



ChatFlow 部署指南

本指南将帮助您从 GitHub 克隆项目并成功部署到您的环境中。



前置要求

- Node.js 18+
- npm 或 yarn
- Cloudflare 账户
- 至少一个 AI 提供商的 API 密钥



快速部署

1. 克隆项目

```
git clone https://github.com/your-username/ChatFlow.git
cd ChatFlow
```

2. 安装依赖

```
# 安装前端依赖
npm install

# 安装 Workers 依赖
cd chatai-workers
npm install
cd ..
```

3. 配置环境变量

```
# 复制环境变量模板
cp .env.example .env.local

# 编辑环境变量文件
# 至少需要设置 NEXT_PUBLIC_CHATAI_WORKER_URL
```

4. 配置 Cloudflare Workers

4.1 登录 Cloudflare

```
cd chatai-workers
```

```
npx wrangler login
```

4.2 创建 KV 命名空间

```
# 创建聊天存储
npx wrangler kv:namespace create "CHAT_STORAGE"
npx wrangler kv:namespace create "CHAT_STORAGE" --preview

# 记录返回的 ID，更新 wrangler.toml
```

4.3 创建 R2 存储桶

```
# 创建文件存储桶
npx wrangler r2 bucket create your-bucket-name

# 更新 wrangler.toml 中的 bucket_name
```

4.4 更新 wrangler.toml

编辑 `chatai-workers/wrangler.toml`，替换占位符：

```
# KV 存储配置
[[kv_namespaces]]
binding = "CHAT_STORAGE"
id = "your-actual-kv-id" # 替换为实际 ID
preview_id = "your-actual-preview-id" # 替换为实际 ID

# R2 存储配置
[[r2_buckets]]
binding = "FILE_STORAGE"
bucket_name = "your-actual-bucket-name" # 替换为实际名称
preview_bucket_name = "your-actual-bucket-name"
```

5. 设置 API 密钥

选择您要使用的 AI 提供商并设置相应的 API 密钥：

5.1 免费/低成本选项（推荐新手）

```
# Google Gemini（有免费额度）
echo "your-gemini-api-key" | npx wrangler secret put GEMINI_API_KEYS

# Groq（免费额度）
echo "your-groq-api-key" | npx wrangler secret put GROQ_API_KEYS
```

5.2 高质量选项

```
# OpenAI GPT
echo "your-openai-api-key" | npx wrangler secret put OPENAI_API_KEYS

# Anthropic Claude
echo "your-anthropic-api-key" | npx wrangler secret put ANTHROPIC_API_KEYS
```

5.3 国内选项

```
# 通义千问
echo "your-qwen-api-key" | npx wrangler secret put QWEN_API_KEYS

# 月之暗面 Kimi
echo "your-moonshot-api-key" | npx wrangler secret put MOONSHOT_API_KEYS

# DeepSeek
echo "your-deepseek-api-key" | npx wrangler secret put DEEPSEEK_API_KEYS
```

6. 部署 Workers

```
# 部署到开发环境
npm run deploy

# 或部署到生产环境
npm run deploy:production
```

7. 更新前端配置

部署成功后，更新 `.env.local` 中的 Workers URL：

```
NEXT_PUBLIC_CHATAI_WORKER_URL=https://your-worker-name.your-subdomain.workers.dev
```

8. 启动前端

```
# 返回项目根目录
cd ..

# 启动开发服务器
npm run dev

# 或构建生产版本
npm run build
npm start
```

多密钥配置

支持为每个提供商配置多个 API 密钥，系统会自动轮换：

```
# 配置多个密钥（用逗号分隔）
echo "key1,key2,key3" | npx wrangler secret put OPENAI_API_KEYS
```

自定义域名

如果您有自定义域名：

1. 在 Cloudflare 中添加域名
2. 更新 `wrangler.toml` 添加路由配置
3. 更新前端环境变量

环境分离

项目支持开发和生产环境分离：

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

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

测试部署

1. 测试 Workers API

```
# 测试基本连接
curl https://your-worker-url.workers.dev/

# 测试聊天功能
curl -X POST https://your-worker-url.workers.dev/api/chat \
  -H "Content-Type: application/json" \
  -H "X-Chat-Session: test-session" \
  -d '{"message": "Hello", "provider": "gemini"}'
```

2. 测试前端

访问 `http://localhost:3000`，确认：

- CloudAI 组件正常显示
- 可以发送消息并收到回复
- 文件上传功能正常
- URL 解析功能正常

故障排除

常见问题

Workers 部署失败

- 检查 `wrangler.toml` 配置是否正确
- 确认 KV 和 R2 资源已创建
- 验证 Cloudflare 账户权限

API 调用失败

- 检查 API 密钥是否正确设置
- 确认密钥格式符合要求
- 验证 API 配额是否充足

前端连接失败

- 确认 Workers URL 配置正确
- 检查 CORS 设置
- 验证网络连接

调试命令

```
# 查看 Workers 日志
npx wrangler tail

# 查看已设置的密钥
npx wrangler secret list

# 测试本地开发
npx wrangler dev
```

监控和维护

使用情况监控

```
# 查看 Workers 指标
```

```
npx wrangler metrics
```

查看 KV 使用情况

```
npx wrangler kv:key list --binding CHAT_STORAGE
```

查看 R2 使用情况

```
npx wrangler r2 object list your-bucket-name
```

定期维护

- 定期检查 API 配额使用情况
- 轮换 API 密钥
- 清理过期的聊天记录和文件
- 更新依赖包

生产环境建议

1. 安全性

- 使用强密码和 2FA
- 定期轮换 API 密钥
- 限制 CORS 域名

2. 性能

- 配置多个 API 密钥
- 使用 CDN 加速静态资源
- 监控响应时间

3. 可靠性

- 配置多个 AI 提供商作为备用
 - 设置适当的错误处理
 - 定期备份重要数据
-