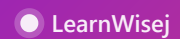


Exercice de code avec l'IA

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.



Briefez-le, lisez-le, puis cassez-le.

Cet exercice comporte trois parties, et seule la première concerne le prompt. Vous briefez un assistant avec la spécification du module et récupérez une solution GlobalDesk ; vous lisez ce qu'il a écrit et décidez s'il fait vraiment ce que le module demandait ; puis vous le poussez plus loin que ce que le module demandait. C'est dans la deuxième et la troisième partie que l'on apprend.

Ce que vous demandez

Enrichissez les ressources partagées de GlobalDesk. Ajoutez Strings.de.resx à côté de Resources/Strings.resx avec exactement les mêmes clés et des valeurs traduites, ajoutez une entrée Customer.LastOrder qui contient deux espaces réservés, et composez la phrase avec string.Format et Application.CurrentCulture au lieu de concaténer des fragments. Faites passer par la classe utilitaire Texts chaque message affiché par l'éditeur client, y compris la confirmation d'enregistrement et les deux messages de validation, et faites en sorte que le service d'enregistrement renvoie une clé de résultat stable comme CustomerEditor.Saved plutôt qu'une phrase anglaise que l'interface affiche telle quelle. Ajoutez ensuite un fichier Resources-de.resx qui redéfinit un libellé système de Wisej, démontrez-le dans une MessageBox, et rédigez une courte note sur la différence entre le texte de votre produit et le texte système de Wisej.

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

- ✓ Strings.de.resx portant les mêmes clés que le fichier neutre, avec uniquement des valeurs traduites
- ✓ Ressource Customer.LastOrder composée avec string.Format et la culture active, jamais par concaténation
- ✓ Service d'enregistrement qui renvoie une clé de ressource ou un code de résultat au lieu d'une phrase anglaise
- ✓ Resources-de.resx qui redéfinit au moins un libellé système de Wisej, montré dans une MessageBox
- ✓ Note de lab distinguant le texte du produit, le texte système de Wisej et le texte de diagnostic

Pourquoi il est conçu ainsi

Le Module 2 a laissé l'éditeur client de GlobalDesk très convaincant en allemand: chaque libellé que possède le designer change avec la culture et la mise en page tient. Puis l'utilisateur appuie sur Save et lit

Customer saved. en anglais, parce que cette phrase n'est la propriété d'aucun contrôle — c'est un littéral dans un gestionnaire de clic, à côté de This field is required. et de la ligne d'état qui compte les contacts chargés.

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
fichier de ressources d'application partagé	Resources/Strings.resx contient le texte visible par l'utilisateur qu'aucun écran conçu ne possède en propre: confirmations d'enregistrement, messages de validation, lignes d'état, titres de rapports et résultats de services.
fichier frère spécifique à une culture	Strings.de.resx et Strings.it.resx se placent à côté du fichier neutre et portent les mêmes clés; un traducteur change les valeurs, jamais les noms de clés.
clé sémantique stable	Un nom comme CustomerEditor.Saved décrit le sens du texte: reformuler la phrase anglaise n'oblige donc jamais à modifier le code ni ne casse les traductions.
Wisej.Resources.ResourceManager	L'objet de recherche qui suit le modèle de ressources.NET: à partir d'un nom de base et d'un assembly, il trouve la valeur pour la culture active et retombe progressivement vers le fichier neutre.
marqueur de clé manquante	Une classe utilitaire qui renvoie [Key] quand une recherche échoue transforme un défaut invisible en quelque chose que l'on voit et que l'on peut rechercher, au lieu d'un label vide.
chaîne de format avec espaces réservés	La phrase entière avec ses {0} et {1} vit dans la ressource et string.Format la remplit, parce que les langues ordonnent le nom, la date et le verbe différemment.
Resources-[LANG].resx	Un fichier de ressources système Wisej.NET distinct, à la racine du projet ou dans un dossier /Resources, qui redéfinit ou ajoute des libellés du framework comme \$Ok, \$Cancel et \$Offline.

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 3. Vous pouvez ouvrir vous-même [module-03.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 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 3 - « Ressources d'application, ResourceManager et textes système de Wisej » - 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-03.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 : Enrichissez les ressources partagées de GlobalDesk. Ajoutez `Strings.de.resx` à côté de `Resources/Strings.resx` avec exactement les mêmes clés et des valeurs traduites, ajoutez une entrée `Customer.LastOrder` qui contient deux espaces réservés, et composez la phrase avec `string.Format` et `Application.CurrentCulture` au lieu de concaténer des fragments. Faites passer par la classe utilitaire `Texts` chaque message affiché par l'éditeur client, y compris la confirmation d'enregistrement et les deux messages de validation, et faites en sorte que le service d'enregistrement renvoie une clé de résultat stable comme `CustomerEditor.Saved` plutôt qu'une phrase anglaise que l'interface affiche telle quelle. Ajoutez ensuite un fichier `Resources-de.resx` qui redéfinit un libellé système de Wisej, démontrez-le dans une `MessageBox`, et rédigez une courte note sur la différence entre le texte de votre produit et le texte système de Wisej.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - `Strings.de.resx` portant les mêmes clés que le fichier neutre, avec uniquement des valeurs traduites
 - Ressource `Customer.LastOrder` composée avec `string.Format` et la culture active, jamais par concaténation
 - Service d'enregistrement qui renvoie une clé de ressource ou un code de résultat au lieu d'une phrase anglaise
 - `Resources-de.resx` qui redéfinit au moins un libellé système de Wisej, montré dans une `MessageBox`
 - Note de lab distinguant le texte du produit, le texte système de Wisej et le texte de diagnostic
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 :

- Supprimez une clé de Strings.de.resx et exécutez en allemand. Décrivez ce que voit l'utilisateur, pourquoi rien ne plante, et comment vous vous en apercevriez dans une vraie application de deux cents clés.
- Votre service d'enregistrement renvoie actuellement CustomerEditor.Saved. Qu'est-ce qui casserait dans les tests unitaires, et dans l'écran allemand, s'il renvoyait la phrase elle-même?
- Un collègue ajoute le mot "Cancel" à Resources-de.resx sous une clé de son invention. Expliquez pourquoi le bouton de la MessageBox ne change pas, et où cette chaîne avait réellement sa place.

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 :

- ✓ Supprimez une clé de Strings.de.resx et exécutez en allemand. Décrivez ce que voit l'utilisateur, pourquoi rien ne plante, et comment vous vous en apercevriez dans une vraie application de deux cents clés.
- ✓ Votre service d'enregistrement renvoie actuellement CustomerEditor.Saved. Qu'est-ce qui casserait dans les tests unitaires, et dans l'écran allemand, s'il renvoyait la phrase elle-même?
- ✓ Un collègue ajoute le mot "Cancel" à Resources-de.resx sous une clé de son invention. Expliquez pourquoi le bouton de la MessageBox ne change pas, et où cette chaîne avait réellement sa place.

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 GlobalDesk 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 GlobalDesk 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 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."*



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.