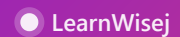


KI-Coding-Übung

Erstellen Sie das Beispielprojekt GlobalDesk 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.



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

Alle anderen Abschnitte dieses Moduls erklären, wie „Rechts-nach-links-Sprachen und lokalisierungssichere Layouts“ 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

Machen Sie GlobalDesk fit für eine Kultur, die von rechts nach links geschrieben wird. Nehmen Sie Arabisch in die Sprachauswahl auf und setzen Sie `rightToLeft` in `Default.json` auf `auto`, lassen Sie die untergeordneten Controls auf `RightToLeft.Inherit` und setzen Sie `RightToLeftLayout` auf dem Dashboard und dem Kundeneditor, damit deren Kinder gespiegelt werden. Lassen Sie genau ein Control von links nach rechts – das Feld für den Kundencode – und begründen Sie in einem Kommentar, warum dieser Bezeichner nicht umgedreht werden darf. Bauen Sie den fest positionierten Teil des Layouts mit Docking, Anchoring und einem Container-Layout neu auf, damit eine lange Übersetzung nicht abgeschnitten werden kann, fügen Sie eine pseudo-lokalisierte Ressourcendatei hinzu, deren Werte verlängert und in Klammern gesetzt sind, und halten Sie fest, was die Kontakt-DataGridView unter RTL tatsächlich tut, statt anzunehmen, dass sie spiegelt. Schließen Sie mit einer Layout-Testliste aus fünf Fällen ab, die Sie auf jedem Bildschirm erneut durchgehen können.

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

- ✓ Eintrag für Arabisch in der Sprachauswahl mit `rightToLeft` auf `auto` in `Default.json`
- ✓ Dashboard und Kundeneditor mit `RightToLeftLayout`, deren Kinder auf `Inherit` belassen sind
- ✓ Ein bewusst von links nach rechts laufendes Control mit einem Kommentar, der die Ausnahme erklärt
- ✓ Mit Docking, Anchoring und einem Container-Layout neu aufgebautes Layout, damit lange Übersetzungen nicht abgeschnitten werden können
- ✓ RTL-Testliste für kurzen Text, langen Text, ein schmales Fenster, eine RTL-Kultur und die Einschränkung der DataGridView

Warum es so aufgebaut ist

GlobalDesk kann die Sprache bereits im laufenden Betrieb wechseln: Die Auswahl setzt `Application.CurrentCulture`, der `Application.CultureChanged` -Handler wendet jede ressourcengestützte Beschriftung erneut an, und der Vorschaubereich formatiert seine Beispiele neu. Dann öffnet ein Tester das Dashboard auf Arabisch. Jedes Wort ist übersetzt und jedes einzelne steht am falschen Platz: Labels schmiegen sich an den linken Rand ihrer Felder, die Navigationsspalte sitzt dort, wo das Auge eines arabischen Lesers zuletzt ankommt, und die langen deutschen Beschriftungen aus dem vorigen Modul schieben schon jetzt eine Schaltfläche aus einem Panel mit fester Breite heraus.

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
<code>RightToLeft</code>	Eine Control-Eigenschaft mit den Werten <code>Inherit</code> , <code>No</code> und <code>Yes</code> , die einem Control sagt, seinen eigenen Text und die Richtung seiner internen Elemente anzupassen.
<code>RightToLeft.Inherit</code>	Die normale Wahl für ein untergeordnetes Control: Es folgt seinem Elternteil, sodass der ganze Bildschirm die Richtung aus einer einzigen Entscheidung an der Spitze ändert.
<code>RightToLeftLayout</code>	Eine andere Eigenschaft, die bei aktivem RTL die waagerechte Position der direkten Kinder eines Containers spiegelt, sodass die visuelle Anordnung umklappt, während Ihr logisches Layout so bleibt, wie entworfen.
Die Anwendungseinstellung <code>rightToLeft</code>	Ein Eintrag in <code>Default.json</code> ; auf "auto" gesetzt lässt sie Wisej.NET das RTL-Verhalten aus der Kultur der Sitzung ableiten statt aus Kulturnamen-Prüfungen, die über Ihre Bildschirme verstreut sind.
<code>Application.RightToLeft</code>	Dieselbe Entscheidung im Code ausgedrückt, für Sitzungen, die bewusst umgeschaltet werden müssen statt über die Kulturerkennung.
Textausdehnung	Eine übersetzte Beschriftung hat selten die Länge der englischen, und deshalb brechen feste Pixelpositionen und von Hand ausgemessene Panelbreiten in einem lokalisierten Build als Erstes.
Dokumentierte RTL-Grenzen	Wisej.NET dokumentiert, dass <code>DataGridView</code> , <code>ListView</code> und <code>PropertyGrid</code> RTL nicht vollständig unterstützen, deshalb muss ein Testplan sie benennen, statt anzunehmen, dass sie spiegeln.

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

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 GlobalDesk 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 – „Rechts-nach-links-Sprachen und lokalisierungssichere Layouts“ – des LearnWisej-Kurses „Lokalisierung 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/localization-in-wisej-net/specs/module-05.md>

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

1. Projekt „GlobalDesk“, 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: Machen Sie GlobalDesk fit für eine Kultur, die von rechts nach links geschrieben wird. Nehmen Sie Arabisch in die Sprachauswahl auf und setzen Sie `rightToLeft` in `Default.json` auf `auto`, lassen Sie die untergeordneten Controls auf `RightToLeft.Inherit` und setzen Sie `RightToLeftLayout` auf dem Dashboard und dem Kundeneditor, damit deren Kinder gespiegelt werden. Lassen Sie genau ein Control von links nach rechts – das Feld für den Kundencode – und begründen Sie in einem Kommentar, warum dieser Bezeichner nicht umgedreht werden darf. Bauen Sie den fest positionierten Teil des Layouts mit Docking, Anchoring und einem Container-Layout neu auf, damit eine lange Übersetzung nicht abgeschnitten werden kann, fügen Sie eine pseudo-lokalisierte Ressourcendatei hinzu, deren Werte verlängert und in Klammern gesetzt sind, und halten Sie fest, was die Kontakt-DataGridView unter RTL tatsächlich tut, statt anzunehmen, dass sie spiegelt. Schließen Sie mit einer Layout-Testliste aus fünf Fällen ab, die Sie auf jedem Bildschirm erneut durchgehen können.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Eintrag für Arabisch in der Sprachauswahl mit `rightToLeft` auf `auto` in `Default.json`
 - Dashboard und Kundeneditor mit `RightToLeftLayout`, deren Kinder auf `Inherit` belassen sind
 - Ein bewusst von links nach rechts laufendes Control mit einem Kommentar, der die Ausnahme erklärt
 - Mit Docking, Anchoring und einem Container-Layout neu aufgebautes Layout, damit lange Übersetzungen nicht abgeschnitten werden können
 - RTL-Testliste für kurzen Text, langen Text, ein schmales Fenster, eine RTL-Kultur und die Einschränkung der DataGridView
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:

- Ein Kollege setzt `RightToLeft.Yes` auf zwanzig Controls, und der arabische Bildschirm sieht richtig aus, der deutsche ist jedoch unverändert. Was hat er tatsächlich konfiguriert, und welche einzelne Eigenschaft hätte die Arbeit für den ganzen Container erledigt?
- Ihr pseudo-lokalisierter Durchgang zeigt drei Beschriftungen weiterhin auf schlichtem Englisch. Nichts wurde abgeschnitten und nichts hat eine Ausnahme ausgelöst. Was sagt Ihnen das über diese drei Beschriftungen, und in welchem früheren Modul klafft eine Lücke?
- Das Kontakt-Grid spiegelt nicht so, wie der Rest des Dashboards es tut. Welche Ihrer Möglichkeiten – es hinnehmen und dokumentieren, das Control ersetzen oder die Spiegelung von Hand schreiben – können Sie gegenüber einem Reviewer vertreten, und was müssten Sie prüfen, bevor Sie eine davon zusagen?

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:

- ✓ Ein Kollege setzt `RightToLeft.Yes` auf zwanzig Controls, und der arabische Bildschirm sieht richtig aus, der deutsche ist jedoch unverändert. Was hat er tatsächlich konfiguriert, und welche einzelne Eigenschaft hätte die Arbeit für den ganzen Container erledigt?
- ✓ Ihr pseudo-lokalisierter Durchgang zeigt drei Beschriftungen weiterhin auf schlichtem Englisch. Nichts wurde abgeschnitten und nichts hat eine Ausnahme ausgelöst. Was sagt Ihnen das über diese drei Beschriftungen, und in welchem früheren Modul klafft eine Lücke?
- ✓ Das Kontakt-Grid spiegelt nicht so, wie der Rest des Dashboards es tut. Welche Ihrer Möglichkeiten — es hinnehmen und dokumentieren, das Control ersetzen oder die Spiegelung von Hand schreiben — können Sie gegenüber einem Reviewer vertreten, und was müssten Sie prüfen, bevor Sie eine davon zusagen?

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.

- ✓ **Machen Sie es fit für einen schlechten Tag.** *"Untersuche dieses GlobalDesk-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 GlobalDesk-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/localization-in-wisej-net/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.