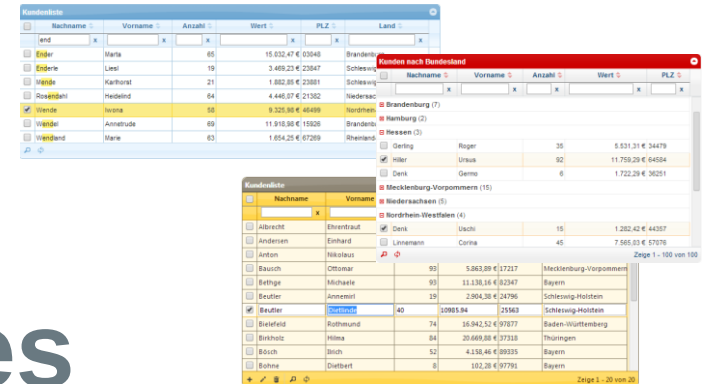


Schick mit Grid – Data Grids in XPages

Knut Herrmann



Nachname	Vorname	Anzahl	Wert	PLZ	Land
End					
Eder	Marla	85	15.032,47 €	03048	Brandenburg
Eder	Liesl	19	3.489,23 €	23847	Schleswig
Eder	Karlrat	21	1.882,85 €	23881	Schleswig
Eder	Hedebel	64	4.448,07 €	21382	Niedersachsen
Eder	Heine	58	8.326,58 €	04499	Niedersachsen
Eder	Annerude	69	11.918,58 €	15626	Brandenburg
Eder	Mare	63	1.854,25 €	07209	Sachsen



NOTES & DOMINO
ENTWICKLERCAMP

- Diplom-Informatiker, Technische Universität Dresden
- seit 1992 Lotus Notes Anwendungsentwicklung
- in letzten Jahren mit Schwerpunkt XPages und Java
- Mitgründer und -Gesellschafter der Leonso GmbH

- Kontakt:
 - E-Mail: knut.herrmann@leonso.de
 - Twitter: @KnutHerrmann
 - Skype: knut.herrmann

Warum Data Grids?

Data Grids in XPages
definieren

Daten lokal
und über
REST/Ajax
liefern

Agenda

In welchen Fällen
sind Data Grids
sinnvoll und wann
Alternativen besser?

Formatieren,
filtern und
sortieren

Kategorisieren mit
Grouping und Tree

Vergleich von
Dojo Data Grid,
jqGrid und
ExtJS Grid

Daten inline bearbeiten

Schickes Aussehen
durch CSS & Co

The screenshot displays an XPages application with a data grid. The grid has columns for 'Nachname', 'Vorname', 'Anzahl', 'Wert', 'PLZ', and 'Land'. It is filtered by 'Land' and shows a list of customers. A modal dialog 'Kunde auswählen' is open, showing a form for selecting a customer. Another modal dialog 'Kunden nach Bundesland' is open, showing a list of customers grouped by state.

- Das sind die Nutzer im Notes Client gewohnt ...

Search in View 'Kundenliste' Show results By relevance Indexed ? x

Search for **Volltextsuche** Search

	Nachname	Vorname	Anrede	Geburtsdag	Anzahl	Wert	Letzte Bestellung
✓	Abel	Anne			14	922,64	31.07.2013 11:53:18
	Fengler	Anne-Kathrin			51	5.498,08	12.04.2012 10:35:32
	Rahm	Anne-Kathrin	Frau	26.07.1933	49	3.294,32	01.02.2014 18:00:37
	Gies	Anne-Kathrin	Frau	08.12.1954	86	18.148,12	10.02.2014 08:51:00
	Stoll	Anne-Marie				32	
✓	Kretzer	Anne-Marie			79	08.02.2013 16:17:34	
	Krumm	Anne-Marie			41	23.04.2013 13:43:51	
✓	Holl	Anne-Rose			13	20.02.2014 20:49:37	
	Käfer	Anne-Rose			38	29.06.2012 14:33:17	
	Knoll	Annedoris			71	21.05.2011 15:30:22	
	Hampe	Annedoris			07	27.11.2009 03:47:54	
	Witzke	Annedoris	Frau	05.05.1977	87	07.06.2013 10:57:54	
	Bohn	Annedoris	Frau	15.10.1989	70	19.054,71	09.06.2009 10:57:04
	Giebel	Annedoris		1942	80	10.018,45	08.10.2010 16:04:05
	Hedrich	Annedoris	Frau	23.04.1951	10	2.351,25	16.11.2009 01:20:07
							23.09.2011 10:46:09

Spaltenumsortierung

(Mehrfach-)Markierung

Find

Find in: Vorname

Starts with: Hemine

Schnellsuche

Scrollbalken

Dokument öffnen

■ ... und vieles geht auch im Browser, aber

- ☐ kein Scrollbalken
- ☐ keine Schnellsuche
- ☐ keine Selektion über alle Dokumente

[← Previous](#)
[Next →](#)
[+ Expand](#)
[- Collapse](#)
[🔍 Search](#)

Kundenliste

Nachname	Vorname	Anrede	Geburts-tag	Anzahl	Wert	Letzte Bestellung
Abel	Anne	Frau	05.12.1926	14	922,64	31.07.2013 11:53:18
Abel	Rebecca	Frau	25.06.1951	25	1.393,79	05.12.2011 03:30:00
Abel	Florentin	Herr	14.07.1946	91	7.270,70	09.10.2013 10:10:24
Abele	Enrico	Herr	18.10.1990	28	5.897,52	06.10.2013 00:41:30

Previous 1 2 3 4 5 6 7 8 ... Next						
Nachname	Vorname	Anrede	Geburts-tag	Anzahl	Wert	Letzte Bestellung
☐ Abel	Anne	Frau	05.12.1926	14.0	922.64	31.07.2013
☐ Abel	Rebecca	Frau	25.06.1951	25.0	1393.79	05.12.2011
☐ Abel	Florentin	Herr	14.07.1946	91.0	7270.7	09.10.2013
☐ Abele	Enrico	Herr	18.10.1990	28.0	5897.52	06.10.2013

- ... da kommen Data Grids ins Spiel

- Data Grids sind Tabellen, die durch CSJS dynamisch werden
 - ☐ Scrollbalken vertikal, auch horizontal
 - ☐ Spalten kann man
 - In der Breite anpassen
 - umsortieren
 - ein- und ausblenden
 - ☐ Zellen editieren
 - ☐ ... und mehr – je nach Data Grid

	 Dojo Data Grid	 jqGrid Grid	 ExtJs Grid
Basis-JS-API	Dojo	jQuery	jQuery
Link	dojotoolkit.org	Trirand.com Trirand.net	Sencha.com
Installation	bereits alles in Domino enthalten	jQuery + jqGrid	jQuery + ExtJs Grid
Lizenz	frei BSD / AFLv2	frei MIT / GPLv2 Commercial	GPL v3 Commercial

Demo

Data Grids in XPages definieren

Dojo Data Grid

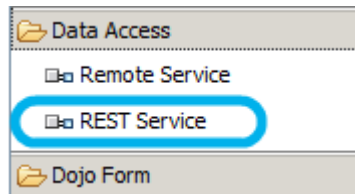
- Voraussetzung
 - ☐ Notes 8.5.3 UP1 oder ExtLib
 - ☐ Notes 9
 - ☐ bereits vorinstalliert – „out of the box“

- Schritte, um Dojo Data Grid in XPages zu erstellen
 1. REST Service einrichten
 2. Dojo Data Grid erstellen und mit REST Service verbinden
 3. Dojo Data Grid Columns einfügen und konfigurieren

Data Grids in XPages definieren

Dojo Data Grid

■ 1. REST Service einrichten



```
<xe:restService
  id="restService1"
  pathInfo="kundenliste">
  <xe:this.service>
    <xe:viewJsonService
      databaseName="EntwCamp/Testdaten.nsf"
      viewName="Kundenliste"
      defaultColumns="true" />
    </xe:this.service>
  </xe:restService>
```

- All Properties > basics > service
„viewJsonService“ hinzufügen

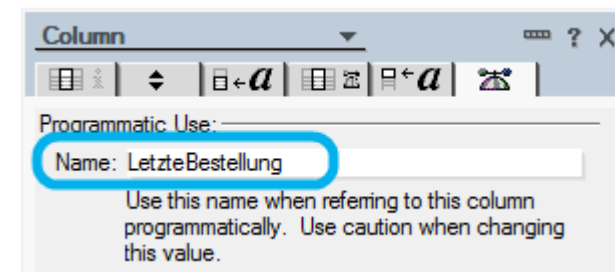
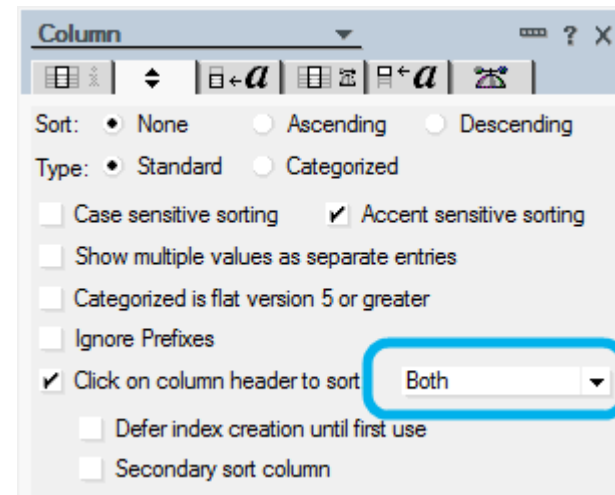
- databaseName = Datenbankpfad/-ReplikId, leer = aktuelle Datenbank
- viewName = Name der Ansicht
- defaultColumns = true

- Zum Testen des Rest Services über URL
pathInfo = Bezeichnung des Services (z.B. „kundenliste“)
Beispiel URL: <http://Server/Datenbank.nsf/Kunden.xsp/kundenliste>

Data Grids in XPages definieren

Dojo Data Grid

- 1. REST Service einrichten
 - sortierbare Spalten für *beide* Richtungen einrichten
 - Spalten einen Namen geben (möglichst Feldnamen)



Data Grids in XPages definieren

Dojo Data Grid

■ 1. REST Service einrichten

□ Antwort-Datenformat des REST-Services viewJsonService

[1. Dokument

```
{
  "@entryid": "1-00F636A5D369B909C1257C8F005C7FA6",
  "@unid": "00F636A5D369B909C1257C8F005C7FA6",
  "@noteid": "117E",
  "@position": "1",
  "@read": true,
  "@siblings": 10000,
  "@form": "Adresse",
```

Standardfelder

```
  "Nachname": "Abel",
  "Vorname": "Anne",
  "Anrede": "Frau",
  "Geburtstag": "1926-12-05T10:56:19Z",
  "Anzahl": 14,
  "Wert": 922.64,
  "LetzteBestellung": "2013-07-31T09:53:18Z",
  "Plz": "35614",
  "Ort": "A\u00DFlar",
  "Strasse": "Rennersdorfer Hauptstra\u00DFe 12",
  "Kreis": "Lahn-Dill-Kreis",
  "Land": "Hessen",
  "Telefon": "",
  "Mail": ""
```

Spalten der Ansicht

```
},
{
```

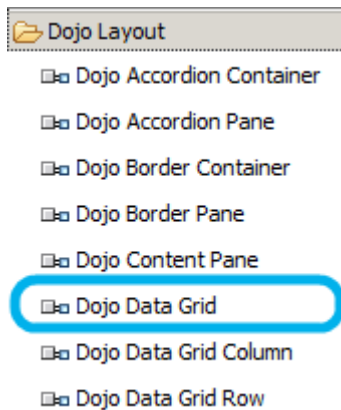
2. Dokument

```
  "@entryid": "2-41B9BCF87B35E1A4C1257C8F005C9676",
  "@unid": "41B9BCF87B35E1A4C1257C8F005C9676",
```

Data Grids in XPages definieren

Dojo Data Grid

■ 2. Dojo Data Grid erstellen und mit REST Service verbinden



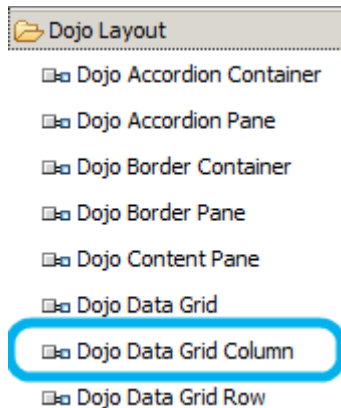
```
<xe:djxDataGrid
  id="djxDataGrid1"
  storeComponentId="restService1">
</xe:djxDataGrid>
```

- All Properties > data > service > storeComponentId
RestService-Id auswählen

Data Grids in XPages definieren

Dojo Data Grid

■ 3. Dojo Data Grid Columns einfügen und konfigurieren



- All Properties > data > field
jeweils Spaltennamen
eintragen

```
<xe:djxDataGrid
  id="djxDataGrid1"
  storeComponentId="restService1">
  <xe:djxDataGridColumn
    id="djxDataGridColumn1"
    field="Nachname" />
  <xe:djxDataGridColumn
    id="djxDataGridColumn2"
    field="Anzahl" />
  <xe:djxDataGridColumn
    id="djxDataGridColumn3"
    field="Wert" />
  <xe:djxDataGridColumn
    id="djxDataGridColumn4"
    field="LetzteBestellung" />
</xe:djxDataGrid>
```

Data Grids in XPages definieren

Dojo Data Grid

■ 4. Spalten formatieren

```
require( [ "dojo/date/locale" ] );

function formatDate(value) {
    return value ? dojo.date.locale.format(new Date(value), {
        formatLength : "short"
    }) : "";
}

function formatEuro(value) {
    return parseFloat(value).toFixed(2).toLocaleString() + " €";
}
```

```
<xe:djxDataGridColumn
    id="djxDataGridColumn6"
    field="Wert"
    formatter="formatEuro">
</xe:djxDataGridColumn>
<xe:djxDataGridColumn
    id="djxDataGridColumn7"
    field="LetzteBestellung"
    formatter="formatDateTime">
</xe:djxDataGridColumn>
```

Data Grids in XPages definieren

Dojo Data Grid

- mit den Grundfunktion können wir bereits
 - ☑ Spalten sortieren
 - ☑ tausende Dokumente durchscrollen
 - ☑ Spaltenbreiten ändern
 - ☑ Zeilen selektieren

Demo

Nachname	Vorname	Anzahl	Wert	LetzteBestellung	Plz	Land
Abel	Lara	15	1661.88 €	22.06.09 20:50	99880	Thüringen
Achatz	Achim	83	22355.72 €	08.08.12 22:53	92363	Bayern
Ackermann	Ishildaaa	50	6157.07 €	12.05.12 09:10	74424	Baden-Württemberg
Adams	Cathleen	29	5991.44 €	17.04.12 08:46	54518	Rheinland-Pfalz
Adler	Wiltrude	35	4436.69 €	03.05.11 16:44	17391	Mecklenburg-Vorpommern
Aigner	Rudger	51	13624.78 €	22.08.09 17:23	06279	Sachsen-Anhalt

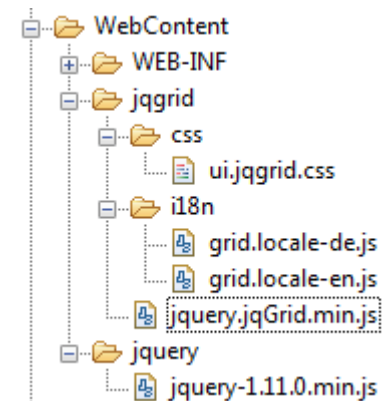
- Funktionalität lässt sich stark erweitern
 - siehe Blog-Serie "Dojo Grids In XPages" von Brad Balassaitis
<http://xcellerant.net/dojo-grids-in-xpages/>

- Alle anderen Data Grids können nicht auf vorgefertigte Custom Controls zurückgreifen

- Notwendige Schritte für Implementierung
 1. JavaScript Bibliotheken integrieren
 2. HTML-Platzhalter für Data Grid definieren
 3. Data Grid als CSJS in XPages definieren

■ 1. JavaScript Bibliotheken für jqGrid integrieren

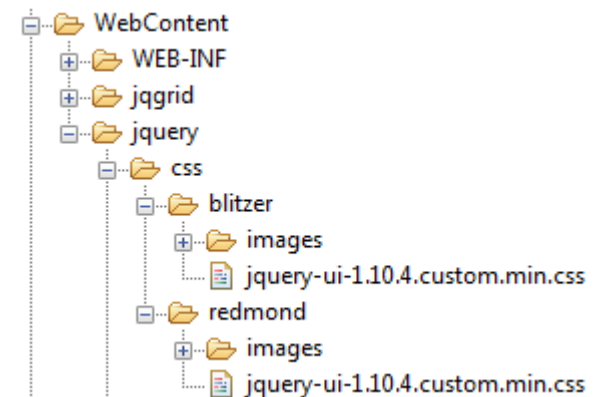
- ☐ Download von http://www.trirand.com/blog/?page_id=6
- ☐ alle Module auswählen und zip downloaden
- ☐ Dateien in WebContent Ordner kopieren
 - jqgrid/css/ui.jqgrid.css
 - jqgrid/i18n/grid.locale-*.js
 - jqgrid/jquery.jqGrid.min.js
 - jquery/jquery-1.11.0.min.js



■ 1. JavaScript Bibliotheken für jqGrid integrieren

- jQueryUi Thema downloaden <http://jqueryui.com/themeroller/>
- Thema-Dateien jeweils in WebContent/jquery/css Ordner kopieren
 - thema/images/*
 - thema/jquery-ui-1-10.4.custom.min.css
- wenn Spaltenverschiebung gewünscht wird, dann auch
 - jquery-ui.js
 - jquery-ui-i18n.js

downloaden und in WebContent/jquery bereitstellen



- 1. JavaScript Bibliotheken für jqGrid integrieren
 - Dateien als Ressource in XPages oder in Thema einbinden

```
<resource>
  <content-type>text/css</content-type>
  <href>jquery/css/redmond/jquery-ui-1.10.4.custom.min.css</href>
</resource>
<resource>
  <content-type>text/css</content-type>
  <href>jqgrid/css/ui.jqgrid.css</href>
</resource>
<resource>
  <content-type>application/x-javascript</content-type>
  <href>jquery/jquery-1.11.0.min.js</href>
</resource>
<resource>
  <content-type>application/x-javascript</content-type>
  <href>jqgrid/i18n/grid.locale-de.js</href>
</resource>
<resource>
  <content-type>application/x-javascript</content-type>
  <href>jqgrid/jquery.jqGrid.min.js</href>
</resource>
```



■ 2. HTML-Platzhalter für Data Grid definieren

- leere Tabelle mit *id* definieren

```
<table id="grid"></table>
```

- optional kann ein Pager als div mit *id* definiert werden

```
<div id="gridpager"></div>
```



■ 3. Data Grid als CSJS in XPages definieren

- JavaScript-Code für jqGrid Definition muss in der Seite geladen und ausgeführt werden, z.B. im onClientLoad-Event:

```
<xp:eventHandler
  event="onClientLoad"
  submit="false">
  <xp:this.script><![CDATA[
    var grid = jQuery("#grid").jqGrid({
      url: "DataService.xsp",
      datatype: "json",
      loadonce: true,
      caption: "Kundenliste",
      pager: "gridpager",
      height: 250,
      forceFit : true,
      colModel:[
        {name:'id',label:'id', width:50, hidden:true},
        {name:'Nachname',label:'Nachname', width:80},
        {name:'Vorname',label:'Vorname', width:80},
```



■ 3. Data Grid als CSJS in XPages definieren

- da der CSJS-Code "nur Text" ist, kann er auf verschiedenste Art erzeugt und bereitgestellt werden, z.B. von einer Java Bean

```
<xp:eventHandler
  event="onClientLoad"
  submit="false"
  script="#{javascript:de.leonso.GridController().getGridDefinition()}" />
```

```
public class GridController implements Serializable {
    private static final long serialVersionUID = 1L;

    public static String getGridDefinition() {
        StringBuilder def = new StringBuilder("jQuery('#grid').jqGrid({\n");
        def.append("    url: 'DataService.xsp',\n");
        def.append("    datatype: 'json',\n");
        def.append("    caption: 'Kundenliste',\n");
        def.append("    pager: 'gridpager',\n");
        def.append("    colModel:[\n");
        def.append("        {name:'id',label:'id', width:50, hidden:true},\n");
        def.append("        {name:'Nachname',label:'Nachname', width:80},\n");
    }
}
```



■ Zwei Ansätze Data Grids mit Daten zu versorgen

□ Daten sofort komplett laden

- in Seite hineinrendern oder REST Service / Ajax
- Daten werden komplett im Browser lokal gespeichert
- + scrollen, umsortieren, filtern, selektieren sehr schnell
- Ladezeit steigt mit Datenmenge
- = ideal für kleine Datenmengen
maximal 1000 Datensätze

□ Daten dynamisch (nach-)laden

- REST Service / Ajax
- nur einige Zeilen werden in Browser geladen – bei Bedarf wird nachgeladen
- + Geschwindigkeit von Datenmenge unabhängig
- scrollen, umsortieren, suchen, filtern erfordert Serverinteraktion
- = ideal für große Datenmengen

■ Daten lokal als Array im CSJS-Code liefern

□ Array direkt in CSJS-Code hineinrechnen

```
var lokaleDaten = [  
    {"id":"64B482081B81C766C1257C90002CF81C", "Nachname":"Abel",...},  
    {"id":"3594868FAD318ABDC1257C90002CF93C", "Nachname":"Achatz",...},  
    {"id":"B21E262ED5158354C1257C90002CF8BF", "Nachname":"Ackermann",...},  
    ...  
];  
  
jQuery("#grid").jqGrid({  
    datatype: "local",  
    data: lokaleDaten,  
    colModel:[  
        {name:'id',label:'id', width:50, hidden:true, key:true},  
        {name:'Nachname',label:'Nachname', width:80},  
        ...  
    ],  
    ...  
});
```



- Daten lokal als Array im CSJS-Code liefern
 - Daten im eingebettetem SSJS aus Ansicht holen

```
var lokaleDaten = [
  #{javascript:
    var db = session.getDatabase(database.getServer(), "EntwCamp/Testdaten.nsf")
    var vw:NotesView = db.getView('Kundenliste');
    var nav:NotesViewNavigator = vw.createViewNav();
    var ve:NotesViewEntry = nav.getFirst();
    var rows = "";
    var i = 1;
    while (ve != null) {
      var cv = ve.getColumnValues();
      rows += ',\n{"id":"' + ve.getUniversalID() + '", "Nachname":"' + cv[0] + ... + '"}';
      var tmpEntry = nav.getNext();
      ve.recycle();
      ve = tmpEntry;
      i++;
    }
    rows.substring(2)
  }];

jQuery("#grid").jqGrid({
  datatype: "local",
  data: lokaleDaten,
  colModel: [ ...
```

Demo



■ Daten über REST/Ajax laden

- jqGrid kann vom Client die Daten dynamisch über Ajax anfordern und unterstützt die Datenformate XML und JSON
- JSON-Daten müssen folgendes Format liefern

```
{
  "page" : "1",
  "total" : "1000",
  "records" : "50",
  "rows" : [ {
    "spalte1" : "wertSpalte1",
    "spalte2" : "wertSpalte2"
  },

  {
    "spalte1" : "wertSpalte1",
    "spalte2" : "wertSpalte2"
  } ]
}
```



■ Daten über REST/Ajax laden

□ 1. Variante:

eigenen Service schreiben, der die Daten wie gewünscht liefert

- xAgent, der Daten einer Ansicht als JSON aufbereitet
- Custom REST Service

□ 2. Variante:

von jqGrid erwartetes Format an verfügbare REST-Services anpassen

- `jsonReader : { root: "items", ...}`
- www.trirand.com/jqgridwiki/doku.php?id=wiki:retrieving_data

```
jsonReader : {  
    // xe:viewItemFileService JSON-Ergebnis an jqGrid anpassen  
    root : "items",  
    total : function(obj) {  
        return obj["@toplevelentries"];  
    },  
    records : function(obj) {  
        return obj["@toplevelentries"];  
    }  
},
```



- Daten über REST/Ajax laden – alle auf einmal
 - mit dem Parameter `loadonce: true` wird nur einen Ajax-Call ausgeführt und es werden dabei alle Daten erwartet

```
jQuery("#grid").jqGrid({  
    url: "RestService.xsp/kundenliste",  
    datatype: "json",  
    loadonce: true,  
    rowNum: 1000,  
    colModel: [  
        {name:'id',label:'id', width:50, hidden:true},  
        {name:'Nachname',label:'Nachname', width:80},  
        ...  
    ],  
    ...  
});
```

- Daten über REST/Ajax laden – dynamisch
 - erfordert Pager
 - scrollen nur innerhalb einer Page
 - rowNum auf höheren Wert setzen, z.B. 500
 - oder Option scroll:true / scroll:1
- jqGrid fordert über Ajax-Calls dynamisch weitere Daten an, wenn z.B. über die bestehenden Daten hinausgescrollt wird
- der gewünschte Datenset wird in der URL als Parameter übergeben
 - `?_search=false&nd=1394373953051&rows=20&page=1&sidx=&sord=asc`
 - `?_search=false&nd=1250348761396&rows=20&page=1&sidx=Name&sord=asc`
 - `?_search=true&rows=20&page=1&sidx=Name&sord=asc&searchField=Nachname&searchString=Meier&searchOper=bw`



■ Daten über REST/Ajax laden – dynamisch

□ 1. Variante

eigener Service ermittelt die Daten entsprechend der URL-Parameter

■ Spaltensortierung

```
view.resortView(param["sidx"].split(' ')[0], true);
```

□ 2. Variante

die URL-Parameter-*Bezeichnungen* werden angepasst

■ prmNames: {sort: "sortIndex", ...}

■ <http://www.trirand.com/jqgridwiki/doku.php?id=wiki:options>

```
prmNames : {  
    // Parameternamen in Ajax URL an xe:viewItemFileService anpassen  
    rows : "count",  
    sort : "sortcolumn",  
    order : "sortorder"  
},
```



■ Daten über REST/Ajax laden – dynamisch

□ 2. Variante

die URL-Parameter-Werte werden angepasst

- `serializeGridData : function(postData) {... postDataNew ...}`

```
serializeGridData : function(postData) {  
    // Parameterwerte in Ajax URL an xe:viewItemFileService anpassen  
    var postDataNew = $.extend( {}, postData);  
    delete postDataNew._search;  
    if (postDataNew.sortcolumn) {  
        postDataNew.sortcolumn = postDataNew.sortcolumn.split(" ")[0];  
        postDataNew.sortorder = postDataNew.sortorder === "asc" ?  
                                "ascending" : "descending";  
    } else {  
        delete postDataNew.sortcolumn;  
        delete postDataNew.sortorder;  
    }  
    ...  
    return postDataNew;  
},
```

Demo



■ Spalten formatieren mit *formatter*

□ vordefinierte Standardformatierer

- länderspezifisch – in grid.locale-xx.js definiert
- formatter: "integer" / "number" / "currency" / "date" / "email" / "link" / ...
- formatoptions: { ... }
- www.trirand.com/jqgridwiki/doku.php?id=wiki:predefined_formatter

■ Beispiel Währung

```
formatter : "currency",  
  formatoptions : {  
    suffix : " €"  
  },
```

■ Beispiel Datum/Uhrzeit – im ISO-Format wegen Sortierung liefern!

```
formatter : "date",  
  formatoptions : {  
    srcformat : 'Y-m-d\\TH:i:s',  
    newformat : 'd.m.Y H:i'  
  }
```



■ Spalten formatieren mit *formatter*

□ individuelle Formatierer

- eigene Funktion schreiben und als *formatter* zuweisen
- www.trirand.com/jqgridwiki/doku.php?id=wiki:custom_formatter

■ Beispiel: Spalte als Grafik anzeigen

```
formatter : function(cellvalue, options, rowObject) {
    return '';
}
```

■ Beispiel: Feld mit Link und Symbol anzeigen

```
formatter : function(cellvalue, options, rowObject) {
    var unid = rowObject['@unid'];
    return '<a href="Kunde.xsp?documentId='
        + unid
        + '&action=openDocument&database='
        + database
        + '" title="Dokument öffnen">'
        + '<span class="ui-icon ui-icon-extlink" style="float:left" />'
        + cellvalue + '</a>';
},
```

Demo



■ Spalten filtern

□ Filterzeile anzeigen

...

`ignoreCase: true,`

...

`})`

```
jQuery("#grid").jqGrid('filterToolbar',  
    {stringResult:true,searchOnEnter:false,defaultSearch:'cn'})
```

Kundenliste							
☐	Nachname	Vorname	Anzahl	Wert	Letzte Bestellung	PLZ	Land
	x	x	x	x	2013 x	x	Rheinland x
☐	Appel	Fridoline	56	6.388,60 €	08.04.2013 04:17	56281	Rheinland-Pfalz
☐	Behrens	Nadeschda	45	10.376,71 €	05.05.2013 13:12	54586	Rheinland-Pfalz
☐	Bollmann	Ingeruth	89	14.865,95 €	01.02.2013 06:16	66909	Rheinland-Pfalz
☐	Brummer	Anett	25	4.564,22 €	07.04.2013 02:16	65626	Rheinland-Pfalz
☐	Ehret	Arwed	68	11.765,97 €	18.06.2013 03:13	55278	Rheinland-Pfalz
☐	Hartwig	Oliver	4	419,06 €	30.10.2013 15:15	66907	Rheinland-Pfalz
☐	Haußmann	Elizabeth	84	9.294,34 €	02.08.2013 17:11	54426	Rheinland-Pfalz
☐	Holtmann	Hartfried	10	730,00 €	22.04.2013 19:13	55291	Rheinland-Pfalz
☐	Klages	Falkhard	26	6.287,06 €	27.12.2013 09:11	54675	Rheinland-Pfalz

■ Spalten filtern

□ Pagerzeile mit erweitertem Filter anzeigen und konfigurieren

```
jQuery("#grid").jqGrid('navGrid','#gridpager', {edit:false,add:false,del:false},
                        {}, {}, {}, {multipleSearch:true, multipleGroup:true});
jQuery("#gridpager_center").hide();
```

☐	Birke	Hieronymus	35	2.411,14 €	09.09.2012 07:11
--------------------------	-------	------------	----	------------	------------------

Suche...

alle

Anzahl größer 50

Wert kleiner 1000

PLZ beginnt mit 0

```
searchoptions : {
  sopt : [ "eq", "ne", "lt", "le", "gt", "ge", "nu", "nn", "in", "ni" ]
}
```

■ Spalten filtern

□ Suchstring in Ergebnissen highlighten

- jquery.highlight-4.js downloaden von [highlight: jQuery plugin](#) und in Ordner WebContent/jquery kopieren
- zu CSS hinzufügen:
`.highlight {border: 1px solid #fcd3a1; background: #fff37a;}`
- JS Code hinzufügen: stackoverflow.com/a/8505932/2065611

Kundenliste							
☐	Nachname	Vorname	Anzahl	Wert	Letzte Bestellung	PLZ	Land
☐	ha x	x	x	x	2012 x	x	x
☐	Achatz	Adelgund	83	22.355,72 €	10.08.2012 05:13	92363	Bayern
☐	Gebhard	Albina	75	11.643,76 €	30.06.2012 06:16	24616	Schleswig-Holstein
☐	Haack	Rosalie	63	764,31 €	18.12.2012 00:10	56307	Rheinland-Pfalz
☐	Hager	Agata	60	5.499,49 €	17.08.2012 02:13	04617	Thüringen
☐	Hain	Sieghard	92	22.435,40 €	14.02.2012 01:09	91456	Bayern
☐	Hampel	Siegfried	36	9.343,73 €	30.03.2012 22:16	60596	Hessen

■ Spalten sortieren

☐ Spalte als nicht sortierbar setzen

- colModel: [{name:"name1", **sortable:false**, ...

☐ Spaltentyp

- colModel: [{name:"name1", **sorttype**: "integer" / "number" / "currency" / "date", ...

☐ Sortierpfeile im Spaltenkopf immer anzeigen

- viewsortcols : [**true**, "vertical", true],

Demo



■ Gruppierung

- ☐ *grouping* im Grid definieren
- ☐ Daten müssen nach Gruppe sortiert sein
- ☐ verschachtelte Gruppen möglich

```
grouping:true,
groupingView:{
  groupField:['Land'],
  groupColumnShow:[false],
  groupText:['<b>{0}</b> ({1})'],
  groupCollapse:true,
  groupOrder:['asc'],
  groupSummary:[false]
},
```

Kundenliste						
☐	Nachname	Vorname	Anzahl	Wert	Letzte Bestellung	PLZ
☒	Hamburg (11)					
☒	Hessen (40)					
☒	Mecklenburg-Vorpommern (63)					
☒	Niedersachsen (54)					
☒	Nordrhein-Westfalen (59)					
☐	Frahm	Katarina	21	2.687,33 €	13.04.2010 07:09	32361
☐	Ritschel	Gaby	74	15.043,69 €	27.12.2009 04:09	32369
☐	Reinhardt	Gertmund	10	2.445,56 €	14.09.2009 00:09	47807
☐	Frech	Dietwulf	34	742,38 €	04.06.2010 11:09	33154
☐	Kamm	Egmont	36	3.332,32 €	09.09.2009 01:10	53925
☐	Hertwig	Rupertus	70	1.705,45 €	15.05.2012 18:10	45133



■ Tree-Grid

- ☐ beliebig tief schachtelbar
- ☐ kann unterschiedlich tief je Datensatz sein
- ☐ vergleichbar mit Notes-Kategorien, die durch "\" getrennt sind
- ☐ treeGrid-Parameter im Grid definieren

```
treeGrid: true,  
treeGridModel: 'adjacency',  
treedatatype: 'local',  
treeIcons: {leaf: 'ui-icon-blank'},  
ExpandColClick: true,  
ExpandColumn: 'Nachname'
```



■ Tree-Grid

- Daten erfordern eine spezielle Struktur, die die Hierarchie abbildet
- Beispiel für JSON-Struktur für treeGridModel "adjacency"

```
[
  {
    "id": "1",
    "Nachname": "Nordrhein-Westfalen",
    "level": "0",
    "parent": "null",
    "isLeaf": false,
    "expanded": false,
    "loaded": true
  },
  {
    "id": "1_1",
    "Nachname": "Meier",
    "level": "1",
    "parent": "1",
    "isLeaf": true,
    "expanded": true,
    "loaded": true
  },
  {
    "id": "1_2",
    "Nachname": "Schulze",
    "level": "1",
    "parent": "1",
    "isLeaf": true,
    "expanded": true,
    "loaded": true
  }
]
```

Demo



■ Zeilen per Klick bearbeiten

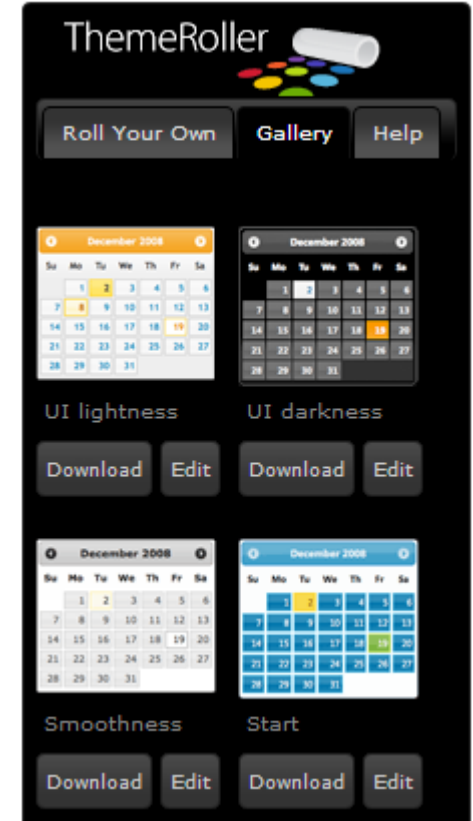
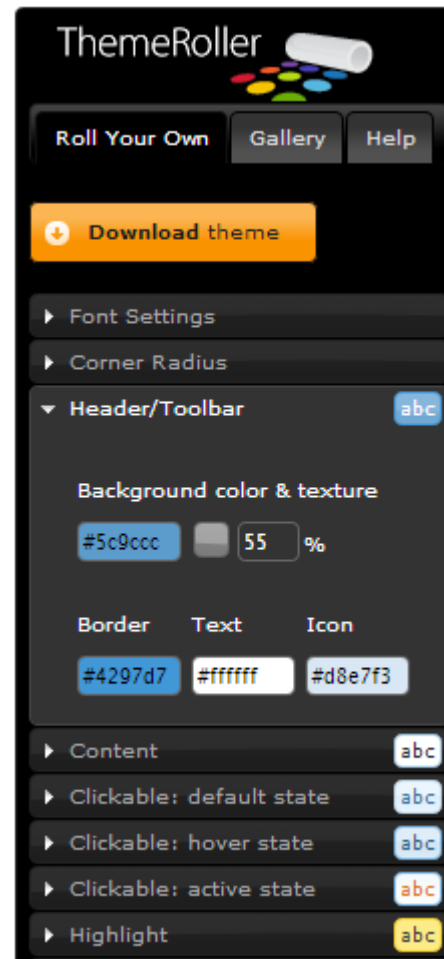
- ☐ Bearbeitbaren Spalten Eigenschaft *editable:true* geben
- ☐ Rest/Ajax Service für Operationen add/edit/del einrichten und als *editurl*: "...service..." angeben
- ☐ bei Ereignis onSelectRow vorhergehende Zeile speichern und aktuelle Zeile auf bearbeitbar setzen

Kundenliste							
☐	Nachname	Vorname	Anzahl	Wert	Letzte Bestellung	PLZ	Land
☐	Abel	Lara	15	1.661,88 €	24.06.2009 02:10	99880	Thüringen
☐	Achatz	Adelgund	83	22.355,72 €	10.08.2012 05:13	92363	Bayern
☒	Ackermann	Ishild	50	6157.07	12.05.2012 10:12	74424	Baden-Württemberg
☐	Adams	Cathleen	29	5.991,44 €	18.04.2012 22:10	54518	Rheinland-Pfalz
☐	Adler	Wiltrude	35	4.436,69 €	04.05.2011 20:11	17391	Mecklenburg-Vorpommern
☐	Aigner	Rudger	51	13.624,78 €	22.08.2009 23:13	06279	Sachsen-Anhalt
☐	Aigner	Clivia	79	22.425,80 €	04.02.2013 05:15	06721	Sachsen-Anhalt

Demo

■ Design wird durch jQueryUi bestimmt

- ☐ Standardthema auswählen
jqueryui.com/themeroller/
- ☐ im Detail anpassen
- ☐ downloaden und nach
WebContent / jquery / ccc
kopieren
- ☐ Thema als Ressource laden



- Zellen individuell per CSS gestalten
 - *class* für alle Zellen einer Spalte zuweisen:
 - `colModel: [ {name:'name1', classes:'my-spec-column', ...`
 - *class* oder *style* in Abhängigkeit des Zelleninhalts zuweisen
 - `colModel: [{name:'name1', cellattr: function(rowId, val, rawObject) {
 if (parseFloat(val) > 1000) {
 return ' class="my-great-number";
 }
}, ...`

Demo



Vergleich der Data Grids

ExtJs Grid

- Integration in .nsf sehr ähnlich zu jqGrid – nur andere Dateien und andere Syntax

```
<div id="grid" />
```

```
Ext.onReady( function() {  
  
    var store = Ext.create('Ext.data.Store', {  
        autoLoad : true,  
        autoDestroy : true,  
        autoSync : false,  
        plugins : buffRend,  
        sorters : {  
            property : 'Nachname',  
            direction : 'ASC'  
        },  
        model : 'Person',  
        proxy : {  
            type : 'rest',  
            url :  
                'RestService.xsp/kundenliste',  
            ...  
        }  
    });  
}
```

```
var grid = Ext.create('Ext.grid.Panel', {  
    renderTo : 'grid',  
    id : 'myPanel',  
    frame : true,  
    features : [ filters ],  
    height : 400,  
    width : 800,  
    title : 'Kundenliste',  
    store : store,  
    iconCls : 'icon-user',  
    columns : [ {  
        header : 'Nachname',  
        dataIndex : 'Nachname',  
        sortable : true,  
        filterable : true  
    }, {  
        text : 'Vorname',  
        ...  
    } ]  
});
```

Vergleich der Data Grids

ExtJs Grid

- Blog-Serie "EXTJS in XPages" von Dr. Mark Roden
 - <http://xomino.com/extjs-in-xpages-the-grid-series/>
 - Beispieldatenbank
<http://demo.xomino.com/xomino/Extjs.nsf/xREST.xsp>



	 Dojo Data Grid	 jqGrid Grid	 ExtJs Grid
Dokumentation	+++	++	+++
Stackoverflow Fragen	300	7.500	100
Community Topics	>30.000	>10.000	>200.000
Funktionalität	++ / +++	++	+++

- Alle Data Grids können
 - ☐ Spalten
 - sortieren
 - verschieben
 - verbergen
 - ☐ Spaltenbreite ändern
 - ☐ Zellen bearbeiten
 - ☐ Zellen individuell formatieren
 - ☐ Daten dynamisch (nach-)laden
- Alle Data Grids unterstützen
 - ☐ Mehrsprachigkeit mit i18n
 - ☐ Pfeil- und Seitentasten
- Alle Data Grids können individuell per CSS gestaltet werden
 - ☐ Themen von Basis-Frameworks (Dojo / jQueryUi)
 - ☐ CSS-Klassen für alle Elemente

					
	Data Grid	Enhanced Grid	dgrid	jqGrid Grid	ExtJs Grid
AMD - JS dynamisch laden	+	+	+	-	+
JS-Array-Daten als Quelle	-	-	+	+	+
Summenzeilen	TreeGrid	-	-	+	+
unbeschränkte Tree-Ebenen	TreeGrid	-	+	+	+
Paging mit Pager	-	+	+	+	+
Integrierte Suche/Filterung	-	+	-	+	+
pdf-Export	-	+	-	-	-
Kontext-Menüs	-	+	-	+	+
Verschachtelte Sortierung	-	+	-	-	+

Wann sind Data Grids sinnvoll und wann Alternativen besser?

- Data Grids sind eine weitere Alternative zu
 - ☐ Repeat Control
 - ☐ View Panel
 - ☐ Data Table
 - ☐ Data View
- ideal für
 - ☐ Dokumentennavigation wie in klassischen Notesansichten
 - ☐ Zahlen-orientierte Tabellen ähnlich zu Spreadsheets
- und nicht so geeignet für
 - ☐ Dokumentenlisten mit mehrzeiligem gegliedertem Inhalt

- Jetzt bitte Fragen stellen...

- ... oder später
 - ☐ E-Mail: knut.herrmann@leonso.de
 - ☐ Twitter: @KnutHerrmann
 - ☐ Skype: knut.herrmann