**Windows Server 部署帮助说明 (方案A: 自托管)**

本指南将详细介绍如何在 Windows Server 环境下，部署“智能收藏夹”的后端服务器及自托管的 MongoDB 数据库。

**第 1 部分：前期准备 (Prerequisites)**

1. **一台 Windows 云服务器**:
   * **推荐系统**: Windows Server 2019 或更高版本。
   * **推荐配置**: 至少 1 核 CPU / 2 GB 内存 / 40 GB SSD 存储 (Windows Server 比 Linux 需要更多资源)。
   * **连接方式**: 您将通过 **远程桌面连接 (RDP)** 来管理服务器。
   * **防火墙**: 确保服务器的**入站规则**已开放 80 (HTTP) 和 443 (HTTPS) 端口。
2. **一个域名 (Domain Name)**:
   * 同样，您需要一个域名（如 api.yourdomain.com）并将其 DNS 解析指向您服务器的公网 IP。
3. **管理员权限**:
   * 确保您用于登录远程桌面的账户拥有管理员权限。

**第 2 部分：MongoDB 数据库部署 (Windows Server)**

1. **下载 MongoDB Server**:
   * 打开服务器上的浏览器，访问 [MongoDB Community Server 下载页面](https://www.mongodb.com/try/download/community)。https://www.mongodb.com/try/download/community
   * 选择最新版本的 MongoDB Community Server，平台选择 "Windows"，包格式选择 "msi"。
   * 下载 MSI 安装包。
2. **安装 MongoDB**:
   * 双击运行下载的 MSI 安装包。
   * 选择 **"Complete" (完整)** 安装。
   * 在 "Service Configuration" (服务配置) 界面，**务必勾选 "Install MongoD as a Service"** 并选择 **"Run service as Network Service user"**。
   * 服务名保持默认的 "MongoDB" 即可。
   * **取消勾选 "Install MongoDB Compass"** 以简化安装过程。
   * 完成安装。
3. **配置 Server 环境变量**:
   * 在服务器的搜索栏中搜索“编辑系统环境变量”并打开它。
   * 点击“环境变量...”。
   * 在“系统变量”下找到 Path 变量，点击“编辑”。
   * 点击“新建”，然后添加 MongoDB Server 的 bin 目录路径，通常是 C:\Program Files\MongoDB\Server\[版本号]\bin。
   * 一路点击“确定”保存。
4. **安全配置 (非常重要！)**:
   * 打开 **命令提示符 (CMD)** 或 **PowerShell** (以管理员身份运行)。
   * 此时，由于 mongosh 还未安装，您暂时无法进行此步骤。请先完成第 2.5 步。
5. **新增：安装并配置 MongoDB Shell (mongosh)**:
   * **下载**: 访问 [MongoDB Shell 下载页面](https://www.mongodb.com/try/download/shell)，https://www.mongodb.com/try/download/shell下载适用于 Windows 的 MSI 安装包。
   * **安装**: 运行下载的 mongodb-shell-\*.msi 文件，按提示完成安装。
   * **配置 Shell 环境变量**: 再次打开“编辑系统环境变量”，在 Path 变量中“新建”一条记录，指向 mongosh 的 bin 目录，通常是 C:\Program Files\MongoDB\mongosh\bin。
   * **验证**: **重新打开一个 CMD 或 PowerShell 窗口**，输入 mongosh --version，如果看到版本号，则表示安装成功。
6. **完成安全配置**:
   * 现在 mongosh 已可用，请**以管理员身份**打开一个新的 CMD 或 PowerShell 窗口。
   * 进入 MongoDB Shell:
   * mongosh
   * 在 Shell 中创建管理员用户:
   * use admin
   * db.createUser({
   * user: "myAdmin",
   * pwd: passwordPrompt(), // 会提示您安全地输入密码
   * roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
   * })
   * exit
   * 启用访问控制。找到 MongoDB 的配置文件 C:\Program Files\MongoDB\Server\[版本号]\bin\mongod.cfg。
   * 用记事本 (以管理员身份) 打开该文件，找到 security: 部分（如果不存在则手动添加），修改为：
   * security:
   * authorization: "enabled"
   * 保存文件后，重启 MongoDB 服务。在“服务”应用中找到 "MongoDB Server (MongoDB)"，右键点击“重新启动”。
7. **为您的应用创建专用用户**:
   * 以管理员身份重新登录 MongoDB Shell:
   * mongosh -u " AiwetalkAdmin" -p --authenticationDatabase "admin"

记录: 用户：AiwetalkAdmin

密码： kjm123#@!

* + 创建应用数据库和用户:
  + use smart\_bookmarker\_prod
  + db.createUser({
  + user: "bookmarkerAppUser",
  + pwd: passwordPrompt(),
  + roles: [ { role: "readWrite", db: "smart\_bookmarker\_prod" } ]
  + })
  + exit
  + **记下您的 MONGO\_URI**: mongodb://bookmarkerAppUser:kjmdb123$%^ @localhost:27017/smart\_bookmarker\_prod

数据库用户: bookmarkerAppUser

密码: kjmdb123$%^

**第 3 部分：服务器端应用程序部署**

**这部分将指导您如何在服务器上设置和运行您的 Node.js 后端应用程序。**

1. **安装必备软件：Node.js 和 Git:**
   * **Node.js: 在服务器的浏览器中，访问** [**Node.js 官网**](https://nodejs.org/)**。下载并安装最新的 LTS (长期支持) 版本。安装过程中请确保勾选 "Add to PATH" 选项。**
   * **Git: 访问** [**Git for Windows 官网**](https://git-scm.com/download/win)**，下载并安装 Git。这将为您提供 git 命令以及一个好用的命令行工具 Git Bash。**
2. **获取您的应用程序代码:**
   * **在服务器上创建一个用于存放应用的文件夹，例如 C:\apps。**
   * **打开 Git Bash (从开始菜单找到它)。**
   * **使用 cd 命令进入您创建的文件夹：**
   * **cd /c/apps**
   * **从您的代码仓库（如 GitHub, Gitee）克隆服务器端程序的代码：**
   * **git clone <your\_repository\_url>**
   * **进入项目目录：**
   * **cd smart-bookmarker-server-prod # 替换为您的项目文件夹名**
3. **安装项目依赖:**
   * **您的项目代码中包含一个 package.json 文件，它定义了所有必需的 Node.js 模块。**
   * **在项目目录下，打开 PowerShell 或 CMD，运行以下命令来安装这些依赖：**
   * **npm install --production**
   * **--production 标志会告诉 npm 跳过 devDependencies（开发依赖），只安装生产环境运行所必需的模块，可以减小安装体积。**
4. **配置环境变量:**
   * **环境变量是存储敏感信息（如数据库密码、API 密钥）的最佳方式，避免将其硬编码在代码中。**
   * **在项目根目录下，创建一个名为 .env 的文本文件。您可以在文件浏览器中右键 -> 新建 -> 文本文档，然后将其重命名为 .env (确保没有 .txt 后缀)。**
   * **用记事本或 VS Code 打开这个 .env 文件，并填入您在前面步骤中准备好的所有配置信息，格式如下：**
   * **# MongoDB 连接字符串 (从第 2 部分获取)**
   * **MONGO\_URI=mongodb://bookmarkerAppUser:YourAppPassword@localhost:27017/smart\_bookmarker\_prod**
   * **# JWT 签名密钥 (请使用一个长且随机的字符串)**
   * **JWT\_SECRET=a\_very\_long\_and\_random\_secret\_string\_for\_production**
   * **# 应用程序运行端口**
   * **PORT=5000**
   * **# 第三方 OAuth 凭证**
   * **GOOGLE\_CLIENT\_ID=YOUR\_GOOGLE\_CLIENT\_ID**
   * **GOOGLE\_CLIENT\_SECRET=YOUR\_GOOGLE\_CLIENT\_SECRET**
   * **GITHUB\_CLIENT\_ID=YOUR\_GITHUB\_CLIENT\_ID**
   * **GITHUB\_CLIENT\_SECRET=YOUR\_GITHUB\_CLIENT\_SECRET**
   * **保存文件。您的 Node.js 程序在启动时会自动加载这些配置。**

**第 4 部分：生产环境运行与反向代理 (IIS)**

**我们将使用 IIS (Internet Information Services) 作为反向代理，并使用 pm2 来管理 Node.js 进程。**

1. **安装 IIS 和所需模块:**
   * **打开“服务器管理器”。**
   * **点击“添加角色和功能”。**
   * **在“服务器角色”中，勾选 "Web 服务器 (IIS)"。**
   * **在弹出的窗口中，点击“添加功能”。**
   * **一直点击“下一步”直到安装完成。**
   * **安装完成后，下载并安装以下两个 IIS 扩展模块：**
     + [**URL Rewrite**](https://www.iis.net/downloads/microsoft/url-rewrite)
     + [**Application Request Routing (ARR)**](https://www.iis.net/downloads/microsoft/application-request-routing)
2. **安装并配置 PM2 (进程管理器):**
   * **2.1 安装 PM2 主程序:**
   * **npm install pm2 -g**
   * **2.2 安装 PM2 Windows 服务模块:**
   * **npm install pm2-windows-service -g**
   * **2.3 (关键步骤) 配置环境变量:**
     + **打开一个 PowerShell 或 CMD 窗口，运行以下命令来查找 npm 的全局模块安装路径：**
     + **npm root -g**
     + **此命令会输出一个路径，通常是 C:\Users\[你的用户名]\AppData\Roaming\npm\node\_modules。您需要的是它的上一级目录，即 C:\Users\[你的用户名]\AppData\Roaming\npm。**
     + **复制这个路径 (不包含 \node\_modules)。**
     + **在服务器的搜索栏中搜索“编辑系统环境变量”并打开它。**
     + **点击“环境变量...”。**
     + **在“系统变量”下找到 Path 变量，点击“编辑”。**
     + **点击“新建”，然后将您刚刚复制的 npm 全局路径粘贴进去。**
     + **一路点击“确定”保存。**
   * **2.4 将 PM2 设置为 Windows 服务:**
     + **非常重要：关闭所有已打开的 PowerShell/CMD 窗口，然后以管理员身份重新打开一个新的 PowerShell 窗口，以确保新的环境变量生效。**
     + **运行安装命令:**
     + **pm2-service-install**
     + **此时会弹出一个确认框，询问是否要为当前用户安装服务。点击“是 (Yes)”。**
     + **它会自动将 PM2 设置为一个在后台运行、并且开机自启的 Windows 服务。**
3. **使用 PM2 启动您的应用:**
   * **在项目目录下，打开 PowerShell 或 CMD，运行：**
   * **pm2 start server.js --name "bookmarker-api"**
   * **保存进程列表: 运行以下命令，PM2 会记住当前正在运行的应用。当服务器重启时，PM2 服务会自动重新启动这些应用。**
   * **pm2 save**
   * **常用 PM2 命令:**
     + **pm2 list: 查看所有应用状态**
     + **pm2 restart bookmarker-api: 重启应用**
     + **pm2 logs bookmarker-api: 查看日志**
     + **pm2 stop bookmarker-api**
     + **pm2 delete bookmarker-api**
4. **配置 IIS 反向代理:**
   * **打开“Internet Information Services (IIS) 管理器”。**
   * **在左侧“连接”窗格中，右键点击“网站”，选择“添加网站”。**
   * **网站名称: bookmarker-api**
   * **物理路径: C:\inetpub\wwwroot (或任何一个空目录)**
   * **主机名: api.yourdomain.com**
   * **点击“确定”。**
   * **选中您刚刚创建的网站，在中间的功能视图中双击 "URL 重写"。**
   * **在右侧“操作”窗格中，点击“添加规则”，选择“反向代理”，确定。**
   * **在“入站规则”中，输入您的 Node.js 应用正在侦听的地址：http://localhost:5000。**
   * **点击“确定”保存。IIS 现在会将所有发往 api.yourdomain.com 的请求转发给您的 Node.js 应用。**
5. **配置 HTTPS (使用 win-acme):**
   * **访问** [**win-acme 官网**](https://www.google.com/search?q=https://www.win-acme.com/downloads) **下载最新的 .zip 压缩包。**
   * **解压到一个固定的位置，例如 C:\win-acme。**
   * **以管理员身份运行 wacs.exe。**
   * **程序会显示一个菜单，选择 N (Create a new certificate)。**
   * **选择 1 (Single binding of an IIS site)，然后选择您刚刚创建的 bookmarker-api 网站。**
   * **按照提示完成验证，win-acme 会自动从 Let's Encrypt 获取证书、配置 IIS 绑定，并创建一个计划任务来自动续期证书。**

**第 5 部分：完成**

**恭喜！您的服务器端程序现在已经成功部署，并通过 https://api.yourdomain.com 安全地对外提供服务。**

**最后一步:**

* **确保在您的 Chrome 扩展程序的前端代码中，将所有 API 请求的地址更新为您部署好的域名。**
* **在 Google/GitHub 等 OAuth 提供商的开发者后台，将您的服务器域名添加到授权的回调 URL 列表中。**