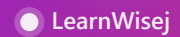


Daten-UX für große Datenmengen, serverseitige Filterung und Stapelverarbeitung

Lektionsleitfaden · Lab- und Prüfungsleitfaden



Lektionsleitfaden

Große Grids, Paging, Filtern, Sortieren, Stapelbearbeitung, Rückmeldung an den Benutzer. Was das für ein Wisej.NET-Team auf Expertenniveau bedeutet – und wie Sie dieses Modul lesen.



Fokus: Produktivbetrieb

Große Grids, Paging, Filtern, Sortieren, Stapelbearbeitung, Rückmeldung an den Benutzer.

1. Das Problem im Produktivbetrieb

Enterprise-Benutzer erwarten schnelle Grids und zuverlässige Massenaktionen. Alles in den Speicher zu laden oder blind neu zu binden, funktioniert in Demos, scheitert aber bei realistischen Datenmengen.

2. Zentrale Konzepte

Konzept	Warum es wichtig ist
serverseitiges Paging	Die Datenbank liefert nur die angeforderte Seite von Zeilen, sodass ein Grid Millionen von Datensätzen durchblättern kann, ohne sie in den Speicher oder den Browser zu laden.
Suchprojektion	Ein schlankes Lesemodell, das nur die Spalten enthält, die ein Suchbildschirm anzeigt, statt vollständiger Entitäten mit sämtlichen Navigationseigenschaften.
Grid-Zustand	Aktuelle Seite, Sortierung, Filter und Auswahl eines Grids, auf dem Server gehalten, damit sie einen Refresh überstehen und sich vorhersagbar wiederherstellen lassen.
virtualisiertes Denken	Jede Liste und jedes Grid so entwerfen, als wäre die Datenmenge unbegrenzt, sodass Zeilen bei Bedarf abgerufen, gerendert und wieder freigegeben werden.
Batch-Command	Eine Serveroperation, die eine Änderung auf einmal auf viele ausgewählte Datensätze anwendet, mit Validierung und Transaktionsbehandlung an einer einzigen Stelle.
Teilfehler	Wenn einige Elemente eines Stapels gelingen und andere scheitern. Die Operation muss genau melden, welche gescheitert sind und warum, statt eines Alles-oder-nichts-Fehlers.
reaktionsschnelles Feedback	Fortschrittsanzeigen, deaktivierte Controls und Statusmeldungen, die Benutzer bei langen Operationen auf dem Laufenden halten, statt den Bildschirm eingefroren wirken zu lassen.

3. Der rote Faden der Lektion

Große Datenmengen sind ein Workflow-Problem

Das Ziel ist nicht, eine Million Zeilen anzuzeigen. Das Ziel ist, dem Benutzer zu helfen, den richtigen Datensatz zu finden, seinen Status zu verstehen und sicher zu handeln. Suche, Filter, gespeicherte Ansichten, Zusammenfassungen und Stapelaktionen zählen mehr als die reine Zeilenzahl.

Projizieren Sie für das Grid

Eine Grid-Zeile braucht oft Anzeigetext, Statussymbole, ein berechnetes Alter, Berechtigungs-Flags und Zähler. Ein Projektionsmodell hält das Grid schnell und übersichtlich, ohne Domänenentitäten mit Darstellungsdetails zu verunreinigen.

Filtern und Paging auf dem Server

Der erste Bildschirm der Enterprise-Arbeitswarteschlange lädt genau eine Seite von Zeilen, und diese Tatsache bestimmt das Design. Das Grid erhält keine Liste, sondern ein `PagedResult` mit den Zeilen der aktuellen Seite und der Gesamtanzahl. Filter, Sortierreihenfolge und Seitennummer werden als `WorkQueueQuery` an den Server gesendet, und der Query-Service übersetzt sie in eine Datenbankabfrage, die nur diese Zeilen zurückgibt. Die Datenbank übernimmt das Filtern, denn sie hat die Indizes und den Speicher dafür.

Gespeicherte Ansichten machen serverseitiges Filtern alltagstauglich. Eine Disponentin, die jeden Morgen überfällige Aufträge mit hoher Priorität prüft, speichert diesen Filter einmal, und die gespeicherte Ansicht ist eine hinterlegte Abfragedefinition, kein zwischengespeichertes Ergebnis. Die gespeicherte Sortierung und die aktuelle Seite sind Teil des Grid-Zustands auf dem Server, sodass ein Refresh dieselbe Ansicht wiederherstellt, statt von vorn zu beginnen.

Die Review-Frage ist die erste des Labs: Wie viele Datensätze werden für den ersten Bildschirm geladen? Lautet die Antwort „mehr als eine Seite“, hat das Design noch gar nicht begonnen.

Projektionen statt Entitäten

Eine Grid-Zeile braucht Anzeigewerte: eine Arbeitsauftragsnummer, einen Kundennamen, eine Statusbezeichnung mit Symbol, ein Alter in Tagen und die Angabe, ob der aktuelle Benutzer den Auftrag neu zuweisen darf. Die Domänenentität hat nichts davon direkt; sie hat Fremdschlüssel, Zeitstempel und eine Status-Enumeration. Wird die Entität an das Grid gebunden, muss der Bildschirm die Anzeigewerte Zeile für Zeile berechnen, und die Entität muss Darstellungseigenschaften bekommen, die sie nicht haben sollte.

Eine Suchprojektion ist eine auf den Bildschirm zugeschnittene Klasse wie `WorkQueueRow`, die der Query-Service mit einer Select-Anweisung füllt, welche die Anzeigewerte in der Datenbank berechnet. Sie enthält nur die Spalten, die das Grid zeigt, sodass eine Seite mit zweihundert Zeilen wenige Kilobyte groß ist statt eines ganzen Entitätsgraphen. Sie enthält außerdem die Berechtigungs-Flags, die das Grid zum Aktivieren oder Deaktivieren von Aktionen braucht, einmal auf dem Server berechnet.

Projektionen und Entitäten ändern sich aus unterschiedlichen Gründen. Eine neue Spalte im Grid ändert die Projektion; eine neue Geschäftsregel ändert die Entität. Hält man sie getrennt, bricht keine der beiden Änderungen die andere.

Stapeloperationen, die nie stillschweigend scheitern

Fünfzig Arbeitsaufträge neu zuzuweisen ist eine Benutzeraktion und fünfzig Geschäftsoperationen. Der Batch-Command im Lab transportiert die ausgewählten Schlüssel und den Zieltechniker, und der Handler verarbeitet sie einzeln: Er validiert jeden, wendet die Änderung an und hält das Ergebnis pro Zeile fest. Eine Zeile kann scheitern, weil der Auftrag inzwischen geschlossen wurde, weil dem Techniker eine Zertifizierung fehlt oder weil ein anderer Benutzer ihn zuerst geändert hat. Nichts davon sollte die übrigen neunundvierzig aufhalten.

Ein Teilfehler ist daher das erwartete Ergebnis, kein Fehlerfall. Der Handler liefert einen Bericht, der jede Zeile mit „erfolgreich“, „fehlgeschlagen“ oder „übersprungen“ samt Grund auflistet. Der Bildschirm zeigt während der Verarbeitung den Fortschritt und am Ende den Bericht und bietet eine Wiederholung nur für die fehlgeschlagenen Zeilen an. Genau das ist der Fehlerpfad im Lab: drei Aufträge neu zuweisen, einer scheitert, und der Benutzer sieht, welcher und warum.

- Wird jede Zeile unabhängig validiert und angewendet, in einer eigenen Transaktion oder in einer dokumentierten Gruppe?
- Nennt das Ergebnis jede Zeile mit ihrem Ausgang?
- Kann der Benutzer die Fehlschläge wiederholen, ohne die Erfolge zu wiederholen?

Wahrgenommene Performance messen

Benutzer beurteilen einen Bildschirm danach, wie lange sie warten, nicht danach, wie viele Zeilen er fassen könnte. Wahrgenommene Performance wird an den Momenten gemessen, die sie bemerken: die Zeit vom Öffnen des Bildschirms bis zum Erscheinen der ersten Zeilen, die Zeit von einer Filteränderung bis zur Aktualisierung des Grids, die Zeit vom Klick auf eine Stapelaktion bis zur ersten Fortschrittsanzeige. Die Performance-Notizen des Labs halten diese Werte für die Arbeitswarteschlange mit realistischen Datenmengen und einem realistischen Netzwerk fest.

Das Tuning folgt den Messungen. Ein langsamer erster Ladevorgang bedeutet meist zu viele Spalten oder einen fehlenden Index für die Standardsortierung. Eine langsame Filteränderung bedeutet meist, dass der Filter nicht in die Datenbank übersetzt wird. Eine Stapelaktion, die eingefroren wirkt, braucht innerhalb der ersten Sekunde eine Fortschrittsanzeige, auch wenn die eigentliche Arbeit eine Minute dauert. Reaktionsschnelles Feedback – ein deaktivierter Button, ein Fortschrittsbalken, eine Statuszeile – kostet fast nichts und verändert, wie lang sich das Warten anfühlt.

Hinter all dem steht die Gewohnheit des virtualisierten Denkens: Entwerfen Sie so, als hätte die Datenmenge keine Obergrenze, damit jede Liste, jede Dropdown-Liste und jeder Export auch dann noch funktioniert, wenn die Daten auf das Zehnfache wachsen.



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:

- ✓ Listenbildschirme für große Datenmengen mit serverseitigem Filtern und Paging entwerfen.
- ✓ Projektionen erstellen, die die Grid-Anzeige von Domänenmodellen trennen.
- ✓ sichere Stapeloperationen mit Fortschritt, Validierung und Behandlung von Teilfehlern implementieren.
- ✓ wahrgenommene Performance messen und optimieren.

Lab- und Prüfungsleitfaden

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



Praxis-Lab

Erstellen Sie einen Bildschirm für die Enterprise-Arbeitswarteschlange mit serverseitigen Filtern, gespeicherten Ansichten, Paging, dauerhaft gespeicherter Sortierung und einer Stapelaktion zur Neuzuweisung. Ergänzen Sie eine Fortschrittsanzeige und einen Ergebnisbericht pro Zeile für Teilfehler.

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

- ✓ Query-Service mit Paging
- ✓ Projektionsmodell für die Suche
- ✓ Definition einer gespeicherten Ansicht
- ✓ Workflow für die Stapel-Neuzuweisung
- ✓ Performance-Notizen

Wiederholungsfragen

- ✓ Wie viele Datensätze werden für den ersten Bildschirm geladen?
- ✓ Was passiert, wenn eine Zeile im Stapel scheitert?
- ✓ Kann der Benutzer die Suche nach einem Refresh wiederholen?



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.