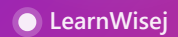


Hybride, PWA-, Offline- und gerätebewusste Wisej.NET-Anwendungen

Lektionsleitfaden · Lab- und Prüfungsleitfaden



Lektionsleitfaden

Online- und Offlinemodus, lokale Daten, Geräte-APIs, Muster für Synchronisationskonflikte, Außendienstszenarios. Was das für ein Wisej.NET-Team auf Expertenniveau bedeutet – und wie Sie dieses Modul lesen.



Fokus: Produktivbetrieb

Online- und Offlinemodus, lokale Daten, Geräte-APIs, Muster für Synchronisationskonflikte, Außendienstszenarios.

1. Das Problem im Produktivbetrieb

Manche Enterprise-Workflows erfordern den Betrieb auf Smartphones, Tablets, im Außendienst, an Kiosk-Terminals oder offline. Erfahrene Wisej.NET-Entwickler müssen entscheiden, wann Hybrid- oder PWA-Muster passen und wie sich ein einheitliches Anwendungsmodell erhalten lässt, ohne die Realität der Geräte zu ignorieren.

2. Zentrale Konzepte

Konzept	Warum es wichtig ist
Wisej.NET Hybrid	Die Unterstützung des Frameworks dafür, dieselbe Anwendung im Browser, als Desktop- oder Mobile-App und mit nativem Gerätezugriff auszuführen.
Verbundener Modus	Das Standardmodell, bei dem die App auf dem Server läuft und das Gerät ein Client ist. Es ist am einfachsten, erfordert aber eine Netzwerkverbindung.
Offline-/lokale App	Die Anwendung läuft lokal auf dem Gerät und funktioniert so auch ohne Verbindung – um den Preis, lokale Daten und die Synchronisation verwalten zu müssen.
Geräteabstraktion	Kamera, Dateisystem, GPS und andere Gerätefunktionen liegen hinter Service-Schnittstellen, damit Bildschirme nicht von einer bestimmten Plattform abhängen.
Lokales SQLite	Eine leichtgewichtige eingebettete Datenbank auf dem Gerät, die im Offlinebetrieb die Arbeitsdaten speichert und die Synchronisations-Warteschlange speist.
Sync-Warteschlange	Eine dauerhafte Liste lokaler Änderungen, die darauf warten, an den Server gesendet zu werden, sobald die Verbindung zurück ist – der Reihe nach verarbeitet und sicher wiederholbar.
Konfliktauflösung	Die Regeln dafür, was passiert, wenn derselbe Datensatz während der Offlinephase lokal und auf dem Server geändert wurde – festgelegt vom Produkt, nicht vom Zufall.
PWA-Shell	Das Manifest der Progressive Web App und der Service Worker, mit denen sich die Anwendung auf einem Gerät installieren und ihre Shell ohne Netzwerk laden lässt.

3. Der rote Faden der Lektion

Offline ist eine Produktentscheidung

Offlineunterstützung ist kein Häkchen in einer Checkbox. Sie verändert Identität, Aktualität der Autorisierung, Datenspeicherung, Konfliktbehandlung, den Zeitpunkt des Audits, Supportprozesse und Deployment. Erfahrene Entwickler beschränken den Offlineumfang auf Workflows, die ihn wirklich brauchen.

Gerätecode gehört hinter Services

Kamera, Scanner, Zwischenablage, Benachrichtigungen und Netzwerkstatus sollten über Services bereitgestellt werden. Der Rest der Anwendung sollte fachlich verständliche Schnittstellen aufrufen, statt gerätespezifischen Code über die Bildschirme zu streuen.

Vier Anwendungsformen

Mit Wisej.NET Hybrid kann ein Anwendungsmodell in vier Formen laufen. Im verbundenen Modus läuft die Anwendung auf einem Server, und der Browser oder eine native Shell um einen Browser ist ein schlanker Client; das Gerät braucht für jede Aktion eine Netzwerkverbindung. Im Remote-Modus verbindet sich eine native Shell auf dem Gerät mit derselben Serveranwendung, erhält über die Shell aber Zugriff auf Gerätefunktionen. Im lokalen Modus wird die Anwendung selbst mit dem Gerät ausgeliefert und führt

ihren Serverprozess auf dem Gerät aus, sodass sie ohne Netzwerk funktioniert. Der Offlinemodus ist der lokale Modus mit lokalen Daten und einer Synchronisationsgrenze zum zentralen System.

Jeder Schritt auf dieser Linie tauscht Einfachheit gegen Unabhängigkeit. Der verbundene Modus hat eine einzige Kopie der Daten und einen einzigen Ort, an dem Regeln durchgesetzt werden. Der Offlinemodus hat eine Kopie auf jedem Gerät, und die Regeln müssen auf jedem davon gelten. Die richtige Form ist die am wenigsten unabhängige, die den Workflow noch erfüllt. Ein Disponent am Schreibtisch bleibt verbunden. Ein Techniker in einem Keller ohne Empfang braucht eine Offline-Warteschlange für die Arbeit, die er dort abschließt.

Legen Sie die Form in der Architekturnotiz fest, bevor Sie Bildschirme entwerfen, denn sie entscheidet, wo Identität, Daten und Berechtigungen liegen.

Offline-Workflows und Synchronisationsgrenzen

Ein offlinefähiger Workflow wird darum herum entworfen, was das Gerät speichern und was es ändern darf. Der Außendienstmodus im Lab speichert die dem Techniker zugewiesenen Arbeitsaufträge in einem lokalen SQLite-Speicher zwischen, und nur diese. Wird ein Arbeitsauftrag offline abgeschlossen, wird der Datensatz nicht aktualisiert; stattdessen wird ein `OfflineCommand` mit dem expliziten `SyncState` „pending“ an eine lokale Warteschlange angehängt. Der Bildschirm zeigt diesen ausstehenden Zustand an, damit der Techniker weiß, dass der Abschluss noch nicht im Büro angekommen ist.

Die Synchronisationsgrenze ist der Punkt, an dem die Warteschlange gegen den Server abgespielt wird. Jeder Command wird der Reihe nach gesendet, von demselben Serverservice ausgeführt, der auch den Onlinefall behandelt, und als synchronisiert oder im Konflikt markiert. Da es derselbe Serverservice ist, gelten jede Regel, jede Berechtigungsprüfung und jeder Audit-Eintrag für die Offlinearbeit genau so wie für die Onlinearbeit. Das Gerät spricht nie direkt mit der Datenbank.

Begrenzen Sie den Umfang konsequent. Jede zusätzliche Entität, die offline zwischengespeichert wird, bedeutet mehr Daten auf einem Gerät, das verloren gehen kann, mehr Synchronisationsfälle zum Testen und mehr Konflikte zum Auflösen.

Gerätefunktionen hinter Service-Schnittstellen

Kamera, Barcodescanner, GPS, lokale Benachrichtigungen und Netzwerkstatus erreichen die Anwendung über die Hybrid-Shell, und jede dieser Funktionen verhält sich auf einem Smartphone, einem Tablet, einem Kiosk-Terminal und in einem Desktop-Browser anders. Das Lab verbirgt sie alle hinter einer Abstraktion `IDeviceServices`. Ein Bildschirm ruft eine Methode wie „Foto aufnehmen“ oder „Barcode lesen“ auf, und die Implementierung für die aktuelle Form entscheidet, ob das die native Kamera, eine Dateiauswahl oder einen Hinweis bedeutet, dass die Funktion nicht verfügbar ist.

Die Abstraktion zahlt sich dreifach aus. Bildschirme bleiben über alle Formen hinweg identisch, sodass ein Formular Büro und Außendienst bedient. Tests können einen gefälschten Geräteservice einsetzen, der ein bekanntes Foto oder einen bekannten Barcode liefert. Und ein neues Gerät oder eine neue Shell erfordert nur eine neue Implementierung, statt jedes Formular nach plattformspezifischen Aufrufen zu durchsuchen.

Behandeln Sie die Ausgaben des Geräts als nicht vertrauenswürdige Eingaben. Ein gescannter Barcode wird auf dem Server validiert wie jeder andere Wert, und eine GPS-Position ist ein Hinweis für den Benutzer, niemals ein Beweis dafür, wo die Arbeit erledigt wurde.

Konflikte, veraltete Berechtigungen und Offlinesicherheit

Ein Konflikt entsteht, wenn derselbe Datensatz auf dem Gerät und auf dem Server geändert wurde, während beide getrennt waren. Im Fehlerpfad des Labs storniert der Disponent einen Arbeitsauftrag, während der Techniker ihn offline abschließt. Keine der beiden Versionen ist einfach falsch, deshalb zeigt der Bildschirm für Synchronisationskonflikte beide an, speichert beide und lässt einen Menschen entscheiden. Die Regeln für die Auflösung – etwa welche Rollen überschreiben dürfen und ob ein stornierter Auftrag noch abgeschlossen werden kann – sind Produktentscheidungen, die dokumentiert werden, bevor der Code entsteht.

Offline veralteten Berechtigungen. Die Rolle des Technikers wurde vielleicht entfernt, während er nicht verbunden war, und das Gerät hält noch die Berechtigungen von gestern. Die Synchronisationsgrenze fängt das ab, indem sie jeden Command in der Warteschlange auf dem Server gegen die aktuellen Berechtigungen neu bewertet. So werden die wartenden Commands eines Benutzers, dem die Rechte entzogen wurden, abgelehnt und auditert statt angewendet. Außerdem aktualisiert das Gerät bei jedem Wiederverbinden seine zwischengespeicherten Berechtigungen.

- Lokale Daten werden im Ruhezustand verschlüsselt und bei der Abmeldung oder nach einer Sperrfrist gelöscht.
- Zwischengespeicherte Anmeldedaten laufen ab, und ein gestohlenes Gerät kann danach keine Commands mehr abspielen.
- Jeder Offline-Command wird auf dem Server mit seinem ursprünglichen Zeitstempel vom Gerät und seinem Synchronisationszeitpunkt auditert.



Expertenregel für den Produktivbetrieb

Hören Sie nicht auf, sobald der Bildschirm korrekt aussieht. Erklären Sie, wem der Zustand gehört, welche Sicherheitsfolgen es gibt, welche Fehlerpfade existieren und wie sich das System im Produktivbetrieb verhält, bevor Sie das Modul als abgeschlossen betrachten.



Lernziele

Am Ende dieses Moduls können Sie:

- ✓ die Hybrid-Anwendungsformen „verbunden“, „remote“, „lokal“ und „offline“ erklären.
- ✓ offlinefähige Workflows mit lokaler Speicherung und Synchronisationsgrenzen entwerfen.
- ✓ Gerätefunktionen über explizite Service-Abstraktionen nutzen.
- ✓ Synchronisationskonflikte, veraltete Berechtigungen und Sicherheitsabwägungen im Offlinebetrieb behandeln.

Lab- und Prüfungsleitfaden

Was Sie im Praxis-Lab bauen, das vorgeschlagene Vorgehen und die geforderten Ergebnisse.



Praxis-Lab

Entwerfen Sie einen Außendienstmodus für EnterpriseOps und erstellen Sie einen Prototyp dafür. Ergänzen Sie ein gerätebewusstes Layout, einen Offline-Cache für Arbeitsaufträge, eine lokale Warteschlange für Abschlüsse, eine Simulation der Synchronisation beim Wiederverbinden und eine Konflikthanzeige.

Vorgeschlagenes Vorgehen

- ✓ Lesen Sie das Problem im Produktivbetrieb und benennen Sie das Architekturrisiko, bevor Sie die UI anfassen.
- ✓ Legen Sie zuerst die betroffene Grenze für Service, Command, Modell oder Komponente an oder aktualisieren Sie sie.
- ✓ Erstellen Sie den Wisej.NET-Bildschirm oder die Komponente nach den etablierten Projektstandards.
- ✓ Demonstrieren Sie auch einen Fehlerpfad, nicht nur den Happy Path.
- ✓ Halten Sie die Änderungen im Lernprotokoll fest und aktualisieren Sie die zugehörige Checkliste.

Geforderte Ergebnisse

- ✓ Architekturnotiz zu Hybrid- und Offlinebetrieb
- ✓ Abstraktion des Geräteservices
- ✓ Modell der Offline-Warteschlange
- ✓ Bildschirm für Synchronisationskonflikte
- ✓ Usability-Checkliste für den Außendienstmodus

Wiederholungsfragen

- ✓ Welche Daten sind offline zulässig?
- ✓ Wie werden Berechtigungen aktualisiert?
- ✓ Was passiert, wenn lokale Änderungen mit Änderungen auf dem Server kollidieren?



Expertenregel für den Produktivbetrieb

Hören Sie nicht auf, sobald der Bildschirm korrekt aussieht. Erklären Sie, wem der Zustand gehört, welche Sicherheitsfolgen es gibt, welche Fehlerpfade existieren und wie sich das System im Produktivbetrieb verhält, bevor Sie das Modul als abgeschlossen betrachten.