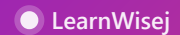


KI-Coding-Übung

Erstellen Sie das Beispielprojekt AccessOps 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 „Schutz von Daten, Uploads, Downloads und Handlern“ 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

Sichern Sie die Dokumentenfunktion von AccessOps ab: Uploads, begrenzt nach Größe und Typ und mit Inhaltsprüfung, außerhalb des Web-Roots unter einem zufälligen Namen gespeichert und mit der zuständigen Abteilung verknüpft; Downloads, die über die Autorisierung von DocumentService und Application.Download laufen; ein eigener Handler für einen signierten Freigabelink, der nach einer Stunde abläuft; und ein Review, das bestätigt, dass jede Abfrage parametrisiert ist und jeder vom Benutzer gelieferte String vor der Anzeige encodiert wird.

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

- ✓ Validierte Upload-Pipeline mit Quarantäne-Ordner
- ✓ Autorisierter Download über den Service
- ✓ Handler für signierte, ablaufende Freigabelinks
- ✓ Review von parametrisierten Abfragen und Ausgabe-Encoding
- ✓ Test mit einer gefälschten Download-Anfrage

Warum es so aufgebaut ist

Der Bereich für vertrauliche Dokumente ist der Teil von AccessOps, auf den es ein Angreifer wirklich abgesehen hat: Gehaltsschreiben, Disziplinarakten, unterschriebene Verträge, Unterlagen für die Geschäftsleitung. Jeder andere Bildschirm verschiebt Zeilen zwischen Grids; dieser hier bewegt Bytes über die Grenze zu einem Browser, den Sie nicht kontrollieren, und wieder zurück. Identität und Berechtigungen wurden in den vorigen Modulen geklärt, und in diesem Modul geht es um die Stellen, an denen eine Datei, eine URL oder ein String daran vorbeiläuft.

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
vom Server gestreamter Download	Der Aufruf von Application.Download im Servercode streamt eine Datei während eines bereits autorisierten Events an den Browser, sodass es keine Adresse gibt, die man als Lesezeichen speichern, weitergeben oder erraten könnte.
Preisgabe statischer Dateien	Alles unterhalb des Web-Roots wird ausgeliefert, bevor eine einzige Zeile Ihres Codes läuft; die Berechtigung für die Datei lautet also genau „wer die URL kennt“.
Uploaded-Event	Das Upload -Control löst es auf dem Server mit den eingehenden Datei-Streams aus – der einzige Ort, an dem eine Datei geprüft werden kann, bevor sie in Ihr System gelangt.
Prüfung der Inhaltssignatur	Die ersten Bytes einer Datei werden mit dem bekannten Header des Formats verglichen, denn Dateiendung und angegebener Content-Type stammen vom Client.
Quarantäne-Speicher	Der angenommene Upload wird außerhalb des Web-Roots unter einem generierten Namen gehalten, bis der Scan unauffällig ist, und erst dann mit der zuständigen Abteilung verknüpft.
eigener HTTP-Handler	Eine bei der Anwendung registrierte Serverklasse, die rohe Requests außerhalb des Event-Flusses erhält; sie hat keine Session-Identität und muss daher alles selbst authentifizieren und validieren.
signierter, ablaufender Link	Eine URL mit Dokument-ID, Ablaufzeit und einem HMAC, der mit einem serverseitig gehaltenen Schlüssel über beide Werte berechnet wird, sodass sich die Werte nicht bearbeiten lassen und der Link von selbst ungültig wird.

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 AccessOps 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 – „Schutz von Daten, Uploads, Downloads und Handlern“ – des LearnWisej-Kurses „Sicherheits- und Authentifizierungsmuster“ 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/security-and-auth-patterns/specs/module-05.md>

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

1. Projekt „AccessOps“, 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: Sichern Sie die Dokumentenfunktion von AccessOps ab: Uploads, begrenzt nach Größe und Typ und mit Inhaltsprüfung, außerhalb des Web-Roots unter einem zufälligen Namen gespeichert und mit der zuständigen Abteilung verknüpft; Downloads, die über die Autorisierung von DocumentService und `Application.Download` laufen; ein eigener Handler für einen signierten Freigabelink, der nach einer Stunde abläuft; und ein Review, das bestätigt, dass jede Abfrage parametrisiert ist und jeder vom Benutzer gelieferte String vor der Anzeige encodiert wird.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Validierte Upload-Pipeline mit Quarantäne-Ordner
 - Autorisierter Download über den Service
 - Handler für signierte, ablaufende Freigabelinks
 - Review von parametrisierten Abfragen und Ausgabe-Encoding
 - Test mit einer gefälschten Download-Anfrage
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:

- Ihr Upload-Handler prüft die Endung gegen eine Allow-List, und die Datei besteht die Prüfung. Erklären Sie, was ein Angreifer trotzdem noch senden kann und was die Signaturprüfung hinzufügt, das die Prüfung der Endung nicht leisten kann.
- Eine Managerin leitet einen Freigabelink vor dessen Ablauf an jemanden außerhalb des Unternehmens weiter. Welche Teile Ihres Designs begrenzen den Schaden, und was müssten Sie ergänzen, um genau diesen einen Link zu widerrufen, ohne den Schlüssel für alle zu ändern?
- Ihr Download-Button ist für Benutzer ohne Leseberechtigung ausgeblendet, und der Test mit der gefälschten Anfrage besteht trotzdem. Erklären Sie genau, welche Codezeile dafür gesorgt hat, dass er besteht, und warum der ausgeblendete Button nichts damit zu tun hatte.

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:

- ✓ Ihr Upload-Handler prüft die Endung gegen eine Allow-List, und die Datei besteht die Prüfung. Erklären Sie, was ein Angreifer trotzdem noch senden kann und was die Signaturprüfung hinzufügt, das die Prüfung der Endung nicht leisten kann.
- ✓ Eine Managerin leitet einen Freigabelink vor dessen Ablauf an jemanden außerhalb des Unternehmens weiter. Welche Teile Ihres Designs begrenzen den Schaden, und was müssten Sie ergänzen, um genau diesen einen Link zu widerrufen, ohne den Schlüssel für alle zu ändern?
- ✓ Ihr Download-Button ist für Benutzer ohne Leseberechtigung ausgeblendet, und der Test mit der gefälschten Anfrage besteht trotzdem. Erklären Sie genau, welche Codezeile dafür gesorgt hat, dass er besteht, und warum der ausgeblendete Button nichts damit zu tun hatte.

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.

- ✓ **Ergänzen Sie den zweiten Fall.** *"Erweitere dieses AccessOps-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/security-and-auth-patterns/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."

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



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.