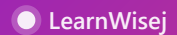


KI-Coding-Übung

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.



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

Alle anderen Abschnitte dieses Moduls erklären, wie „Geräte-APIs über die Hybrid-Bridge“ 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

Ergänzen Sie FieldOps um Gerätefunktionen: eine Aktion Take Photo, die in der nativen Shell die Gerätekamera nutzt und im Browser auf ein Upload-Control zurückfällt, eine Unterschriftserfassung, die als Bild hochgeladen wird, eine Aktion Navigate, die den Gerätestandort liest und die Karten-App öffnet, eine Push-Benachrichtigung bei der Zuweisung eines Arbeitsauftrags und einen Capability-Check, der jede Aktion pro Shell ein- oder ausblendet.

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

- ✓ Kameraaufnahme mit Upload-Fallback im Browser
- ✓ Unterschriftserfassung und Upload
- ✓ Geolocation und Übergabe an die Karten-App
- ✓ Push-Benachrichtigung bei Zuweisung
- ✓ Capability-Check-Matrix über alle Shells

Warum es so aufgebaut ist

Ein FieldOps-Techniker fotografiert ein korrodiertes Ventil, lässt sich eine Unterschrift auf dem Display geben und fährt zur nächsten Adresse – während der Disponent, der den Auftrag vergeben hat, denselben Arbeitsauftrag in einem Desktop-Browser verfolgt. Jedes Control in FieldOps ist ein C#-Objekt auf Ihrem Server, und der kommt weder an die Kamera noch an den GPS-Chip noch an den Benachrichtigungsbereich heran. Dieses Modul behandelt den Hin- und Rückweg, der diese Lücke schließt – und was der Bildschirm dort tut, wo es diesen Weg nicht gibt.

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
Geräte-Bridge	Der Nachrichtenkanal, den eine hybride Shell der gehosteten Seite bereitstellt: Ihr Server bittet die Shell, eine Geräteoperation auszuführen, und die Shell antwortet mit einer Nachricht, die die App wieder in ein serverseitiges Event verwandelt.
Host-Objekt	Das Objekt, das eine WebView in die gehostete Seite injiziert, damit JavaScript dort nativen Code aufrufen kann. Serverseitiger C#-Code erreicht es nur über das JavaScript-Interop der App.
Capability-Deskriptor	Die Funktionen, die eine Shell beim Start einer Session meldet – Kamera, Standort, Benachrichtigungen, Biometrie –, zwischengespeichert und abgefragt, bevor irgendeine Geräteaktion angezeigt wird.
Progressive Enhancement	Jeden Bildschirm so entwerfen, dass der Weg im einfachen Browser eine vollständige Funktion ist und der native Weg eine bessere Version davon – nie der einzige, der funktioniert.
Just-in-Time-Berechtigung	Die Plattform genau in dem Moment um Zugriff auf Kamera, Standort oder Benachrichtigungen bitten, in dem der Benutzer den Button drückt, der ihn braucht – direkt nach einem Satz, der erklärt, warum.
Bridge-Ergebnis-Event	Die asynchrone Antwort – Fotodaten, Koordinaten oder eine Ablehnung –, die mit einer späteren Anfrage ankommt als der Klick, der sie angefordert hat, und daher einen eigenen Push braucht.
Speicherung außerhalb des Web-Roots	Aufgenommene Fotos und Unterschriften dort ablegen, wo keine URL auf sie verweist, und sie über einen Handler ausliefern, der prüft, ob diese Session den Arbeitsauftrag sehen darf.

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 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 5 – „Geräte-APIs über die Hybrid-Bridge“ – 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-05.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: Ergänzen Sie FieldOps um Gerätefunktionen: eine Aktion Take Photo, die in der nativen Shell die Gerätekamera nutzt und im Browser auf ein Upload-Control zurückfällt, eine Unterschriftserfassung, die als Bild hochgeladen wird, eine Aktion Navigate, die den Gerätestandort liest und die Karten-App öffnet, eine Push-Benachrichtigung bei der Zuweisung eines Arbeitsauftrags und einen Capability-Check, der jede Aktion pro Shell ein- oder ausblendet.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Kameraaufnahme mit Upload-Fallback im Browser
 - Unterschriftserfassung und Upload
 - Geolocation und Übergabe an die Karten-App
 - Push-Benachrichtigung bei Zuweisung
 - Capability-Check-Matrix über alle Shells
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:

- Das Kameraergebnis kommt vierzig Sekunden nach dem Klick an, der es angefordert hat. Erklären Sie, mit welcher Anfrage es ankommt, welchen Session-Zustand es vorfinden muss, um zu wissen, was mit den Bytes zu tun ist, und warum die Bildanzeige ohne expliziten Push nicht im Browser erscheint.
- Ein Techniker hat die Standortberechtigung vor drei Wochen auf seinem eigenen Smartphone dauerhaft verweigert. Beschreiben Sie genau, was ein Druck auf Navigate heute bewirken soll, und nennen Sie die eine Sache, die die App darauf niemals tun darf.
- Take Photo funktioniert in der mobilen Shell, und der Upload-Fallback funktioniert in Chrome auf dem Desktop. Nennen Sie einen Punkt, der sich für den Benutzer zwischen diesen beiden Wegen wirklich unterscheidet, und begründen Sie, ob dieser Unterschied für einen Außendiensttechniker akzeptabel ist.

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:

- ✓ Das Kameraergebnis kommt vierzig Sekunden nach dem Klick an, der es angefordert hat. Erklären Sie, mit welcher Anfrage es ankommt, welchen Session-Zustand es vorfinden muss, um zu wissen, was mit den Bytes zu tun ist, und warum die Bildanzeige ohne expliziten Push nicht im Browser erscheint.
- ✓ Ein Techniker hat die Standortberechtigung vor drei Wochen auf seinem eigenen Smartphone dauerhaft verweigert. Beschreiben Sie genau, was ein Druck auf Navigate heute bewirken soll, und nennen Sie die eine Sache, die die App darauf niemals tun darf.
- ✓ Take Photo funktioniert in der mobilen Shell, und der Upload-Fallback funktioniert in Chrome auf dem Desktop. Nennen Sie einen Punkt, der sich für den Benutzer zwischen diesen beiden Wegen wirklich unterscheidet, und begründen Sie, ob dieser Unterschied für einen Außendiensttechniker akzeptabel ist.

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.

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



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.