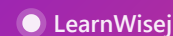


Exercice de code avec l'IA

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.



Le même module, construit comme vous construirez vraiment.

Toutes les autres sections de ce module expliquent comment fonctionne « Protéger les données, les uploads, les téléchargements et les handlers ». Celle-ci vous demande de le produire — en pilotant un assistant avec la spécification du module elle-même, puis en vérifiant que le résultat s'exécute et fait ce que décrit l'énoncé du lab. C'est la méthode de travail que vous utiliserez sur de vrais projets, mise en pratique sur un sujet dont vous connaissez déjà la bonne réponse.

Ce que vous demandez

Sécurisez la fonctionnalité de documents d'AccessOps : des uploads limités en taille et en type avec inspection du contenu, stockés hors de la racine web sous un nom aléatoire et rattachés au département propriétaire, des téléchargements qui passent par l'autorisation de DocumentService et par Application.Download, un handler personnalisé pour un lien de partage signé qui expire au bout d'une heure, et une revue vérifiant que chaque requête est paramétrée et que chaque chaîne fournie par l'utilisateur est encodée avant affichage.

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

- ✓ Chaîne d'upload validée avec dossier de quarantaine
- ✓ Téléchargement autorisé via le service
- ✓ Handler de lien de partage signé et à expiration
- ✓ Revue des requêtes paramétrées et de l'encodage des sorties
- ✓ Test avec une requête de téléchargement falsifiée

Pourquoi il est conçu ainsi

L'espace des documents confidentiels est la partie d'AccessOps qu'un attaquant convoite réellement : lettres de salaire, dossiers disciplinaires, contrats signés, dossiers du conseil d'administration. Tous les autres écrans déplacent des lignes entre des grilles; celui-ci fait franchir la frontière à des octets, vers un navigateur que vous ne contrôlez pas, et dans l'autre sens. L'identité et les permissions ont été réglées dans les modules précédents; celui-ci traite des endroits où un fichier, une URL ou une chaîne les contourne.

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
téléchargement diffusé par le serveur	Appeler <code>Application.Download</code> depuis le code serveur envoie un fichier au navigateur pendant un événement que vous avez déjà autorisé: il n'a donc aucune adresse à mettre en favori, à partager ou à deviner.
exposition des fichiers statiques	Tout ce qui se trouve sous la racine web est servi avant qu'une seule ligne de votre code ne s'exécute: les permissions du fichier se résument donc exactement à « quiconque connaît l'URL ».
événement <code>Uploaded</code>	Le contrôle Upload le déclenche sur le serveur avec les flux des fichiers reçus — le seul endroit où un fichier peut être inspecté avant d'entrer dans votre système.
vérification de la signature du contenu	Comparer les premiers octets d'un fichier avec l'en-tête connu du format, car l'extension et le type de contenu déclaré sont fournis par le client.
stockage en quarantaine	Conserver l'upload accepté hors de la racine web, sous un nom généré, jusqu'à ce que l'analyse soit saine, et seulement ensuite le rattacher au département propriétaire.
handler HTTP personnalisé	Une classe serveur enregistrée auprès de l'application qui reçoit des requêtes brutes en dehors du flux d'événements: elle n'a aucune identité de session et doit tout authentifier et tout valider elle-même.
lien signé à expiration	Une URL portant un id de document, une date d'expiration et un HMAC calculé sur les deux avec une clé détenue par le serveur, de sorte que les valeurs ne peuvent pas être modifiées et que le lien cesse de fonctionner de lui-même.

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 5. Vous pouvez ouvrir vous-même [module-05.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 5 – « Protéger les données, les uploads, les téléchargements et les handlers » – 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-05.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 : Sécurisez la fonctionnalité de documents d'AccessOps : des uploads limités en taille et en type avec inspection du contenu, stockés hors de la racine web sous un nom aléatoire et rattachés au département propriétaire, des téléchargements qui passent par l'autorisation de `DocumentService` et par `Application.Download`, un handler personnalisé pour un lien de partage signé qui expire au bout d'une heure, et une revue vérifiant que chaque requête est paramétrée et que chaque chaîne fournie par l'utilisateur est encodée avant affichage.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Chaîne d'upload validée avec dossier de quarantaine
 - Téléchargement autorisé via le service
 - Handler de lien de partage signé et à expiration
 - Revue des requêtes paramétrées et de l'encodage des sorties
 - Test avec une requête de téléchargement falsifiée
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 gestionnaire d'upload vérifie l'extension par rapport à une liste d'autorisation, et le fichier passe. Expliquez ce qu'un attaquant peut encore envoyer, et ce que la vérification de signature apporte que la vérification d'extension ne peut pas apporter.
- Un manager transmet un lien de partage à une personne extérieure à l'entreprise avant son expiration. Quelles parties de votre conception limitent les dégâts, et que faudrait-il ajouter pour révoquer ce seul lien sans changer la clé pour tout le monde?
- Votre bouton de téléchargement est masqué pour les utilisateurs sans permission de lecture, et le test de requête falsifiée passe toujours. Expliquez précisément quelle ligne de code le fait passer, et pourquoi le bouton masqué n'y est pour rien.

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 gestionnaire d'upload vérifie l'extension par rapport à une liste d'autorisation, et le fichier passe. Expliquez ce qu'un attaquant peut encore envoyer, et ce que la vérification de signature apporte que la vérification d'extension ne peut pas apporter.
- ✓ Un manager transmet un lien de partage à une personne extérieure à l'entreprise avant son expiration. Quelles parties de votre conception limitent les dégâts, et que faudrait-il ajouter pour révoquer ce seul lien sans changer la clé pour tout le monde?
- ✓ Votre bouton de téléchargement est masqué pour les utilisateurs sans permission de lecture, et le test de requête falsifiée passe toujours. Expliquez précisément quelle ligne de code le fait passer, et pourquoi le bouton masqué n'y est pour rien.

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.

- ✓ **Ajoutez le deuxième cas.** *"Étends ce projet AccessOps avec un deuxième cas qui suit le même modèle que le premier — une autre entité, un autre écran ou un autre contrôle de même forme, câblé de la même manière. Extrais dans une classe de base ou un service ce qui est réellement partagé, mais seulement là où cela se justifie ; deux choses similaires n'appellent pas toujours une abstraction. Montre-moi le diff et justifie chaque élément que tu as extrait."*
- ✓ **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-05.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."*



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.