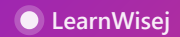


KI-Coding-Übung

Erstellen Sie das Beispielprojekt FieldOps 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.



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

Schließen Sie das Abschlussprojekt FieldOps ab: Signieren Sie den Windows-Installer sowie die Android- und iOS-Pakete, bereiten Sie eine Unternehmensverteilung für die mobilen Shells und einen Entwurf für den Store-Eintrag vor, definieren Sie eine Kompatibilitätsregel zwischen Server- und Shell-Versionen mit einer Prüfung der Mindestversion beim Start, ergänzen Sie Absturzberichte und ein Nutzungs-Event für jede Gerätefunktion, und liefern Sie eine Release-Checkliste, die alle vier Bereitstellungsmodelle abdeckt.

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

- ✓ Signierte Pakete für Windows, Android und iOS
- ✓ Einrichtung der Unternehmensverteilung und Entwurf des Store-Eintrags
- ✓ Kompatibilitätsprüfung zwischen Server- und Shell-Version
- ✓ Absturz- und Nutzungstelemetrie
- ✓ Release-Checkliste für alle Bereitstellungsmodelle

Warum es so aufgebaut ist

FieldOps beendet diesen Kurs als ein Server mit vier Zugangswegen: ein Browser-Tab, eine installierte PWA, eine Windows-Desktop-Shell und eine mobile Shell unter Android und iOS. Die drei paketierte Varianten sind Dateien, deren Installation ein Betriebssystem verweigert, sofern sie nicht von einer vertrauenswürdigen Identität signiert wurden, und jede nimmt einen anderen Kanal mit eigener Prüfwarteschlange und eigenem Rollout-Tempo. Den Server können Sie noch vor der Mittagspause neu deployen; der iOS-Build, den ein Techniker in der Hand hält, liegt womöglich drei Wochen zurück – und muss trotzdem funktionieren.

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
Authenticode-Signatur	Die Signatur, die ein Codesignaturzertifikat auf einen Windows-Installer oder ein MSIX anbringt; sie belegt, dass die Datei nach dem Signieren nicht verändert wurde, und nennt einen geprüften Herausgeber.
Zeitstempel der Gegensignatur	Ein vertrauenswürdiger Zeitstempel, der beim Signieren festgehalten wird, damit der Installer auch nach Ablauf des Signaturzertifikats gültig bleibt.
Android-Keystore und Upload-Schlüssel	Der Schlüssel, der einen Android-Build signiert; mit Play App Signing signieren Sie mit einem Upload-Schlüssel, und Google signiert mit dem App-Signaturschlüssel, den es für dieses Paket verwahrt, neu.
Apple-Distributionszertifikat und Provisioning-Profil	Das Zertifikat identifiziert Ihr Team; das Profil verknüpft eine App-ID, dieses Zertifikat, die Entitlements und – außerhalb des App Store – die zugelassenen Geräte.
Unternehmensverteilung	Einen privaten Build an die Geräte einer einzelnen Organisation ausliefern – Managed Google Play, benutzerdefinierte Apps über Apple Business Manager, ein per Intune verteiltes MSIX –, statt ihn öffentlich zu veröffentlichen.
Minimale Shell-Version	Der älteste Shell-Build pro Plattform, den der aktuelle Server noch unterstützt; alles darunter wird beim Verbinden gestoppt, bevor es an irgendwelche Daten gelangt.
Nutzungs-Event für Gerätefunktionen	Ein Datensatz, der bei jeder Ausführung einer Shell-Fähigkeit geschrieben wird – Kamera, Unterschrift, Standort, Push, Leeren der Offline-Warteschlange – mit dem Ergebnis sowie Plattform und Version der Shell.

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 6 geht. Sie können [module-06.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 FieldOps 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 6 – „Signierung, Verteilung, Updates und das Abschlussprojekt FieldOps“ – des LearnWisej-Kurses „Hybride und native Paketierung“ 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/hybrid-native-packaging/specs/module-06.md>

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

1. Projekt „FieldOps“, 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: Schließen Sie das Abschlussprojekt FieldOps ab: Signieren Sie den Windows-Installer sowie die Android- und iOS-Pakete, bereiten Sie eine Unternehmensverteilung für die mobilen Shells und einen Entwurf für den Store-Eintrag vor, definieren Sie eine Kompatibilitätsregel zwischen Server- und Shell-Versionen mit einer Prüfung der Mindestversion beim Start, ergänzen Sie Absturzberichte und ein Nutzungs-Event für jede Gerätefunktion, und liefern Sie eine Release-Checkliste, die alle vier Bereitstellungsmodelle abdeckt.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Signierte Pakete für Windows, Android und iOS
 - Einrichtung der Unternehmensverteilung und Entwurf des Store-Eintrags
 - Kompatibilitätsprüfung zwischen Server- und Shell-Version
 - Absturz- und Nutzungstelemetrie
 - Release-Checkliste für alle Bereitstellungsmodelle
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 Android-Shell eines Technikers ist sechs Wochen alt, und der Server wurde seitdem viermal ausgeliefert. Welche dieser vier Serveränderungen hätte die minimale Shell-Version anheben müssen, und woran hätten Sie das zum jeweiligen Zeitpunkt erkannt?
- Der Windows-Installer wurde signiert, aber ohne Zeitstempel, und das Zertifikat ist letzten Monat abgelaufen. Was passiert mit bestehenden Installationen, was mit neuen Downloads, und was ist der kürzeste Weg zurück zu einem vertrauenswürdigen Installer?
- Das Nutzungs-Event der Kamera zeigt einen starken Anstieg des Browser-Fallback-Wegs – ausschließlich unter iOS. Nennen Sie drei Erklärungen, die zu den Daten passen, und sagen Sie, welches zusätzliche Feld im Event sie voneinander unterscheiden würde.

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 Android-Shell eines Technikers ist sechs Wochen alt, und der Server wurde seitdem viermal ausgeliefert. Welche dieser vier Serveränderungen hätte die minimale Shell-Version anheben müssen, und woran hätten Sie das zum jeweiligen Zeitpunkt erkannt?
- ✓ Der Windows-Installer wurde signiert, aber ohne Zeitstempel, und das Zertifikat ist letzten Monat abgelaufen. Was passiert mit bestehenden Installationen, was mit neuen Downloads, und was ist der kürzeste Weg zurück zu einem vertrauenswürdigen Installer?
- ✓ Das Nutzungs-Event der Kamera zeigt einen starken Anstieg des Browser-Fallback-Wegs – ausschließlich unter iOS. Nennen Sie drei Erklärungen, die zu den Daten passen, und sagen Sie, welches zusätzliche Feld im Event sie voneinander unterscheiden würde.

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 FieldOps-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 FieldOps-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 FieldOps-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.