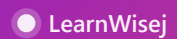


Exercice de code avec l'IA

Construisez l'exemple AdaptiveOps 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 — y compris en le comparant à notre propre solution de référence sur GitHub.



L'assistant, ce sont les mains rapides. L'architecte, c'est toujours vous.

Un assistant peut produire l'exemple de ce module en quelques minutes. Ce qu'il ne peut pas faire, c'est décider si le résultat est la bonne conception pour votre application — cette décision vous revient. C'est vous qui apportez le jugement : briefez-le correctement, puis vérifiez ce qu'il vous renvoie à la lumière de ce que le Module 7 vient de vous apprendre.

Ce que vous demandez

Terminez le projet de synthèse Adaptive Operations Console : un thème personnalisé avec des tokens nommés et au moins quatre appearances sémantiques, un usage ciblé de CssClass, un shell de bureau avec des régions Flow, Table et Flex, un ClientProfiles.json avec un comportement téléphone, tablette, petit bureau et bureau, des propriétés responsives ainsi qu'un gestionnaire ResponsiveProfileChanged idempotent, et un indicateur de profil visible en mode test. Menez la revue d'accessibilité des cadres de focus, des états invalides, du contraste, des zones tactiles et de l'ordre de tabulation, remplissez la matrice de QA responsive avec des captures pour chaque profil, et rédigez la note d'architecture et le grounding pack qui expliquent chaque décision de thème, de mise en page et de profil.

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

- ✓ Adaptive Operations Console terminée avec thème personnalisé, appearances sémantiques, CSS ciblé et régions Flow, Table et Flex
- ✓ ClientProfiles.json, propriétés responsives et gestionnaire ResponsiveProfileChanged idempotent avec un indicateur de profil en mode test
- ✓ Revue d'accessibilité couvrant les cadres de focus, les états invalides, le contraste, les zones tactiles et l'ordre de tabulation
- ✓ Matrice de QA responsive avec des captures bureau, tablette et téléphone dans les deux orientations
- ✓ Note d'architecture et grounding pack pour l'IA expliquant chaque décision de thème, de mise en page et de profil

Pourquoi il est conçu ainsi

Six modules de travail vivent désormais dans une seule solution: un thème dérivé d'un thème de base, du CSS ciblé, un shell de conteneurs imbriqués, des régions Flow, Table et Flex, un ClientProfiles.json et un gestionnaire de profil. Ce qui transforme cela en produit n'est pas une fonctionnalité de plus. C'est la preuve que la console est utilisable sur chaque profil, qu'un utilisateur au clavier ou au toucher peut s'en servir, qu'un redimensionnement ne déclenche pas de reconstructions coûteuses, et que le développeur suivant, ou la prochaine mise à jour de Wisej.NET, ne peut pas casser sans bruit le système visuel. Ce module apporte cette preuve, ainsi que le grounding pack qui permet à un assistant IA de répondre aux questions sur les décisions que vous avez prises.

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
ensemble d'écrans du projet de synthèse	Adaptive Operations Console terminée: thème personnalisé, appearances sémantiques, CSS ciblé, un shell bureau, un shell tablette adaptatif, un shell téléphone, une grille, un éditeur, un retour de validation, une navigation et un indicateur de profil visible en mode test.
revue d'accessibilité visuelle	Un passage sur les cadres de focus, les états désactivés et invalides, le contraste, l'espacement tactile, les infobulles des boutons à icône seule et l'ordre de tabulation au clavier, réalisé en utilisant l'écran plutôt qu'en lisant le thème.
gestionnaire de profil idempotent	Un gestionnaire ResponsiveProfileChanged qui attribue des valeurs finales pour le profil courant, si bien que l'exécuter deux fois, ou après un rafraîchissement, produit exactement la même interface.
stabilité de la mise en page	MinimumSize et MaximumSize empêchent les régions de s'effondrer, et AutoSize reste désactivé sur les grandes hiérarchies de conteneurs, pour des redimensionnements prévisibles et peu coûteux.
gouvernance du thème	Le JSON du thème, la feuille de style ciblée et ClientProfiles.json sont versionnés, relus sous forme de diffs, documentés en tokens et jamais modifiés à l'intérieur d'un thème intégré.
personnalisation résistante aux mises à jour	Les appearances documentées, les flux du Theme Builder et les propriétés côté serveur survivent aux mises à jour; les sélecteurs CSS qui dépendent du DOM interne d'un widget peut-être pas, ils sont donc ciblés, documentés ou évités.
matrice de QA responsive et grounding pack	La matrice consigne les largeurs, orientations, profil attendu et vérifications de chaque scénario; le grounding pack consigne l'arbre de décision, les règles de réponse et la carte des sources à partir desquels un assistant répond.

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 7. Vous pouvez ouvrir vous-même [module-07.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 AdaptiveOps 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 7 – « Projet de synthèse prêt pour le mobile, accessibilité, performances et gouvernance en production » – du cours LearnWisej « Thèmes et mises en page responsives », 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/theming-and-responsive-layouts/specs/module-07.md>

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

1. Projet « AdaptiveOps », 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 Adaptive Operations Console : un thème personnalisé avec des tokens nommés et au moins quatre appareances sémantiques, un usage ciblé de `CssClass`, un shell de bureau avec des régions Flow, Table et Flex, un `ClientProfiles.json` avec un comportement téléphone, tablette, petit bureau et bureau, des propriétés responsives ainsi qu'un gestionnaire `ResponsiveProfileChanged` idempotent, et un indicateur de profil visible en mode test. Menez la revue d'accessibilité des cadres de focus, des états invalides, du contraste, des zones tactiles et de l'ordre de tabulation, remplissez la matrice de QA responsive avec des captures pour chaque profil, et rédigez la note d'architecture et le grounding pack qui expliquent chaque décision de thème, de mise en page et de profil.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Adaptive Operations Console terminée avec thème personnalisé, appareances sémantiques, CSS ciblé et régions Flow, Table et Flex
 - `ClientProfiles.json`, propriétés responsives et gestionnaire `ResponsiveProfileChanged` idempotent avec un indicateur de profil en mode test
 - Revue d'accessibilité couvrant les cadres de focus, les états invalides, le contraste, les zones tactiles et l'ordre de tabulation
 - Matrice de QA responsive avec des captures bureau, tablette et téléphone dans les deux orientations
 - Note d'architecture et grounding pack pour l'IA expliquant chaque décision de thème, de mise en page et de profil
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 ResponsiveProfileChanged s'exécute deux fois pour le profil Tablet après un rafraîchissement du navigateur. Quelle ligne de votre code garantit que la seconde exécution ne change rien, et que verrait l'utilisateur si cette ligne manquait?
- Un relecteur vous demande de retirer le cadre de focus d' action-button parce qu'il « fait daté ». Que modifiez-vous à la place, quel token le pilote, et quels deux autres états revérifiez-vous ensuite dans le Theme Builder?
- L'en-tête de la grille des tickets a besoin d'un ajustement que seul un sélecteur sur le DOM interne du widget peut atteindre. Qu'essayez-vous en premier, qu'est-ce qui doit être vrai avant que le sélecteur soit autorisé, et que consignez-vous pour que la prochaine mise à jour ne surprenne pas 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 gestionnaire ResponsiveProfileChanged s'exécute deux fois pour le profil Tablet après un rafraîchissement du navigateur. Quelle ligne de votre code garantit que la seconde exécution ne change rien, et que verrait l'utilisateur si cette ligne manquait?
- ✓ Un relecteur vous demande de retirer le cadre de focus d' action-button parce qu'il « fait daté ». Que modifiez-vous à la place, quel token le pilote, et quels deux autres états revérifiez-vous ensuite dans le Theme Builder?
- ✓ L'en-tête de la grille des tickets a besoin d'un ajustement que seul un sélecteur sur le DOM interne du widget peut atteindre. Qu'essayez-vous en premier, qu'est-ce qui doit être vrai avant que le sélecteur soit autorisé, et que consignez-vous pour que la prochaine mise à jour ne surprenne pas 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.



Nous l'avons déjà construit — comparez votre version à la nôtre

L'implémentation de référence du Module 7 se trouve dans le dépôt public des exemples. Construisez d'abord la vôtre, puis ouvrez la nôtre : ce sont les différences qui sont intéressantes.



[Ouvrir l'exemple du Module 7 sur GitHub](#)

Ou clonez l'ensemble avec `git clone https://github.com/iceteagroup/LearnWisej-Samples` et ouvrez `Theming and Responsive Layouts Course/Module 7`.



Notre exemple est un point de départ, pas un produit fini

Il a été construit de la même manière que vous construisez le vôtre — par prompt, puis corrigé — et il s'arrête volontairement là où s'arrête le module. Attendez-vous à des éléments provisoires là où irait du code de production, à une gestion des erreurs minimale et à l'absence de tests. Voyez-y une invitation : utilisez ce que vous avez appris dans ce module, avec l'assistant, pour aller plus loin. Améliorer du code que vous n'avez pas écrit, c'est précisément la compétence que ce cours enseigne.

Trois prompts pour aller plus loin

Orientez votre assistant vers le projet que vous venez de construire — ou vers le nôtre 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/theming-and-responsive-layouts/specs/module-07.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 AdaptiveOps 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 AdaptiveOps 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.