**服务器端程序部署说明**

本指南将引导您完成“智能收藏夹”后端服务器及 MongoDB 数据库的完整部署流程。

**第 1 部分：前期准备 (Prerequisites)**

在开始之前，请确保您已准备好以下资源：

1. **一台云服务器**:
   * **推荐配置**: 至少 1 核 CPU / 1 GB 内存 / 25 GB SSD 存储。
   * **推荐操作系统**: Ubuntu 22.04 LTS 或 CentOS 7+。
   * **服务商**: 阿里云 (ECS)、腾讯云 (CVM)、华为云或任何其他云服务提供商。
   * **安全组/防火墙**: 确保服务器的 22 (SSH), 80 (HTTP), 443 (HTTPS) 端口是开放的。
2. **一个域名 (Domain Name)**:
   * 您需要一个域名用于访问您的 API，例如 api.yourdomain.com。
   * 请确保您已将该域名的 DNS 解析指向您服务器的公网 IP 地址。
3. **SSH 客户端**:
   * 用于连接到您的云服务器。Windows 用户可以使用 PuTTY、Xshell 或 Windows Terminal；macOS/Linux 用户可以直接使用终端。
4. **Node.js 环境**:
   * 服务器上需要安装 Node.js (推荐 v18.x 或更高版本) 和 npm。

**第 2 部分：MongoDB 数据库部署**

请根据您的偏好选择 **方案 A** 或 **方案 B** 中的一种。

**方案 A: 在自己的服务器上安装 MongoDB (Self-Hosted)**

1. **通过 SSH 连接到您的服务器**。
2. **安装 MongoDB**:
   * **对于 Ubuntu 22.04**:
   * sudo apt-get update
   * sudo apt-get install -y mongodb
   * **对于 CentOS 7**:
   * # 添加 MongoDB 的 yum 仓库
   * sudo tee /etc/yum.repos.d/mongodb-org-6.0.repo <<EOL
   * [mongodb-org-6.0]
   * name=MongoDB Repository
   * baseurl=https://repo.mongodb.org/yum/redhat/7/mongodb-org/6.0/x86\_64/
   * gpgcheck=1
   * enabled=1
   * gpgkey=https://www.mongodb.org/static/pgp/server-6.0.asc
   * EOL
   * # 安装 MongoDB
   * sudo yum install -y mongodb-org
3. **启动并设置开机自启**:
4. sudo systemctl start mongod
5. sudo systemctl enable mongod
6. **安全配置 (非常重要！)**:
   * 默认安装的 MongoDB 可能没有启用访问控制。我们需要创建一个管理员用户并启用认证。
   * 进入 MongoDB Shell:
   * mongosh
   * 在 MongoDB Shell 中执行以下命令:
   * // 1. 切换到 admin 数据库
   * use admin
   * // 2. 创建一个管理员用户 (请务必替换 'YourAdminPassword')
   * db.createUser({
   * user: "myAdmin",
   * pwd: passwordPrompt(), // 会提示您输入密码，更安全
   * roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
   * })
   * // 3. 退出 shell
   * exit
   * 编辑 MongoDB 配置文件以启用认证:
   * sudo nano /etc/mongod.conf
   * 找到 #security: 部分，取消注释并修改为:
   * security:
   * authorization: "enabled"
   * 重启 MongoDB 服务使配置生效:
   * sudo systemctl restart mongod
7. **为您的应用创建专用用户**:
   * 使用管理员账号重新登录 MongoDB Shell:
   * mongosh -u "myAdmin" -p --authenticationDatabase "admin"
   * 在 Shell 中为您的应用创建数据库和用户 (请替换 YourAppPassword):
   * // 1. 切换到您的应用数据库 (如果不存在会自动创建)
   * use smart\_bookmarker\_prod
   * // 2. 为该数据库创建一个专用用户
   * db.createUser({
   * user: "bookmarkerAppUser",
   * pwd: passwordPrompt(),
   * roles: [ { role: "readWrite", db: "smart\_bookmarker\_prod" } ]
   * })
   * // 3. 退出
   * exit
   * **记下您的数据库连接字符串 (MONGO\_URI)**，它将用于后续配置: mongodb://bookmarkerAppUser:YourAppPassword@localhost:27017/smart\_bookmarker\_prod

**方案 B: 使用 MongoDB Atlas (云数据库 - 推荐)**

1. 访问 [MongoDB Atlas 官网](https://www.mongodb.com/cloud/atlas) 并注册一个免费账户。
2. **创建一个免费集群 (Cluster)**: 按照引导，选择一个云服务商和离您用户最近的区域（例如 AWS / Asia Pacific (Hong Kong)），创建一个 M0 级别的免费集群。
3. **创建数据库用户**: 在左侧菜单的 "Database Access" 下，创建一个新的数据库用户。记下**用户名**和**密码**。
4. **配置网络访问**: 在 "Network Access" 下，添加允许访问的 IP 地址。为了开发方便，可以先添加 0.0.0.0/0 (允许任何 IP 访问)，但在生产环境中，应限制为您的服务器 IP。
5. **获取连接字符串**: 返回到集群概览页面，点击 "Connect"，选择 "Connect your application"，然后选择 Node.js 版本。复制提供的连接字符串。
6. **记下您的 MONGO\_URI**: 将连接字符串中的 <password> 替换为您刚刚创建的数据库用户密码。它看起来像这样: mongodb+srv://YourUsername:YourPassword@cluster0.xxxxx.mongodb.net/smart\_bookmarker\_prod?retryWrites=true&w=majority

**第 3 部分：服务器端应用程序部署**

1. **安装 Node.js 和 Git**:
2. # 对于 Ubuntu
3. sudo apt-get update
4. sudo apt-get install -y git nodejs npm
5. # 对于 CentOS
6. sudo yum install -y git nodejs
7. **克隆您的代码**:
8. git clone <your\_repository\_url>
9. cd smart-bookmarker-server-prod
10. **安装依赖**:
11. npm install --production # --production 标志会跳过 devDependencies
12. **配置环境变量**:
    * 在项目根目录创建一个 .env 文件:
    * nano .env
    * 将您之前准备好的所有环境变量填入其中:
    * MONGO\_URI=... (根据您选择的方案 A 或 B 填写)
    * JWT\_SECRET=... (生成一个长且随机的字符串)
    * PORT=5000
    * GOOGLE\_CLIENT\_ID=...
    * GOOGLE\_CLIENT\_SECRET=...
    * GITHUB\_CLIENT\_ID=...
    * GITHUB\_CLIENT\_SECRET=...

**第 4 部分：生产环境运行与反向代理**

直接使用 node server.js 运行应用是不稳定的，如果程序崩溃或服务器重启，应用就会停止。我们需要一个进程管理器，并使用 Nginx 作为反向代理来暴露服务和处理 HTTPS。

1. **安装 PM2 (进程管理器)**:
2. sudo npm install pm2 -g
3. **使用 PM2 启动您的应用**:
4. pm2 start server.js --name "bookmarker-api"
   * **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 startup 和 pm2 save: 设置 PM2 开机自启
5. **安装 Nginx**:
6. # 对于 Ubuntu
7. sudo apt-get install -y nginx
8. # 对于 CentOS
9. sudo yum install -y epel-release && sudo yum install -y nginx
10. **配置 Nginx 反向代理**:
    * 为您的 API 域名创建一个新的 Nginx 配置文件:
    * sudo nano /etc/nginx/sites-available/api.yourdomain.com
    * 将以下配置粘贴进去，并替换 api.yourdomain.com 和 PORT (5000):
    * server {
    * listen 80;
    * server\_name api.yourdomain.com;
    * location / {
    * proxy\_pass http://localhost:5000; # 将请求转发给 Node.js 应用
    * proxy\_http\_version 1.1;
    * proxy\_set\_header Upgrade $http\_upgrade;
    * proxy\_set\_header Connection 'upgrade';
    * proxy\_set\_header Host $host;
    * proxy\_cache\_bypass $http\_upgrade;
    * }
    * }
    * 启用该配置:
    * sudo ln -s /etc/nginx/sites-available/api.yourdomain.com /etc/nginx/sites-enabled/
11. **配置 HTTPS (使用 Let's Encrypt)**:
    * 安装 Certbot:
    * # 对于 Ubuntu
    * sudo apt-get install -y certbot python3-certbot-nginx
    * # 对于 CentOS
    * sudo yum install -y certbot python2-certbot-nginx
    * 自动获取并配置 SSL 证书:
    * sudo certbot --nginx -d api.yourdomain.com
    * 按照提示操作，Certbot 会自动修改您的 Nginx 配置以启用 HTTPS 并设置自动续期。
12. **重启 Nginx**:
13. sudo systemctl restart nginx

**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。