

绿区专案 — AI Browser（企业管控浏览器） 主要功能测试剧本

产品：AI Browser（企业管控浏览器）

说明：企业内部的 AI 管控浏览器，员工通过它访问放行的 AI 网址

核心功能：白名单管控网址访问、内置书签预设云端/地端 AI 网址

地端 AI：AvaryGPT（内网部署）

云端 AI：配置中允许的云端 AI 网站

测试对象：AI Browser 使用者

三维度：CASE → SCENARIO → SCRIPT

建立日期：2026-07-20 / ZDT CONFIDENTIAL

 测试总览

#	功能	CASE 数	SCENARIO 数	SCRIPT 数
1	网址访问白名单管控	4	2	1
2	内置书签预设（云端/地端 AI）	3	1	1
3	个人聊天记录隔离	3	1	1
4	DLP 防泄漏	4	1	1
5	跨模块串接验证	2	2	1
6	安全与防绕过	3	1	1
	合计	19	8	6

 测试角色

角色	身份	权限
USR-Bob	一般员工	标准权限，可访问白名单网址
ADM-Admin	浏览器管理员	管理白名单、更新书签
EXT-Eve	外部人员	无公司凭证，不可使用

场景 1：网址访问白名单管控

CASE

CASE-BROWSER-URL-01：访问放行的云端 AI 网址

- **输入：**Bob 在 AI Browser 地址栏输入允许的云端 AI 网址（如 `https://chatgpt.com`）
- **预期：**正常打开，网页加载完成
- **不通过条件：**被拦截或无法加载

CASE-BROWSER-URL-02：访问地端 AI（AvaryGPT）网址

- **输入：**Bob 在地址栏输入地端 AvaryGPT 网址（内网 URL）
- **预期：**正常打开，可在内网使用 AvaryGPT
- **不通过条件：**访问失败或跳转到外网

CASE-BROWSER-URL-03：访问未放行的网址

- **输入：**Bob 在地址栏输入未在白名单中的网站（如 `https://www.youtube.com`）
- **预期：**弹出拦截页面「此网址不在允许列表中，如需访问请联系管理员」
- **不通过条件：**可正常访问

CASE-BROWSER-URL-04：管理员新增/移除白名单网址

- **输入：**Admin 在管理后台新增一个网址到白名单
- **预期：**新增后 Bob 即刻可以访问该网址；移除后无法访问
- **不通过条件：**新增后仍被拦截，或移除后仍可访问

SCENARIO

SCENARIO-BROWSER-URL-A：白名单放行 vs 阻挡

参与者：USR-Bob

剧情：

1. Bob 打开 AI Browser
2. 在地址栏输入 `https://chatgpt.com`（云端 AI）→  正常打开
3. 输入地端 AvaryGPT 网址 →  正常打开
4. 输入 `https://www.youtube.com` →  弹出拦截页
5. 输入 `https://www.facebook.com` →  弹出拦截页
6. 拦截页显示「此网址不在允许列表中，请联系管理员」

验证点：白名单内外网址正确放行/拦截

SCENARIO-BROWSER-URL-B: 白名单变更即时生效

参与者: ADM-Admin + USR-Bob

剧情:

1. Admin 在管理后台将 `https://claude.ai` 加入白名单
2. Bob 刷新浏览器 → 输入 `https://claude.ai` →  正常访问
3. Admin 将 `https://claude.ai` 从白名单移除
4. Bob 刷新 → 再次访问 →  被拦截
5. 变更即时生效, 无需重启浏览器

验证点: 白名单变更即时生效

SCRIPT

SCRIPT-BROWSER-URL-1: 白名单功能验证

```
# browser_script_01_whitelist.ps1
# 在 AI Browser 中手动执行
Write-Host "=== 场景 1: 网址访问白名单管控 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host "🕒 路径 A - 放行网址测试:" -ForegroundColor Yellow
Write-Host "  1. 在地址栏输入云端 AI 网址 (如 chatgpt.com)" -ForegroundColor Yellow
Write-Host "  2. 确认页面正常加载  " -ForegroundColor Yellow
Write-Host "  3. 在地址栏输入地端 AvaryGPT 网址" -ForegroundColor Yellow
Write-Host "  4. 确认可正常使用 AvaryGPT  " -ForegroundColor Yellow

Write-Host ""
Write-Host "🕒 路径 B - 未放行网址测试:" -ForegroundColor Yellow
Write-Host "  1. 输入 youtube.com → 确认弹出拦截页" -ForegroundColor Yellow
Write-Host "  2. 输入 facebook.com → 确认弹出拦截页" -ForegroundColor Yellow
Write-Host "  3. 拦截页显示引导文字「请联系管理员」" -ForegroundColor Yellow

Write-Host ""
Write-Host "🕒 路径 C - 白名单变更测试 (需管理员配合):" -ForegroundColor Yellow
Write-Host "  1. 管理员新增一个网址到白名单" -ForegroundColor Yellow
Write-Host "  2. 确认即刻可访问  " -ForegroundColor Yellow
Write-Host "  3. 管理员移除该网址" -ForegroundColor Yellow
Write-Host "  4. 确认被拦截  " -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

场景 2：内置书签预设（云端/地端 AI）

CASE

CASE-BROWSER-BMK-01：书签栏默认包含云端 AI 网址

- **输入：**Bob 首次打开 AI Browser，查看书签栏
- **预期：**书签栏中默认包含放行的云端 AI 网址（如 ChatGPT、Claude 等）
- **不通过条件：**书签栏为空或缺少云端 AI 书签

CASE-BROWSER-BMK-02：书签栏默认包含地端 AI 网址（AvaryGPT）

- **输入：**Bob 查看书签栏
- **预期：**书签中包含地端 AvaryGPT 的链接
- **不通过条件：**缺少 AvaryGPT 书签

CASE-BROWSER-BMK-03：书签点击即可访问

- **输入：**Bob 点击书签栏上的任一 AI 网址书签
- **预期：**直接跳转到该网址，正常打开
- **不通过条件：**点击无反应或跳转后无法访问

SCENARIO

SCENARIO-BROWSER-BMK-A：书签栏预设 + 一键访问

参与者：USR-Bob

剧情：

1. Bob 打开 AI Browser
2. 书签栏显示预设的云端 AI 网址（如 ChatGPT、Claude 等）
3. 书签栏显示地端 AI 网址（AvaryGPT）
4. 点击 ChatGPT 书签 → 直接跳转到 ChatGPT，自动打开
5. 点击 AvaryGPT 书签 → 直接跳转到内网 AvaryGPT，自动打开
6. 用户不用记网址，点书签即可开始使用 AI

验证点：书签完整、一键可达

SCRIPT

SCRIPT-BROWSER-BMK-1：书签功能验证

```
# browser_script_02_bookmarks.ps1
Write-Host "=== 场景 2：内置书签预设 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host " ⌚ 请手动操作：" -ForegroundColor Yellow
Write-Host " 1. 打开 AI Browser" -ForegroundColor Yellow
Write-Host " 2. 查看书签栏 → 截图记录所有预设书签" -ForegroundColor Yellow
Write-Host " 3. 确认包含云端 AI 网址书签（如 ChatGPT、Claude 等）" -ForegroundColor Yellow
Write-Host " 4. 确认包含地端 AI 网址书签（AvaryGPT）" -ForegroundColor Yellow
Write-Host " 5. 点击每个书签 → 确认都能正常打开对应网站" -ForegroundColor Yellow
Write-Host " 6. 记录书签数量：云端 AI ____ 个，地端 AI ____ 个" -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

场景 3：个人聊天记录隔离

CASE

CASE-BROWSER-ISOL-01：不同用户在 AI 网站的聊天记录独立

- **输入：**Bob 和 Alice 分别用各自账号登入 AI Browser，访问同一 AI 网站
- **预期：**两人各自的聊天记录不同，互不干扰
- **不通过条件：**Bob 看到 Alice 的记录

CASE-BROWSER-ISOL-02：同一用户不同设备同步

- **输入：**Bob 在电脑 A 上打开 ChatGPT 聊了几句，再到电脑 B 打开 AI Browser 访问 ChatGPT
- **预期：**电脑 B 看到电脑 A 的历史记录（企业账号绑定）
- **不通过条件：**电脑 B 无历史记录

CASE-BROWSER-ISOL-03：用户无法使用个人账号登录 AI 网站

- **输入：**Bob 在 ChatGPT 页面尝试点击「Log in with personal account」
- **预期：**被阻止或提示「请使用企业账号，无法使用个人账号登录」
- **不通过条件：**可成功登录个人账号

SCENARIO

SCENARIO-BROWSER-ISOL-A: 聊天记录隔离

参与者: USR-Bob + USR-Alice

剧情:

1. Bob 登录 AI Browser, 打开 ChatGPT 书签
2. 自动使用企业 SSO 登录 ChatGPT (无需手动输入)
3. Bob 发送「我的项目进度报告」
4. Bob 退出登录
5. Alice 登录 AI Browser, 打开 ChatGPT
6. Alice 看到的是自己的空对话, 看不到 Bob 的记录
7. Alice 发送「我的数据分析需求」
8. Bob 再次登录 → 仍看到自己的「项目进度报告」对话, 看不到 Alice 的

验证点: 聊天记录严格按用户隔离

SCRIPT

SCRIPT-BROWSER-ISOL-1: 聊天记录隔离验证

```
# browser_script_03_isolation.ps1
Write-Host "=== 场景 3 : 个人聊天记录隔离 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host "🕒 需要 Bob 和 Alice 配合测试:" -ForegroundColor Yellow
Write-Host ""
Write-Host "步骤 A – Bob 创建对话:" -ForegroundColor Yellow
Write-Host "  1. Bob 登入 AI Browser" -ForegroundColor Yellow
Write-Host "  2. 点击 ChatGPT 书签 → 自动 SSO 登入" -ForegroundColor Yellow
Write-Host "  3. 发送一条消息「我的项目进度报告」" -ForegroundColor Yellow
Write-Host "  4. Bob 退出" -ForegroundColor Yellow
Write-Host ""
Write-Host "步骤 B – Alice 查看:" -ForegroundColor Yellow
Write-Host "  1. Alice 登入 AI Browser" -ForegroundColor Yellow
Write-Host "  2. 点击 ChatGPT 书签 → 自动 SSO 登入" -ForegroundColor Yellow
Write-Host "  3. 确认看不到 Bob 的对话 (空对话)" -ForegroundColor Yellow
Write-Host "  4. Alice 发送「我的数据分析需求」" -ForegroundColor Yellow
Write-Host "  5. Alice 退出" -ForegroundColor Yellow
Write-Host ""
Write-Host "步骤 C – Bob 再登入:" -ForegroundColor Yellow
Write-Host "  1. Bob 再次登入" -ForegroundColor Yellow
Write-Host "  2. 确认仍看到自己的「项目进度报告」对话" -ForegroundColor Yellow
Write-Host "  3. 确认看不到 Alice 的对话 ✅" -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

场景 4：DLP 防泄漏

CASE

CASE-BROWSER-DLP-01：输入侧 DLP — 粘贴内容检测

- **输入：**Bob 从内部系统复制一段带「机密」「ZD-CONFIDENTIAL」标记的文字，粘贴到 ChatGPT 对话输入框
- **预期：**AI Browser 检测到粘贴内容含敏感标记，弹出提示「检测到机密信息，已阻止发送」
- **不通过条件：**机密信息成功发送到 AI 网站

CASE-BROWSER-DLP-02：输入侧 DLP — 文件上传检测

- **输入：**Bob 在 ChatGPT 上传一个文件名含「机密」或「薪资」的文件
- **预期：**AI Browser 拦截上传，提示「该文件含有敏感信息，禁止上传」
- **不通过条件：**文件成功上传

CASE-BROWSER-DLP-03：输入侧 DLP — 手机号/身份证正则拦截

- **输入：**Bob 在对话框中输入「我的身份证 A123456789，帮我分析」
- **预期：**AI Browser 探测到身份证号，弹窗提示「检测到个人身份信息，是否脱敏后发送？」
- **不通过条件：**身份证号原文发送到 AI 网站

CASE-BROWSER-DLP-04：输出侧 DLP — AI 回应含敏感词警告

- **输入：**Alice 在 AI 网站问「生成一份完整的客户名单」
- **预期：**AI 生成后，AI Browser 探测到输出含客户名单，弹出提示「此输出可能包含客户隐私数据，请注意使用」
- **不通过条件：**无任何提示直接显示

SCENARIO

SCENARIO-BROWSER-DLP-A：DLP 输入 + 输出防护

参与者：USR-Bob

剧情：

1. Bob 在 AI Browser 中打开 ChatGPT
2. 从内部系统复制一段含「机密」文字到 ChatGPT → 拦截并提示
3. 尝试上传一个 `薪资报表.xlsx` → 拦截
4. 输入「我的身份证 A123456789」→ 提示脱敏

- 5. 点「脱敏后发送」→ 身份证被遮罩为 A12***89 后发送
- 6. AI 回应中生成客户名单 → 浏览器弹窗警告「输出含客户隐私数据」

验证点：输入侧拦截 + 输出侧警告

SCRIPT

SCRIPT-BROWSER-DLP-1: DLP 功能验证

```
# browser_script_dlp.ps1
Write-Host "=== 场景 4 : DLP 防泄漏 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host "🕒 输入侧 DLP 测试：" -ForegroundColor Yellow
Write-Host ""
Write-Host "1. 文本粘贴检测：" -ForegroundColor Yellow
Write-Host "    复制「机密数据 ZD-CONFIDENTIAL」→ 贴到 ChatGPT 输入框" -ForegroundColor Yellow
Write-Host "    预期：弹窗提示「检测到机密信息，已阻止发送」✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "2. 文件上传检测：" -ForegroundColor Yellow
Write-Host "    上传一个名为「薪资报表_2026.xlsx」的文件" -ForegroundColor Yellow
Write-Host "    预期：拦截上传，提示含敏感信息✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "3. PII 检测：" -ForegroundColor Yellow
Write-Host "    输入「我的身份证 A123456789」" -ForegroundColor Yellow
Write-Host "    预期：弹窗提示「检测到个人信息，是否脱敏后发送？」✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "🕒 输出侧 DLP 测试：" -ForegroundColor Yellow
Write-Host ""
Write-Host "4. 在 AI 中间「生成一份完整的客户名单」" -ForegroundColor Yellow
Write-Host "    预期：输出弹出警告「此输出含客户隐私数据」✅" -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

场景 5：跨模块串接验证

说明

绿区专案 4 个模块需要串接验证才能确认完整链路正常：

模块	角色	说明
VDI	用户桌面	开启桌面环境
AI Browser	企业浏览器	白名单 + DLP + 自动登录
CSP	SSO 代理 + base_url	免登录 + 聊天隔离 + API 统一入口
LLM Gateway	模型服务	模型路由 + 配额 + 安全

CASE

CASE-XMOD-01: VDI → AI Browser → CSP → ChatGPT 全链路

- **输入:** Bob 在 VDI 中打开 AI Browser → 点击 ChatGPT 书签 → CSP 自动登录后提问
- **预期:**
 - VDI 中 AI Browser 正常启动
 - ChatGPT 通过 CSP 自动登录 (SSO 无感)
 - 提问正常, AI 回应
- **不通过条件:** 任一环节失败

CASE-XMOD-02: Vibe Coding → CSP base_url → LLM Gateway

- **输入:** Alice 在 VDI 中的 VS Code 配置 `base_url=https://csp.zd.com/v1`
- **预期:** Vibe Coding 工具正常使用所有模型 (Fable-5、Kimi-K3 等)
- **不通过条件:** 无法连接或模型调用失败

SCENARIO

SCENARIO-XMOD-A: 完整绿区工作流程验证

参与者: USR-Bob → VDI → AI Browser → CSP → AI 网站

剧情:

1. Bob 登入 VDI (开发者桌面)
2. 打开 AI Browser → 书签选中
3. ChatGPT → CSP 自动登录 → 提问「帮我分析这份报告」
4. 从本地拖入一个文件到 VDI → 弹窗「是否拷入绿区? 」→ 点「是」
5. 主管在钉钉审批通过
6. VDI 文件、CSP 登录、AI Browser 使用三条链路同时正常

SCENARIO-XMOD-B: 开发者全链路

参与者: DEV-Alice → VDI → VS Code → CSP base_url

剧情:

1. Alice 在 VDI 中打开 VS Code

- 2. 配置 AI 插件: `base_url=https://csp.zd.com/v1`
- 3. 选模型 Fable-5 → 写代码补全正常
- 4. 切换模型 Kimi-K3 → 补全正常
- 5. Alice 在 Terminal 中用 `curl` 测试 CSP API
- 6. 确认所有模型通过同一 `base_url` 可用

SCRIPT

SCRIPT-XMOD-1: 全链路验证

```
# browser_script_xmod.ps1
Write-Host "=== 场景 5：跨模块串接验证 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host "🕒 路径 A - VDI → AI Browser → CSP → AI：" -ForegroundColor Yellow
Write-Host "  1. 登入 VDI" -ForegroundColor Yellow
Write-Host "  2. 打开 AI Browser → 书签点击 ChatGPT" -ForegroundColor Yellow
Write-Host "  3. 确认 CSP 自动登录（无登录页）✅" -ForegroundColor Yellow
Write-Host "  4. 输入问题 → 确认 AI 回应 ✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "🕒 路径 B - VDI + VS Code → CSP base_url：" -ForegroundColor Yellow
Write-Host "  1. 在 VDI 中打开 VS Code" -ForegroundColor Yellow
Write-Host "  2. AI 插件配置 base_url=https://csp.zd.com/v1" -ForegroundColor Yellow
Write-Host "  3. 选 Fable-5 → 确认代码补全正常 ✅" -ForegroundColor Yellow
Write-Host "  4. 切 Kimi-K3 → 确认补全正常 ✅" -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

场景 6：安全与防绕过

CASE

CASE-BROWSER-SEC-01: 无法通过代理绕过白名单

- **输入:** Bob 在 AI Browser 中配置代理服务器，尝试通过代理访问被禁网址
- **预期:** 浏览器检测到代理设置，拦截访问或忽略代理配置
- **不通过条件:** 通过代理成功访问被禁网站

CASE-BROWSER-SEC-02: 无法安装第三方插件绕过

- **输入:** Bob 尝试在 AI Browser 中安装 VPN/代理类浏览器扩展
- **预期:** 被阻止，提示「此扩展不被允许」
- **不通过条件:** 成功安装并启用

CASE-BROWSER-SEC-03: 下载/上传文件管控

- **输入:** Bob 从某允许的 AI 网站下载文件
- **预期:** 下载的文件经过安全检查
- **不通过条件:** 未检查即允许下载

SCENARIO

SCENARIO-BROWSER-SEC-A: 防绕过测试

参与者: USR-Bob

剧情:

1. Bob 尝试在浏览器设置中打开代理 → 设置被锁定
2. Bob 尝试安装 VPN 插件 → 被阻止, 提示不允许
3. Bob 尝试通过 IP 直连访问被禁网站 → 仍被拦截
4. Bob 尝试用隐私/无痕模式 → 白名单规则仍生效
5. 所有绕过尝试均失败

SCRIPT

SCRIPT-BROWSER-SEC-1: 防绕过验证

```
# browser_script_04_security.ps1
Write-Host "=== 场景 4 : 安全与防绕过 ===" -ForegroundColor Cyan

Write-Host ""
Write-Host " 🕒 请测试以下绕过方式是否均被阻止 : " -ForegroundColor Yellow
Write-Host ""
Write-Host "1. 代理绕过 : " -ForegroundColor Yellow
Write-Host "   在浏览器设置中添加 HTTP 代理" -ForegroundColor Yellow
Write-Host "   结果 : 设置被锁定 ⚠️ 忽略代理配置 ✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "2. 扩展绕过 : " -ForegroundColor Yellow
Write-Host "   尝试安装 VPN 类浏览器扩展" -ForegroundColor Yellow
Write-Host "   结果 : 被阻止 ⚠️ 提示不允许 ✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "3. 隐私模式 : " -ForegroundColor Yellow
Write-Host "   打开无痕/隐私窗口 → 访问被禁网站" -ForegroundColor Yellow
Write-Host "   结果 : 仍被拦截 ✅" -ForegroundColor Yellow
Write-Host ""
Write-Host "4. IP 直连 : " -ForegroundColor Yellow
Write-Host "   直接输入被禁网站的 IP 地址" -ForegroundColor Yellow
Write-Host "   结果 : 仍被拦截 ✅" -ForegroundColor Yellow

Read-Host "按 Enter 继续"
```

附录：准备清单

#	准备项目	负责人	完成
1	AI Browser 已部署并可启动	开发	☐
2	网址白名单已配置（含放行/阻挡规则）	管理员	☐
3	云端 AI 网址书签已预设	管理员	☐
4	地端 AI / AvaryGPT 网址书签已预设	管理员	☐
5	企业 SSO 自动登入已对接 AI 网站	开发	☐
6	聊天记录隔离已启用	开发	☐
7	代理/插件/隐私模式管控已启用	管理员	☐

文件结束

版本：v1.0 · 2026-07-20

场景数：4 个主要功能 · 13 CASE · 5 SCENARIO · 4 SCRIPT