

---

## گزارش کار - راهاندازی Windows Authentication با Backend و Frontend روی IIS

### مقدمه

- هدف امروز این بود که سایت **HR** به گونه‌ای راهاندازی شود که:
- کاربران بدون نیاز به ورود دستی، با نام کاربری ویندوز خود شناسایی شوند. (Windows Authentication).
  - سایت (صفحه اصلی و صفحات کاربران) روی IIS سرو شود.
  - فقط برای API‌ها استفاده شود و درخواست‌های .../api/... از IIS عبور کنند.

این فرآیند ابتدا روی سیستم شخصی با ویندوز معمولی (**Home Edition**) تست شد تا ببینیم قابلیت Windows Authentication و نحوه اتصال Backend و Frontend چطور کار می‌کند.

---

### مراحل انجام شده

- #### ۱ آماده‌سازی سایت در IIS
- یک سایت جدید به نام **HR** در IIS Manager ساخته شد.
  - مسیر فایل‌های IIS Frontend (HTML/JS/CSS) به IIS معرفی شد.
  - پورت IIS به طور موقت روی **8080** تنظیم شد تا تست راحت انجام شود.
  - نکته: ویندوز **Home** Windows Authentication ندارد، بنابراین در این مرحله صرفاً محیط شبیه‌سازی شد و نام کاربر واقعی ویندوز قابل شناسایی نبود.
- 

#### ۲ Authentication سازی

- غیرفعال شد تا همه درخواست‌ها تحت مدیریت IIS قرار گیرند. البته چون **Anonymous Authentication** بعدش متوجه شدیم Windows Authentication نداریم، مجدد فعالش کردیم تا بتونیم بدون اون درخواست بزنیم و تست کنیم.
- در ویندوز **Home**، گزینه Windows Authentication وجود نداشت؛ بنابراین در این محیط نمی‌توانستیم نام واقعی کاربر را از سیستم دریافت کنیم.

- برای شبیه‌سازی، از یک Backend FastAPI استفاده شد تا **Fake Header (x-windows-user)** بتواند نام کاربر را بخواند و فرآیند Authentication را تست کنیم.
- 

### ۱۳) نصب و پیکربندی Reverse Proxy

- برای اینکه IIS درخواست‌های API (/api/...) را به FastAPI هدایت کند، Rule URL Rewrite نصب و فعال شد.
  - ایجاد شد تا:
    - مسیرهایی که با /api/ شروع می‌شوند → به Backend FastAPI روی پورت 8000 هدایت شوند.
    - مسیرهای دیگر (صفحه اصلی و صفحات Frontend) توسط IIS سرو شوند.  - گزینه Enable SSL Offloading فعال شد (در صورت نیاز).
  - این مرحله باعث شد که API و Frontend در یک پورت (8080) برای کاربر نهایی ظاهر شوند، اما همچنان روی 8000 باقی بماند.
- 

### ۱۴) جرای Backend

- روی پورت 8000 Backend FastAPI اجرا شد و آمده پاسخ‌دهی به مسیرهای API بود.
  - با استفاده از FastAPI، کد توanst نام کاربر شبیه‌سازی‌شده را بخواند و Authentication را تست کند.
  - مسیر /api/hr/users و سایر API‌ها با موفقیت از IIS عبور کردند و پاسخ مناسب بازگشت داده شد.
- 

### ۱۵) تست مسیرها

- IIS (8080):
  - صفحه اصلی Frontend بالا آمد
  - درخواست JSON هدایت شد و اطلاعات Backend /api/hr/users کاربران دریافت شد
- مرورگر مستقیم روی Backend (8000):
  - مرورگر روی:

- مسیر API پاسخ داد
- مسیر / یا صفحه اصلی 404 داد، چون FastAPI دیگر Frontend را سرو نمی‌کند و IIS مسیر Frontend را مدیریت می‌کند.

نکته مهم: این رفتار طبیعی است و در محیط واقعی Production نیز همین ساختار پیاده می‌شود: کاربران هیچوقت مستقیم به وصل نمی‌شوند، همه چیز از IIS عبور می‌کند.

---

#### نکات مهم و محدودیت‌ها

##### 1. ویندوز Home

- این نسخه از ویندوز Windows Authentication ندارد.
- به همین دلیل تست واقعی احراز هویت با نام کاربری ویندوز امکان‌پذیر نبود.
- برای شبیه‌سازی، از Fake Header استفاده شد.

##### 2. پورت‌ها و مسیرها

- آرروی IIS روی API Frontend 8080 را سرو می‌کند.
- آرروی Backend روی 8000 فقط API را پاسخ می‌دهد و Frontend روی آن سرو نمی‌شود.
- همه کاربران واقعی باید از IIS عبور کنند تا Frontend Authentication و سرو Frontend درست کار کند.

##### 3. URL Rewrite محدود به /api/ است

- این کار باعث شد که Frontend (صفحه اصلی و صفحات کاربران) همچنان توسط IIS سرو شود.
- API‌ها بدون تغییر به Backend هدایت شوند.
- اگر بخواهیم همه مسیرها مستقیم به mount Frontend برond، FastAPI Backend باید فایل‌های IIS (Authentication)، کش و سرعت (از بین می‌رود).

##### 4. شبیه‌سازی DEV

- اجازه داد مسیر API و دریافت نام کاربر شبیه‌سازی شود.
- آرروی IIS روی Frontend 8080 پورت تست شدند.

---

## جمع‌بندی

- با هم هماهنگ شدن، به طوری که:
  - کاربران با ورود به (8080) IIS همه صفحات و API‌ها را می‌بینند
  - فقط برای API کاربرد دارد و Frontend روی آن سرو نمی‌شود
- در محیط واقعی روی IIS فعال خواهد شد، اما روی ویندوز Home Windows Authentication ندارد و تنها شبیه‌سازی شد.
- معماری نهایی به گونه‌ای است که:
  - IIS → Frontend + Reverse Proxy /api/... → Backend
  - کاربران هیچ وقت مستقیم به Backend وصل نمی‌شوند.