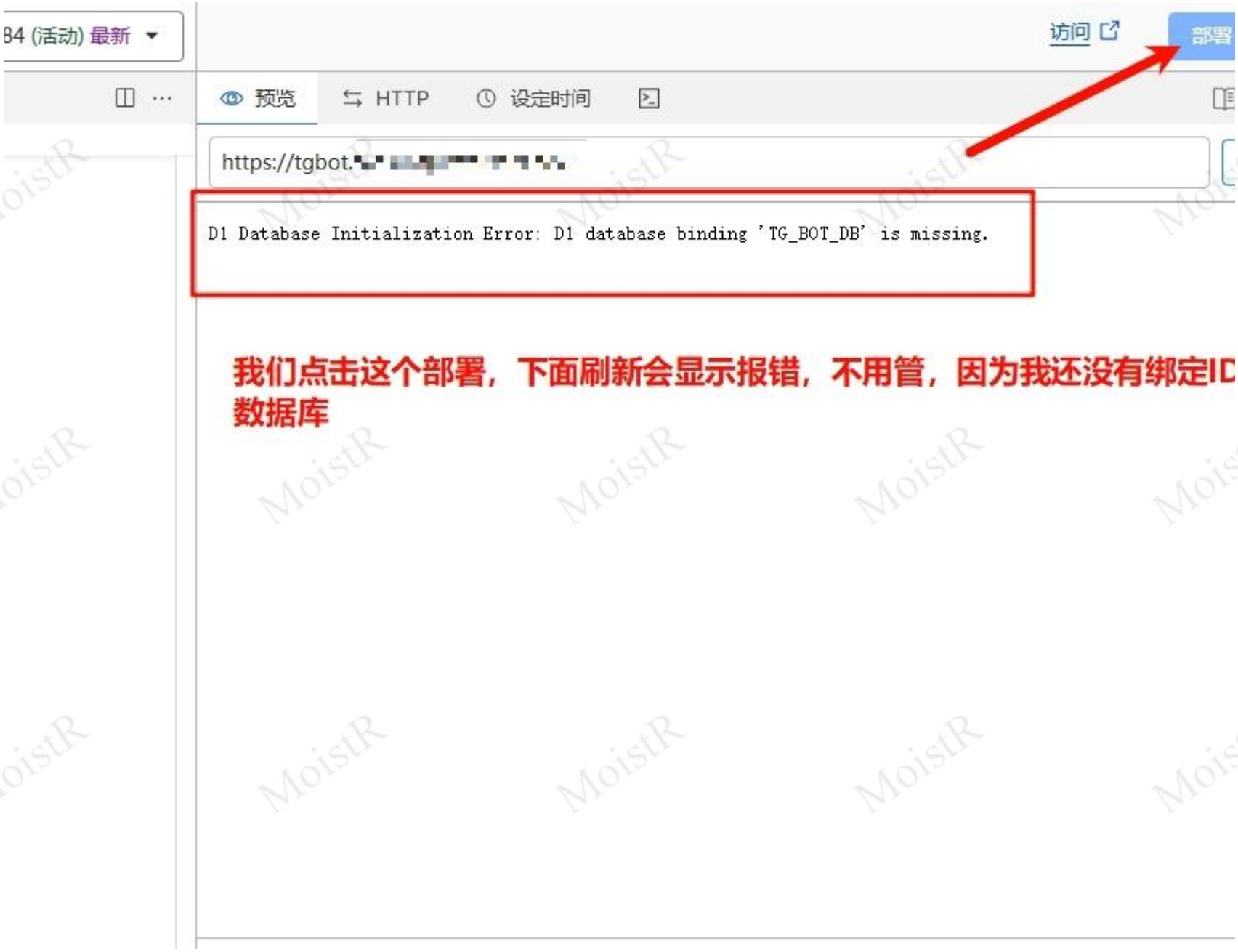
No releases published
Create a new release

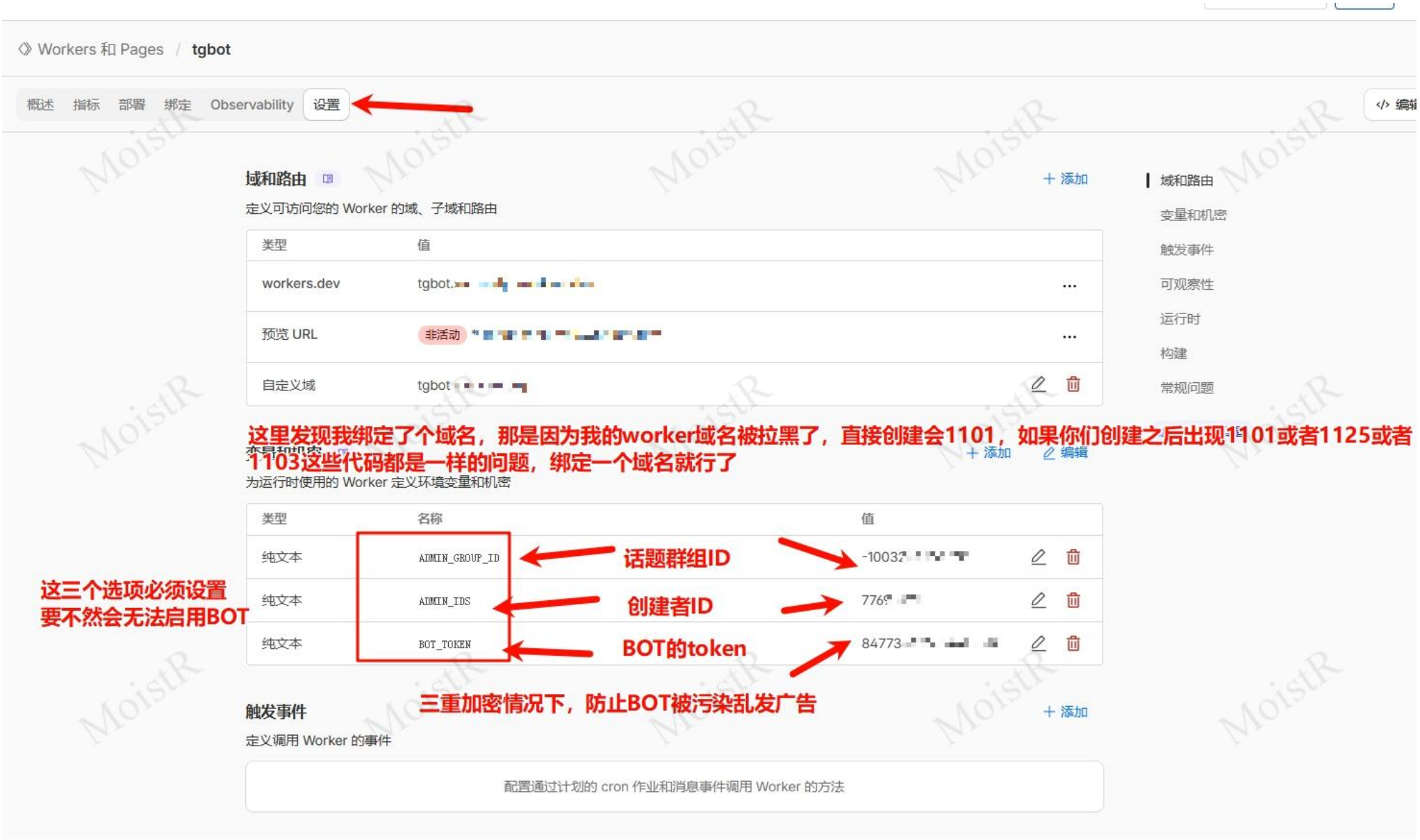
Packages

No packages published
Publish your first package


编辑代码
访问
创建完毕是这个界面，我们点击编辑代码
1分钟前
CPU 时间
0 ms
域和路由
自定义域
—
路由
—
后续步骤
连接自定义域
使用您自己的域为您的 Worker 提供专业的外观。


← tgbot
worker.js
worker.js > default
/**
* Welcome to Cloudflare Workers! This is your first worker.
*
* - Run "npm run dev" in your terminal to start a development server
* - Open a browser tab at http://localhost:8787/ to see your worker in action
* - Run "npm run deploy" to publish your worker
*
* Learn more at https://developers.cloudflare.com/workers/
*/
export default {
async fetch(request, env, ctx) {
// You can view your logs in the Observability dashboard
console.info({ message: 'Hello World Worker received a request!' });
return new Response('Hello World!');
}
};
全选，把这个代码全部删除掉，然后把刚刚复制的代码粘贴进来


84 (活动) 最新
访问
部署
预览
HTTP
设定时间
https://tgbot.
D1 Database Initialization Error: D1 database binding 'TG_BOT_DB' is missing.
我们点击这个部署，下面刷新会显示报错，不用管，因为我还没有绑定ID
数据库


Workers 和 Pages / tgbot
概述 指标 部署 绑定 Observability 设置
域和路由
+ 添加
定义可访问您的 Worker 的域、子域和路由
类型 值
workers.dev tgbot.
预览 URL 非活动
自定义域 tgbot
域和路由
变量和机密
触发事件
可观察性
运行时
构建
常规问题
这里发现我绑定了个域名，那是因为我的worker域名被拉黑了，直接创建会1101，如果你们创建之后出现1101或者1125或者1103这些代码都是一样的问题，绑定一个域名就行了
+ 添加
编辑
为运行时使用的 Worker 定义环境变量和机密
类型 名称 值
纯文本 ADMIN_GROUP_ID -10032
纯文本 ADMIN_IDS 776
纯文本 BOT_TOKEN 84773
这三个选项必须设置
要不然会无法启用BOT
话题群组ID
创建者ID
BOT的token
三重加密情况下，防止BOT被污染乱发广告
触发事件
+ 添加
定义调用 Worker 的事件
配置通过计划的 cron 作业和消息事件调用 Worker 的方法


Workers 和 Pages / tgbot
概述 指标 部署 绑定 Observability 设置
绑定
添加外部资源并将其连接到 Worker，无需管理权限或 API 密钥。
查看文档
添加绑定 +
已连接绑定
这一步我们绑定D1存储，点击添加绑定
tgbot
+ 绑定
添加绑定


添加绑定
添加外部资源并将其连接到 Worker 应用程序。
Analytics Engine
浏览器呈现
D1 数据库
耐用对象
Hyperdrive
Images
KV 命名空间
mTLS 证书
速率限制器
Secrets Store
服务绑定
Vectorize 索引
版本元数据
Workers AI
工作流
发现更多
分派命名空间
Queue
R2 存储桶
D1 数据库
查看文档
使用无服务器 SQL 数据库存储关系数据
D1 是 Cloudflare 的本机无服务器数据库，让您能够构建无需额外成本即可处理大量用户的应用程序。
type OrderRow = {
Id: string;
CustomerName: string;
OrderDate: number;
};
export default {
async fetch(request, env) {
const result = await env.MY_DB.prepare(
"SELECT Id, CustomerName, OrderDate FROM [Order] ORDER BY S
).run<OrderRow>();
return new Response(JSON.stringify(result));
}
}
选择D1数据库，然后添加绑定
取消
添加绑定


添加 D1 数据库 绑定
变量名称必须是这个，不能乱设置
变量名称
TG_BOT_DB
用于引用此绑定的名称。
D1 数据库
tg-bot-db
此绑定连接的 D1 数据库。
export default {
async fetch(request, env) {
const result = await env.TG_BOT_DB.prepare(
"SELECT * FROM [Order] ORDER LIMIT 100",
).run();
return new Response(JSON.stringify(result));
}
}
然后点击这个下箭头，绑定这个数据库
返回
添加绑定


Workers 和 Pages / tgbot
概述
指标
部署
绑定
Observabl
点击这个概述，然后点击这个位置的域名，如果提示OK 那就说明部署成功了，如果是1101，那就去绑定一个域名，用绑定域名访问
已部署版本 4b0cccb4
指标
最后一个 24 小时
请求
0
错误
0
绑定

下一步，激活CF和BOT直接的通讯，这里我们打开浏览器的地址栏，按照下面的格式填到地址栏
https://api.telegram.org/bot<您的BOT_TOKEN>/setWebhook?url=<您的Worker服务URL>

回车之后如果提示{"ok":true,"result":true,"description":"Webhook was set"}，则表示部署成功

如果提示404或者error，返回之前的部署步骤，检查是否有遗漏或者错误


@erdaitgbot_bot
二代机器人传话筒
机器人
全局搜索结果
二代机器人传话筒
@erdaitgbot_bot
这时候我们就可以回到tg上，对BOT进行设置了，输入用户名，点击开始
开始


@erdaitgbot_bot
全局搜索结果
二代机器人传话筒
@erdaitgbot_bot
二代机器人传话筒
机器人
11月14日
/start 15:49
机器人主配置菜单
请选择要管理的配置类别： 15:49
基础配置(验证问答)
自动回复管理
关键词屏蔽管理
按类型过滤管理
协管员授权设置
备份群组设置
刷新主菜单
输入消息...
弹出设置列表就说明设置成功了，每一个功能都比较简单，点开每一项都有说明所以怎么设置就不交代了，看自己需求进行设置就好


传话筒
2 位成员
2 条消息
General
用其他的号或者朋友给你的机器人发一条消息，第一次验证会出现认证两次start，因为第一次是创建新的数据库，第二次会正常使用，后续没有再次验证了
二代机器人传话筒
管理员
用户资料卡
15:52
屏蔽此人 (Block)
置顶此消息 (Pin Card)
二代机器人传话筒 管理员
111
15:52
输入消息...