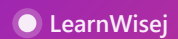


## KI-Coding-Übung

Erstellen Sie das Beispielprojekt EnterpriseOps 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.



### **Dieses Beispiel tippen Sie nicht ab. Sie steuern es.**

Kaum jemand schreibt Wisej.NET-Code heute noch in einer leeren Datei: Ein Assistent liefert den ersten Entwurf, und der Entwickler entscheidet, ob er stimmt. Das funktioniert aber nur, wenn Sie den Stoff beherrschen – und genau den hat Ihnen der Rest dieses Moduls vermittelt. Hier geben Sie dieses Wissen an ChatGPT oder Claude weiter, erhalten das Beispielprojekt des Moduls als lauffähige Visual-Studio-Projektmappe zurück und prüfen es so, wie Sie den Pull-Request eines Kollegen prüfen würden.

## Worum Sie bitten

Legen Sie die Basis der Lösung EnterpriseOps Command Center an. Fügen Sie Projekte oder Ordner für UI, gemeinsame Controls, Domäne, Anwendungsservices, Datenzugriff, Integrationen, Sicherheit, Diagnose, Ressourcen und Deployment hinzu. Ergänzen Sie einen Ordner für Architecture Decision Records und schreiben Sie ADR-001 zur gewählten Lösungsstruktur.

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

- ✓ Screenshot oder Baumansicht der Lösungs- und Ordnerstruktur
- ✓ ADR-001 mit der Entscheidung zur Lösungsstruktur
- ✓ Seite mit den Coding-Standards des Teams
- ✓ Beispiel für die Benennung eines Referenzbildschirms und für eine Service-Kapselung
- ✓ Auf den ersten Bildschirm angewendete Code-Review-Checkliste

## Warum es so aufgebaut ist

Bei fortgeschrittener Arbeit mit Wisej.NET geht es nicht mehr darum, ob sich ein Bildschirm bauen lässt. Es geht darum, ob ein Team eine große Anwendung über Jahre weiterentwickeln kann, ohne dass Produktivität zu versteckten technischen Schulden wird.

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                                                                                                                                                            |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enterprise-Lösungsstruktur       | Teilt die Lösung in Projekte für UI, Anwendungsservices, Domäne und Infrastruktur auf, damit Bildschirme schlank bleiben und Geschäftsregeln genau einen Ort haben.             |
| Architecture Decision Records    | Kurze Dokumente, die Kontext, Optionen, Entscheidung und Konsequenzen einer Designentscheidung festhalten, damit das Team sie später prüfen und neu bewerten kann.              |
| designerfreundliche Abgrenzungen | Halten Forms und Pages im Designer bearbeitbar, während Workflow, Validierung und Datenzugriff in Services liegen, die der Designer nie anfasst.                                |
| Review-Gates                     | Festgelegte Prüfpunkte, an denen Code, Architektur und Sicherheit geprüft werden, bevor ein Feature gemergt wird – so wird Qualität durchgesetzt, statt nur erhofft.            |
| Teamkonventionen                 | Gemeinsame Regeln für Benennung, Ordner und Service-Registrierung, durch die jeder Bildschirm jedem Entwickler im Team vertraut vorkommt.                                       |
| Kontrolle technischer Schulden   | Praktiken wie Schuldenprotokolle, Refactoring-Budgets und Regeln für Deprecation, die Abkürzungen sichtbar und eingeplant halten, statt sie in Vergessenheit geraten zu lassen. |

## 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 1 geht. Sie können [module-01.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 EnterpriseOps 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 1 – „Enterprise-Architektur mit Wisej.NET, technische Führung und Governance“ – des LearnWisej-Kurses „Enterprise Wisej.NET: von der Architektur in die Cloud“ 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/azure-docker/specs/module-01.md>

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

1. Projekt „EnterpriseOps“, 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: Legen Sie die Basis der Lösung EnterpriseOps Command Center an. Fügen Sie Projekte oder Ordner für UI, gemeinsame Controls, Domäne, Anwendungsservices, Datenzugriff, Integrationen, Sicherheit, Diagnose, Ressourcen und Deployment hinzu. Ergänzen Sie einen Ordner für Architecture Decision Records und schreiben Sie ADR-001 zur gewählten Lösungsstruktur.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
  - Screenshot oder Baumansicht der Lösungs- und Ordnerstruktur
  - ADR-001 mit der Entscheidung zur Lösungsstruktur
  - Seite mit den Coding-Standards des Teams
  - Beispiel für die Benennung eines Referenzbildschirms und für eine Service-Kapselung
  - Auf den ersten Bildschirm angewendete Code-Review-Checkliste
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:

- Findet ein neuer Senior-Entwickler den Workflow-Service zu einem Bildschirm in weniger als einer Minute?
- Kann das Team erklären, warum eine Grenze existiert?
- Ist der Designer nach der Aufteilung der Architektur noch nutzbar?

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:

- ✓ Findet ein neuer Senior-Entwickler den Workflow-Service zu einem Bildschirm in weniger als einer Minute?
- ✓ Kann das Team erklären, warum eine Grenze existiert?
- ✓ Ist der Designer nach der Aufteilung der Architektur noch nutzbar?

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 1 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 1 auf GitHub öffnen](#)

Oder klonen Sie alle Beispiele mit `git clone https://github.com/iceteagroup/LearnWisej-Samples` und öffnen Sie [Enterprise Wisej.NET Course/Module 1](#).



### **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.

- ✓ **Prüfen Sie es wie einen Pull-Request.** *"Führ mich durch diesen EnterpriseOps-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 EnterpriseOps-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 EnterpriseOps-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.