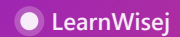


Exercice de code avec l'IA

Construisez l'exemple FieldOps 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

Terminez le projet de synthèse FieldOps : signez l'installateur Windows et les packages Android et iOS, préparez une distribution en entreprise pour les shells mobiles et un brouillon de fiche store, définissez une règle de compatibilité entre les versions du serveur et celles des shells avec une vérification de version minimale au démarrage, ajoutez le signalement des plantages et un événement d'usage pour chaque fonctionnalité de l'appareil, et livrez une checklist de publication couvrant les quatre modèles de distribution.

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

- ✓ Packages Windows, Android et iOS signés
- ✓ Configuration de la distribution en entreprise et brouillon de fiche store
- ✓ Vérification de compatibilité des versions du serveur et des shells
- ✓ Télémétrie des plantages et de l'usage
- ✓ Checklist de publication pour tous les modèles de distribution

Pourquoi il est conçu ainsi

À la fin de ce cours, FieldOps est un serveur et quatre façons de l'atteindre: un onglet de navigateur, une PWA installée, un shell desktop Windows et un shell mobile sur Android et iOS. Les trois versions packagées sont des fichiers qu'un système d'exploitation refuse d'installer tant qu'une identité en laquelle il a confiance ne les a pas signés, et chacune emprunte un canal différent avec sa propre file de revue et sa propre vitesse de déploiement. Vous pouvez redéployer le serveur avant le déjeuner; le build iOS qu'a en main un technicien peut avoir trois semaines de retard, et il doit quand même fonctionner.

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
signature Authenticode	La signature qu'un certificat de signature de code applique à un installateur Windows ou à un MSIX, prouvant que le fichier n'a pas été modifié après la signature et nommant un éditeur vérifié.
horodatage par contre-signature	Un horodatage de confiance enregistré au moment de la signature, pour que l'installateur continue d'être validé après l'expiration du certificat de signature lui-même.
keystore Android et clé d'import	La clé qui signe un build Android; avec Play App Signing, vous signez avec une clé d'import et Google re-signe avec la clé de signature de l'application qu'il détient pour ce package.
certificat de distribution et profil de provisionnement Apple	Le certificat identifie votre équipe; le profil relie un App ID, ce certificat, les entitlements et, en dehors de l'App Store, les appareils autorisés.
distribution en entreprise	Livrer un build privé aux appareils d'une organisation — Google Play géré, applications personnalisées Apple Business Manager, un MSIX poussé par Intune — au lieu de le publier publiquement.
version minimale du shell	Le plus ancien build de shell, par plateforme, que le serveur actuel prend encore en charge; tout ce qui est en dessous est arrêté à la connexion avant d'atteindre la moindre donnée.
événement d'usage d'une fonctionnalité de l'appareil	Un enregistrement écrit chaque fois qu'une capacité du shell s'exécute — caméra, signature, localisation, push, vidage de la file hors ligne — portant le résultat ainsi que la plateforme et la version du shell.

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 6. Vous pouvez ouvrir vous-même [module-06.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 FieldOps 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 6 – « Signature, distribution, mises à jour et projet de synthèse FieldOps » – du cours LearnWisej « Packaging hybride et natif », 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/hybrid-native-packaging/specs/module-06.md>

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

1. Projet « FieldOps », 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 : Terminez le projet de synthèse FieldOps : signez l'installateur Windows et les packages Android et iOS, préparez une distribution en entreprise pour les shells mobiles et un brouillon de fiche store, définissez une règle de compatibilité entre les versions du serveur et celles des shells avec une vérification de version minimale au démarrage, ajoutez le signalement des plantages et un événement d'usage pour chaque fonctionnalité de l'appareil, et livrez une checklist de publication couvrant les quatre modèles de distribution.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Packages Windows, Android et iOS signés
 - Configuration de la distribution en entreprise et brouillon de fiche store
 - Vérification de compatibilité des versions du serveur et des shells
 - Télémétrie des plantages et de l'usage
 - Checklist de publication pour tous les modèles de distribution
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 :

- Le shell Android d'un technicien a six semaines et le serveur a été livré quatre fois depuis. Laquelle de ces quatre modifications du serveur aurait dû relever la version minimale du shell, et comment l'auriez-vous su à ce moment-là?
- L'installateur Windows a été signé mais pas horodaté, et le certificat a expiré le mois dernier. Que font les installations existantes, que font les nouveaux téléchargements, et quel est le chemin le plus court pour retrouver un installateur de confiance?
- L'événement d'usage de la caméra montre une forte hausse du chemin de repli navigateur, uniquement sur iOS. Nommez trois explications compatibles avec les données, et indiquez quel champ supplémentaire de l'événement permettrait de les départager.

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 :

- ✓ Le shell Android d'un technicien a six semaines et le serveur a été livré quatre fois depuis. Laquelle de ces quatre modifications du serveur aurait dû relever la version minimale du shell, et comment l'auriez-vous su à ce moment-là?
- ✓ L'installateur Windows a été signé mais pas horodaté, et le certificat a expiré le mois dernier. Que font les installations existantes, que font les nouveaux téléchargements, et quel est le chemin le plus court pour retrouver un installateur de confiance?
- ✓ L'événement d'usage de la caméra montre une forte hausse du chemin de repli navigateur, uniquement sur iOS. Nommez trois explications compatibles avec les données, et indiquez quel champ supplémentaire de l'événement permettrait de les départager.

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.

- ✓ **Remplacez une pièce sans déplacer les jointures.** *"Choisis la partie de ce projet FieldOps 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."*
- ✓ **Faites-le résister aux mauvais jours.** *"Examine ce projet FieldOps pour repérer les modes de défaillance qu'il ne gère pas encore — une saisie erronée ou manquante, une dépendance lente ou*

indisponible, un traitement qui aboutit après que l'utilisateur a changé de page, et deux actions qui se chevauchent. Pour chacun, dis-moi d'abord s'il s'agit d'un risque réel dans Wisej.NET 4 et pourquoi, et seulement ensuite écris la correction. Là où tu estimes qu'aucune gestion n'est nécessaire, dis-le et explique ton raisonnement au lieu d'ajouter du code défensif."

- ✓ **Ajoutez le deuxième cas.** *"Étends ce projet FieldOps 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."*



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.