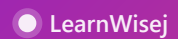


Exercice de code avec l'IA

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.



Le même module, construit comme vous construirez vraiment.

Toutes les autres sections de ce module expliquent comment fonctionne « FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel et propriétés de mise en page étendues ». 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

Construisez trois versions de la zone de contenu d'Adaptive Operations Console : une zone de filtres et de cartes de métriques en FlowLayoutPanel qui passe à la ligne et utilise FillWeight et FlowBreak, un éditeur de ticket en TableLayoutPanel dont les labels et les éditeurs sont alignés grâce à RowStyles et ColumnStyles, et un shell de tableau de bord en FlexLayoutPanel où la liste et les détails se partagent l'espace par FillWeight, avec AlignX, AlignY, MinimumSize et MaximumSize définis. Construisez l'une des trois en code avec les extensions fluent de Wisej.Web.Markup et comparez la façon dont chacune résiste au redimensionnement du navigateur.

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

- ✓ Zone de filtres et de cartes de métriques en FlowLayoutPanel utilisant WrapContents, FillWeight et FlowBreak
- ✓ Éditeur de ticket en TableLayoutPanel avec RowStyles, ColumnStyles et fusions pour des champs alignés
- ✓ Shell de tableau de bord en FlexLayoutPanel avec FillWeight, AlignX, AlignY, MinimumSize et MaximumSize
- ✓ Une région construite en code avec les extensions fluent de Wisej.Web.Markup
- ✓ Captures comparatives au redimensionnement et note indiquant quel conteneur convient à chaque région

Pourquoi il est conçu ainsi

Le Module 4 a doté Adaptive Operations Console d'un shell de régions ancrées, mais le docking décide seulement de l'emplacement d'une région, pas du comportement de son contenu quand le navigateur fait

1400 pixels de large le lundi et 700 sur une tablette le mardi. Une rangée de cartes de métriques doit passer à la ligne, un éditeur de ticket doit garder ses labels et ses éditeurs alignés quelle que soit la longueur de la traduction allemande, et la liste et les détails doivent se partager l'espace de travail dans la proportion que vous décidez. Chacun de ces cas est un comportement de contenu différent, et Wisej.NET fournit un conteneur pour chacun: FlowLayoutPanel, TableLayoutPanel et FlexLayoutPanel. Choisir le mauvais, c'est ainsi que les équipes se remettent à écrire du code de redimensionnement.

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
FlowLayoutPanel	Dispose les enfants en lignes ou colonnes automatiques selon FlowDirection, passe à la ligne quand WrapContents est à true et ignore Location, Dock et Anchor de chaque enfant, si bien que pastilles et cartes se réorganisent d'elles-mêmes.
TableLayoutPanel	Place les enfants dans des cellules adressées par Row et Column, avec des fusions pour les cellules qui s'étendent, si bien que labels et éditeurs restent alignés tandis que chaque enfant peut encore utiliser Dock ou Anchor dans sa cellule.
FlexLayoutPanel	Dispose les enfants sur une ligne ou une colonne, remplit toujours sa zone cliente et partage l'espace restant par poids, ce qui convient aux grandes régions qui divisent l'espace de travail proportionnellement.
FillWeight	Une propriété enfant étendue lue par les moteurs flow et flex: un poids supérieur à zéro étire l'enfant dans l'espace restant, proportionnellement aux autres enfants pondérés; zéro conserve sa propre taille.
FlowBreak	Une propriété enfant étendue de FlowLayoutPanel qui force l'enfant suivant sur une nouvelle ligne ou colonne, pour qu'une barre de filtres termine une ligne délibérément au lieu de dépendre de la largeur.
Styles de lignes et de colonnes	RowStyles et ColumnStyles déclarent si une ligne ou une colonne s'ajuste automatiquement, prend une taille fixe en pixels ou partage le reste en pourcentage; c'est là que vivent les proportions d'un formulaire.
Margin et Padding	Padding est l'espace intérieur du conteneur autour de tous les enfants; Margin est la distance d'un enfant à ses voisins, et seuls les moteurs qui respectent les marges l'utilisent.

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



Utiliser Claude

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.

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 5 – « FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel et propriétés de mise en page étendues » – 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-05.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 : Construisez trois versions de la zone de contenu d'Adaptive Operations Console : une zone de filtres et de cartes de métriques en FlowLayoutPanel qui passe à la ligne et utilise FillWeight et FlowBreak, un éditeur de ticket en TableLayoutPanel dont les labels et les éditeurs sont alignés grâce à RowStyles et ColumnStyles, et un shell de tableau de bord en FlexLayoutPanel où la liste et les détails se partagent l'espace par FillWeight, avec AlignX, AlignY, MinimumSize et MaximumSize définis. Construisez l'une des trois en code avec les extensions fluent de Wisej.Web.Markup et comparez la façon dont chacune résiste au redimensionnement du navigateur.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Zone de filtres et de cartes de métriques en FlowLayoutPanel utilisant WrapContents, FillWeight et FlowBreak
 - Éditeur de ticket en TableLayoutPanel avec RowStyles, ColumnStyles et fusions pour des champs alignés
 - Shell de tableau de bord en FlexLayoutPanel avec FillWeight, AlignX, AlignY, MinimumSize et MaximumSize
 - Une région construite en code avec les extensions fluent de Wisej.Web.Markup
 - Captures comparatives au redimensionnement et note indiquant quel conteneur convient à chaque région
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 :

- Vous avez donné un FillWeight de 1 à la région de détails mais aucun MinimumSize, et à 700 pixels elle a disparu. Expliquez ce que le moteur flex a fait de l'espace et pourquoi la correction est une contrainte de taille plutôt qu'un autre poids.

- Le bouton Apply passe à la deuxième ligne à 1024 pixels alors qu'il y a de la place pour lui. Quelle propriété étendue en est la cause, et comment la FilterBar se comporterait-elle si vous la retiriez et mettiez WrapContents à false?
- Une Margin de 8 sur les cartes du FlowLayoutPanel a créé des espacements, mais la même Margin sur le rail de navigation ancré n'a rien fait. Quel moteur respecte les marges dans chaque cas, et quelle propriété crée l'espacement pour le rail?

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 :

- ✓ Vous avez donné un FillWeight de 1 à la région de détails mais aucun MinimumSize, et à 700 pixels elle a disparu. Expliquez ce que le moteur flex a fait de l'espace et pourquoi la correction est une contrainte de taille plutôt qu'un autre poids.
- ✓ Le bouton Apply passe à la deuxième ligne à 1024 pixels alors qu'il y a de la place pour lui. Quelle propriété étendue en est la cause, et comment la FilterBar se comporterait-elle si vous la retiriez et mettiez WrapContents à false?
- ✓ Une Margin de 8 sur les cartes du FlowLayoutPanel a créé des espacements, mais la même Margin sur le rail de navigation ancré n'a rien fait. Quel moteur respecte les marges dans chaque cas, et quelle propriété crée l'espacement pour le rail?

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 5 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 5 sur GitHub](#)

Ou clonez l'ensemble avec `git clone https://github.com/iceteagroup/LearnWisej-Samples` et ouvrez `Theming and Responsive Layouts Course/Module 5`.



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.

- ✓ **Faites-le résister aux mauvais jours.** *"Examine ce projet AdaptiveOps 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 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-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."*



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.