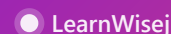


Exercice de code avec l'IA

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.



Le même module, construit comme vous construirez vraiment.

Toutes les autres sections de ce module expliquent comment fonctionne « API de l'appareil via le bridge hybride ». 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

Ajoutez des fonctionnalités de l'appareil à FieldOps : une action Take Photo qui utilise la caméra de l'appareil dans le shell natif et se replie sur un contrôle Upload dans le navigateur, une capture de signature envoyée sous forme d'image, une action Navigate qui lit la position de l'appareil et ouvre l'application de cartographie, une notification push lorsqu'un ordre de travail est attribué, et une vérification de capacités qui affiche ou masque chaque action selon le shell.

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

- ✓ Capture photo avec repli sur l'envoi de fichier dans le navigateur
- ✓ Capture et envoi de la signature
- ✓ Géolocalisation et transfert vers l'application de cartographie
- ✓ Notification push à l'attribution
- ✓ Matrice de vérification des capacités par shell

Pourquoi il est conçu ainsi

Un technicien FieldOps photographie une vanne corrodée, recueille une signature sur l'écran et part vers l'adresse suivante — pendant que le répartiteur qui a attribué l'intervention suit le même ordre de travail dans un navigateur desktop. Chaque contrôle de FieldOps est un objet C# sur votre serveur, qui ne peut toucher ni la caméra, ni la puce GPS, ni la zone de notifications. Ce module est l'aller-retour qui comble cet écart, et ce que fait l'écran là où cet aller-retour n'existe pas.

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
bridge vers l'appareil	Le canal de messages qu'un shell hybride expose à la page qu'il héberge: votre serveur demande au shell d'effectuer une opération sur l'appareil, et le shell répond par un message que l'application retransforme en événement côté serveur.
objet hôte	L'objet qu'une WebView injecte dans la page hébergée pour que le JavaScript de cette page puisse appeler du code natif. Le C# côté serveur ne l'atteint qu'à travers l'interopérabilité JavaScript de l'application.
descripteur de capacités	Les fonctionnalités qu'un shell annonce au démarrage d'une session — caméra, localisation, notifications, biométrie — mises en cache et consultées avant d'afficher toute action sur l'appareil.
amélioration progressive	Concevoir chaque écran pour que le chemin du simple navigateur soit une fonctionnalité complète et que le chemin natif en soit une meilleure version, jamais le seul qui fonctionne.
autorisation au moment opportun	Demander à la plateforme l'accès à la caméra, à la localisation ou aux notifications au moment où l'utilisateur appuie sur le bouton qui en a besoin, juste après une phrase qui explique pourquoi.
événement de résultat du bridge	La réponse asynchrone — octets de la photo, coordonnées ou refus — qui arrive sur une requête ultérieure à celle du clic qui l'a demandée, et qui a donc besoin de son propre push.
stockage hors de la racine web	Écrire les photos et signatures capturées là où aucune URL ne mène, puis les servir via un gestionnaire qui vérifie si cette session a le droit de voir cet ordre de travail.

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 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 5 - « API de l'appareil via le bridge hybride » - 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-05.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 : Ajoutez des fonctionnalités de l'appareil à FieldOps : une action Take Photo qui utilise la caméra de l'appareil dans le shell natif et se replie sur un contrôle Upload dans le navigateur, une capture de signature envoyée sous forme d'image, une action Navigate qui lit la position de l'appareil et ouvre l'application de cartographie, une notification push lorsqu'un ordre de travail est attribué, et une vérification de capacités qui affiche ou masque chaque action selon le shell.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Capture photo avec repli sur l'envoi de fichier dans le navigateur
 - Capture et envoi de la signature
 - Géolocalisation et transfert vers l'application de cartographie
 - Notification push à l'attribution
 - Matrice de vérification des capacités par shell
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 résultat de la caméra arrive quarante secondes après le clic qui l'a demandé. Expliquez sur quelle requête il arrive, quel état de session il doit trouver pour savoir quoi faire des octets, et pourquoi l'image n'apparaît pas dans le navigateur sans push explicite.
- Un technicien a refusé définitivement l'autorisation de localisation il y a trois semaines sur son propre téléphone. Décrivez précisément ce que doit faire aujourd'hui un appui sur Navigate, et nommez la seule chose que l'application ne doit jamais faire en réponse.
- Take Photo fonctionne dans le shell mobile et le repli Upload fonctionne dans Chrome sur desktop. Nommez une chose réellement différente pour l'utilisateur entre ces deux chemins, et argumentez sur le caractère acceptable ou non de cette différence pour un technicien de

terrain.

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 résultat de la caméra arrive quarante secondes après le clic qui l'a demandé. Expliquez sur quelle requête il arrive, quel état de session il doit trouver pour savoir quoi faire des octets, et pourquoi l'image n'apparaît pas dans le navigateur sans push explicite.
- ✓ Un technicien a refusé définitivement l'autorisation de localisation il y a trois semaines sur son propre téléphone. Décrivez précisément ce que doit faire aujourd'hui un appui sur Navigate, et nommez la seule chose que l'application ne doit jamais faire en réponse.
- ✓ Take Photo fonctionne dans le shell mobile et le repli Upload fonctionne dans Chrome sur desktop. Nommez une chose réellement différente pour l'utilisateur entre ces deux chemins, et argumentez sur le caractère acceptable ou non de cette différence pour un technicien de terrain.

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.

- ✓ **Relisez-le comme une pull request.** *"Fais-moi parcourir ce code FieldOps 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 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."*



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.