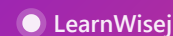


Exercice de code avec l'IA

Construisez l'exemple AccessOps de ce module avec ChatGPT ou Claude à partir d'un prompt prêt à l'emploi qui oriente le modèle vers la spécification du module, puis relisez, exécutez et étendez ce qu'il vous renvoie.



Vous ne taperez pas cet exemple. Vous le piloterez.

Presque plus personne n'écrit du code Wisej.NET à partir d'un fichier vide — un assistant rédige le premier jet et le développeur décide s'il est correct. Cela ne fonctionne que si vous maîtrisez le sujet, et c'est ce que le reste de ce module vous a apporté. Ici, vous confiez ces connaissances à ChatGPT ou à Claude, vous récupérez l'exemple du module sous forme de solution Visual Studio fonctionnelle, et vous le relisez comme vous reliriez la pull request d'un collègue.

Ce que vous demandez

Créez la solution AccessOps et rédigez son modèle de menaces : tracez les frontières de confiance, listez les actifs (dossiers des employés, documents, approbations), recensez les points d'entrée (connexion, uploads, téléchargements, endpoints personnalisés, l'assistant IA), évaluez chaque menace et associez chaque mesure d'atténuation au module de ce cours qui la met en œuvre. Construisez le shell avec une page de connexion provisoire.

Le module demande ces livrables, et le prompt ci-dessous demande exactement ceux-là à l'assistant :

- ✓ Solution AccessOps avec une page de connexion provisoire
- ✓ Diagramme des frontières de confiance
- ✓ Inventaire des actifs et des points d'entrée
- ✓ Tableau des menaces avec probabilité, impact et atténuation
- ✓ Carte de placement des contrôles de sécurité

Pourquoi il est conçu ainsi

AccessOps est un portail d'exploitation interne: départements, employés, documents confidentiels, demandes d'approbation et console d'administration. Tous ceux qui peuvent y accéder travaillent déjà pour l'entreprise, et c'est précisément pour cela que le travail de sécurité est repoussé — et qu'un seul portail finit par contenir les grilles de salaires, les courriers disciplinaires et la trace de chaque dérogation jamais accordée. Avant d'écrire la moindre ligne de code d'authentification, il vous faut trois réponses: ce que détient le navigateur, où passent les frontières, et lesquels des incidents possibles méritent d'être traités en premier.

Voici les termes que l'assistant doit maîtriser. Si ce qu'il vous renvoie les emploie de façon approximative, il n'a pas compris le module :

Terme	Ce qu'il signifie ici
modèle d'UI côté serveur	Les contrôles sont des objets C# dans le processus serveur; le navigateur détient un miroir rendu et un canal d'événements, si bien que votre code et vos requêtes ne sont jamais téléchargés.
état sérialisé des widgets	Les valeurs de propriétés que Wisej.NET sérialise pour que le navigateur puisse dessiner un contrôle. Tout ce que vous affectez circule sur le réseau, y compris sur les contrôles que l'utilisateur ne voit pas.
frontière de confiance	Une ligne où les données passent entre des parties de niveaux de confiance différents. Tout ce qui la franchit est révérifié par le côté le plus fiable, car l'autre côté peut mentir.
frontière de session	La limite entre un navigateur et son propre état serveur dans Application.Session. Les décisions reposent sur l'état situé côté serveur de cette limite, jamais sur une valeur venue du navigateur.
point d'entrée	Toute voie d'accès qu'un attaquant peut atteindre: la page de connexion, les gestionnaires d'événements, les uploads, les téléchargements, les handlers HTTP personnalisés, les appels d'interop et le prompt de l'assistant IA.
STRIDE	Six catégories pour nommer les menaces — usurpation d'identité, falsification, répudiation, divulgation d'informations, déni de service, élévation de privilèges — afin qu'une revue soit systématique.
placement des contrôles	Choisir la frontière à laquelle une règle peut être appliquée. Une règle qui n'existe que dans l'UI est une suggestion; la même règle dans un service est un contrôle.

Construisez-le — choisissez votre assistant

Les deux boutons ouvrent un nouvel onglet avec le **même prompt prêt à l'emploi** déjà rempli. Le prompt oriente le modèle vers la spécification de ce module — un unique fichier Markdown qui contient la leçon, l'énoncé du lab et les livrables — pour qu'il travaille sur le vrai contenu au lieu de deviner de quoi parle le Module 1. Vous pouvez ouvrir vous-même [module-01.md](#) pour voir exactement ce qu'il lira.



Utiliser ChatGPT

Ouvre [chatgpt.com](#) avec le prompt de ce module déjà chargé. Fonctionne avec GPT-5 ou Codex ; toute offre capable de naviguer sur le web récupérera la spécification.



Utiliser Claude

Ouvre [claude.ai](#) avec le même prompt déjà chargé. Si vous utilisez Claude Code, vous pouvez plutôt le coller dans un terminal ouvert sur un dossier vide.



Envoyez-le tel quel et vous obtenez un projet, pas un extrait

Le prompt se termine en demandant toute la solution sous forme d'un unique `.zip`, avec les étapes pour la compiler et l'exécuter. Téléchargez-le, ouvrez la solution dans Visual Studio, appuyez sur F5 et AccessOps s'ouvre sur `localhost`. Il vous faut le SDK .NET et le package NuGet `Wisej-4`; tout le reste, le prompt demande à l'assistant de l'écrire lui-même.

Je suis le Module 1 – « Le modèle de sécurité de Wisej.NET » – du cours LearnWisej « Modèles de sécurité et d'authentification », et je veux construire avec toi l'exemple de ce module. Réponds-moi en français.

Commence par lire la spécification du module. C'est un unique fichier Markdown (en anglais) qui contient la leçon complète, l'énoncé du lab et les livrables :

<https://learnwisej.com/assets/courses/security-and-auth-patterns/specs/module-01.md>

Construis ensuite un projet Wisej.NET 4 complet que je puisse ouvrir dans Visual Studio :

1. Projet « AccessOps », C#, net8.0 ou ultérieur, package NuGet Wisej-4 4.1.x, hôte Kestrel qui appelle `app.UseWisej()`.
2. Construis ce que demande le lab de la spécification : Créez la solution AccessOps et rédigez son modèle de menaces : tracez les frontières de confiance, listez les actifs (dossiers des employés, documents, approbations), recensez les points d'entrée (connexion, uploads, téléchargements, endpoints personnalisés, l'assistant IA), évaluez chaque menace et associez chaque mesure d'atténuation au module de ce cours qui la met en œuvre. Construisez le shell avec une page de connexion provisoire.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Solution AccessOps avec une page de connexion provisoire
 - Diagramme des frontières de confiance
 - Inventaire des actifs et des points d'entrée
 - Tableau des menaces avec probabilité, impact et atténuation
 - Carte de placement des contrôles de sécurité
4. Suis la spécification partout où elle nomme une classe, une propriété ou un événement Wisej.NET – utilise exactement celui-là. Là où elle laisse le choix, dis-moi quelle API tu as choisie et pourquoi.
5. Garde l'autorité côté serveur : l'état et la validation vivent dans .NET, le navigateur affiche et remonte les informations. Ne déplace jamais une décision dans JavaScript parce que c'est plus simple là-bas.
6. Rends le résultat démontrable : je dois pouvoir l'exécuter et voir le comportement dont traite ce module, y compris ce qui se passe quand quelque chose tourne mal.

Avant d'écrire la moindre ligne de code, reformule avec tes propres mots ce que tu vas construire et où vivra chaque élément, pour que je puisse corriger la conception tant que cela ne coûte pas cher.

Quand le code est terminé, rassemble toute la solution dans un unique fichier .zip téléchargeable et donne-moi les étapes exactes pour la compiler et l'exécuter.

Enfin, réponds à ces questions de la spécification en t'appuyant sur ta propre implémentation :

- Votre grille des employés est liée aux dossiers complets des employés et la colonne du salaire est masquée dans le designer. Décrivez ce que détient le navigateur, et nommez la modification qui retire réellement le salaire de ce qui circule sur le réseau.
- Deux personnes sont connectées à AccessOps sur le même serveur. Expliquez quelles parties de l'état elles partagent par défaut, et ce qui devrait être vrai pour que l'une d'elles voie le département courant de l'autre.
- Choisissez la menace de votre tableau dont l'impact est le plus élevé et défendez la thèse inverse: que faudrait-il qu'il soit vrai d'AccessOps pour que cette atténuation soit une perte de temps pour l'équipe?

Pose-moi des questions sur tout ce qui est ambigu avant de commencer.

Puis relisez ce qui vous revient

Un code qui compile n'est pas encore un code juste. Les questions intéressantes sont des questions de conception : est-ce bien le contrôle Wisej.NET que la tâche exige, suit-il le cycle de vie que le framework vous offre, l'état de référence est-il sur le serveur ou a-t-il été laissé dans le navigateur parce que c'était pratique ? Lisez le résultat en relecteur, pas en client.

Ce module vous donne la checklist. Répondez à chaque point en vous appuyant sur le code que vous venez de recevoir :

- ✓ Votre grille des employés est liée aux dossiers complets des employés et la colonne du salaire est masquée dans le designer. Décrivez ce que détient le navigateur, et nommez la modification qui retire réellement le salaire de ce qui circule sur le réseau.
- ✓ Deux personnes sont connectées à AccessOps sur le même serveur. Expliquez quelles parties de l'état elles partagent par défaut, et ce qui devrait être vrai pour que l'une d'elles voie le département courant de l'autre.
- ✓ Choisissez la menace de votre tableau dont l'impact est le plus élevé et défendez la thèse inverse: que faudrait-il qu'il soit vrai d'AccessOps pour que cette atténuation soit une perte de temps pour l'équipe?

Tout ce que vous avez dû corriger mérite d'être noté. Une courte note des conventions et des décisions retenues, collée dans le premier message de votre prochaine session, vaut plus que n'importe quel prompt astucieux.



Pas encore d'exemple de référence pour ce cours — le vôtre est le premier

Le [dépôt des exemples LearnWisej](#) contient des solutions complètes pour la plupart des cours, mais pas encore pour celui-ci. Il n'y a donc rien à quoi vous comparer et aucun raccourci : la spécification et votre propre relecture sont votre seul filet de sécurité. Gardez ce que vous construisez — et si le résultat est réussi, nous aimerions le voir.

Trois prompts pour aller plus loin

Orientez votre assistant vers le projet que vous venez de construire et essayez ceux-ci. Chacun est un exercice de relecture et d'extension, pas une réécriture.

- ✓ **Prouvez que ça fonctionne vraiment.** *"Passe en revue ce projet livrable par livrable par rapport à la spécification du module sur <https://learnwisej.com/assets/courses/security-and-auth-patterns/specs/module-01.md> et montre-moi la preuve que chacun est réellement terminé — le chemin dans le code, où cliquer et ce que je dois voir. Là où un livrable n'est que partiellement satisfait, dis-le clairement au lieu de le défendre. Ensuite, ajoute ce qui manque."*
- ✓ **Relisez-le comme une pull request.** *"Fais-moi parcourir ce code AccessOps comme si tu relisais la pull request d'un collègue et que tu ne l'avais pas écrit. Où vit l'état, que se passe-t-il de bout en bout lors d'une action de l'utilisateur, que renverrais-tu pour correction, et quelles parties ne serais-*

tu pas à l'aise de montrer à un client ? Sois précis et cite les lignes dont tu parles — ne me résume pas la conception."

- ✓ **Remplacez une pièce sans déplacer les jointures.** *"Choisis la partie de ce projet AccessOps qui remplace le plus manifestement quelque chose de réel — une source de données factice, un stockage en mémoire, un service simplifié — et remplace-la par une implémentation digne de la production. Le contrat de part et d'autre ne doit pas changer : mêmes types, mêmes appels, même comportement en cas d'échec. Dis-moi ce que l'implémentation réelle t'a obligé à gérer et que la version provisoire ne gérait pas."*



Emportez-le dans le lab

Le lab pratique qui clôt ce module demande les mêmes livrables que ceux que vous venez de faire produire à l'assistant. Servez-vous de ce que vous avez ici comme forme de votre réponse et comme preuve à l'appui — pas comme la réponse elle-même. Le lab est noté sur votre raisonnement, et cette partie doit venir de vous.