

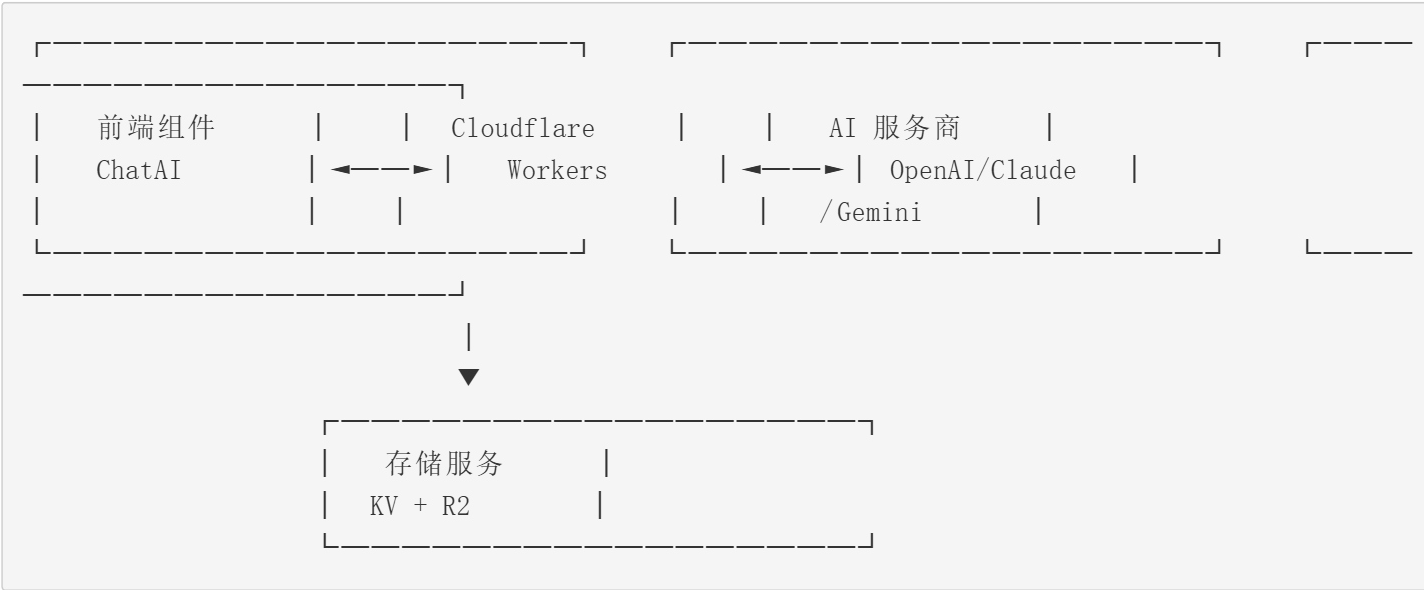
ChatAI Workers

基于 Cloudflare Workers 的智能聊天助手后端服务，为博客提供 AI 对话、文件上传、URL 解析等功能。

🌟 功能特性

- 🤖 **多 AI 提供商支持**: OpenAI、Anthropic Claude、Google Gemini
- 📁 **文件上传识别**: 支持文本、图片、代码等多种文件类型
- 🔗 **URL 内容解析**: 自动提取和分析网站链接内容
- 💬 **实时聊天**: WebSocket 支持实时消息推送
- 📄 **数据持久化**: KV 存储聊天记录，R2 存储文件
- 🔄 **自动清理**: 定时清理过期数据
- 🚀 **高性能**: 全球边缘计算，低延迟响应

🏗️ 架构设计



🚀 快速开始

1. 环境准备

```
# 安装 Node.js 和 npm
# 安装 Wrangler CLI
npm install -g wrangler

# 登录 Cloudflare
wrangler login
```

2. 克隆和配置

```
# 进入项目目录
cd chatai-workers

# 安装依赖
npm install

# 复制配置文件
cp wrangler.toml.example wrangler.toml
```

3. 配置 wrangler.toml

```
name = "chatai-workers"
main = "src/index.js"
compatibility_date = "2024-01-01"

# 更新你的 KV 和 R2 配置
[[kv_namespaces]]
binding = "CHAT_STORAGE"
id = "6d6592bc35fb4e2a985d29f7ef1d6e73" # 生产用 ID
preview_id = "6d6592bc35fb4e2a985d29f7ef1d6e73" # 预览用 ID（如果你没创建单独的预览命名空间，就用同一个）

[[r2_buckets]]
binding = "FILE_STORAGE"
bucket_name = "your-r2-bucket-name"
```

4. 一键部署

```
# 开发环境部署
chmod +x deploy.sh
./deploy.sh development

# 生产环境部署
./deploy.sh production
```

手动部署

创建存储资源

```
# 创建 KV 存储
wrangler kv:namespace create "CHAT_STORAGE"
wrangler kv:namespace create "CHAT_STORAGE" --preview
```

```
# 创建 R2 存储
wrangler r2 bucket create chatai-files
wrangler r2 bucket create chatai-files-preview
```

设置环境变量

支持多密钥轮换，当一个密钥配额用完时自动切换到下一个密钥：

```
# 国外AI平台（支持多密钥，用逗号分隔）
echo "key1,key2,key3" | wrangler secret put OPENAI_API_KEYS
echo "key1,key2,key3" | wrangler secret put ANTHROPIC_API_KEYS
echo "key1,key2,key3" | wrangler secret put GEMINI_API_KEYS

# 中国国内AI平台（支持多密钥）
echo "key1,key2,key3" | wrangler secret put QWEN_API_KEYS           # 阿里通义千问
echo "client_id1:secret1,client_id2:secret2" | wrangler secret put BAIDU_API_KEYS    # 百度文心一言
echo "key1,key2,key3" | wrangler secret put ZHIPU_API_KEYS         # 智谱AI
echo "key1,key2,key3" | wrangler secret put MOONSHOT_API_KEYS      # 月之暗面
echo "key1,key2,key3" | wrangler secret put DEEPSEEK_API_KEYS      # DeepSeek
echo "key1,key2,key3" | wrangler secret put MINIMAX_API_KEYS       # MiniMax

# 兼容单密钥格式（向后兼容）
echo "your-openai-key" | wrangler secret put OPENAI_API_KEY
echo "your-anthropic-key" | wrangler secret put ANTHROPIC_API_KEY
echo "your-gemini-key" | wrangler secret put GEMINI_API_KEY
```

密钥轮换机制

-  **自动轮换**：当密钥配额用完或出错时，自动切换到下一个可用密钥
-  **智能选择**：根据内容类型自动选择最适合的AI模型
- **CN 国内优先**：优先使用国内AI平台，速度更快，成本更低
-  **配额管理**：实时监控各平台配额使用情况
-  **容错机制**：单个密钥失败不影响整体服务

部署服务

```
# 开发环境
wrangler deploy --env development

# 生产环境
wrangler deploy --env production
```

聊天接口

```
# 发送消息
POST /api/chat/send
{
  "message": "Hello",
  "chatId": "optional-chat-id",
  "files": [],
  "urls": []
}

# 获取聊天历史
GET /api/chat/history/:chatId

# 创建新聊天
POST /api/chat/new
{
  "title": "New Chat"
}
```

文件接口

```
# 上传文件
POST /api/file/upload
Content-Type: multipart/form-data

# 获取文件
GET /api/file/:fileId
```

URL 解析

```
# 解析 URL
POST /api/url/parse
{
  "url": "https://example.com"
}
```

前端集成

1. 安装依赖

```
npm install framer-motion
```

2. 环境变量

```
# .env.local
NEXT_PUBLIC_CHATAI_WORKER_URL=https://chatai-workers.your-subdomain.workers.dev
```

3. 使用组件

```
import ChatAIWidget from '@components/ChatAI/ChatAIWidget'

function App() {
  return (
    <div>
      {/* 你的应用内容 */}

      <ChatAIWidget
        config={{
          name: 'ChatAI',
          avatar: '/images/avatar/chatai-avatar.png',
          welcomeMessage: '👋 Hello, what can I do for you?',
          position: 'bottom-right',
          workerUrl: process.env.NEXT_PUBLIC_CHATAI_WORKER_URL
        }}
      />
    </div>
  )
}
```

配置选项

主题配置 (themes/matery/config.js)

```
const CONFIG = {
  // ChatAI 配置
  CHATAI_ENABLE: true,
  CHATAI_NAME: 'ChatAI',
  CHATAI_AVATAR: '/images/avatar/chatai-avatar.png',
  CHATAI_WELCOME: '👋 Hello, what can I do for you?',
  CHATAI_POSITION: 'bottom-right', // 或 'bottom-left'
  CHATAI_WORKER_URL: 'https://chatai-workers.your-subdomain.workers.dev'
}
```

开发调试

```
# 本地开发
wrangler dev
```

```
# 查看日志
wrangler tail

# 测试 API
curl https://chatai-workers.your-subdomain.workers.dev/
```

监控和维护

查看使用情况

```
# Workers 使用统计
wrangler metrics

# KV 使用情况
wrangler kv:key list --binding CHAT_STORAGE

# R2 使用情况
wrangler r2 object list chatai-files
```

数据清理

```
# 手动触发清理
curl -X POST https://chatai-workers.your-subdomain.workers.dev/api/admin/cleanup
```

安全考虑

1. **API 密钥管理**: 使用 Wrangler secrets 管理敏感信息
2. **CORS 配置**: 限制允许的域名
3. **速率限制**: 防止 API 滥用
4. **数据加密**: 敏感数据加密存储

成本估算

- **Workers**: 免费额度 100,000 请求/天
- **KV**: 免费额度 100,000 读取/天, 1,000 写入/天
- **R2**: 免费额度 10GB 存储, 1,000,000 A 类操作/月

贡献指南

1. Fork 项目

2. 创建功能分支
3. 提交更改
4. 推送到分支
5. 创建 Pull Request

许可证

MIT License

故障排除

常见问题

1. **部署失败**: 检查 wrangler.toml 配置
2. **API 调用失败**: 验证 CORS 设置
3. **文件上传失败**: 检查 R2 权限配置
4. **聊天记录丢失**: 验证 KV 存储配置

获取帮助

- 查看 [Cloudflare Workers 文档](#)

 享受你的 ChatAI 聊天助手!