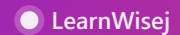


KI-Coding-Übung

Erstellen Sie das Beispielprojekt IconDesk dieses Moduls mit ChatGPT oder Claude aus einem fertigen Prompt, der das Modell auf die Spezifikation des Moduls verweist – und prüfen, starten und erweitern Sie dann, was Sie erhalten.



Der Assistent liefert die schnellen Hände. Der Architekt bleiben Sie.

Ein Assistent erstellt das Beispielprojekt dieses Moduls in wenigen Minuten. Was er nicht kann, ist zu entscheiden, ob das Ergebnis der richtige Entwurf für Ihre Anwendung ist – diese Entscheidung bleibt bei Ihnen. Das Urteilsvermögen bringen Sie mit: Instruieren Sie ihn sorgfältig und prüfen Sie dann, was zurückkommt, anhand dessen, was Ihnen Modul 2 gerade vermittelt hat.

Worum Sie bitten

Bauen Sie in IconDesk eine Seite ImageLab, die die zwei Pipelines nebeneinander sichtbar macht. Zeigen Sie ein Rasterbild viermal mit SizeMode auf Normal, Zoom, Cover und StretchImage, damit die Unterschiede bei Zuschchnitt und Verzerrung offensichtlich werden, ergänzen Sie eine PictureBox, die ein entferntes JPEG mit LoadAsync holt und die Seite währenddessen bedienbar hält, und platzieren Sie ein über ImageSource zugewiesenes SVG in 16, 24, 32 und 48 Pixeln, um zu belegen, dass es scharf bleibt. Schließen Sie mit einem Handler ab, der für jede PictureBox der Seite meldet, ob ihr Bild aus Image oder aus ImageSource stammt.

Das Modul verlangt diese Ergebnisse, und der Prompt unten fordert genau diese vom Assistenten an:

- ✓ Seite ImageLab, die ein Rasterbild in den Size Modes Normal, Zoom, Cover und StretchImage zeigt
- ✓ PictureBox, die ein entferntes JPEG mit LoadAsync holt, ohne die Seite zu blockieren
- ✓ Ein über ImageSource zugewiesenes SVG, gerendert in 16, 24, 32 und 48 Pixeln
- ✓ Report-Handler, der pro PictureBox benennt, ob Image oder ImageSource das Bild geliefert hat
- ✓ Format-Notiz, die auflistet, welche Lab-Assets über System.Drawing laufen und welche Ressourcen-Strings bleiben

Warum es so aufgebaut ist

„Welche Bildformate unterstützt Wisej.NET?“ ist die Frage, die jedes Team in der ersten Woche stellt, und die ehrliche Antwort hängt davon ab, welchen API-Weg das Bild nimmt. Beantworten Sie sie mit einer vagen Liste, bekommen Sie ein verzerrtes Logo auf einem High-DPI-Laptop, eine Seite, die einfriert, während ein Produktfoto von einem langsamen Host geladen wird, und eine Fehlermeldung, weil sich eine.ico -Datei nicht über einen Ressourcen-String auflösen lässt. Dieses Modul ersetzt die vage Liste durch zwei präzise, jede fest verbunden mit der Eigenschaft des Steuerelements, die sie verwendet.

Diese Begriffe muss der Assistent richtig verwenden. Geht die Antwort nachlässig mit ihnen um, hat er das Modul nicht verstanden:

Begriff	Was er hier bedeutet
System.Drawing-Pipeline	Die Eigenschaft Image hält ein System.Drawing.Image im Serverspeicher; Wisej.NET sendet es als PNG-Daten an den Browser. Ihre dokumentierten Quellen sind Bitmap-, Metafile-, Icon-, JPEG-, GIF- und PNG-Dateien.
Bildquellen-Pipeline	Ein String, den der Client selbst auflöst: eine relative oder absolute URL, ein benanntes Theme-Bild oder ein resource.wx -Pfad. Die dokumentierten Bild-Erweiterungen für resource.wx sind.jpg,.jpeg,.png,.gif,.svg und.bmp.
PictureBox.Image	Der Einstiegspunkt für Pixel, die Ihr C#-Code bereits besitzt oder gerade erzeugt hat, etwa ein mit Graphics gezeichnetes Diagramm.
PictureBox.ImageSource	Der Einstiegspunkt für ein Bild, das der Browser selbst holen kann. Statt Bildbytes reist ein kurzer String – deshalb gehört SVG hierher.
PictureBox.SizeMode	Die sechs Werte Normal, AutoSize, CenterImage, StretchImage, Zoom und Cover, die entscheiden, ob ein Bild beschnitten, zentriert, verzerrt, eingepasst oder zugeschnitten wird.
Load und LoadAsync	Load(url) aktualisiert das Bild synchron aus einer URL; LoadAsync(url) holt dieselbe Datei asynchron, sodass ein langsamer Host den Bildschirm nicht blockiert.
Vektor gegenüber Raster	Ein SVG ist Markup, das in der vom Layout gewünschten Größe gezeichnet wird, und geeignete einfarbige Grafiken lassen sich umfärben; eine Rasterdatei ist ein festes Pixelraster, das immer nur neu berechnet wird.

Selbst bauen – wählen Sie Ihren Assistenten

Beide Buttons öffnen einen neuen Tab, in dem **derselbe fertige Prompt** bereits eingetragen ist. Der Prompt verweist das Modell auf die Spezifikation dieses Moduls – eine einzelne Markdown-Datei mit der Lektion, der Lab-Aufgabe und den geforderten Ergebnissen –, sodass es mit dem echten Material arbeitet, statt zu raten, worum es in Modul 2 geht. Sie können [module-02.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden

Öffnet [chatgpt.com](#) mit dem Prompt dieses Moduls. Funktioniert mit GPT-5 oder Codex; jeder Tarif mit Websuche ruft die Spezifikation ab.



Claude verwenden

Öffnet [claude.ai](#) mit demselben Prompt. Wenn Sie Claude Code nutzen, können Sie ihn stattdessen in einem leeren Ordner in ein Terminal einfügen.



Unverändert abgeschickt, erhalten Sie ein Projekt, kein Code-Fragment

Der Prompt verlangt am Ende die gesamte Projektmappe als einzelne `.zip`-Datei mit Anleitung zum Kompilieren und Starten. Laden Sie sie herunter, öffnen Sie die Projektmappe in Visual Studio, drücken Sie F5, und IconDesk startet auf `localhost`. Sie benötigen das .NET SDK und das NuGet-Paket `Wisej-4`; alles andere soll der Assistent laut Prompt selbst schreiben.

Ich arbeite gerade Modul 2 – „Bitmaps, SVGs und PictureBox: Formate, Größenanpassung und Laden“ – des LearnWisej-Kurses „Icons und Bilder in Wisej.NET“ durch und möchte das Beispielprojekt des Moduls gemeinsam mit dir erstellen. Bitte antworte auf Deutsch.

Lies zuerst die Modulspezifikation. Sie ist eine einzelne Markdown-Datei (auf Englisch) mit der vollständigen Lektion, der Lab-Aufgabe und den geforderten Ergebnissen:

<https://learnwisej.com/assets/courses/icons-and-images/specs/module-02.md>

Erstelle dann ein vollständiges Wisej.NET-4-Projekt, das ich in Visual Studio öffnen kann:

1. Projekt „IconDesk“, C#, net8.0 oder neuer, NuGet-Paket Wisej-4 4.1.x, Kestrel-Host, der `app.UseWisej()` aufruft.
2. Setze um, was das Lab der Spezifikation verlangt: Bauen Sie in IconDesk eine Seite `ImageLab`, die die zwei Pipelines nebeneinander sichtbar macht. Zeigen Sie ein Rasterbild viermal mit `SizeMode` auf `Normal`, `Zoom`, `Cover` und `StretchImage`, damit die Unterschiede bei Zuschchnitt und Verzerrung offensichtlich werden, ergänzen Sie eine `PictureBox`, die ein entferntes JPEG mit `LoadAsync` holt und die Seite währenddessen bedienbar hält, und platzieren Sie ein über `ImageSource` zugewiesenes SVG in 16, 24, 32 und 48 Pixeln, um zu belegen, dass es scharf bleibt. Schließen Sie mit einem Handler ab, der für jede `PictureBox` der Seite meldet, ob ihr Bild aus `Image` oder aus `ImageSource` stammt.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Seite `ImageLab`, die ein Rasterbild in den Size Modes `Normal`, `Zoom`, `Cover` und `StretchImage` zeigt
 - `PictureBox`, die ein entferntes JPEG mit `LoadAsync` holt, ohne die Seite zu blockieren
 - Ein über `ImageSource` zugewiesenes SVG, gerendert in 16, 24, 32 und 48 Pixeln
 - Report-Handler, der pro `PictureBox` benennt, ob `Image` oder `ImageSource` das Bild geliefert hat
 - Format-Notiz, die auflistet, welche Lab-Assets über `System.Drawing` laufen und welche Ressourcen-Strings bleiben
4. Halte dich an die Spezifikation, wo immer sie eine Wisej.NET-Klasse, -Eigenschaft oder ein -Event nennt – verwende genau diese. Wo sie eine Wahl offenlässt, sag mir, welche API du gewählt hast und warum.
5. Der Server bleibt maßgeblich: Zustand und Validierung leben in .NET, der Browser stellt dar und meldet zurück. Verlagere nie eine Entscheidung nach JavaScript, nur weil es dort einfacher ist.
6. Mach das Ergebnis vorführbar: Ich muss es starten und das Verhalten sehen können, um das es in diesem Modul geht – auch, was passiert, wenn etwas schiefgeht.

Bevor du Code schreibst, fasse in eigenen Worten zusammen, was du bauen wirst und wo jedes Teil liegen wird, damit ich den Entwurf korrigieren kann, solange das noch wenig kostet.

Wenn der Code fertig ist, packe die gesamte Projektmappe in eine einzelne .zip-Datei zum Herunterladen und nenne mir die genauen Schritte, um sie zu kompilieren und auszuführen.

Beantworte zum Schluss diese Fragen aus der Spezifikation und nutze dabei deine eigene Implementierung als Beleg:

- Ihr Rasterbild ist breiter als hoch, und die Box ist quadratisch. Welcher der beiden Modi `Zoom` und `Cover` verliert einen Teil des Bildes, welcher lässt freien Raum, und welchen würden Sie für ein Kundenlogo verteidigen?
- Der Host des entfernten JPEG ist zehn Sekunden lang nicht erreichbar. Was erleben die Benutzer Ihrer Seite mit `LoadAsync`, und was hätten sie mit dem synchronen Load erlebt?
- Ihr Report-Handler meldet, das Bild eines Steuerelements stamme aus `ImageSource`, aber die

Designer-Datei weist auf demselben Steuerelement auch Image zu. Welches wird tatsächlich gezeichnet, und wie würden Sie das belegen, statt zu raten?

Frag mich nach allem, was unklar ist, bevor du anfängst.

Prüfen Sie dann, was zurückkommt

Code, der kompiliert, ist noch lange kein richtiger Code. Die spannenden Fragen sind Designfragen: Ist das wirklich das Wisej.NET-Control, das die Aufgabe verlangt? Folgt der Code dem Lebenszyklus, den das Framework vorgibt? Liegt der maßgebliche Zustand auf dem Server – oder wurde er im Browser abgelegt, weil das bequemer war? Lesen Sie das Ergebnis als Reviewer, nicht als Kunde.

Die Checkliste liefert Ihnen dieses Modul. Beantworten Sie jeden Punkt anhand des Codes, den Sie gerade erhalten haben:

- ✓ Ihr Rasterbild ist breiter als hoch, und die Box ist quadratisch. Welcher der beiden Modi Zoom und Cover verliert einen Teil des Bildes, welcher lässt freien Raum, und welchen würden Sie für ein Kundenlogo verteidigen?
- ✓ Der Host des entfernten JPEG ist zehn Sekunden lang nicht erreichbar. Was erleben die Benutzer Ihrer Seite mit LoadAsync, und was hätten sie mit dem synchronen Load erlebt?
- ✓ Ihr Report-Handler meldet, das Bild eines Steuerelements stamme aus ImageSource, aber die Designer-Datei weist auf demselben Steuerelement auch Image zu. Welches wird tatsächlich gezeichnet, und wie würden Sie das belegen, statt zu raten?

Alles, was Sie korrigieren mussten, sollten Sie festhalten. Eine kurze Notiz zu den Konventionen und Entscheidungen, auf die Sie sich festgelegt haben – eingefügt in die erste Nachricht Ihrer nächsten Sitzung –, ist mehr wert als jeder noch so clevere Prompt.



Für diesen Kurs gibt es noch kein Referenzbeispiel – Ihres ist das erste

Das [LearnWisej-Beispiel-Repository](#) enthält ausgearbeitete Builds für die meisten Kurse, für diesen aber noch nicht. Es gibt also nichts zum Vergleichen und keine Abkürzung: Die Spezifikation und Ihr eigenes Review sind das ganze Sicherheitsnetz. Behalten Sie, was Sie bauen – und wenn es gut wird, würden wir es gern sehen.

Drei Prompts für den nächsten Schritt

Richten Sie Ihren Assistenten auf das Projekt, das Sie gerade gebaut haben, und probieren Sie diese Prompts aus. Jeder ist eine Übung zum Prüfen und Erweitern, kein Neuschreiben.

- ✓ **Prüfen Sie es wie einen Pull-Request.** *"Führ mich durch diesen IconDesk-Code, als würdest du den Pull-Request eines Kollegen prüfen und hättest ihn nicht selbst geschrieben. Wo lebt der Zustand, was passiert bei einer einzelnen Benutzeraktion von Anfang bis Ende, was würdest du zur Überarbeitung zurückgeben, und welche Teile würdest du nicht guten Gewissens einem Kunden vorlegen? Sei konkret und zitiere die Zeilen, die du meinst – fasse mir nicht den Entwurf zusammen."*
- ✓ **Tauschen Sie ein Teil aus, ohne die Schnittstellen zu verschieben.** *"Wähle den Teil dieses IconDesk-Projekts, der am offensichtlichsten nur ein Platzhalter für etwas Echtes ist – eine Fake-*

Datenquelle, ein In-Memory-Speicher, ein vereinfachter Service – und ersetze ihn durch eine praxisnahe Implementierung. Der Vertrag auf beiden Seiten darf sich nicht ändern: dieselben Typen, dieselben Aufrufe, dasselbe Verhalten im Fehlerfall. Sag mir, was du durch die echte Implementierung behandeln musstest, was der Platzhalter nicht behandelt hat."

- ✓ **Machen Sie es fit für einen schlechten Tag.** *"Untersuche dieses IconDesk-Projekt auf Fehlerszenarien, die es noch nicht behandelt – fehlerhafte oder fehlende Eingaben, eine Abhängigkeit, die langsam oder nicht erreichbar ist, Arbeit, die eintrifft, nachdem der Benutzer die Seite verlassen hat, und zwei Aktionen, die sich gegenseitig in die Quere kommen. Sag mir für jedes zuerst, ob es in Wisej.NET 4 ein echtes Risiko ist und warum, und schreib erst dann die Korrektur. Wo du entscheidest, dass nichts behandelt werden muss, sag das und begründe es, statt defensiven Code hinzuzufügen."*



Nehmen Sie es mit ins Lab

Das praktische Lab am Ende dieses Moduls verlangt dieselben Ergebnisse, die Sie gerade vom Assistenten haben erstellen lassen. Nutzen Sie, was Sie hier haben, als Gerüst für Ihre Antwort und als Beleg dafür – nicht als die Antwort selbst. Bewertet wird im Lab Ihre Argumentation, und dieser Teil muss von Ihnen stammen.