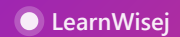


Exercice de code avec l'IA

Construisez l'exemple IconDesk 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.



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 2 vient de vous apprendre.

Ce que vous demandez

Construisez dans IconDesk une page ImageLab qui rend les deux pipelines visibles côte à côte. Affichez un même visuel raster quatre fois avec SizeMode défini sur Normal, Zoom, Cover et StretchImage pour que les différences de rognage et de déformation sautent aux yeux, ajoutez un PictureBox qui récupère un JPEG distant avec LoadAsync et garde la page réactive pendant son arrivée, et placez un SVG affecté via ImageSource à 16, 24, 32 et 48 pixels pour prouver qu'il reste net. Terminez par un gestionnaire qui indique, pour chaque PictureBox de la page, si son image vient d'Image ou d'ImageSource.

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

- ✓ Page ImageLab affichant un même visuel raster dans les modes de dimensionnement Normal, Zoom, Cover et StretchImage
- ✓ PictureBox qui récupère un JPEG distant avec LoadAsync sans bloquer la page
- ✓ Un SVG affecté via ImageSource et rendu à 16, 24, 32 et 48 pixels
- ✓ Gestionnaire de rapport qui nomme, pour chaque PictureBox, si l'image vient d'Image ou d'ImageSource
- ✓ Note de formats listant les ressources du lab qui passent par System.Drawing et celles qui restent des chaînes de ressource

Pourquoi il est conçu ainsi

« Quels formats d'image Wisej.NET prend-il en charge? » est la question que toute équipe pose dès la première semaine, et la réponse honnête dépend du chemin d'API qu'emprunte l'image. Répondez par une seule liste vague et vous obtiendrez un logo étiré sur un portable haute densité, une page qui se fige pendant qu'une photo de produit se télécharge depuis un hôte lent, et un rapport de bug parce qu'un fichier.ico refuse de se résoudre via une chaîne de ressource. Ce module remplace la liste vague par deux listes précises, chacune attachée à la propriété de contrôle qui la consomme.

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
pipeline System.Drawing	La propriété Image contient un System.Drawing.Image dans la mémoire du serveur; Wisej.NET l'envoie au navigateur sous forme de données PNG. Ses sources documentées sont les fichiers bitmap, métafichier, icône, JPEG, GIF et PNG.
pipeline des sources d'image	Une chaîne que le client résout lui-même: une URL relative ou absolue, une image de thème nommée, ou un chemin resource.wx. Les extensions d'image resource.wx documentées sont.jpg,.jpeg,.png,.gif,.svg et.bmp.
PictureBox.Image	Le point d'entrée pour des pixels que votre code C# possède déjà ou vient de générer, par exemple un graphique dessiné avec Graphics.
PictureBox.ImageSource	Le point d'entrée pour une image que le navigateur peut récupérer lui-même. Une courte chaîne voyage à la place des octets de l'image, et c'est pourquoi le SVG a sa place ici.
PictureBox.SizeMode	Les six valeurs Normal, AutoSize, CenterImage, StretchImage, Zoom et Cover, qui décident si une image est coupée, centrée, déformée, ajustée ou rognée.
Load et LoadAsync	Load(url) met à jour l'image depuis une URL de façon synchrone; LoadAsync(url) effectue la même récupération de façon asynchrone, si bien qu'un hôte lent ne bloque pas l'écran.
vectoriel contre raster	Un SVG est du markup dessiné à la taille que réclame la mise en page, et un visuel monochrome adapté peut être recoloré; un fichier raster est une grille de pixels figée qui ne fait jamais que se rééchantillonner.

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 2. Vous pouvez ouvrir vous-même [module-02.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 IconDesk 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 2 - « Bitmaps, SVG et PictureBox : formats, dimensionnement et chargement » - du cours LearnWisej « Icônes et images 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/icons-and-images/specs/module-02.md>

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

1. Projet « IconDesk », 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 dans IconDesk une page ImageLab qui rend les deux pipelines visibles côte à côte. Affichez un même visuel raster quatre fois avec `SizeMode` défini sur Normal, Zoom, Cover et StretchImage pour que les différences de rognage et de déformation sautent aux yeux, ajoutez un PictureBox qui récupère un JPEG distant avec `LoadAsync` et garde la page réactive pendant son arrivée, et placez un SVG affecté via `ImageSource` à 16, 24, 32 et 48 pixels pour prouver qu'il reste net. Terminez par un gestionnaire qui indique, pour chaque PictureBox de la page, si son image vient d'Image ou d'ImageSource.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Page ImageLab affichant un même visuel raster dans les modes de dimensionnement Normal, Zoom, Cover et StretchImage
 - PictureBox qui récupère un JPEG distant avec `LoadAsync` sans bloquer la page
 - Un SVG affecté via `ImageSource` et rendu à 16, 24, 32 et 48 pixels
 - Gestionnaire de rapport qui nomme, pour chaque PictureBox, si l'image vient d'Image ou d'ImageSource
 - Note de formats listant les ressources du lab qui passent par `System.Drawing` et celles qui restent des chaînes de ressource
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 visuel raster est plus large que haut et la zone est carrée. Lequel de Zoom et de Cover perd une partie de l'image, lequel laisse de l'espace vide, et lequel défendriez-vous pour le logo d'un client?
- L'hôte du JPEG distant est inaccessible pendant dix secondes. Que vit l'utilisateur de votre page avec `LoadAsync`, et qu'aurait-il vécu avec le Load synchrone?

– Votre gestionnaire de rapport indique que l'image d'un contrôle vient d' ImageSource, mais le fichier designer affecte aussi Image sur le même contrôle. Laquelle est réellement dessinée, et comment le prouveriez-vous plutôt que de le deviner?

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 visuel raster est plus large que haut et la zone est carrée. Lequel de Zoom et de Cover perd une partie de l'image, lequel laisse de l'espace vide, et lequel défendriez-vous pour le logo d'un client?
- ✓ L'hôte du JPEG distant est inaccessible pendant dix secondes. Que vit l'utilisateur de votre page avec LoadAsync, et qu'aurait-il vécu avec le Load synchrone?
- ✓ Votre gestionnaire de rapport indique que l'image d'un contrôle vient d' ImageSource, mais le fichier designer affecte aussi Image sur le même contrôle. Laquelle est réellement dessinée, et comment le prouveriez-vous plutôt que de le deviner?

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 IconDesk 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 IconDesk 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 IconDesk 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.