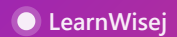


Exercice de code avec l'IA

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.



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

Ajoutez à Adaptive Operations Console un ClientProfiles.json avec les profils Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop et Desktop, ordonnés du plus étroit au plus large. Dans le designer, utilisez les propriétés responsives pour masquer le rail de navigation et le panneau de détails sur téléphone, passer les boutons de la barre d'outils en Display icône seule et ré-ancrer la région de détails sous la grille sur tablette. Gérez ResponsiveProfileChanged pour afficher le nom du profil actif dans le label d'état, journaliser chaque changement et ouvrir l'éditeur de détails sous forme de formulaire modal sur téléphone au lieu d'un panneau latéral.

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

- ✓ ClientProfiles.json avec les profils Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop et Desktop dans l'ordre de première correspondance
- ✓ Valeurs de propriétés responsives attribuées dans le designer pour masquer le rail de navigation et le panneau de détails sur téléphone
- ✓ Display icône seule de la barre d'outils et région de détails ré-ancrée pour la tablette, définis via les propriétés responsives
- ✓ Gestionnaire ResponsiveProfileChanged qui affiche et journalise le nom du profil actif
- ✓ Éditeur de détails ouvert sous forme de formulaire modal sur téléphone via le gestionnaire de profil

Pourquoi il est conçu ainsi

Adaptive Operations Console dispose maintenant d'un shell thématé construit avec des conteneurs Flow, Table et Flex qui résiste au redimensionnement sans gestionnaire Resize, mais il montre le même écran à

tout le monde: sur un téléphone de 390 pixels, le rail de navigation, la grille des tickets et l'éditeur de détails se disputent la même largeur. Les media queries CSS ne peuvent pas régler ce problème, car les contrôles vivent sur le serveur: masquer un panneau, passer un bouton en icône seule ou ré-ancrer une région est un changement de propriété côté serveur, et un utilisateur de téléphone qui perd le panneau de détails a besoin d'une autre façon de modifier un ticket, pas d'une copie réduite de la version bureau. Ce module ajoute le mécanisme que Wisej.NET fournit pour cela: les profils client, les propriétés responsives et l'événement ResponsiveProfileChanged.

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
ClientProfile	Un ensemble nommé de règles de correspondance liées à l'appareil (largeur du navigateur, largeur d'écran, type d'appareil, user-agent, paysage) qui décrit la façon dont une catégorie de clients est utilisée, pas un nom commercial d'appareil.
ClientProfiles.json	Le fichier que lit Wisej.NET; les profils sont évalués de haut en bas et le premier qui correspond l'emporte, l'ordre des entrées fait donc partie de la configuration.
Application.ActiveProfile	Le profil actuellement sélectionné pour cette session; son Name est ce qu'affiche le label d'état et ce sur quoi se base le gestionnaire de profil.
propriété responsive	Une propriété.NET ordinaire marquée par ResponsivePropertyAttribute qui contient une valeur par profil client et applique la valeur correspondante chaque fois que le profil actif change.
adaptation par le designer d'abord	Les valeurs responsives ne s'attribuent que dans le designer de Visual Studio, ce qui les garde sérialisées, visibles et déterministes; le code sert aux adaptations qui dépendent des données ou de l'état d'exécution.
ResponsiveProfileChanged	L'événement déclenché sur Application, sur les conteneurs comme Form, Page, Desktop et UserControl, et sur DataGridView et ListView quand le profil actif change; ResponsiveProfileChangedEventArgs.CurrentProfile transporte le nouveau profil.
matrice de QA responsive	Le plan de test qui liste la largeur, la hauteur, l'orientation, le type d'appareil, le tactile, les barres de défilement, la mise à l'échelle des polices et le nom de profil attendu pour chaque taille.

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 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 6 - « ClientProfile et propriétés responsives » - 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-06.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 : Ajoutez à Adaptive Operations Console un `ClientProfiles.json` avec les profils Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop et Desktop, ordonnés du plus étroit au plus large. Dans le designer, utilisez les propriétés responsives pour masquer le rail de navigation et le panneau de détails sur téléphone, passer les boutons de la barre d'outils en Display icône seule et ré-ancrer la région de détails sous la grille sur tablette. Gérez `ResponsiveProfileChanged` pour afficher le nom du profil actif dans le label d'état, journaliser chaque changement et ouvrir l'éditeur de détails sous forme de formulaire modal sur téléphone au lieu d'un panneau latéral.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - `ClientProfiles.json` avec les profils Phone, Phone Landscape, Tablet, Tablet Landscape, Small Desktop et Desktop dans l'ordre de première correspondance
 - Valeurs de propriétés responsives attribuées dans le designer pour masquer le rail de navigation et le panneau de détails sur téléphone
 - Display icône seule de la barre d'outils et région de détails ré-ancrée pour la tablette, définis via les propriétés responsives
 - Gestionnaire `ResponsiveProfileChanged` qui affiche et journalise le nom du profil actif
 - Éditeur de détails ouvert sous forme de formulaire modal sur téléphone via le gestionnaire 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 :

- Si Small Desktop était déplacé en tête de `ClientProfiles.json`, lesquels des cinq autres profils pourraient encore être retenus, et que verrait une tablette en paysage?
- Le ré-ancrage tablette vit dans le designer et l'éditeur modal téléphone dans le code.

Qu'est-ce qui décide quel mécanisme porte chaque changement, et qu'est-ce qui tournerait mal si vous les échangiez?

– Un testeur a fait pivoter un téléphone quatre fois et votre gestionnaire s'est exécuté quatre fois. Quel état doit être identique après la quatrième exécution et après la première, et comment votre code le garantit-il?

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 :

- ✓ Si Small Desktop était déplacé en tête de ClientProfiles.json, lesquels des cinq autres profils pourraient encore être retenus, et que verrait une tablette en paysage?
- ✓ Le ré-ancrage tablette vit dans le designer et l'éditeur modal téléphone dans le code. Qu'est-ce qui décide quel mécanisme porte chaque changement, et qu'est-ce qui tournerait mal si vous les échangiez?
- ✓ Un testeur a fait pivoter un téléphone quatre fois et votre gestionnaire s'est exécuté quatre fois. Quel état doit être identique après la quatrième exécution et après la première, et comment votre code le garantit-il?

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 6 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 6 sur GitHub](#)

Ou clonez l'ensemble avec `git clone https://github.com/iceteagroup/LearnWisej-Samples` et ouvrez `Theming and Responsive Layouts Course/Module 6`.



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.

- ✓ **Ajoutez le deuxième cas.** *"Étends ce projet AdaptiveOps 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/theming-and-responsive-layouts/specs/module-06.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."*



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.