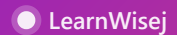


KI-Coding-Übung

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.



Instruieren, lesen, dann auf die Probe stellen.

Diese Übung hat drei Teile, und nur der erste dreht sich um den Prompt. Sie instruieren einen Assistenten mit der Modulspezifikation und erhalten eine IconDesk-Projektmappe zurück; Sie lesen, was er geschrieben hat, und entscheiden, ob es wirklich tut, was das Modul verlangt; dann treiben Sie es weiter, als das Modul verlangt hat. Im zweiten und dritten Teil steckt das eigentliche Lernen.

Worum Sie bitten

Füllen Sie in IconDesk eine Seite IconGallery, indem Sie den Image Selector des Designers verwenden, statt Quell-Strings von Hand zu tippen. Weisen Sie fünf Steuerelementen Bilder zu: ein benanntes Theme-Bild, ein Projekt-SVG über eine relative URL, eine absolute URL und mindestens zwei Assets aus einem Icon-Pack, falls eines installiert ist. Färben Sie ein einfarbiges SVG mit dem Suffix `?color=` und dem Namen einer Theme-Farbe um, öffnen Sie die erzeugte Designer.cs und lesen Sie die Bildquellen-Strings, die der Selector geschrieben hat; schalten Sie die Anwendung dann zwischen zwei Wisej-Themes um und halten Sie fest, welche Assets sich geändert haben und welche exakt gleich geblieben sind.

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

- ✓ Seite IconGallery mit fünf Steuerelementen, deren Bilder über den Image Selector des Designers zugewiesen wurden
- ✓ Darunter ein benanntes Theme-Bild, ein relatives Projekt-SVG, eine absolute URL und mindestens zwei Icon-Pack-Assets
- ✓ Einfarbiges SVG, das über das Suffix `?color=` mit dem Namen einer Theme-Farbe umgefärbt wurde
- ✓ Auszug aus Designer.cs mit den Bildquellen-Strings, die der Selector erzeugt hat
- ✓ Notiz zum Theme-Wechsel, die festhält, welche Assets dem Theme gefolgt sind und welche unverändert blieben

Warum es so aufgebaut ist

IconDesk hat inzwischen ein Befehls-Panel und ein Bildlabor, und jedes Bild darin ist eine ausgelieferte Datei plus ein von Hand getippter Pfad. Das trägt, bis die Anwendung ein neues Theme bekommt, unter einem anderen virtuellen Verzeichnis ausgeliefert wird oder dasselbe Symbol auf einem Bildschirm in einer Warnfarbe und auf einem anderen neutral zeigen soll. Dieses Modul verlegt den Ursprung eines Bildes aus

dem Projektordner in das Theme, in eine URL oder in den Designer, damit der String, den ein Steuerelement trägt, auch nach einem Wechsel von Grafik, Theme oder Server noch etwas bedeutet.

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
benanntes Theme-Bild	Ein semantischer Name wie icon-print, den das aktive Theme auf eine Grafik abbildet, sodass ein anderes Theme ohne C#-Änderung ein anderes Bild liefern kann.
Theme-Cache des Clients	Theme-Bilder reisen mit dem Theme und liegen bereits im Browser, wenn ein Steuerelement eines anfordert – das macht ein Theme zum richtigen Ort für einen kleinen Satz von Befehlssymbolen und zum falschen für Tausende.
Quelle mit relativer URL	Ein Bildquellen-String, der gegen die Anwendung aufgelöst wird: die einfachste Form für ein Logo, das mit der Anwendung ausgeliefert wird.
Quelle mit absoluter URL	Eine vollständige externe Adresse, die das Bild zu einer Abhängigkeit von einem anderen System macht – mit Folgen für Verfügbarkeit, Datenschutz, CORS und Content Security Policy.
eingebettetes base64-Bild	Bildbytes, die in die Theme-Definition selbst geschrieben werden, sodass die Grafik keine zweite Anfrage braucht – zum Preis eines größeren Themes.
Image Selector	Der Auswahldialog zur Entwurfszeit in Visual Studio, der Projektbilder, Theme-Bilder und referenzierte Icon-Pack-Ressourcen durchsucht und den gewählten Quell-String dann in Designer.cs serialisiert.
Suffix?color=	Ein Suffix an einer Bildquelle, das Wisej.NET bittet, ein geeignetes einfarbiges SVG mit einer CSS-Farbe oder dem Namen einer Theme-Farbe umzufärben.

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 3 geht. Sie können [module-03.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 3 – „Theme-Bilder, URLs, SVG-Farbe und der Image Selector im Designer“ – 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-03.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: Füllen Sie in IconDesk eine Seite IconGallery, indem Sie den Image Selector des Designers verwenden, statt Quell-Strings von Hand zu tippen. Weisen Sie fünf Steuerelementen Bilder zu: ein benanntes Theme-Bild, ein Projekt-SVG über eine relative URL, eine absolute URL und mindestens zwei Assets aus einem Icon-Pack, falls eines installiert ist. Färben Sie ein einfarbiges SVG mit dem Suffix `?color=` und dem Namen einer Theme-Farbe um, öffnen Sie die erzeugte Designer.cs und lesen Sie die Bildquellen-Strings, die der Selector geschrieben hat; schalten Sie die Anwendung dann zwischen zwei Wisej-Themes um und halten Sie fest, welche Assets sich geändert haben und welche exakt gleich geblieben sind.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Seite IconGallery mit fünf Steuerelementen, deren Bilder über den Image Selector des Designers zugewiesen wurden
 - Darunter ein benanntes Theme-Bild, ein relatives Projekt-SVG, eine absolute URL und mindestens zwei Icon-Pack-Assets
 - Einfarbiges SVG, das über das Suffix `?color=` mit dem Namen einer Theme-Farbe umgefärbt wurde
 - Auszug aus Designer.cs mit den Bildquellen-Strings, die der Selector erzeugt hat
 - Notiz zum Theme-Wechsel, die festhält, welche Assets dem Theme gefolgt sind und welche unverändert blieben
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:

- Die Galerie erscheint auf Ihrem Rechner korrekt, zeigt aber auf dem einer Kollegin oder eines Kollegen nichts für die absolute URL. Welcher der fünf Mechanismen könnte diesen Unterschied erzeugen, und was würden Sie zuerst prüfen?
- Zwei Ihrer Icons ändern ihre Farbe nicht, obwohl das Suffix angehängt ist. Was steckt in

diesen Dateien, und wann ist es die falsche Antwort, sie zu bearbeiten?

– Eine Designerin verlangt dasselbe Symbol in der Galerie, in einer Toolbar und in einer Grid-Zelle. Was kostet später am wenigsten Aufwand beim Ändern – der Theme-Schlüssel, die relative URL oder die Pack-Ressource – und warum?

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:

- ✓ Die Galerie erscheint auf Ihrem Rechner korrekt, zeigt aber auf dem einer Kollegin oder eines Kollegen nichts für die absolute URL. Welcher der fünf Mechanismen könnte diesen Unterschied erzeugen, und was würden Sie zuerst prüfen?
- ✓ Zwei Ihrer Icons ändern ihre Farbe nicht, obwohl das Suffix angehängt ist. Was steckt in diesen Dateien, und wann ist es die falsche Antwort, sie zu bearbeiten?
- ✓ Eine Designerin verlangt dasselbe Symbol in der Galerie, in einer Toolbar und in einer Grid-Zelle. Was kostet später am wenigsten Aufwand beim Ändern – der Theme-Schlüssel, die relative URL oder die Pack-Ressource – und warum?

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.

- ✓ **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."*
- ✓ **Ergänzen Sie den zweiten Fall.** *"Erweitere dieses IconDesk-Projekt um einen zweiten Fall, der demselben Muster folgt wie der erste – eine weitere Entität, Maske oder ein Control derselben Form, auf dieselbe Weise verdrahtet. Lagere wirklich Gemeinsames in eine Basisklasse oder einen Service aus, aber nur dort, wo es sich lohnt; zwei ähnliche Dinge brauchen nicht immer eine Abstraktion. Zeig mir den Diff und begründe jede Auslagerung."*



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.