

Caddy에 웹 서비스를 등록하는 방법

| 작성: 2026-09-07

| 갱신: 2026-09-16 — 2026-09-15 GCP VM 이전 사례 추가

| 대상: Windows의 C:\Developments\LivekitDev, Caddy 2.10.2

| 4절·5절의 새 도메인과 포트는 설명용 예제입니다. 12절은 실제로 적용된 설정입니다.

1. 무엇을 등록하는가

Caddy 설정은 사용자가 요청한 도메인과 경로를 실제 파일 또는 앱 프로세스에 연결합니다. DNS는 서버의 주소를 알려 주고, Caddy는 해당 서버에 도착한 요청을 처리합니다.

사용자: <https://docs.dev.tictechtoeai.com/>

- > DNS: 도메인을 서버 공인 IP로 해석
- > 서버의 Caddy :443: HTTPS 처리
- > 정적 파일 또는 로컬 앱으로 전달
- > 응답을 사용자에게 반환

구분	준비해야 할 것	Caddy 설정
HTML, Astro 정적 사이트, Flutter 웹 빌드	빌드된 HTML/CSS/JS 파일	root와 file_server
Next.js 서버, Node API, Python 웹 서버	포트를 열고 실행 중인 앱	reverse_proxy
한 도메인에 웹과 API 함께 제공	정적 파일과 API 프로세스	경로별 handle

Caddy가 앱을 빌드하거나 앱 프로세스를 실행해 주지는 않습니다. 프록시 대상 앱은 별도로 시작하고 유지해야 합니다. 현재 서비스 목록은 Caddy 서비스 현황에서 확인합니다.

2. 이 프로젝트에서 수정할 파일

경로	역할
caddy.exe	Caddy 실행 파일
Caddyfile	전체 설정의 진입점
proxyserver/*.caddy	도메인별 추가 설정
homepage/dist/	기존 홈페이지 정적 빌드 (dev.tictechtoeai.com)
docs-site/dist/	문서 사이트 정적 빌드
C:/Developments/tictechtoeai/dist/	회사 사이트 정적 빌드 (tictechtoeai.com)
C:/Developments/VisionIvy/.web/build/client/	VisionIvy 정적 프런트
legal/	별도 HTML 법무 문서

루트 Caddyfile에 다음 줄이 이미 있습니다.

```
import C:/Developments/LivekitDev/proxyserver/*.caddy
```

새 도메인은 proxyserver/새도메인.caddy에 사이트 블록을 작성합니다. *.caddy에 맞는 파일이 모두 포함되므로, 백업은 이 폴더 밖에 저장하거나 .caddy.bak처럼 다른 확장자를 사용합니다. old.caddy라는 백업도 활성화 설정으로 읽힙니다. 같은 도메인을 여러 파일에 중복 등록하지 않습니다. Caddy import 문서

기존 도메인에 경로만 추가하려면 그 도메인을 정의한 기존 블록을 수정합니다. 새 파일에 같은 도메인의 블록을 하나 더 만들지 않습니다.

3. 도메인과 네트워크 준비

예를 들어 문서 사이트를 docs.dev.tictechtoeai.com으로 제공한다고 가정합니다.

1. DNS 관리 화면에서 이 이름의 A 레코드를 서버의 현재 공인 IPv4로 지정합니다. 관리 중인 DNS zone에 따라 입력할 이름은 docs 또는 docs.dev처럼 달라집니다.
2. AAAA 레코드가 있다면 IPv6도 실제로 이 서버에 연결되어야 합니다. 잘못된 AAAA는 일부 접속이나 인증서 발급을 실패하게 할 수 있습니다.
3. 공유기 뒤에 있다면 TCP 80과 443을 Caddy 머신으로 전달합니다. Windows 방화벽과 클라우드 보안 그룹도 확인합니다.
4. Caddy 서비스 계정이 인증서 저장소를 지속적으로 읽고 쓸 수 있어야 합니다.

도메인 이름을 설정하면 Caddy는 조건을 충족한 공개 도메인의 인증서를 자동 발급·갱신하고 HTTP를 HTTPS로 전환합니다. 기본 HTTP-01 검증은 80, TLS-ALPN 검증은 443을 사용합니다. DNS 검증은 별도의 제공자 모듈과 설정이 필요한 방식입니다. 자동 HTTPS 문서

```
Resolve-DnsName docs.dev.tictechtoeai.com -Type A
Resolve-DnsName docs.dev.tictechtoeai.com -Type AAAA
Get-Service LivekitDevCaddy
Get-NetTCPConnection -State Listen -LocalPort 80,443
```

기존 서버에서는 80/443 규칙을 재사용할 수 있습니다. 프록시 대상 앱을 127.0.0.1에 바인딩하면 Caddy가 내부로 접속하므로 앱 포트까지 외부에 열 필요는 없습니다. WebRTC 미디어처럼 별도 통신이 필요한 경우는 해당 서비스의 포트 설계를 따릅니다.

4. 정적 사이트 등록 예제: Astro 문서

4.1 빌드

```
cd C:\Developments\LivekitDev\docs-site
npm run build
if ($LASTEXITCODE -ne 0) { throw '문서 빌드 실패' }
Test-Path .\dist\index.html
```

이 프로젝트는 빌드 전에 Markdown을 동기화합니다. 빌드 성공과 dist/index.html 존재를 확인한 뒤 다음 단계로 진행합니다.

4.2 도메인 설정 파일 작성

C:\Developments\LivekitDev\proxyserver\docs.dev.tictechtoeai.com.caddy에 다음 내용을 저장합니다.

```
docs.dev.tictechtoeai.com {
  encode gzip
  root * C:/Developments/LivekitDev/docs-site/dist
  file_server
}
```

root는 파일의 기준 폴더이고 file_server가 실제 요청을 처리합니다. 예를 들어 /guides/getting-started/는 빌드된 해당 디렉터리의 index.html로 연결됩니다. 서비스 계정에 이 폴더의 읽기 권한이 있어야 합니다. 정적 파일 제공 문서

도메인만 적은 사이트 블록은 기본적으로 HTTPS를 사용합니다. 기존 프로젝트처럼 301 리다이렉트를 명시하려면 다음 블록도 추가할 수 있습니다. 자동 HTTPS만으로도 HTTP 리다이렉트는 제공됩니다.

```
http://docs.dev.tictechtoeai.com {
```

```
redir https://docs.dev.tictechtoeai.com{uri} permanent
}
```

설정 저장만으로 적용되지는 않습니다. 7절의 검증과 reload를 실행해야 합니다.

4.3 문서를 수정한 뒤

기존 root 경로를 유지한다면 문서 수정 후 다시 빌드하면 됩니다. Caddy 설정 변경이 없으면 파일 갱신만으로 새 콘텐츠가 제공됩니다. 운영 중 빌드하면 잠깐 파일이 비는 구간이 생길 수 있으므로, 중요한 사이트는 별도 배포 폴더에서 완성된 빌드를 준비한 뒤 전환합니다.

Astro의 여러 페이지로 구성된 문서 사이트에 무조건 /index.html로 보내는 SPA fallback을 추가하지 않습니다. 존재하지 않는 문서는 404로 남겨야 링크 오류를 찾기 쉽습니다.

5. 앱 또는 API 프록시 등록 예제

예를 들어 앱이 127.0.0.1:8100에서 실행 중이라고 가정합니다.

```
Invoke-WebRequest http://127.0.0.1:8100/ -UseBasicParsing
```

먼저 앱의 실제 경로가 정상 응답하는지 확인합니다. 그런 다음 proxyserver/app.dev.tictechtoeai.com.caddy를 작성합니다.

```
app.dev.tictechtoeai.com {
  encode gzip
  reverse_proxy 127.0.0.1:8100
}
```

사용자에게는 HTTPS 443으로 보이지만 Caddy와 같은 머신의 앱은 HTTP 8100으로 통신합니다. 기본 프록시는 요청 경로와 메서드를 유지하며, WebSocket 연결도 지원합니다. reverse_proxy 문서

운영 앱은 터미널을 닫아도 유지되도록 Windows 서비스 등으로 관리합니다. Caddy 서비스가 Running이어도 앱이 종료되면 502가 발생합니다. 이 프로젝트의 서비스 구성은 Windows 자동 시작 가이드를 참고합니다.

6. 기존 도메인에 경로 추가

6.1 접두사를 유지하는 handle

백엔드에 실제로 /api/health라는 경로가 있다면 다음처럼 사용합니다. 기존 사이트 블록 안에 넣는 예시입니다.

```
handle /api/* {
  reverse_proxy 127.0.0.1:8100
}
```

외부 /api/health가 내부 /api/health로 전달됩니다.

6.2 접두사를 제거하는 handle_path

백엔드에는 /health만 있고 외부에서 /api/health로 제공하려면 다음처럼 사용합니다.

```
handle_path /api/* {
  reverse_proxy 127.0.0.1:8100
}
```

외부 /api/health가 내부 /health로 전달됩니다. 같은 경로에 위 두 방식을 동시에 넣지 않습니다. /api/*는 /api 자체와는 다르므로 필요한 경우 /api를 /api/로 리다이렉트합니다. handle_path 문서

6.3 실제 사례: CelloDex 개인정보처리방침

현재 API 도메인에는 다음 설정이 있습니다.

```
handle_path /legal/* {
  root * C:/Developments/LivekitDev/legal
  file_server
}
```

따라서 `https://api.dev.tictechtoeai.com/legal/cellotutor/privacy.html`은 `legal/cellotutor/privacy.html` 파일로 연결됩니다. 페이지에 표시되는 앱 이름은 CelloDex이며, 기존 URL의 `cellotutor` 경로는 유지하고 있습니다. HTML 본문 수정만으로 반영되며 Caddy reload는 필요하지 않습니다.

6.4 웹과 API를 함께 제공

다음은 독립적인 새 도메인의 전체 블록 예제입니다.

```
app.dev.tictechtoeai.com {
  encode gzip
  handle /api/* {
    reverse_proxy 127.0.0.1:8100
  }
  handle {
    root * C:/Developments/LivekitDev/homepage/dist
    file_server
  }
}
```

마지막 handle은 나머지 요청을 처리합니다. API 요청을 먼저 분리하면 API 오류가 웹의 `index.html`로 바뀌는 문제를 피하기 쉽습니다. 공통 설정 패턴

하위 경로에 앱 전체를 올리면 앱이 생성하는 자산 URL과 리다이렉트도 그 경로를 알아야 합니다. Caddy의 접두사 제거만으로 HTML 안의 URL까지 수정되지는 않습니다. 문서 사이트는 독립 서브도메인으로 시작하면 경로 설정이 단순합니다.

7. 백업, 검증, 적용

7.1 변경 전 백업

설정을 수정하기 전에 프로젝트 루트에서 실행합니다. 같은 PowerShell 창에서 `$backupDir` 값을 유지합니다.

```
cd C:\Developments\LivekitDev
$stamp = Get-Date -Format 'yyyyMMdd-HH:mm:ss'
$backupDir = Join-Path (Get-Location) "logs\caddy-backup-$stamp"
New-Item -ItemType Directory -Path $backupDir | Out-Null
Copy-Item -LiteralPath .\Caddyfile -Destination $backupDir
Copy-Item -Path .\proxyservers\*.caddy -Destination $backupDir
```

7.2 변경 후 전체 설정 검증

추가한 파일만 검증하지 말고 import가 포함된 루트 설정 전체를 검증합니다.

```
.\Caddy.exe validate --config .\Caddyfile --adapter caddyfile
if ($LASTEXITCODE -ne 0) { throw 'Caddy 검증 실패: 적용 중단' }
```

`adapt`는 설정 변환을, `validate`는 설정 로딩과 모듈 준비까지 검사합니다. 검증 성공이 DNS·인증서 발급·백엔드 API 성공을 보장하지는 않습니다. Caddy 명령행 문서

7.3 실행 중인 Caddy에 적용

```
.%caddy.exe reload --config .%Caddyfile --adapter caddyfile
if ($LASTEXITCODE -ne 0) { throw 'Caddy reload 실패: 오류 확인' }
Get-Service LivekitDevCaddy
```

reload는 실행 중인 Caddy의 관리 API를 통해 설정을 교체합니다. 일반 설정 변경에는 서비스 재설치나 별도 caddy run 실행이 필요하지 않습니다. 이 프로젝트의 기존 서비스 이름은 LivekitDevCaddy입니다.

관리 API에 연결할 수 없다는 오류가 나면 서비스 상태와 실제 관리 주소를 확인합니다. 이 프로젝트의 기본 조회 예시는 다음과 같습니다.

```
Invoke-RestMethod http://127.0.0.1:2019/config/
```

이는 현재 활성 설정을 조회하는 명령입니다. 파일을 수정한 내용과 활성 설정은 reload 전까지 다를 수 있습니다. Caddy 관리 API

7.4 문제 발생 시 되돌리기

기존 파일을 수정했다면 백업본을 해당 원래 경로에 복원합니다. 새 파일을 추가했다면 그 파일도 import 대상에서 제외해야 합니다. 루트 Caddyfile만 복원해도 새 .caddy 파일은 계속 읽힙니다.

이번에 새로 추가한 문서 도메인 파일을 되돌리는 예시:

```
Rename-Item -LiteralPath .%proxyserver%docs.dev.tictechtoeai.com.caddy `
-NewName docs.dev.tictechtoeai.com.caddy.disabled
.%caddy.exe validate --config .%Caddyfile --adapter caddyfile
if ($LASTEXITCODE -ne 0) { throw '복원 설정 검증 실패' }
.%caddy.exe reload --config .%Caddyfile --adapter caddyfile
if ($LASTEXITCODE -ne 0) { throw '복원 설정 적용 실패' }
```

8. 적용 후 확인

1. HTTPS 페이지가 열리고 인증서 이름이 도메인과 일치하는지 확인합니다.
2. HTTP 주소가 HTTPS로 이동하는지 확인합니다.
3. 홈뿐 아니라 하위 페이지, CSS/JS, 다운로드 파일도 확인합니다.
4. API는 실제 헬스 경로와 필요한 인증을 사용해 확인합니다.
5. 다른 네트워크의 장치에서도 접속해 DNS 및 외부 포트 접근을 확인합니다.

```
curl.exe -I https://docs.dev.tictechtoeai.com/
curl.exe -I http://docs.dev.tictechtoeai.com/
curl.exe -I https://docs.dev.tictechtoeai.com/guides/getting-started/
```

DNS 전파 전 Caddy 라우팅만 점검하려면, 인증서가 준비된 환경에서 아래처럼 주소 해석을 강제로 지정할 수 있습니다. 이 검사는 실제 공개 DNS 결과를 검증하지 않습니다.

```
curl.exe --resolve docs.dev.tictechtoeai.com:443:127.0.0.1 `
-I https://docs.dev.tictechtoeai.com/
```

이 머신의 일부 자동화 실행 환경에서는 curl의 Schannel 자격 증명 오류가 발생할 수 있습니다. 이 경우 Chrome에서도 확인해 요청 도구의 오류와 서버의 오류를 구분합니다.

9. 자주 발생하는 문제

증상	확인할 내용
DNS 이름을 찾지 못함	DNS zone, A/AAAA 레코드, 전파 상태
연결 시간 초과	공인 IP, 포트 전달, 방화벽, Caddy 실행
인증서 발급 실패	DNS 대상, 80/443 접근, Caddy 로그, 저장소 권한
502 Bad Gateway	프록시 대상 앱 종료, 잘못된 포트, HTTP/HTTPS 혼동
404	root 경로, 빌드 파일, 요청 경로, handle_path 접두사
HTML은 뜨지만 화면이 깨짐	CSS/JS URL, 앱 base 경로, 빠진 빌드 파일
변경이 보이지 않음	설정은 reload, 소스는 빌드, 브라우저 캐시 확인
포트 사용 중 오류	기존 서비스와 수동 실행한 Caddy 중복 여부
새 도메인 설정 충돌	루트 및 import 파일의 중복 사이트 블록

기존 Caddy 설치 스크립트의 로그 경로 예시:

```
Get-Content .\logs\caddy-service-stderr.log -Tail 50
Get-Content .\logs\api-proxy.log -Tail 50
```

서비스 설치 방식에 따라 로그 파일명은 달라질 수 있습니다. API 접근 로그는 해당 API 도메인 요청에 대한 기록이며 전체 도메인의 로그가 아닙니다.

10. 변경 종류별 필요한 작업

변경	빌드	Caddy reload	앱 재시작
legal/의 HTML 글자 수정	불필요	불필요	불필요
Astro 또는 Flutter 소스 수정	필요	root가 같으면 불필요	정적 사이트는 불필요
새 도메인 또는 경로 추가	대상에 따라 필요	필요	대상 앱 준비 필요
reverse_proxy 포트 변경	보통 불필요	필요	새 포트에서 앱 실행 필요
서버 앱 코드·환경 변수 변경	앱에 따라 필요	대상 주소가 같으면 불필요	앱의 배포 방식에 따라 필요

11. 완료 체크리스트

- [] 새 도메인이 올바른 서버 주소를 가리킴
- [] 정적 빌드 또는 백엔드가 준비됨
- [] 설정 파일이 import 대상에 들어감
- [] 기존 도메인 및 경로와 중복되지 않음
- [] 전체 설정 validate 성공 후 reload 성공
- [] HTTPS, 하위 페이지, 자산, API 응답 확인
- [] 외부 장치에서도 접속 확인
- [] Caddy 서비스 현황 문서 갱신

문서 사이트를 실제 공개할 경우 docs-site/public/downloads/의 다운로드 파일도 공개 범위에 포함됩니다. 게시할 콘텐츠 범위를 확인한 뒤 도메인을 연결합니다.

12. 실제 사례: 2026-09-15 GCP VM에서 이전

GCP VM(sparklabsacademy, 34.47.127.124)에서 돌리던 회사 사이트와 VisionIvy를 이 머신의 LivekitDevCaddy로 옮겼습니다. 현재 연결은 Caddy 서비스를 기준으로 합니다.

12.1 운영에는 Astro 개발 서버를 두지 않는다

VM의 tictechtoeai.com은 astro dev로 서비스되고 있었습니다. 개발 서버는 모든 응답에 Vite 클라이언트를 실어 404 하나가 약 130 KB였고, /@fs/로 프로젝트 소스가 인터넷에서 읽혔습니다. 이 머신에서는 정적 빌드만 제공합니다.

```
proxyserver/tictechtoeai.com.caddy
root * C:/Developments/tictechtoeai/dist
file_server
handle_errors { 404 -> /404.html }
```

소스 위치는 LivekitDev 밖입니다. WSL Ubuntu /home/administrator/sites/bearnie에서 npx astro build한 뒤 C:/Developments/tictechtoeai/dist로 rsync 합니다. root가 같으면 재배포 후 Caddy reload는 필요 없습니다.

dev.tictechtoeai.com 홈페이지(homepage/dist)와 apex tictechtoeai.com은 다른 빌드입니다. 한 폴더로 합치지 않습니다.

12.2 같은 문서는 한 주소에서만 낸다

Astro site는 하나뿐입니다. canonical과 sitemap이 https://tictechtoeai.com으로 박혀 있으므로 www와 sparklabsacademy.dev / .com은 내용을 복제하지 않고 apex로 301 이동합니다.

```
www.tictechtoeai.com {
  redir https://tictechtoeai.com{uri} permanent
}

sparklabsacademy.dev, www.sparklabsacademy.dev, sparklabsacademy.com, www.sparklabsacademy.com {
  redir https://tictechtoeai.com{uri} permanent
}
```

www DNS는 A가 아니라 apex를 가리키는 CNAME입니다. 이 머신 공인 IP는 DHCP(임대 2시간)라, IP가 바뀌면 apex A만 고치면 따릅니다. www.sparklabsacademy.com은 이전에는 자기 자신을 가리키는 CNAME 루프였고 이전 당일 고쳤습니다.

12.3 정적 파일과 백엔드를 한 도메인에서 나눈다

VisionIvy는 Reflex입니다. 프론트는 reflex export 정적 파일이고, 상태 동기화는 127.0.0.1:6000입니다. VM처럼 프론트 개발 서버(3000)를 프록시하지 않습니다. 이 머신의 3000은 Meet입니다.

```
handle /_event/* { reverse_proxy 127.0.0.1:6000 }
handle /ping { reverse_proxy 127.0.0.1:6000 }
handle /_upload/* { reverse_proxy 127.0.0.1:6000 }
handle {
  root * C:/Developments/VisionIvy/.web/build/client
  try_files {path} {path}.html {path}/index.html /404.html
  file_server
}
```

Windows에서 localhost는 ::1로 먼저 풀릴 수 있습니다. 백엔드가 IPv4만 들으면 127.0.0.1로 적습니다.

백엔드는 Caddy가 띄우지 않습니다. 작업 스케줄러 VisionIvy-Ensure가

C:\Developments\VisionIvy\scripts\ensure_visionivy.ps1을 부팅 시와 5분 주기로 실행합니다. 백엔드가 꺼져도 정적 화면은 열리고, /ping과 /_event/*만 502가 됩니다.

아직 GCP VM에 남은 호스트는 mbti, saju, visionivy2입니다. 백엔드가 없고 소스도 VM에 없습니다. 이 머신 Caddy에 넣지 않습니다.