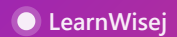


KI-Coding-Übung

Erstellen Sie das Beispielprojekt AdaptiveOps 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, auch im Vergleich mit unserem eigenen Referenz-Build auf GitHub.



Dasselbe Modul – so gebaut, wie Sie in der Praxis wirklich bauen.

Alle anderen Abschnitte dieses Moduls erklären, wie „FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel und erweiterte Layout-Eigenschaften“ funktioniert. Dieser fordert Sie auf, es herzustellen: indem Sie einen Assistenten mit der Spezifikation des Moduls steuern und dann prüfen, ob das Ergebnis läuft und tut, was die Lab-Aufgabe beschreibt. Es ist der Arbeitsablauf, den Sie in echten Projekten nutzen werden – geübt an etwas, dessen richtige Lösung Sie bereits kennen.

Worum Sie bitten

Bauen Sie drei Varianten des Inhaltsbereichs der Adaptive Operations Console: einen Filter- und Kennzahlenkartenbereich mit FlowLayoutPanel, der umbricht und FillWeight sowie FlowBreak nutzt, einen Ticket-Editor mit TableLayoutPanel, dessen Labels und Editoren über RowStyles und ColumnStyles ausgerichtet sind, und eine Dashboard-Shell mit FlexLayoutPanel, in der sich Listen- und Detailbereich den Platz per FillWeight teilen, mit gesetztem AlignX, AlignY, MinimumSize und MaximumSize. Bauen Sie einen der drei Bereiche im Code mit den fluenten Erweiterungen aus Wisej.Web.Markup und vergleichen Sie, wie jeder eine Größenänderung des Browsers übersteht.

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

- ✓ Filter- und Kennzahlenkartenbereich mit FlowLayoutPanel unter Verwendung von WrapContents, FillWeight und FlowBreak
- ✓ Ticket-Editor mit TableLayoutPanel, RowStyles, ColumnStyles und Spans für ausgerichtete Felder
- ✓ Dashboard-Shell mit FlexLayoutPanel, FillWeight, AlignX, AlignY, MinimumSize und MaximumSize
- ✓ Ein Bereich im Code mit den fluenten Erweiterungen aus Wisej.Web.Markup gebaut
- ✓ Screenshots zum Größenänderungsvergleich und eine Notiz, welcher Container zu welchem Bereich passt

Warum es so aufgebaut ist

Modul 4 hat der Adaptive Operations Console eine Shell aus gedockten Bereichen gegeben, doch Docking entscheidet nur, wo ein Bereich sitzt, nicht, wie sich der Inhalt darin verhält, wenn der Browser am Montag 1400 Pixel breit ist und am Dienstag auf einem Tablet 700. Eine Reihe von Kennzahlenkarten soll

umbrechen, ein Ticket-Editor soll Labels und Editoren ausgerichtet halten, wie lang die deutsche Übersetzung auch sein mag, und Listen- und Detailbereich sollen sich den Arbeitsbereich in einem Verhältnis teilen, das Sie festlegen. Jedes davon ist ein anderes Inhaltsverhalten, und Wisej.NET liefert für jedes einen Container: FlowLayoutPanel, TableLayoutPanel und FlexLayoutPanel. Wer den falschen wählt, schreibt am Ende wieder Resize-Code.

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
FlowLayoutPanel	Ordnet Children in automatischen Zeilen oder Spalten entlang FlowDirection an, bricht um, wenn WrapContents true ist, und ignoriert Location, Dock und Anchor jedes Childs, sodass Chips und Karten sich selbst neu anordnen.
TableLayoutPanel	Platziert Children in Zellen, adressiert über Row und Column, mit Spans für Zellen, die sich erstrecken; so bleiben Labels und Editoren ausgerichtet, während jedes Child innerhalb seiner Zelle weiterhin docken oder verankern kann.
FlexLayoutPanel	Ordnet Children in einer Zeile oder einer Spalte an, füllt stets seinen Client-Bereich und verteilt den verbleibenden Platz nach Gewichtung – passend für Hauptbereiche, die den Arbeitsbereich proportional aufteilen.
FillWeight	Eine erweiterte Child-Eigenschaft, die die Flow- und Flex-Engines lesen: Ein Gewicht über null dehnt das Child im Verhältnis zu den anderen gewichteten Children in den verbleibenden Platz; null behält seine eigene Größe.
FlowBreak	Eine erweiterte Child-Eigenschaft des FlowLayoutPanel, die das nächste Child in eine neue Zeile oder Spalte zwingt, sodass eine Filterleiste eine Zeile bewusst beendet, statt sich auf die Breite zu verlassen.
Zeilen- und Spalten-Styles	RowStyles und ColumnStyles legen fest, ob eine Zeile oder Spalte ihre Größe automatisch anpasst, eine feste Pixelgröße erhält oder sich den Rest prozentual teilt; hier liegen die Proportionen eines Formulars.
Margin und Padding	Padding ist der Innenraum des Containers um alle Children; Margin ist der Abstand des Childs zu seinen Nachbarn, und nur Engines, die Margins berücksichtigen, verwenden ihn.

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 5 geht. Sie können [module-05.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden



Claude verwenden

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

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 AdaptiveOps 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 5 – „FlowLayoutPanel, TableLayoutPanel, FlexLayoutPanel und erweiterte Layout-Eigenschaften“ – des LearnWisej-Kurses „Theming und responsive Layouts“ 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/theming-and-responsive-layouts/specs/module-05.md>

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

1. Projekt „AdaptiveOps“, 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 drei Varianten des Inhaltsbereichs der Adaptive Operations Console: einen Filter- und Kennzahlenkartenbereich mit FlowLayoutPanel, der umbricht und FillWeight sowie FlowBreak nutzt, einen Ticket-Editor mit TableLayoutPanel, dessen Labels und Editoren über RowStyles und ColumnStyles ausgerichtet sind, und eine Dashboard-Shell mit FlexLayoutPanel, in der sich Listen- und Detailbereich den Platz per FillWeight teilen, mit gesetztem AlignX, AlignY, MinimumSize und MaximumSize. Bauen Sie einen der drei Bereiche im Code mit den fluenten Erweiterungen aus Wisej.Web.Markup und vergleichen Sie, wie jeder eine Größenänderung des Browsers übersteht.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Filter- und Kennzahlenkartenbereich mit FlowLayoutPanel unter Verwendung von WrapContents, FillWeight und FlowBreak
 - Ticket-Editor mit TableLayoutPanel, RowStyles, ColumnStyles und Spans für ausgerichtete Felder
 - Dashboard-Shell mit FlexLayoutPanel, FillWeight, AlignX, AlignY, MinimumSize und MaximumSize
 - Ein Bereich im Code mit den fluenten Erweiterungen aus Wisej.Web.Markup gebaut
 - Screenshots zum Größenänderungsvergleich und eine Notiz, welcher Container zu welchem Bereich passt
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:

- Sie haben dem Detailbereich FillWeight 1, aber keine MinimumSize gegeben, und bei 700 Pixeln ist er verschwunden. Erklären Sie, was die Flex-Engine mit dem Platz gemacht hat und warum die Lösung eine Größenbeschränkung ist und kein anderes Gewicht.
- Der Apply-Button bricht bei 1024 Pixeln in die zweite Zeile um, obwohl Platz für ihn wäre.

Welche erweiterte Eigenschaft hat das bewirkt, und wie würde sich die FilterBar verhalten, wenn Sie sie entfernen und `WrapContents` auf `false` setzen?

– Ein Margin von 8 an den Karten im `FlowLayoutPanel` hat Abstände erzeugt, derselbe Margin an der gedockten Navigationsleiste hat nichts bewirkt. Welche Engine berücksichtigt Margins im jeweiligen Fall, und welche Eigenschaft erzeugt den Abstand für die Leiste?

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:

- ✓ Sie haben dem Detailbereich `FillWeight` 1, aber keine `MinimumSize` gegeben, und bei 700 Pixeln ist er verschwunden. Erklären Sie, was die Flex-Engine mit dem Platz gemacht hat und warum die Lösung eine Größenbeschränkung ist und kein anderes Gewicht.
- ✓ Der Apply-Button bricht bei 1024 Pixeln in die zweite Zeile um, obwohl Platz für ihn wäre. Welche erweiterte Eigenschaft hat das bewirkt, und wie würde sich die FilterBar verhalten, wenn Sie sie entfernen und `WrapContents` auf `false` setzen?
- ✓ Ein Margin von 8 an den Karten im `FlowLayoutPanel` hat Abstände erzeugt, derselbe Margin an der gedockten Navigationsleiste hat nichts bewirkt. Welche Engine berücksichtigt Margins im jeweiligen Fall, und welche Eigenschaft erzeugt den Abstand für die Leiste?

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.



Dieses Beispiel haben wir schon gebaut – vergleichen Sie Ihres mit unserem

Die Referenzimplementierung zu Modul 5 liegt im öffentlichen Beispiel-Repository. Bauen Sie zuerst Ihre eigene Version und öffnen Sie dann unsere: Die Unterschiede sind der interessante Teil.



[Beispiel zu Modul 5 auf GitHub öffnen](#)

Oder klonen Sie alle Beispiele mit `git clone https://github.com/iceteagroup/LearnWisej-Samples` und öffnen Sie [Theming and Responsive Layouts Course/Module 5](#).



Unser Beispiel ist ein Ausgangspunkt, kein fertiges Produkt

Es ist genauso entstanden wie Ihres – per Prompt und anschließend korrigiert – und endet bewusst dort, wo das Modul endet. Rechnen Sie mit Platzhaltern, wo Produktivcode hingehört, mit knapper Fehlerbehandlung und ohne Tests. Verstehen Sie das als Einladung: Nutzen Sie, was Sie in diesem Modul gelernt haben, und dazu den Assistenten, um es weiterzuentwickeln. Code zu verbessern, den Sie nicht selbst geschrieben haben, ist die Fähigkeit, die dieser Kurs eigentlich vermittelt.

Drei Prompts für den nächsten Schritt

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

- ✓ **Machen Sie es fit für einen schlechten Tag.** *"Untersuche dieses AdaptiveOps-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 AdaptiveOps-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."*
- ✓ **Beweisen Sie, dass es wirklich funktioniert.** *"Geh dieses Projekt Ergebnis für Ergebnis anhand der Modulspezifikation unter <https://learnwisej.com/assets/courses/theming-and-responsive-layouts/specs/module-05.md> durch und zeig mir den Beleg, dass jedes wirklich erledigt ist – den Codepfad, worauf ich klicken muss und was ich sehen sollte. Wo ein Ergebnis nur teilweise erfüllt ist, sag das klar, statt es schönzureden. Ergänze dann, was fehlt."*



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.