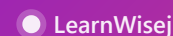


Esercizio di coding con l'IA

Costruisci il progetto EnterpriseOps di questo modulo con ChatGPT o Claude partendo da un prompt già pronto che indirizza il modello alla specifica del modulo, poi rivedi, esegui ed estendi quello che ti restituisce — confrontandolo anche con la nostra build di riferimento su GitHub.



Lo stesso modulo, costruito come costruirai davvero le cose.

Tutte le altre sezioni di questo modulo spiegano come funziona "Architettura di sicurezza: identità, SSO, autorizzazione, audit e rilascio sicuro". Questa ti chiede di produrlo: dirigendo un assistente con la specifica del modulo stesso, poi verificando che il risultato giri e faccia quello che descrive la traccia del lab. È il flusso di lavoro che userai sul lavoro vero, provato su qualcosa di cui conosci già la risposta giusta.

Che cosa stai chiedendo

Aggiungi una simulazione di autenticazione enterprise, la mappatura dei claims sui permessi, l'imposizione dei ruoli per tenant, l'autorizzazione a livello di servizio, la revisione dell'HTML sicuro, l'approvazione delle esportazioni e il log di audit per le azioni sensibili.

Il modulo chiede questi deliverable, e il prompt qui sotto li chiede all'assistente esattamente così:

- ✓ Progetto della mappatura delle identità
- ✓ Matrice dei permessi
- ✓ Implementazione del servizio dei permessi
- ✓ Schermata del log di audit
- ✓ Checklist di hardening della sicurezza

Perché è fatto così

Un'architettura lato server riduce molte superfici di attacco del browser, ma i sistemi avanzati hanno comunque bisogno di integrazione con l'identità, autorizzazione a livello di servizio, audit trail, rendering sicuro e una configurazione operativa sicura.

Questi sono i termini che l'assistente deve azzeccare. Se quello che ti restituisce li usa con leggerezza, non ha capito il modulo:

Termine	Che cosa significa qui
confine OIDC/SSO	Il punto in cui un identity provider esterno consegna all'applicazione un'identità verificata. Tutto ciò che viene dopo quel confine è responsabilità dell'applicazione.
mappatura dei claims	Tradurre i claims specifici del provider nei ruoli e nei permessi propri dell'applicazione, così il codice di autorizzazione non dipende mai dal formato di un singolo identity provider.
servizio dei permessi	Un unico servizio che risponde se l'utente corrente può eseguire un'azione, così la regola vive in un solo punto invece che in ogni schermata.
autorizzazione a livello di servizio	Imporre i permessi dentro i servizi applicativi, non solo nella UI, così un pulsante nascosto non è mai l'unica cosa a proteggere un'operazione.
log di audit	Una registrazione a prova di manomissione di chi ha fatto che cosa e quando per le azioni importanti, così gli eventi di sicurezza si possono indagare e la conformità si può dimostrare.
revisione di AllowHtml	Verificare ogni punto in cui un controllo ammette HTML grezzo per assicurarsi che il contenuto fornito dall'utente sia sanificato, perché l'HTML non revisionato è un rischio di cross-site scripting.
gestione dei segreti	Tenere stringhe di connessione, chiavi e token in una configurazione sicura o in un vault, mai nel controllo del codice sorgente o in impostazioni visibili dal client.
header di sicurezza	Header di risposta HTTP come la content security policy e la strict transport security, che riducono la superficie di attacco del browser per l'applicazione.

Costruiscilo — scegli il tuo assistente

Entrambi i pulsanti aprono una nuova scheda con lo **stesso prompt già pronto**. Il prompt indirizza il modello alla specifica di questo modulo — un unico file Markdown con la lezione, la traccia del lab e i deliverable — così lavora sul materiale vero invece di tirare a indovinare che cosa sia il Modulo 10. Puoi aprire tu stesso [module-10.md](#) per vedere esattamente che cosa leggerà.



Usa ChatGPT

Apri [chatgpt.com](#) con il prompt di questo modulo già caricato. Funziona con GPT-5 o Codex; qualsiasi piano che possa navigare il web recupererà la specifica.



Usa Claude

Apri [claude.ai](#) con lo stesso prompt già caricato. Chi usa Claude Code può invece incollarlo in un terminale dentro una cartella vuota.



Mandalo così com'è e ottieni un progetto, non un frammento

Il prompt si chiude chiedendo l'intera soluzione in un unico `.zip` con i passi per compilarla ed eseguirla. Scaricalo, apri la soluzione in Visual Studio, premi F5 e EnterpriseOps si apre su `localhost`. Ti servono l'SDK .NET e il pacchetto NuGet `Wisej-4`; tutto il resto il prompt chiede all'assistente di scriverlo da sé.

Il prompt, per intero — leggilo, poi modificalo se vuoi un progetto diverso

Sto seguendo il Modulo 10 – "Architettura di sicurezza: identità, SSO, autorizzazione, audit e rilascio sicuro" – del corso LearnWisej "Wisej.NET enterprise: dall'architettura al cloud", e voglio costruire con te il progetto del modulo. Rispondimi in italiano.

Comincia leggendo la specifica del modulo. È un unico file Markdown (in inglese) che contiene l'intera lezione, la traccia del lab e i deliverable:
<https://learnwisej.com/assets/courses/azure-docker/specs/module-10.md>

Poi costruisci un progetto Wisej.NET 4 completo che io possa aprire in Visual Studio:

1. Progetto "EnterpriseOps", C#, net8.0 o successivo, pacchetto NuGet Wisej-4 4.1.x, host Kestrel che chiama `app.UseWisej()`.
2. Costruisci quello che chiede il lab della specifica: Aggiungi una simulazione di autenticazione enterprise, la mappatura dei claims sui permessi, l'imposizione dei ruoli per tenant, l'autorizzazione a livello di servizio, la revisione dell'HTML sicuro, l'approvazione delle esportazioni e il log di audit per le azioni sensibili.
3. Produci questi deliverable e dimmi dove si trova ciascuno nella soluzione:
 - Progetto della mappatura delle identità
 - Matrice dei permessi
 - Implementazione del servizio dei permessi
 - Schermata del log di audit
 - Checklist di hardening della sicurezza
4. Segui la specifica ovunque nomini una classe, una proprietà o un evento di Wisej.NET – usa esattamente quello. Dove lascia una scelta aperta, dimmi quale API hai scelto e perché.
5. Tieni l'autorità sul server: stato e validazione vivono in .NET, il browser disegna e riferisce. Non spostare mai una decisione in JavaScript solo perché lì è più comodo.
6. Rendi il risultato dimostrabile: devo poterlo eseguire e vedere il comportamento di cui parla questo modulo, compreso che cosa succede quando qualcosa va storto.

Prima di scrivere codice, riformula con parole tue che cosa costruirai e dove starà ogni pezzo, così posso correggere il progetto finché costa poco.

Quando il codice è pronto, impacchetta l'intera soluzione in un unico file `.zip` scaricabile e dammi i passi esatti per compilarla ed eseguirla.

Infine rispondi a queste domande della specifica, usando come prova la tua implementazione:

- Dove viene imposta l'autorizzazione?
- Quali elementi della UI visualizzano HTML?
- Un pulsante abilitato può aggirare le regole del servizio?

Chiedimi qualsiasi cosa non ti sia chiara prima di cominciare.

Poi rivedi quello che ti è tornato indietro

Codice che compila non è ancora codice giusto. Le domande interessanti sono domande di progetto: è questo il controllo Wisej.NET che il compito richiede davvero, segue il ciclo di vita che il framework ti offre, lo stato autorevole sta sul server o è stato parcheggiato nel browser perché era comodo? Leggi il risultato da revisore, non da cliente.

Questo modulo ti dà la checklist. Rispondi a ciascuna domanda guardando il codice che hai appena ricevuto:

- ✓ Dove viene imposta l'autorizzazione?
- ✓ Quali elementi della UI visualizzano HTML?
- ✓ Un pulsante abilitato può aggirare le regole del servizio?

Tutto ciò che hai dovuto correggere vale la pena di scriverlo. Una breve nota con le convenzioni e le decisioni a cui sei arrivato, incollata nel primo messaggio della sessione successiva, vale più di qualsiasi prompt ingegnoso.



Questo l'abbiamo già costruito — confronta il tuo con il nostro

L'implementazione di riferimento del Modulo 10 si trova nel repository pubblico degli esempi. Costruisci prima il tuo, poi apri il nostro: la parte interessante sono le differenze.



[Apri l'esempio del Modulo 10 su GitHub](#)

Oppure clona tutto con `git clone https://github.com/iceteagroup/LearnWisej-Samples` e apri `Enterprise Wisej.NET Course/Module 10`.



Il nostro esempio è un punto di partenza, non un prodotto finito

È stato costruito nello stesso modo in cui stai costruendo il tuo — con un prompt, poi corretto — e si ferma deliberatamente dove si ferma il modulo. Aspettati soluzioni provvisorie dove andrebbe codice di produzione, gestione degli errori essenziale e nessun test. Prendilo come un invito: usa quello che hai imparato in questo modulo, più l'assistente, per portarlo più avanti. Migliorare codice che non hai scritto tu è l'abilità che questo corso insegna davvero.

Tre prompt per andare oltre

Punta il tuo assistente sul progetto che hai appena costruito — o sul nostro e prova questi. Ognuno è un esercizio di revisione ed estensione, non una riscrittura.

- ✓ **Dimostra che funziona davvero.** *"Passa in rassegna questo progetto deliverable per deliverable rispetto alla specifica del modulo su <https://learnwisej.com/assets/courses/azure-docker/specs/module-10.md> e mostrami la prova che ciascuno è davvero fatto — il percorso nel codice, su che cosa cliccare e che cosa dovrei vedere. Dove un deliverable è soddisfatto solo in parte, dillo chiaramente invece di arrampicarti sugli specchi. Poi aggiungi quello che manca."*
- ✓ **Rivedilo come faresti con una pull request.** *"Accompagnami attraverso questo codice EnterpriseOps come se stessi rivedendo la pull request di un collega e non l'avessi scritto tu. Dove vive lo stato, che cosa succede dall'inizio alla fine su una singola azione dell'utente, che cosa rimanderesti indietro per modifiche, e quali parti non ti sentiresti tranquillo a mettere davanti a un cliente? Sii specifico e cita le righe di cui parli — non riassumermi il progetto."*
- ✓ **Sostituisci un pezzo senza spostare le giunzioni.** *"Scegli la parte di questo progetto EnterpriseOps che più evidentemente sta al posto di qualcosa di reale — una sorgente dati finta, un archivio in memoria, un servizio semplificato — e sostituiscila con un'implementazione di forma produttiva. Il contratto ai due lati non deve cambiare: stessi tipi, stesse chiamate, stesso comportamento in caso di errore. Dimmi che cosa l'implementazione reale ti ha costretto a gestire e che la versione provvisoria non gestiva."*



Portalo dentro il lab

Il lab pratico che chiude questo modulo chiede gli stessi deliverable che hai appena fatto produrre all'assistente. Usa quello che hai qui come forma della tua risposta e come prova a sostegno — non come la risposta in sé. Il lab viene valutato sul tuo ragionamento, e quella parte deve essere tua.