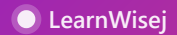


Exercice de code avec l'IA

Construisez l'exemple GlobalDesk 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 « Langues de droite à gauche et mises en page compatibles avec la localisation ». 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

Faites en sorte que GlobalDesk survive à une culture de droite à gauche. Ajoutez l'arabe au sélecteur de langue et mettez `rightToLeft` sur `auto` dans `Default.json`, laissez les contrôles enfants sur `RightToLeft.Inherit`, et activez `RightToLeftLayout` sur le tableau de bord et l'éditeur client pour que leurs enfants soient mis en miroir. Gardez exactement un contrôle de gauche à droite, le champ du code client, et expliquez dans un commentaire pourquoi cet identifiant ne doit pas être inversé. Reconstituez avec du docking, de l'ancrage et un conteneur de mise en page la partie de l'écran placée à des positions fixes, pour qu'une traduction longue ne puisse pas être tronquée, ajoutez un fichier de ressources pseudo-localisé dont les valeurs sont allongées et encadrées de crochets, et notez ce que fait réellement la `DataGridView` des contacts en RTL au lieu de supposer qu'elle se met en miroir. Terminez par une liste de tests de mise en page en cinq cas que vous pourrez rejouer sur n'importe quel écran.

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

- ✓ Entrée arabe dans le sélecteur de langue avec `rightToLeft` sur `auto` dans `Default.json`
- ✓ Tableau de bord et éditeur client utilisant `RightToLeftLayout`, leurs enfants restant sur `Inherit`
- ✓ Un contrôle délibérément laissé de gauche à droite, avec un commentaire expliquant l'exception
- ✓ Mise en page reconstruite avec docking, ancrage et un conteneur de mise en page pour qu'une traduction longue ne soit jamais tronquée
- ✓ Liste de tests RTL couvrant un texte court, un texte long, une fenêtre étroite, une culture RTL et la limitation de la `DataGridView`

Pourquoi il est conçu ainsi

GlobalDesk sait déjà changer de langue en cours d'exécution: le sélecteur définit `Application.CurrentCulture`, le gestionnaire `Application.CultureChanged` réapplique chaque libellé issu des ressources, et le panneau d'aperçu remet ses exemples en forme. Puis un testeur ouvre le tableau de bord

en arabe. Chaque mot est traduit et chacun d'eux est au mauvais endroit: les labels se collent au bord gauche de leurs champs, la colonne de navigation se trouve là où l'œil d'un lecteur arabe arrive en dernier, et les longs libellés allemands du module précédent poussent déjà un bouton hors d'un panneau à largeur fixe.

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
RightToLeft	Une propriété de contrôle aux valeurs Inherit, No et Yes qui demande à un contrôle d'adapter son propre texte et le sens de ses éléments internes.
RightToLeft.Inherit	Le choix normal pour un contrôle enfant: il suit son parent, si bien que tout l'écran change de sens à partir d'une seule décision prise en haut.
RightToLeftLayout	Une autre propriété, qui met en miroir la position horizontale des enfants directs d'un conteneur quand le RTL est actif, de sorte que l'agencement visuel s'inverse pendant que votre mise en page logique reste telle que conçue.
Le paramètre d'application rightToLeft	Une entrée de Default.json; réglée sur "auto", elle fait déduire par Wisej.NET le comportement RTL de la culture de la session, au lieu de tests de nom de culture éparpillés dans vos écrans.
Application.RightToLeft	La même décision exprimée dans le code, pour les sessions qui doivent être basculées délibérément plutôt que par détection de la culture.
Allongement du texte	Un libellé traduit a rarement la longueur de l'anglais, et c'est pourquoi les positions en pixels fixes et les largeurs de panneaux mesurées à la main cassent en premier dans une version localisée.
Limites RTL documentées	Wisej.NET documente que DataGridView, ListView et PropertyGrid ne prennent pas totalement en charge le RTL: un plan de tests doit donc les nommer au lieu de supposer qu'ils se mettent en miroir.

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



Utiliser Claude

Ouvre [claude.ai](#) avec le même prompt déjà chargé. Si vous utilisez Claude Code,

naviguer sur le web récupérera la spécification.

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 GlobalDesk 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 - « Langues de droite à gauche et mises en page compatibles avec la localisation » - du cours LearnWisej « Localisation dans Wisej.NET », 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/localization-in-wisej-net/specs/module-05.md>

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

1. Projet « GlobalDesk », 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 : Faites en sorte que GlobalDesk survive à une culture de droite à gauche. Ajoutez l'arabe au sélecteur de langue et mettez `rightToLeft` sur `auto` dans `Default.json`, laissez les contrôles enfants sur `RightToLeft.Inherit`, et activez `RightToLeftLayout` sur le tableau de bord et l'éditeur client pour que leurs enfants soient mis en miroir. Gardez exactement un contrôle de gauche à droite, le champ du code client, et expliquez dans un commentaire pourquoi cet identifiant ne doit pas être inversé. Reconstituez avec du docking, de l'ancrage et un conteneur de mise en page la partie de l'écran placée à des positions fixes, pour qu'une traduction longue ne puisse pas être tronquée, ajoutez un fichier de ressources pseudo-localisé dont les valeurs sont allongées et encadrées de crochets, et notez ce que fait réellement la `DataGridView` des contacts en RTL au lieu de supposer qu'elle se met en miroir. Terminez par une liste de tests de mise en page en cinq cas que vous pourrez rejouer sur n'importe quel écran.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Entrée arabe dans le sélecteur de langue avec `rightToLeft` sur `auto` dans `Default.json`
 - Tableau de bord et éditeur client utilisant `RightToLeftLayout`, leurs enfants restant sur `Inherit`
 - Un contrôle délibérément laissé de gauche à droite, avec un commentaire expliquant l'exception
 - Mise en page reconstruite avec docking, ancrage et un conteneur de mise en page pour qu'une traduction longue ne soit jamais tronquée
 - Liste de tests RTL couvrant un texte court, un texte long, une fenêtre étroite, une culture RTL et la limitation de la `DataGridView`
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 :

- Un collègue met RightToLeft.Yes sur vingt contrôles et l'écran arabe paraît correct, mais l'écran allemand n'a pas changé. Qu'a-t-il réellement configuré, et quelle propriété unique aurait fait le travail pour tout le conteneur?
- Votre passe pseudo-localisée montre trois libellés encore en anglais courant. Rien n'a été tronqué et rien n'a levé d'exception. Que cela vous apprend-il sur ces trois libellés, et le travail de quel module précédent comporte-t-il une lacune?
- La grille des contacts ne se met pas en miroir comme le reste du tableau de bord. Laquelle de vos options — l'accepter et la documenter, remplacer le contrôle, ou écrire la mise en miroir à la main — pouvez-vous défendre devant un relecteur, et que devriez-vous vérifier avant de promettre l'une d'elles?

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 :

- ✓ Un collègue met RightToLeft.Yes sur vingt contrôles et l'écran arabe paraît correct, mais l'écran allemand n'a pas changé. Qu'a-t-il réellement configuré, et quelle propriété unique aurait fait le travail pour tout le conteneur?
- ✓ Votre passe pseudo-localisée montre trois libellés encore en anglais courant. Rien n'a été tronqué et rien n'a levé d'exception. Que cela vous apprend-il sur ces trois libellés, et le travail de quel module précédent comporte-t-il une lacune?
- ✓ La grille des contacts ne se met pas en miroir comme le reste du tableau de bord. Laquelle de vos options — l'accepter et la documenter, remplacer le contrôle, ou écrire la mise en miroir à la main — pouvez-vous défendre devant un relecteur, et que devriez-vous vérifier avant de promettre l'une d'elles?

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.

- ✓ **Faites-le résister aux mauvais jours.** *"Examine ce projet GlobalDesk 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 GlobalDesk 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/localization-in-wisej-net/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.