

```
1 @echo off
2 chcp 65001 >nul
3 setlocal enabledelayedexpansion
4
5 :: 设置标题
6 title GreenWeb 启动工具
7
8 echo =====
9 echo          GreenWeb 启动工具
10 echo =====
11 echo.
12 ";
13 :: 检查Node.js和npm是否安装
14 echo 检查环境依赖 ...
15 where node >nul 2>nul
16 if %errorlevel% neq 0 (
17     echo Node.js未安装, 准备自动安装 ...
18     call :InstallNodeJS
19 ) else (
20     for /f "tokens=*" %i in ('node -v') do set nodeVersion=%i
21     echo 已安装Node.js版本: !nodeVersion!
22 )
23
24 where npm >nul 2>nul
25 if %errorlevel% neq 0 (
26     echo npm未安装, 可能需要重新安装Node.js ...
27     call :InstallNodeJS
28 ) else (
29     for /f "tokens=*" %i in ('npm -v') do set npmVersion=%i
30     echo 已安装npm版本: !npmVersion!
31 )
32
33 :: 检查项目依赖是否安装
34 if not exist "node_modules" (
35     echo 项目依赖未安装, 准备安装 ...
36
37     :: 检查package.json文件是否存在
38     if not exist "package.json" (
39         echo 错误: 未找到package.json文件, 请确保在正确的项目目录中执行此脚本。
40         goto End
41     )
42
43     echo 正在安装项目依赖, 这可能需要几分钟 ...
44     call npm install
45     call cd server
46     call npm install
47     call cd ..
48     if !errorlevel! neq 0 (
49         echo 依赖安装失败, 请检查网络连接或手动运行 npm install
50         goto End
51     )
52     echo 依赖安装完成!
53 ) else if not exist "server/node_modules" (
```

```
53     echo 正在安装项目依赖，这可能需要几分钟 ...
54     call cd server
55     call npm install
56     call cd ..
57     echo 依赖安装完成!
58 )else (
59     echo 项目依赖已安装
60 )
61
62 :Menu
63 echo.
64 echo =====
65 echo                    请选择启动模式
66 echo =====
67 echo 1. 开发模式（前后端分离）
68 echo 2. 生产模式（整合前后端）
69 echo 3. 仅构建项目
70 echo 4. 更新项目依赖
71 echo 5. 退出
72 echo =====
73 echo.
74
75 set /p choice=请选择操作（1-5）:
76
77 if "%choice%"=="1" (
78     call :StartDevMode
79 ) else if "%choice%"=="2" (
80     call :StartProdMode
81 ) else if "%choice%"=="3" (
82     call :BuildProject
83 ) else if "%choice%"=="4" (
84     call :UpdateDependencies
85 ) else if "%choice%"=="5" (
86     echo 退出程序 ...
87     exit /b
88 ) else (
89     echo 无效的选择，请重新输入!
90     goto Menu
91 )
92
93 goto End
94
95 :InstallNodeJS
96 echo.
97 echo 准备安装Node.js ...
98 echo 正在下载Node.js安装程序 ...
99
100 :: 创建临时目录
101 set tempDir=%temp%\NodeJSInstall
102 mkdir %tempDir% 2>nul
103
104 :: 检查是否已安装curl
105 where curl >nul 2>nul
106 if %errorlevel% neq 0 (
107     echo 未找到curl工具，使用PowerShell下载 ...
```

```
108
109     echo 使用PowerShell下载Node.js ...
110     powershell -Command "& {[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12; Invoke-WebRequest -Uri
'https://nodejs.org/dist/v18.17.0/node-v18.17.0-x64.msi' -OutFile
'%tempDir%\nodejs_setup.msi'}"
111 ) else (
112     :: 使用curl下载Node.js安装程序
113     curl -L -o "%tempDir%\nodejs_setup.msi"
'https://nodejs.org/dist/v18.17.0/node-v18.17.0-x64.msi'
114 )
115
116 if not exist "%tempDir%\nodejs_setup.msi" (
117     echo 下载Node.js失败, 请手动访问 https://nodejs.org/ 下载并安装
118     start https://nodejs.org/
119     goto End
120 )
121
122 echo 下载完成, 开始安装Node.js ...
123 :: 静默安装Node.js
124 msixexec /i "%tempDir%\nodejs_setup.msi" /qn
125 if %errorlevel% neq 0 (
126     echo Node.js安装失败, 尝试手动安装 ...
127     start "" "%tempDir%\nodejs_setup.msi"
128     goto End
129 )
130
131 echo Node.js安装完成!
132 echo 请重新启动此脚本以继续操作 ...
133 goto End
134
135 :StartDevMode
136 echo 启动开发模式 ...
137
138 :: 检查并关闭占用端口的进程
139 call :CheckAndClosePort 3000
140 call :CheckAndClosePort 5173
141
142 :: 确保package.json中有相应的脚本命令
143 findstr /c:"\"server:dev\""" package.json >nul
144 if %errorlevel% neq 0 (
145     echo 警告: package.json 中未找到 "server:dev" 脚本。
146     echo 请确保项目配置正确。
147     goto End
148 )
149
150 findstr /c:"\"dev\""" package.json >nul
151 if %errorlevel% neq 0 (
152     echo 警告: package.json 中未找到 "dev" 脚本。
153     echo 请确保项目配置正确。
154     goto End
155 )
156
157 echo 1. 启动后端服务器 ...
158 start cmd /k "cd /d %~dp0 && npm run server:dev"
```

```
159 echo 2. 启动前端开发服务器 ...
160 start cmd /k "cd /d %~dp0 && npm run dev"
161 echo 服务已启动! 请在浏览器中打开 http://localhost:5173
162 goto End
163
164 :StartProdMode
165 echo 启动生产模式 ...
166
167 :: 检查并关闭占用端口的进程
168 call :CheckAndClosePort 3000
169
170 :: 确保package.json中有相应的脚本命令
171 findstr /c:"\"build\"" package.json >nul
172 if %errorlevel% neq 0 (
173     echo 警告: package.json 中未找到 "build" 脚本。
174     echo 请确保项目配置正确。
175     goto End
176 )
177
178 findstr /c:"\"start\"" package.json >nul
179 if %errorlevel% neq 0 (
180     echo 警告: package.json 中未找到 "start" 脚本。
181     echo 请确保项目配置正确。
182     goto End
183 )
184
185 echo 1. 构建前端应用 ...
186 call npm run build
187 if %errorlevel% neq 0 (
188     echo 构建失败, 请检查错误信息
189     goto End
190 )
191
192 echo 2. 启动整合服务器 ...
193 start cmd /k "cd /d %~dp0 && npm run start"
194
195 echo 服务已启动! 请在浏览器中打开 http://localhost:3000
196 goto End
197
198 :BuildProject
199 echo 执行项目构建 ...
200
201 :: 确保package.json中有相应的脚本命令
202 findstr /c:"\"build\"" package.json >nul
203 if %errorlevel% neq 0 (
204     echo 警告: package.json 中未找到 "build" 脚本。
205     echo 请确保项目配置正确。
206     goto End
207 )
208
209 call npm run build
210 if %errorlevel% neq 0 (
211     echo 构建失败, 请检查错误信息
212 ) else (
213     echo 构建完成! 可以通过 npm run start 启动服务
```

```
214 )
215 goto End
216
217 :UpdateDependencies
218 echo 更新项目依赖 ...
219 call npm update
220 if %errorlevel% neq 0 (
221     echo 依赖更新失败，请检查网络连接或手动运行 npm update
222 ) else (
223     echo 依赖更新完成!
224 )
225 goto End
226
227 :CheckAndClosePort
228 echo 检查端口 %1 占用情况 ...
229 FOR /F "tokens=5" %%P IN ('netstat -ano ^| findstr :%1 ^| findstr LISTENING')
DO (
230     echo 发现进程占用端口 %1: %%P, 尝试关闭 ...
231     taskkill /F /PID %%P 2>NUL
232     if !errorlevel! equ 0 (
233         echo 端口 %1 已释放
234     ) else (
235         echo 无法释放端口 %1, 可能需要手动关闭进程
236     )
237 )
238 goto :eof
239
240 :End
241 echo.
242 echo 按任意键继续 ...
243 pause >nul
```