

# Listen, Listen, Listen

Thomas Kestler, elevato GmbH

**Nahezu jede Software enthält Listen, das Thema ist doch schon steinalt, warum macht sich da noch einer die Mühe, einen Artikel zu schreiben? Doch was zunächst langweilig klingt, wird bei näherer Betrachtung richtig spannend.**

Ja, es gibt sie in tausend Formen und Ausprägungen: Listen. Mit oder ohne Blätterfunktion, mit oder ohne Sortierung, dynamisch oder starr. Und in jeder Anwendung sind sie etwas anders (teilweise sogar innerhalb einer Anwendung). Da ein IT-Anwender heute mit einer Vielzahl von Anwendung arbeitet, muss er auch mit unterschiedlichen Listen arbeiten. Wenn diese nun unterschiedlich funktionieren, führt das zu Umgewöhnungsaufwand (teils sogar zu latentem Stress). Natürlich wäre es illusorisch, eine Homogenisierung von Listen in den verschiedenen Anwendungen zu erwarten. Andererseits, bei all der Patternitis, warum gibt es für Listen eigentlich noch kein Pattern?

## Bestandsaufnahme

Es gibt die unterschiedlichsten Formen von Listen: paginiert/unpaginiert, lang/kurz, dynamisch/starr, begrenzt/unbegrenzt erweiternd, etc.

### Dynamisch/Starr

Eine dynamische Liste verändert Ihren Inhalt während der Benutzer sie verwendet: löscht er z.B. ein Element daraus, so verschwindet es aus der Liste (löscht jemand anderes ein Element, so verschwindet es auch, z.B. nach vor- und zurück blättern). Dynamische Listen eignen sich für dynamische Anwendungen, z.B. eBay. Eine starre Liste hält den Zustand zum Zeitpunkt der Suchabfrage und muss erst wieder durch erneute Abfrage aktualisiert werden. Starre Listen eignen sich dann, wenn sich Daten kaum während der Betrachtung ändern. Google und Bugzilla sind Beispiele für starre Listen. Produktkataloge fallen auch in diese Kategorie.

### Begrenzt/Unbegrenzt

Listen, die sehr lange werden können, sind aus Performancegründen oftmals begrenzt. Andernfalls könnte eine ungeschickte Abfrage (ohne Suchkriterien) zu Millionen von Treffern führen (z.B. Einwohnermeldeamt). Google begrenzt seine Ergebnisliste derzeit auf 1000 Ergebnisse. Eine solche Begrenzung ist immer zulässig, wenn die Vollständigkeit der Liste nicht als Beweis dienen muss. Für die Stückliste eines Airbus wäre das ggfs. nicht zulässig.

### Paginiert/Unpaginiert

Gerade in Web-Anwendungen sind Listen oftmals paginiert, d.h. man sieht zunächst die ersten 10 (20, 50, 100) Treffer und kann dann z.B. mit Links wie Next/Previous navigieren. Meist gibt es ein Navigationselement für erste und letzte Seite. Einzelne Seiten können angesprungen werden, teilweise werden aber nur eine begrenzte Anzahl von Seitenlinks. Schon die Berechnung der Seitenlinks ist ein Schmankehl, denn man möchte z.B. die aktuelle Seite (hier 14) in der Mitte haben:

[|<](#) [<<](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [>>](#) [|>](#)

Im obigen Beispiel sollen also je drei Seiten rechts und links neben der aktuellen Seite stehen, also bis zu 7 direkte Seitenlinks und diese umrahmt von den Links für die allgemeine Navigation (erste, vorige, nächste, letzte). Mittlerweile erwartet der Anwender, dass seine Einstellung für die Listengröße erhalten bleibt (z.B. Speicherung in Cookie, Voreinstellungen/Profil).

### Erweiternde Listen

Eine erweiternde Liste zeigt anfangs nur eine gewisse Menge von Daten an, je weiter man Richtung Ende der Liste blättern, desto mehr Daten werden nachgeladen. Das kennen wir z.B. von Microsoft Access. Je nach Implementierung sieht man die Gesamtanzahl der Ergebnisse von Anfang an (1-10/1053) oder erst, wenn man wirklich an das Ende navigiert hat.

Wenn es für den Anwender nicht unbedingt nötig ist, bereits zu Beginn die Gesamtzahl zu wissen, kann man dadurch die Datenbankzugriffe optimieren und zunächst nur einen Teil einlesen.

### Lange Listen

Bei Listen, die sehr lange werden können, ist es aus Performancegründen nicht sinnvoll, bei jedem Blättern alle Daten aus der Datenbank zu lesen. Auch das Caching in der Anwendung (z.B. in der Session bei einer Web-Anwendung) ist bei großen Listen ungeschickt (denken wir an Clusterbetrieb und Serialisierung der Session, dann würde das mit serialisiert werden). Bei sehr langen Listen (> 1000 Ergebnisse) sollte man also darüber nachdenken, immer nur den wirklich zu Anzeige benötigten Teil der Daten aus der Datenbank zu lesen. Das ist aber in der Praxis nicht trivial. Zwar unterstützen viele Datenbanken eine Begrenzung der Ergebnismenge (z.B. SQLServer TOP, ORACLE rowcount). Aber nur wenige Datenbanken unterstützen das Aufsetzen bei einem bestimmten Datensatz und dann stellt sich immer noch die Frage ob das wirklich performant passiert.

Eine Möglichkeit bei starren Listen besteht darin, dass man die Abfrage in eine temporäre Tabelle umleitet mit vorangestellter Nummerierung. Beim Lesen zur Anzeige erfolgt der Zugriff auf diese temporäre Tabelle über diese Ordnungsnummer. Damit der Zugriff über die Ordnungsnummer performant ist, müsste eigentlich ein Index darauf angelegt werden und nur wenige Datenbanksysteme unterstützen Indices auf temporären Tabellen, abgesehen davon, dass auch das Konstrukt der (pro Connection existierenden) temporären Tabelle nicht in allen Datenbanksystemen (identisch) existiert. Bei Änderung der Such- und/oder Sortierkriterien muss die temporäre Tabelle immer neu gefüllt werden.

Eine sehr einfache Variante kann so aussehen, dass man im DAO unterscheidet, ob man sich auf Seite 1-N befindet oder dahinter (z.B. N=10, jede Seite hat 20 Einträge). Bei einer Seitennummer < N liest man die Daten mit

```
SELECT TOP 200 ... from ...
```

Bei einer größeren Seitennummer ohne diese Begrenzung. Hierbei geht man davon aus, dass sich der Benutzer in 95% der Fälle auf den ersten 10 Seiten der Liste aufhält und bei sehr großen Listen ggfs. besser durch genauere Suchkriterien einschränkt.

Auch eine Variante: wenn Liste ohne (explizite) Suchkriterien aufgerufen wird, dann nur X Datensätze lesen (z.B. 500), dann aber bitte den Benutzer auch darüber informieren und die Möglichkeit geben, darüber hinaus zu blättern (oder Begrenzung abzuschalten, vgl. DbVisualizer). Die Liste wurde also implizit in eine erweiternde Liste (s.o.) umgewandelt.

### **Selektionen und Aktionen**

Der Benutzer möchte mit den Listen etwas tun und zwar meist bestimmte Elemente selektieren. Dies kann per Single- oder Multi-Select erfolgen oder per Checkbox. Meist wünscht man sich auch eine Select All bzw. DeSelect All Funktionalität, die oftmals als Checkbox oberhalb der Liste visualisiert wird. Sofern eine Liste paginiert ist, soll die Selektion dabei möglichst erhalten bleiben.

### **Suchkriterien/Sortierung**

Die Suchkriterien sind untrennbar mit der Liste verknüpft. Hier sollte es dem Benutzer möglichst leicht gemacht werden: Wildcardsuche mit \*,?,%,\_ möglich, Auswahllisten, ggfs. Mehrfachauswahl zulassen.

Da der Benutzer meist nach einem festen Schema arbeitet (z.B. ein Mitarbeiter ruft alle Neukunden an), sollten die Suchkriterien ganz oder teilweise erhalten bleiben, auch wenn er die Liste verlassen hatte und wieder zurückkehrt (vgl. Google merkt sich die gewünschte Anzahl Treffer pro Seite im Cookie).

Ein Reset-Button sollte das zurücksetzen der Suchkriterien auf Defaultwerte ermöglichen.

Das gleiche gilt für die Sortierung. Für beides gilt: der Benutzer erwartet, dass die Liste bei Änderung der Suchkriterien oder Sortierung wieder bei Seite 1 beginnt.

### **Verfeinern**

Bei der Suche in Produktkatalogen (vgl. eBay, Electronic Scout) besteht die Möglichkeit der Verfeinerung einer Suche. Beispiel: Der Benutzer sucht einen HD-Rekorder, dazu existieren zahlreiche Geräte und noch mehr Angebote. Der Benutzer erhält eine Möglichkeit zur Verfeinerung:

**Suchergebnis zu HD Recorder**

Sie befinden sich in der empfohlenen Kategorie: **DVD- & HDD-Rekorder (11 Produkte)**  
 Weitere Kategorien mit Treffern zu Ihrer Suche:

**LCD-TV (18) | Digital Receiver (21)**

▶ [Alle 24 Kategorien mit 133 Treffern einblenden](#)

**DVD- & HDD-Rekorder einschränken nach:**

Hersteller	Preis	System Zusatzinformation	Weitere Einschränkungen
<a href="#">Panasonic</a> (6)	180 € - 250 € (3)	<a href="#">inkl. Festplatte 40-80 GB</a> (5)	<a href="#">Bitrate</a> <a href="#">Video-D/A-Wandler</a>
<a href="#">SAMSUNG</a> (1)	250 € - 300 € (1)	<a href="#">inkl. Festplatte 81-160 GB</a> (3)	<a href="#">Taktung</a> <a href="#">Video-D/A-Wandler</a>
<a href="#">Sony</a> (3)	300 € - 350 € (2)	<a href="#">inkl. Festplatte 161-250 GB</a> (2)	<a href="#">Video-D/A-Wandler</a>
<a href="#">THOMSON</a> (1)	350 € - 400 € (2)	<a href="#">inkl. Festplatte 251-320 GB</a> (2)	<a href="#">Farbe</a>
	<a href="#">über 400 €</a> (3)		

**Alle Suchergebnisse** Seite 1 von 2

1 - 10 Artikel Seite 1 | 2 >> |

Vergleichen	↕ Produktname	↕ Angebote	
<input type="checkbox"/>	 <p><b>Panasonic DMR-EX77EG-S</b>                      DVD-Rekorder mit 160GB Festplatte                      Bewertung: nicht bewertet</p>	<p>18 Angebote</p> <p>von <del>300 €</del>                      bis <del>300 €</del></p>	<p><b>Angebote ansehen</b></p> <p>oder für <del>300 €</del> bei <del>300 €</del></p>

Abb.: Screenshot Suchergebnis electronicscout24.de

Der Benutzer kann nun die Suche nach Hersteller, Preis, Ausstattung oder anderen Kriterien verfeinern. Die Verfeinerung ist aber nur temporär, bei der nächsten Suche muss der Benutzer ggfs. erneut verfeinern (oder er gibt z.B. den Hersteller im Suchbegriff mit ein).

Die Verfeinerung erfordert aber meist eine entsprechend komplexe Suchmaschine, die im Suchergebnis auch gleich Kategorisierungen mit liefert. Dafür kann man kommerzielle Produkte einsetzen (FAST, Qwiser, etc.) oder man bereitet die linearen Suchergebnisse mit einem Merge Join auf eine Kategorientabelle entsprechend auf. Letzterer Ansatz ließe sich z.B. mit Lucene von Apache einfach aufsetzen.

### Testfälle für Listen

Wer eine Listenfunktion implementiert muss sich auch Gedanken machen, wie diese auf Korrektheit getestet. Hier ein paar Anregungen (unvollständig):

- Vollständigkeit – Überprüfen, ob die Liste auch alle die Datensätze anzeigt, die auch die SQL-Abfrage in der Datenbank liefert. Bei Listen mit Suchkriterien auf Einschlussfälle achten ( $\geq$ ,  $\leq$ ).
- Überlappung – wird beim Blättern ein Datensatz der letzten Seite auf der nächsten Seite angezeigt (sollte nicht).
- Navigation – Navigieren letzte Seite, beliebige Seite, Letzte Seite, seitenweise durchnavigieren. Es müssen bei allen Navigationselementen immer die erwarteten Daten angezeigt werden (Beispiel: 7 Seiten, bei Klick auf 7 und bei Klick auf „Letzte“ muss der gleiche Inhalt dargestellt werden).
- Sortierung – funktionieren alle Sortierungen auf- und absteigend? Bleibt diese auch beim Blättern erhalten. Ist die Sortierreihenfolge wie in Spezifikation

gefordert (dort könnte z.B. stehen Sortierung nach Telefonbuch, oder A-Z, a-z, ...). Wird bei Änderung der Sortierung wieder auf Seite 1 gesprungen, falls in Spezifikation nicht anders gefordert?

- Änderung Suchkriterien – Verhalten wie spezifiziert? Sprung auf Seite 1 nach Änderung? Falls gefordert, bleiben Suchkriterien dauerhaft erhalten (Session, falls gefordert)
- Dynamische Listen: wird Änderung während Blättern erkannt? Auf Seite 1 gehen, Datensatz von Seite 2 parallel ändern, auf Seite 2 gehen, prüfen, usw.

## Praxisbeispiel: Displaytag

Es gibt eine ganze Reihe von Tools und Bibliotheken, welche das Handling von Listen vereinfachen sollen. Exemplarisch wird hier `displaytag` herausgegriffen, ein JSP-Tag, das als OpenSource unter [sf.net](http://sf.net) verfügbar ist. `Displaytag` ist sehr praktisch und erlaubt eine sehr einfache und flexible Handhabung von Listen in JSP Seiten:

```
<display:table name="test">
  <display:column property="id" title="ID" />
  <display:column property="name" />
  <display:column property="email" />
  <display:column property="status" />
  <display:column property="description" title="Comments"/>
</display:table>
```

Die eigentliche Liste wird als Request-Attribut („test“) übergeben. Der Zugriff auf die Listenobjekte kann auch durch einen Decorator gekapselt werden. Spalten können auch sortiert bar gekennzeichnet werden, die Paginierung ist automatisch enthalten.

Und genau hier liegt ein Stolperstrick: in der aktuellen Version ist es nicht möglich, die aktuelle Sortierung und Seite (z.B. in der Session) beizubehalten (siehe dazu <http://jira.codehaus.org/browse/DISPL-125>). Um nun aber die Sortierung und Seitennummer dauerhaft (z.B. in der Session) beizubehalten, so dass man auch wieder auf der gleichen Seite bei gleicher Sortierung landet, wenn man auf die Liste zurück springt, muss man das Feature External Paging von `Displaytag` nutzen und sich selbst um das halten der Informationen in der Session kümmern (Abhilfe wird in `Displaytag 1.2` kommen).

External Paging ist dann angesagt, wenn die Liste sehr lange werden kann. Ohne External Paging bekommt `Displaytag` die gesamte Liste und macht das Paging selbst. Wird die Liste aber nicht als `java.util.List`, an das Tag übergeben, sondern als `org.displaytag.pagination.PaginatedList` so erkennt `Displaytag` das External Paging und zeigt nur die in diesem Objekt für diese Seite hinterlegten Daten an (mehr Info siehe `Displaytag Website`).

## Praxistipp: Ext JS

Die Javascript Bibliothek **Ext JS** wurde im letzten Jahr deutlich weiter entwickelt und sollte heute für Webprojekte durchaus in Betracht gezogen werden. `Extjs` bringt in puncto Listen die sogenannten Grids mit, die auch paginiert sein können. Bei `Ext` besticht vor allem die Optik (Themes) und die Ajax Integration. Nützliche Features wie Ein- und Ausblenden von Spalten sind damit sehr einfach machbar.

ExtJS.com - Browse Forums			
Topic	Author	Replies	Last Post
<b>Leverage the reusability of ToolbarLayout.</b> Ext: Feature Requests Forum Right now, the Buttons in a Panel's footer area are rendered by a block of HTML-creation code in Panel::onRender: if(this.buttons && this.buttons.length > 0){ // tables are required to maintain order and for correct IE layout var tb = this.footer.createChild({c...	Animal	0	May 23, 2008, 9:35 am by Animal
<b>Binding ComboBox Dynamically</b> Ext: Help Forum I am working on binding combobox dynamically. I am getting the JSON Output from the url in the form below: <pre>[[{"1","Hospital"}, {"2","HospitalName"}]]</pre> which has to be binded to the ComboBox in ExtJS.... My code is as follows: new Ext.form.ComboBox({ fieldLabel: 'Manage...	vijayakumari	2	May 23, 2008, 9:34 am by tomcheng76
<b>Problem in IE: height set to -1</b> Ext: Help Forum Hello all, I have a modal window to which I disabled the open / close animation by calling show(null). However after disabling the animation, IE throws an error because the framework tries to set a negative height to the window's shadow (my guess). The error is in Shadow.realign at line: s.height...	daniel.rucareanu	0	May 23, 2008, 9:29 am by daniel.rucareanu
<b>Is there a greedy spacer for the buttons in the Panel footer?</b> Ext: Help Forum Well first, as far as I can tell, I cannot modify the Panel footer directly. I only see how to auto create it when I specify buttons in the Panel config. So, the real question is: If I have three buttons, how can I get two to align to the right and one to align to the left? A greedy spacer would do...	rtconner	3	May 23, 2008, 9:28 am by Animal
<b>add remove items in portal layout</b> Ext: Help Forum I am quite new to EXTJS, find it gr8, although really in need for some answers I have a page with portal layout, with draggable widgets, My code for portal is : var myportal = { xtype:'portal', region:'center', // margins:'35 5 5 0', items:[{ ...	Richa	0	May 23, 2008, 9:23 am by Richa

Page 1 of 1280 | Show Preview | Displaying topics 1 - 25 of 31996

## Checkliste

Bei der Pflichtenhefterstellung sollte der Autor in Hinblick auf Listen kurz folgende Checkliste durchgehen:

- Suchkriterien: welche? Merken (wo, wie lange)? Wildcards?
- dynamisch oder starr?
- begrenzt oder unbegrenzt?
- Erweiternde Liste, lange Liste? Performanceaspekte: wie viele Daten sind zu erwarten? Maximalbegrenzung? Warnung bei unqualifizierter Suche? Abbruchmöglichkeit?
- paginiert, unpaginiert (scrolling)? Anzahl Treffer pro Seite merken (Session, Cookie, Preferences)?
- Paginierung: Ausgestaltung, wie viele Links, Tooltips, etc?
- Listeneinstellungen global für alle Listen oder je Liste unterschiedlich merken?
- Aktuelle Seite und/oder Sortierung in Session merken?
- Verfeinerung, Kategorisierung benötigt (spezieller Anwendungsfall)?
- Testfälle skizzieren

## Abgrenzung zu Tabelle

Eine Liste dient der Selektion von Elementen. Eine Tabelle dient darüber hinaus auch neben der grundlegenden Visualisierung (Stammdaten) auch für weitergehende Aktionen wie In-Place-Editing und Copy/Paste. Listen werden meist in modalen Dialogen eingesetzt, während Tabellen oftmals der Ausgangspunkt für die Bearbeitungsschritte des Benutzers darstellen. Ein Mail-Client ist ein gutes Beispiel

für eine Anwendung, die auf einer Tabelle basiert (Posteingang). Auch Tabellen können paginiert sein und viele der Anforderungen die für Listen gelten, lassen sich auch hier übertragen. Von der Konzeption her sollte man sich immer überlegen, welches der beiden Elemente man für welchen Anwendungszweck einsetzen will (natürlich mit fließenden Übergängen).

## Fazit

Auch ein scheinbares Standardthema wie Listen kann es in sich haben. Entscheidend ist, dass man sich bereits beim Schreiben der Spezifikation Gedanken über die in der Anwendung zu verwendenden Listen macht und diese dann überall gleich umsetzt.

Thomas Kestler ist Geschäftsführer der elevato GmbH.

<http://www.elevato.de/>