

Chrome Starting Page Dashboard v1.0.2

version v1.0.2

license MIT

Node.js v18+

dependencies zero

나만의 크롬 시작 페이지 대시보드. Node.js 로컬 서버 기반으로, 외부 서비스 의존 없이 내 컴퓨터에서만 실행됩니다.

Zero-dependency — npm install 없이 Node.js 내장 모듈만 사용합니다.

목차

1. [주요 기능](#)
2. [시작하기 전에](#)
3. [설치](#)
4. [크롬 시작 페이지로 설정](#)
5. [사용 방법](#)
6. [데이터 백업과 복원](#)
7. [문제 해결](#)
8. [완전 제거](#)
9. [프로젝트 구조](#)
10. [기술 사양](#)
11. [변경 이력](#)

주요 기능

📌 북마크 관리

- 카테고리 기반 카드 레이아웃 — 북마크를 카테고리별로 분류하여 카드로 표시
- 멀티 페이지 — 여러 페이지를 만들어 탭으로 전환 (이모지 탭 이름 지원)
- 드래그 앤 드롭 — 북마크 순서 변경, 카테고리 간 이동
- URL 드롭 — 브라우저 주소창에서 카드로 직접 드래그하여 추가
- 스마트 페이스트 — URL 붙여넣기 시 카테고리 선택 팝업 자동 표시
- NEW 배지 — 7일 이내 추가된 북마크에 배지 표시
- 사용 통계 — URL 기반 클릭 횟수, 시간대별 사용 패턴 추적
- 추천 시스템 — 자주 사용하는 북마크 상단에 추천 표시
- 데드링크 감지 — 접속 불가 사이트를 자동 감지 (3회 연속 실패 시 표시)

검색

- 멀티 검색 엔진 — Google, Naver, YouTube 등 원클릭 전환
- 스포트라이트 검색 (Ctrl+F) — 북마크, 할 일, 메모, D-Day 통합 검색
- 초성 검색 — 한글 초성만으로 북마크 검색 (ㅋㄹㄷ → 클라우드)
- 퍼지 매칭 — 오타가 있어도 유사한 결과 표시

메모장

- 복수 메모 카드 — 원하는 만큼 메모장 카드 생성
- 카드별 줄 수 설정 — 메모장마다 라인 수 조정 가능
- 실시간 자동 저장 — 입력 즉시 서버에 저장

할 일 (Todo)

- 우선순위 — 높음/보통/낮음 3단계
- 태그 분류 — 태그별 필터링
- 마감일 — 날짜 지정 및 기한 경과 표시
- 반복 설정 — 매일/매주/매월 자동 재생성
- 완료 기록 — 최근 90일 완료 이력 보존

달력 & 일정

- 월간/주간 보기 — 달력 카드에서 일정 확인
- 일정 등록 — 날짜, 시간, 반복, 카테고리 설정
- 알림 — 등록된 일정 시작 시 브라우저 알림

습관 트래커

- 일일 체크리스트 — 매일 반복할 습관 등록 및 체크
- 자동 리셋 — 매일 자정 체크 상태 초기화

D-Day

- 복수 D-Day — 여러 개의 중요한 날짜 카운트다운
- 위젯 표시 — 대시보드 상단에 남은 일수 표시

시계 & 날씨

- 멀티 시계 — 여러 시간대 시계 동시 표시
- 시간 오버레이 — 배경 위에 현재 시각 대형 표시
- 날씨 위젯 — OpenWeather API 연동 (API 키 필요)

테마 & 디자인

- **테마** — 다크 / 라이트 / 시간 자동 전환
- **강조 색상** — 8종 (블루, 퍼플, 그린, 오렌지 등)
- **글래스 프리셋** — 투명 / 기본 / 불투명 3단계
- **블러 강도** — 카드 배경 블러 수치 조정
- **배경** — 단색, 커스텀 이미지, 슬라이드쇼 (자동 전환)
- **레이아웃 프리셋** — 기본/컴팩트/와이드 전환
- **커스텀 CSS** — 직접 CSS를 작성하여 추가 커스터マイ징

단축키 & 조작

- **키보드 네비게이션** — 방향키로 북마크 탐색
- **넘버키 바로가기** — 1~9 숫자 키로 첫 카드의 북마크 열기
- **우클릭 컨텍스트 메뉴** — 북마크 편집, 이동, 삭제
- **멀티 셀렉트** — Ctrl/Shift 클릭으로 여러 북마크 동시 선택
- **인라인 편집** — 더블클릭으로 북마크 이름 즉시 수정
- **포커스 모드** — 북마크만 표시, 기타 위젯 숨김

데이터 & 안정성

- **자동 백업** — 스마트 보관 (7일 전체, 8~30일 1일 1개, 30일+ 삭제, 50개 캡)
- **프로필 관리** — 여러 프로필 전환 (직장/개인 등)
- **JSON 내보내기/가져오기** — 전체 데이터 백업 및 복원
- **드래그 가져오기** — JSON 파일을 페이지에 드래그하여 복원
- **휴지통** — 삭제된 항목 복구 가능
- **오프라인 지원** — Service Worker 캐싱으로 서버 중단 시에도 열람
- **데이터 무결성 검사** — 부팅 시 자동 검증
- **온보딩** — 최초 실행 시 안내 표시

서버

- **Gzip 압축** — 응답 압축으로 전송 속도 최적화
- **ETag 캐싱** — 변경되지 않은 파일은 304 응답
- **인메모리 캐시** — 정적 파일 캐싱으로 디스크 I/O 최소화
- **Graceful Shutdown** — 종료 시 자동 백업 후 안전 종료
- **에러 로깅** — server.error.log에 오류 기록 (10MB 자동 로테이션)
- **백그라운드 실행** — PowerShell 히든 윈도우로 실행 (콘솔 창 없음)
- **MCP 서버 보호** — 재시작 시 대시보드 포트만 종료 (다른 Node 프로세스 유지)

시작하기 전에

필요한 것

- **Node.js** (v18 이상) — <https://nodejs.org>에서 LTS 버전 다운로드
- 설치 확인: 터미널에서 `node -v` 입력 → `v18.x.x` 이상이면 OK

이 프로그램이 하는 일

- **포트 1111**에서 로컬 웹 서버를 실행합니다
- 모든 데이터는 프로젝트 폴더 안의 `data/` 디렉토리에 JSON 파일로 저장됩니다
- 인터넷 연결 없이도 동작합니다 (날씨 위젯 제외)
- 외부로 데이터를 전송하지 않습니다

이 프로그램이 하지 않는 일

- 시스템 레지스트리를 수정하지 않습니다
- 다른 프로그램에 영향을 주지 않습니다
- 포트 1111 외의 네트워크 리소스를 사용하지 않습니다
- 백그라운드에서 CPU/메모리를 거의 사용하지 않습니다 (대기 시 ~15MB)

설치

Windows

방법 1: 자동 설치 (권장)

1. 프로젝트 폴더를 원하는 위치에 배치합니다 (예: `C:\Users\내이름\Dashboard`)
2. `setup_windows.bat` 파일을 더블클릭합니다
3. 완료 메시지가 나타나면 `http://localhost:1111`로 접속합니다

자동 설치가 하는 일:

- Node.js 설치 여부를 확인합니다
- 서버를 백그라운드로 시작합니다 (PowerShell 히든 윈도우)
- Windows 시작 프로그램에 등록합니다 (PC 켜 때 자동 실행)

방법 2: 수동 실행

```
cd C:\Users\내이름\Dashboard  
node server.js
```

터미널에 `[Dashboard v1.0.2] http://localhost:1111/`이 표시되면 브라우저에서 해당 주소로 접속합니다. 이 방법은 터미널을 닫으면 서버가 종료됩니다.

서버 재시작

```
restart.bat
```

포트 1111을 사용하는 프로세스만 종료 후 백그라운드로 재시작합니다. 다른 Node.js 프로세스(MCP 서버 등)에 영향을 주지 않습니다.

macOS

방법 1: 자동 설치 (권장)

```
cd ~/Dashboard  
chmod +x setup_mac.sh  
./setup_mac.sh
```

자동 설치가 하는 일:

- Node.js 설치 여부를 확인합니다
- macOS LaunchAgent를 등록합니다 (로그인 시 자동 실행)
- 서버를 즉시 시작합니다

방법 2: 수동 실행

```
cd ~/Dashboard  
node server.js
```

Linux

```
cd ~/Dashboard  
node server.js
```

자동 시작을 원하면 systemd 서비스를 등록합니다:

```
sudo tee /etc/systemd/system/dashboard.service << EOF  
[Unit]  
Description=Chrome Starting Page Dashboard  
After=network.target  
  
[Service]  
Type=simple  
User=$USER  
WorkingDirectory=$HOME/Dashboard  
ExecStart=$(which node) server.js  
Restart=on-failure  
  
[Install]  
WantedBy=multi-user.target  
EOF  
  
sudo systemctl enable dashboard  
sudo systemctl start dashboard
```

크롬 시작 페이지로 설정

기본 설정

1. 크롬 주소창에 `chrome://settings/onStartup` 입력
2. "특정 페이지 또는 페이지 모음 열기" 선택
3. "새 페이지 추가" 클릭
4. `http://localhost:1111` 입력 후 추가

새 탭 페이지로도 사용하기 (선택)

크롬 확장 프로그램 [New Tab Redirect](#)를 설치한 뒤, 리다이렉트 URL을 `http://localhost:1111`로 설정하면 새 탭을 열 때마다 대시보드가 표시됩니다.

사용 방법

키보드 단축키

단축키	기능
/ 또는 <code>ctrl+K</code>	검색창 포커스
<code>ctrl+F</code>	스포트라이트 검색 (전체 통합 검색)
S	설정 열기
E	편집 모드 토글
T	테마 전환 (다크/라이트)
1 ~ 9	첫 번째 카드의 n번째 북마크 열기
← → ↑ ↓	북마크 탐색
<code>Escape</code>	모달/검색 닫기

북마크 관리

- **추가:** 카드 하단 입력창에 URL 붙여넣기, 또는 브라우저 주소창에서 카드로 드래그
- **정렬:** 드래그하여 순서 변경 또는 다른 카드로 이동
- **편집:** 더블클릭으로 이름 인라인 편집, 우클릭으로 상세 편집
- **삭제:** 마우스 오버 시 × 버튼 (휴지통으로 이동, Undo 가능)
- **멀티 셀렉트:** `Ctrl+클릭`으로 여러 개 선택 후 일괄 이동/삭제

카드 종류

카드	설명
북마크	카테고리별 즐겨찾기 링크 모음
메모장	간단한 메모 (여러 개 생성, 줄 수 조정 가능)
할 일	우선순위, 태그, 마감일, 반복 설정
달력	월간/주간 달력, 일정 등록 및 알림
습관	일일 습관 체크리스트 (자동 자동 리셋)

하단 영역의 + 카드 추가 버튼으로 원하는 카드를 추가합니다.

설정 (S 키)

- 테마: 다크/라이트/자동, 강조 색상 8종
- 클래스 효과: 투명도, 블러 강도, 프리셋
- 배경: 단색, 이미지 업로드, 슬라이드쇼
- 날씨: OpenWeather API 키 + 도시명 입력
- 검색: 기본 검색 엔진 변경, 커스텀 키워드 추가
- D-Day: 중요한 날짜 카운트다운
- 데이터: 내보내기/가져오기, 프로필 관리, 데드링크 초기화

데이터 백업과 복원

자동 백업

서버가 자동으로 주기적 백업을 생성합니다 (기본 24시간 간격).

스마트 보관 정책:

- 최근 7일: 모든 백업 보관
- 8~30일: 하루에 1개만 보관
- 30일 이후: 자동 삭제
- 전체 상한: 50개

수동 백업

설정(S) → 데이터 → 내보내기 버튼 → JSON 파일 다운로드

복원

설정(S) → 데이터 → 가져오기 버튼으로 JSON 파일 불러오기. 또는 JSON 파일을 페이지에 드래그 앤 드롭.

문제 해결

"포트 1111이 이미 사용 중" 오류

Windows:

```
netstat -ano | findstr :1111  
taskkill /PID <PID번호> /F
```

macOS/Linux:

```
lsof -i :1111  
kill <PID번호>
```

또는 `server.js`의 `const PORT = 1111;` 을 다른 번호로 변경하고, 크롬 시작 페이지 URL도 함께 변경합니다.

서버가 실행되지 않음

1. Node.js 설치 확인: `node -v`
2. 프로젝트 폴더에서 실행: `cd Dashboard && node server.js`
3. `data/` 폴더 존재 확인 (없으면 서버가 자동 생성)

페이지가 표시되지 않음

1. 서버 실행 중인지 확인 (터미널에 로그 출력 여부)
2. 주소가 `http://localhost:1111` 인지 확인 (`https`가 아닌 `http`)
3. 브라우저 개발자 도구(F12) → Console 탭에서 에러 확인

아이콘이 표시되지 않음

아이콘은 4단계 외부 서비스를 통해 로딩됩니다 (Google → DuckDuckGo → icon.horse → 직접 favicon.ico). 인터넷 연결이 필요하며, 모든 소스가 실패하면 이름 앞 2글자의 컬러 글래스 아이콘으로 대체됩니다.

데이터가 저장되지 않음

- 화면 우하단 "⚠ 저장 실패" 표시 → 서버 연결 문제
- 서버 재시작: `restart.bat` 실행
- `data/` 폴더 쓰기 권한 확인

완전 제거

시스템 레지스트리, 글로벌 설정, 숨겨진 폴더 등에 아무것도 남기지 않습니다. 아래 두 가지만 제거하면 됩니다.

Windows

```
:: 1. 자동 시작 제거  
del "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\Dashboard_StartPage.lnk"  
  
:: 2. 서버 종료  
netstat -ano | findstr :1111 | findstr LISTENING  
taskkill /PID <PID번호> /F  
  
:: 3. 프로젝트 폴더 삭제  
:: 4. chrome://settings/onStartup 에서 localhost:1111 제거
```

macOS

```
launchctl unload ~/Library/LaunchAgents/com.dashboard.startpage.plist  
rm ~/Library/LaunchAgents/com.dashboard.startpage.plist  
rm -rf ~/Dashboard
```

Linux (systemd)

```
sudo systemctl stop dashboard  
sudo systemctl disable dashboard  
sudo rm /etc/systemd/system/dashboard.service  
sudo systemctl daemon-reload  
rm -rf ~/Dashboard
```

프로젝트 구조

```
Chrome_Starting-Page/  
├── server.js           ← Node.js 서버 (API + 정적 파일 + Gzip + 캐시)  
├── index.html          ← 페이지 구조  
├── style.css           ← 스타일 (테마, 클래스, 애니메이션)  
├── script.js           ← 프론트엔드 로직 (~2,965줄)  
├── sw.js                ← Service worker (오프라인 캐싱)  
├── manifest.webmanifest ← PWA 매니페스트  
├── package.json         ← 프로젝트 메타데이터  
├── setup_windows.bat    ← windows 자동 설치  
├── setup_mac.sh         ← macOS 자동 설치  
├── restart.bat          ← 서버 재시작 (포트 1111만 종료)  
├── run_server_background.bat ← 백그라운드 실행  
├── assets/              ← 배경 이미지, PWA 아이콘, 슬라이드쇼  
└── data/                ← 사용자 데이터 (자동 생성)  
    ├── bookmarks.json  
    ├── config.json  
    ├── notes.json  
    ├── todos.json  
    ├── events.json  
    └── ddays.json
```

```
|--- usage.json  
|--- trash.json  
|--- icons/  
|--- profiles/  
|--- backups/
```

기술 사양

항목	내용
서버	Node.js (순수 내장 모듈, npm install 불필요)
프론트엔드	Vanilla JS + CSS (프레임워크 없음)
포트	1111 (변경 가능)
바인딩	127.0.0.1 (localhost 전용, 외부 접근 불가)
데이터 저장	로컬 JSON 파일
자동 백업	스마트 보관 (7일/30일/50개 캡)
압축	Gzip (HTML, CSS, JS, JSON)
캐싱	ETag + 인메모리 정적 파일 캐시
오프라인	Service Worker 캐싱
대기 메모리	~15MB
아이콘	4단계 fallback + 메모리 캐시
부팅	3단계 (즉시 렌더 → 지연 초기화 → 백그라운드 체크)

변경 이력

[CHANGELOG.md](#) 파일에서 전체 버전 이력을 확인하세요.