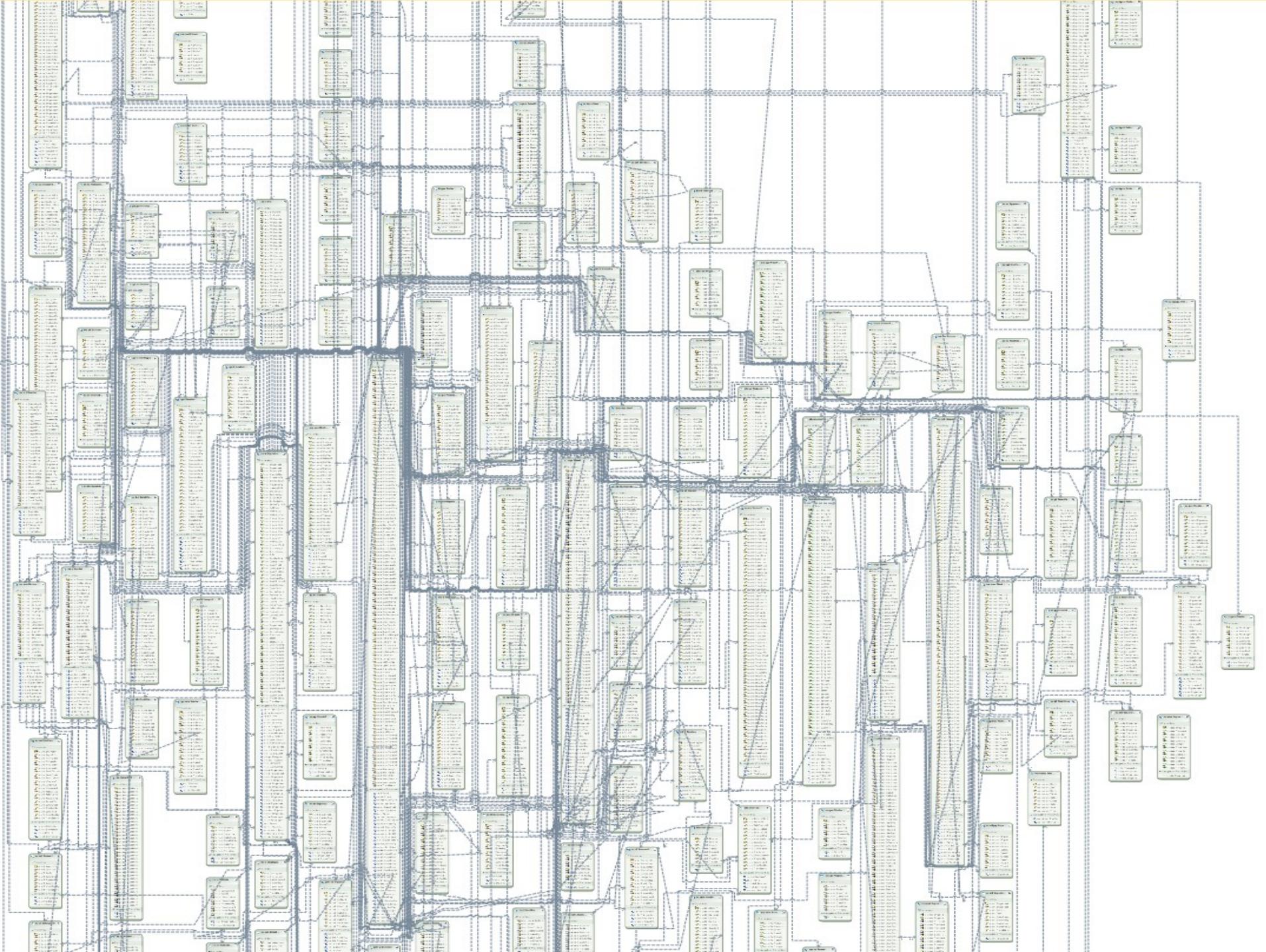


# Buffered Store Internals

## ExtJS 6



000000

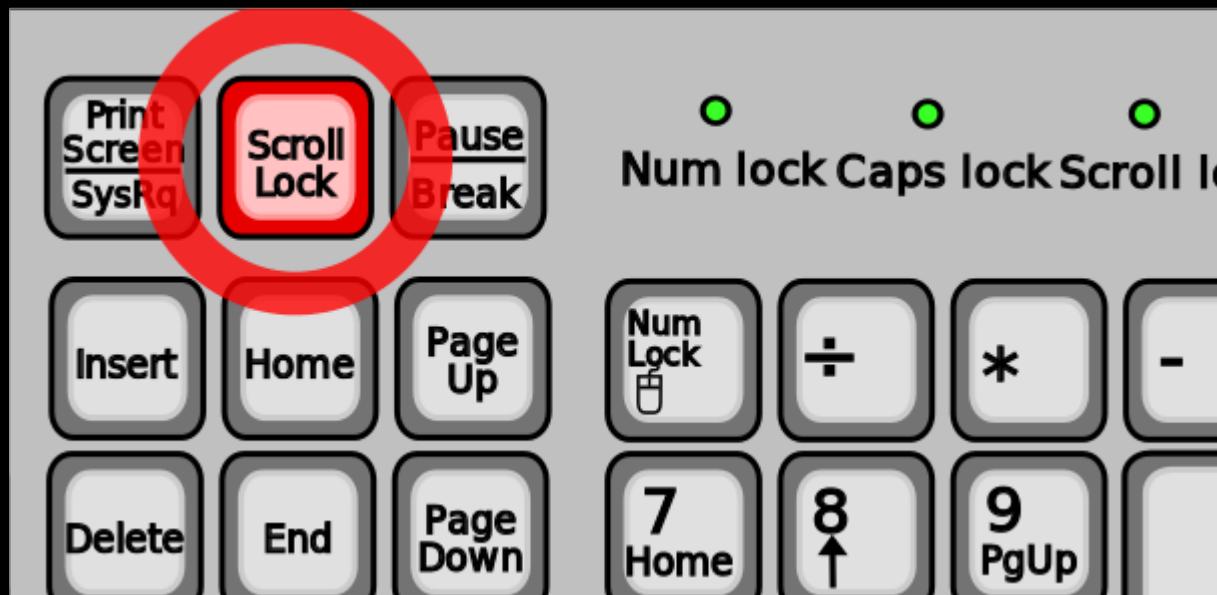
TIME 014

F T D T

MoOn

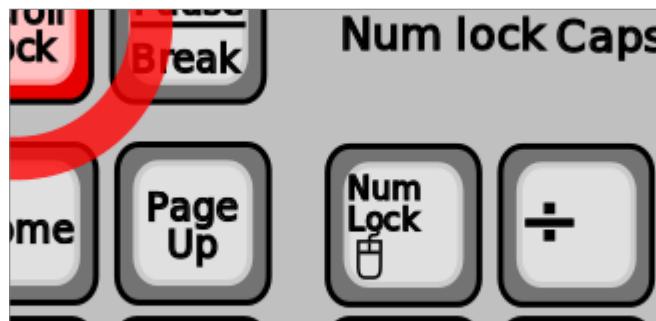
©1982 IREM CORP.





Num lock Caps lock Scroll lock

| Application Users |                    |                        |
|-------------------|--------------------|------------------------|
| Name              | Email Address      | Phone Number           |
| Lisa              | lisa@simpsons.com  | 555-111-1224           |
| Bart              | bart@simpsons.com  | 555-222-1234           |
| Homer             | home@simpsons.com  | 555-222-1244           |
| Marge             | marge@simpsons.com | 555-222-1254           |
|                   |                    |                        |
| ◀◀                | Page               | 2 of 3 ▶▶              |
|                   |                    | Displaying 5 - 8 of 12 |



| Application Users |                    |              |
|-------------------|--------------------|--------------|
| Name              | Email Address      | Phone Number |
| Lisa              | lisa@simpsons.com  | 555-111-1224 |
| Bart              | bart@simpsons.com  | 555-222-1234 |
| Homer             | home@simpsons.com  | 555-222-1244 |
| Marge             | marge@simpsons.com | 555-222-1254 |

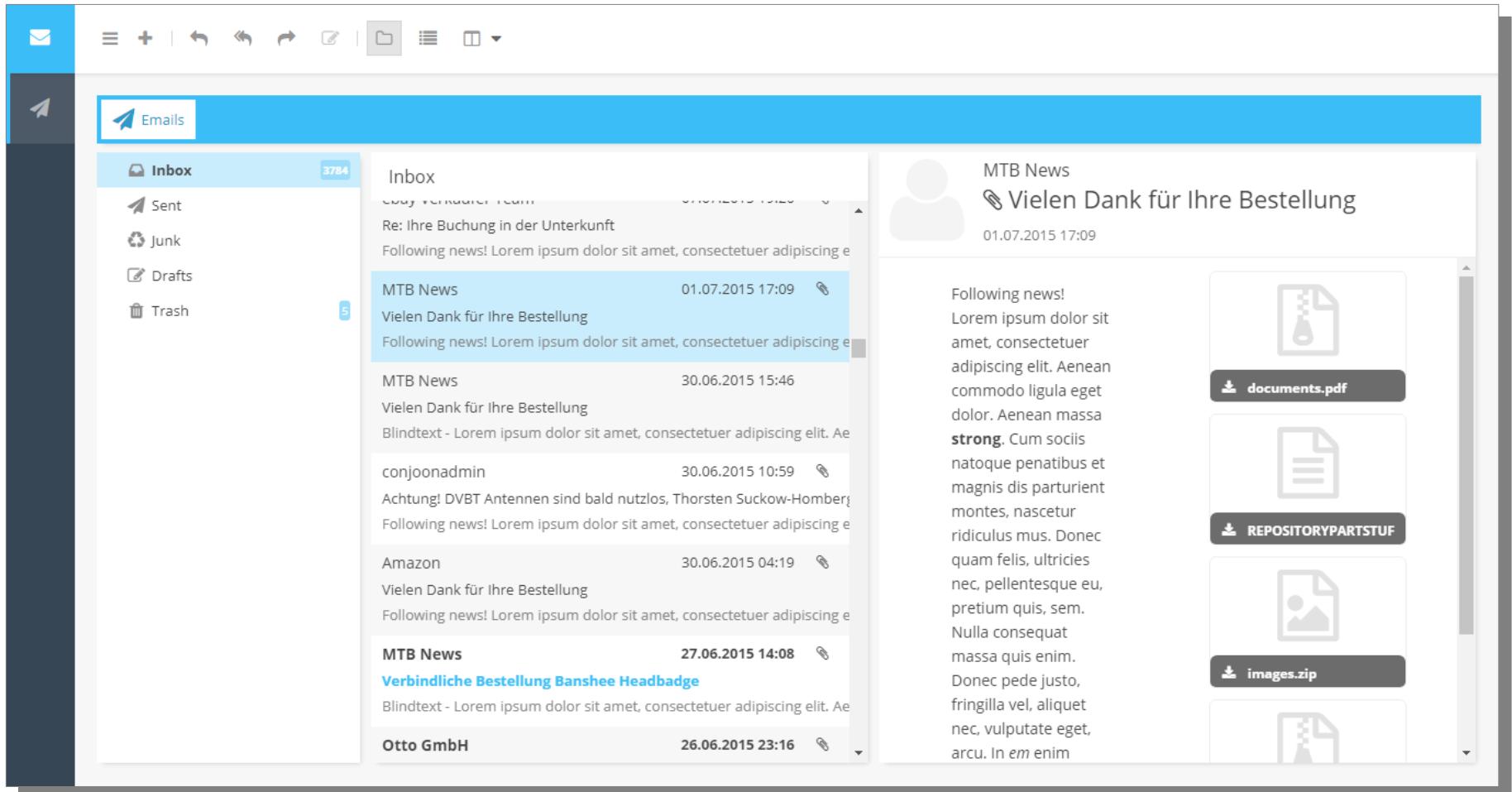
Page 2 of 3 | ▶ | 🔍 | Displaying 5 - 8 of 12



Browsing



○ = Information



The screenshot shows a desktop application window for managing emails. The top bar includes standard icons for file operations like new, open, save, and search. Below this is a navigation bar with tabs: 'Emails' (selected), 'Inbox' (with 3784 notifications), 'Sent', 'Junk', 'Drafts', and 'Trash'. The main area displays the 'Inbox' with several messages listed:

- MTB News** - 01.07.2015 17:09: Vielen Dank für Ihre Bestellung. Following news! Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- MTB News** - 30.06.2015 15:46: Vielen Dank für Ihre Bestellung. Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean massa.
- conjoonadmin** - 30.06.2015 10:59: Achtung! DVBT Antennen sind bald nutzlos, Thorsten Suckow-Homberg. Following news! Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Amazon** - 30.06.2015 04:19: Vielen Dank für Ihre Bestellung. Following news! Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- MTB News** - 27.06.2015 14:08: Verbindliche Bestellung Banshee Headbadge. Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean massa.
- Otto GmbH** - 26.06.2015 23:16: Following news!

The right side of the interface shows the details of the selected email from 'MTB News':

**MTB News**  
**Vielen Dank für Ihre Bestellung**  
01.07.2015 17:09

Following news!  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In em enim

Associated files in the sidebar:

-  documents.pdf
-  REPOSITORYPARTSTUF
-  images.zip
- 

| Application Users |                    |              |
|-------------------|--------------------|--------------|
| Name              | Email Address      | Phone Number |
| Lisa              | lisa@simpsons.com  | 555-111-1224 |
| Bart              | bart@simpsons.com  | 555-222-1234 |
| Homer             | home@simpsons.com  | 555-222-1244 |
| Marge             | marge@simpsons.com | 555-222-1254 |

←

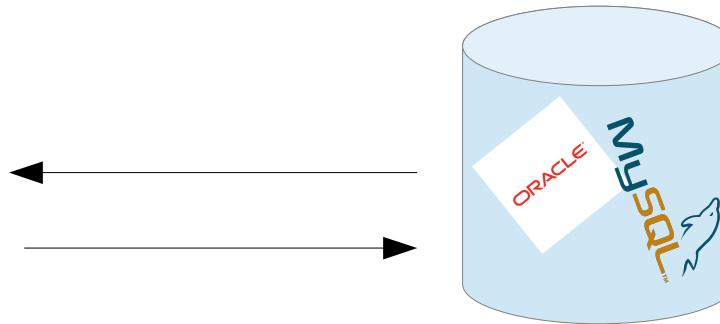
Page 2 of 3 | Displaying 5 - 8 of 12



1 .. n

| Application Users |                    |              |
|-------------------|--------------------|--------------|
| Name              | Email Address      | Phone Number |
| Lisa              | lisa@simpsons.com  | 555-111-1224 |
| Bart              | bart@simpsons.com  | 555-222-1234 |
| Homer             | home@simpsons.com  | 555-222-1244 |
| Marge             | marge@simpsons.com | 555-222-1254 |

Page 2 of 3     Displaying 5 - 8 of 12



1 .. n

### Pros:

- easy to maintain
- view on snapshots of data
- do not ask for much code wise
- user stays focused

### Cons:

- browsing through/operations on data require more user interaction
- ux not consistent (e.g. data does not shift)

| Large table |          |        |                     |   |
|-------------|----------|--------|---------------------|---|
| #           | Number ▲ | String | Date                |   |
| 1730        | 1729     | BNN    | 1980-02-21 15:36:40 |   |
| 1731        | 1730     | BNO    | 1972-02-29 02:06:53 |   |
| 1732        | 1731     | BNP    | 1974-11-11 05:41:40 |   |
| 1733        | 1732     | BNQ    | 1970-06-10 08:23:13 |   |
| 1734        | 1733     | BNR    | 1977-09-11 18:10:20 |   |
| 1735        | 1734     | BNS    | 1991-12-28 19:40:37 |   |
| 1736        | 1735     | BNT    |                     |   |
| 1737        | 1736     | BNU    |                     | <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> Buffering. Please wait...</div> |
| 1738        | 1737     | BNV    | 1994-02-08 10:49:33 |   |
| 1739        | 1738     | BNW    | 1984-10-31 06:42:54 |   |
| 1740        | 1739     | BNX    | 1978-12-29 05:02:28 |   |
| 1741        | 1740     | BNY    | 1971-03-17 12:56:12 |   |
| 1742        | 1741     | BNZ    | 1971-05-27 20:27:22 |   |
| 1743        | 1742     | BOA    | 1992-02-03 22:52:57 |   |
| 1744        | 1743     | BOB    | 1987-02-27 17:02:19 |   |

GridView (1902)  
Store  
RowSelectionModel  
Toolbar  
GridPanel  
CheckboxSelectionModel  
EditorGridPanel  
JsonReader  
DragZone

```
// skip recalculating the row index if we are currently buffering, but not if we
// are just pre-buffering
// ... also, skip recalculating if the store is currently invalid
```

| Large table |        |     |                     |
|-------------|--------|-----|---------------------|
| #           | Number |     | Date                |
| 1730        | 1729   | BNB | 1960-02-21 15:36:40 |
| 1731        | 1730   | BNO | 1972-02-29 02:06:53 |
| 1732        | 1731   | BNP | 1974-11-11 05:41:40 |
| 1733        | 1732   | BNQ | 1970-06-10 08:23:13 |
| 1734        | 1733   | BR  | 1977-09-11 18:10:20 |
| 1735        | 1734   | BNT | 1974-02-06 10:49:33 |
| 1736        | 1735   | BNV | 1976-06-13 03:02:20 |
| 1737        | 1736   | BNX | 1976-12-29 03:02:20 |
| 1738        | 1737   | BNY | 1971-03-17 12:58:12 |
| 1739        | 1738   | BNZ | 1971-05-27 20:27:22 |
| 1740        | 1739   | BOA | 1992-02-03 22:52:57 |
| 1741        | 1740   | BOB | 1987-02-27 17:02:19 |
| 1742        | 1741   |     |                     |
| 1743        | 1742   |     |                     |
| 1744        | 1743   |     |                     |

**Minor:**

- Rendering Performance

(Solution: only render the rows which can be rendered into the currently visible rectangle)

**Major:**

Mirroring state of data was key!

- Delete
- Add
- Update
- MultiSelect (Range)

## ExtJS 4:

The Sencha Platform homepage features a large background image of a man working on a laptop. Overlaid on the top right is a navigation bar with links: SUPPORT, FORUM, RESOURCES, BLOG, CONTACT US, TRY FOR FREE, and BUY NOW. To the left of the navigation is a sidebar with categories: Integrated Products, Frameworks, Tools, Management, and Sencha Market. The main content area has a heading "Sencha Platform" and a sub-section "Manage Your Web Application Lifecycle". Below this are images of various devices (laptop, tablet, smartphone) displaying different web-based dashboards and data visualizations.

- BufferedRenderer (rendering performance)
- BufferedStore

Rapidly design, develop, and manage cross-platform applications.



Genentech



# ExtJS 4:

SUPPORT FORUM RESOURCES BLOG CONTACT US

Products Services Training Customers Company Store TRY FOR FREE BUY NOW

- BufferedRenderer (rendering performance)
- BufferedStore

```
add: function () {
    Ext.raise('add method may not be called on a buffered store - the store is a map of remote data');
},
insert: function () {
    Ext.raise('insert method may not be called on a buffered store - the store is a map of remote data');
},
```



Genentech



add

MTB News

Vielen Dank für Ihre Bestellung

01.07.2015 17:09

Following news!  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In em enim

remove

27.06.2015 14:08

**Verbindliche Bestellung Banshee Headbadge**

Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In em enim

update

CISCO

EXPLORE THE SENCHA PLATFORM

ExtJS 6

## Requirements

- Grid should immediately show new data while keeping BufferedStore functionality
- Grid should immediately show changes made to data
- Grid should immediately update its View properly when data was removed



## **Update**

- Does not change data set (i.e. number of data)
- Changes data values, reflected by implemented renderer



## **Improvements**

- consider manual sort of data once data was changed (`remoteSort : true`), since this does not happen automatically

add

MTB News

Vielen Dank für Ihre Bestellung

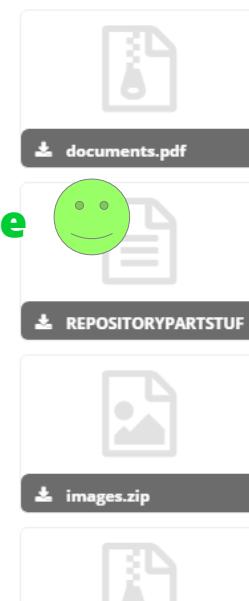
01.07.2015 17:09

Following news!  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In em enim

remove



update



Inbox

RECENTLY VIEWED

- Re: Ihre Buchung in der Unterkunft  
Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e
- MTB News 01.07.2015 17:09  
Vielen Dank für Ihre Bestellung  
Following news! Lorem ipsum dolor sit amet, consectetur adipisci
- MTB News 30.06.2015 15:46  
Vielen Dank für Ihre Bestellung  
Blindtext - Lorem ipsum dolor sit amet, consectetur adipisci
- conjoonadmin 30.06.2015 10:59  
Achtung! ➔ Antennen sind bald nutzlos, Thorsten Suckow-Homber  
Following news! Lorem ipsum dolor sit amet, consectetur adipisci
- Amazon 30.06.2015 04:19  
Vielen Dank für Ihre Bestellung  
Following news! Lorem ipsum dolor sit amet, consectetur adipisci
- MTB News 27.06.2015 14:08  
**Verbindliche Bestellung Banshee Headbadge**  
Blindtext - Lorem ipsum dolor sit amet, consectetur adipisci
- Otto GmbH 26.06.2015 23:16



Inbox

3784

Sent

Junk

Drafts

Trash

5

CISCO

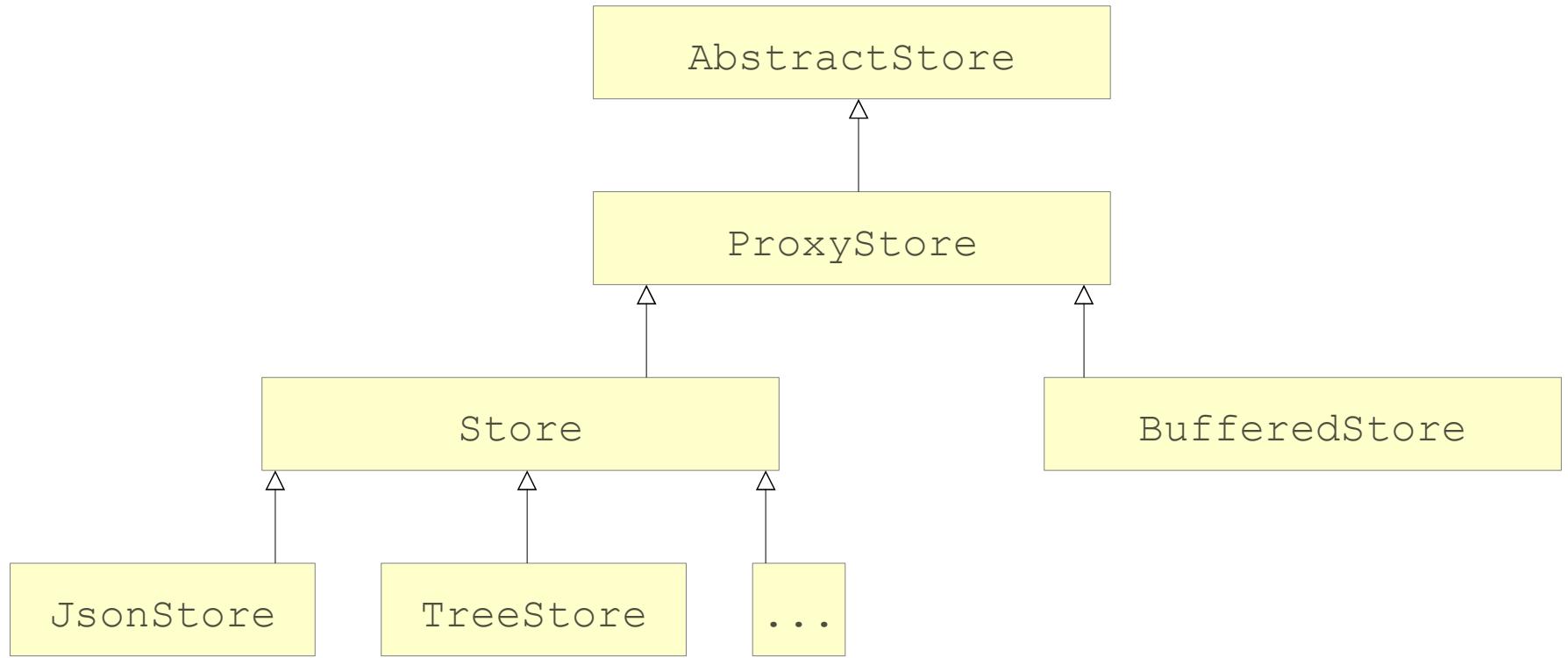
TCS

KLM

## Add

- Changes data set (i.e. number of data)
- Challenge: Find data position in the local store and hope assume it is the same on the server (remote operations such as „sorting“ active)
- Find data which needs to be swapped out by the newly added data (BufferedStore contains extracts from remote data)





Store

applyData :  
Ext.util.Collection

add()  
insert()  
removeAt()

BufferedStore

Store

applyData :  
Ext.util.Collection

add()  
insert()  
removeAt()

BufferedStore

applyData :  
Ext.data.PageMap

Ext.util.LruCache

Ext.util.HashMap

## Ext.util.LruCache

- maintains most recently accessed items and discards Least Recently Used (maxSize-property)
- maps pages, not store data:  
scrolling → reusePage/loadPage/discardPage
- adding/ removing data solely works on pages, not „store data“

## Ext.util.LruCache

```
store.loadPage(1);
console.log(store.getData().map)

1: {prev: {...}, next: {...}, key: 1, value: Array(25)}
2: {prev: {...}, next: {...}, key: 2, value: Array(25)}
3: {prev: {...}, next: {...}, key: 3, value: Array(25)}
4: {prev: {...}, next: {...}, key: 4, value: Array(25)}
5: {prev: null, next: {...}, key: 5, value: Array(25)}
...
12: {prev: {...}, next: null, key: 12, value: Array(25)}
```

## Ext.util.LruCache

Pages

```
store.loadPage(1);
console.log(store.getData().map)

1: {prev: {...}, next: {...}, key: 1, value: Array(25)}
2: {prev: {...}, next: {...}, key: 2, value: Array(25)}
3: {prev: {...}, next: {...}, key: 3, value: Array(25)}
4: {prev: {...}, next: {...}, key: 4, value: Array(25)}
5: {prev: null, next: {...}, key: 5, value: Array(25)}
...
12: {prev: {...}, next: null, key: 12, value: Array(25)}
```

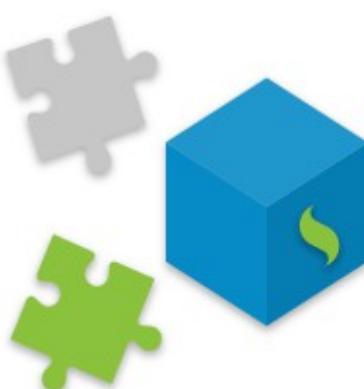
Ext.data.Model

## Ext.util.LruCache

```
store.loadPage(1);
console.log(store.getData().indexMap)

{1 : 0, 2 : 1, 3 : 2, 4 : 3, 5 : 4, ..., 300 : 299}
```

```
<table id="tableview-1018-record-2" ... data-recordid="2" data-recordindex="3" class="...">
```



## Add (concrete)

- Get Sorter – sort property, direction
- Find insert index in LruCache.map:
  - iterate through pages
  - compare values based on sort information
  - if insert position found, add new record, remove one record (based on insert strategy, e.g. insert at page 2, index 5, remove record at position 25 to keep pageSize)
- Update indexMap so view can work with newly added data
- Refresh view

```
addSorted : function(record) {
    var me          = this,
        data         = me.getData(),
        index        = me.findInsertIndexInPagesForRecord(record),
        storeIndex   = 0,
        page, pos, values;

    if (index === null) {
        return null;
    }

    page  = index[0];
    pos   = index[1];
    values = data.map[page].value;

    Ext.Array.splice(values, pos, 0, record);
    values.pop();

    // Update the indexMap
    for (var startIdx in data.map) {
        for (var i = 0, len = data.map[startIdx].value.length; i < len; i++) {
            data.indexMap[data.map[startIdx].value[i].internalId] = storeIndex++;
        }
    }

    return record;
},
```





## map:

```
addSorted : function(record) {
    var me          = this,
        data         = me.getData(),
        index        = me.findInsertIndexInPagesForRecord(re
        storeIndex = 0,
        page, pos, values;

    if (index === null) {
        return null;
    }

    page   = index[0];
    pos    = index[1];
    values = data.map[page].value;

    Ext.Array.splice(values, pos, 0, record);
    values.pop();

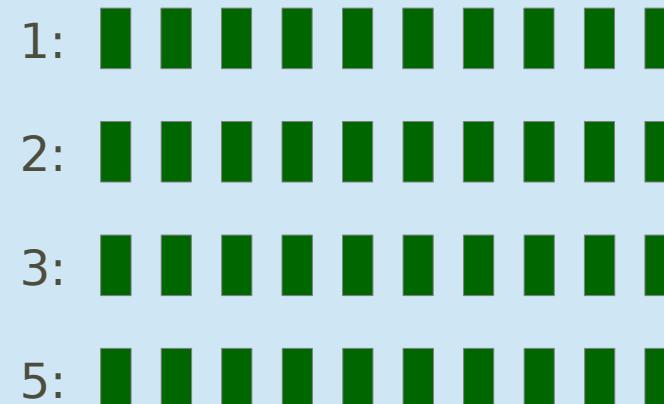
    // Update the indexMap
    for (var startIdx in data.map) {
        for (var i = 0, len = data.map[startIdx].value.length; i < len; i++) {
            data.indexMap[data.map[startIdx].value[i].internalId] = storeIndex++;
        }
    }

    return record;
},
```

## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

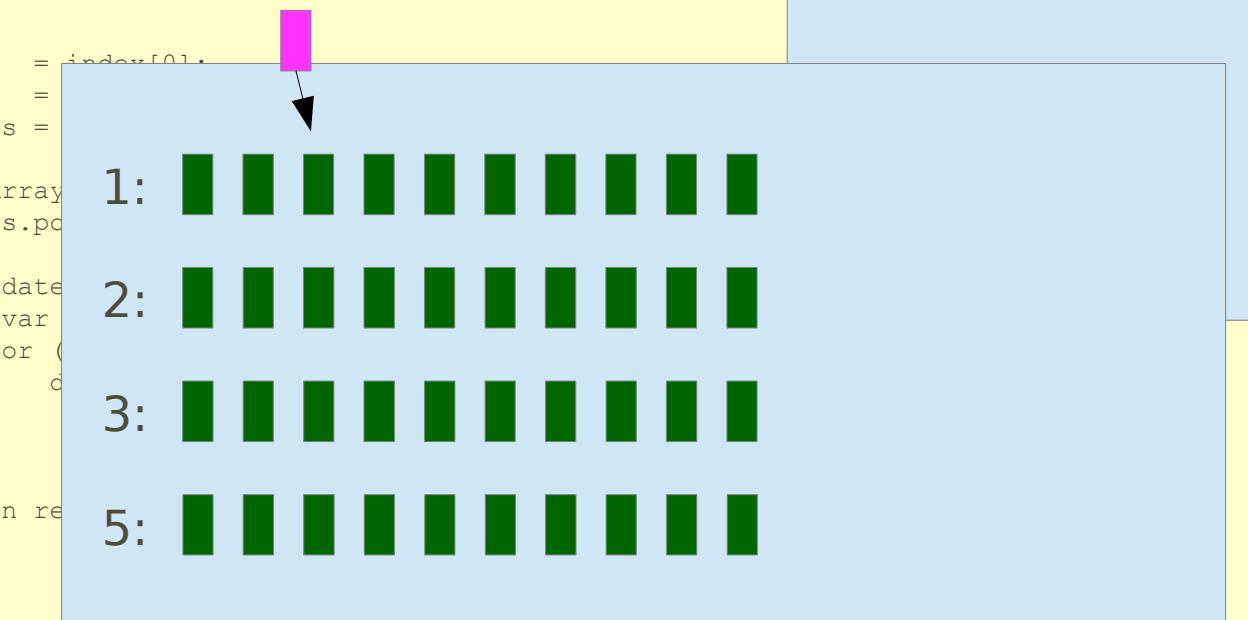
```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var i = 0; i < page.length; i++) {  
        for (var j = 0; j < page[i].length; j++) {  
            page[i][j] = values[i];  
        }  
    }  
  
    return re  
},
```



## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

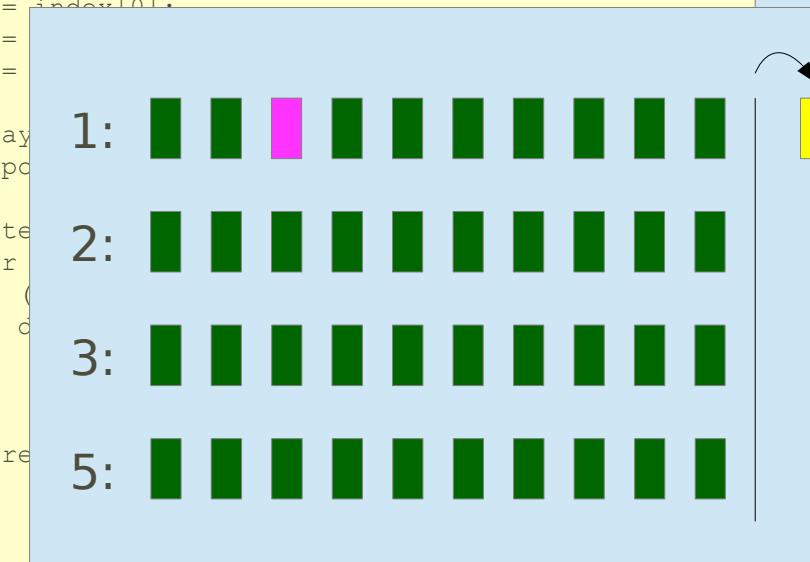
```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var  
        for (  
            c  
        }  
    }  
  
    return re  
},
```



## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

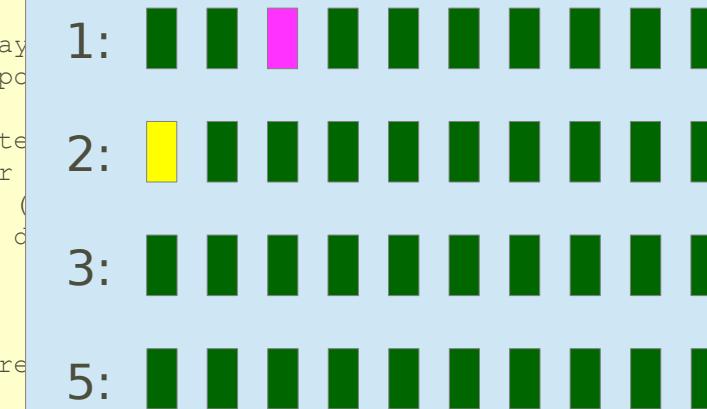
```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var i = 0; i < page.length; i++) {  
        for (var j = 0; j < page[i].length; j++) {  
            if (page[i][j] >= record) {  
                page[i][j] = record;  
                break;  
            }  
        }  
    }  
  
    return re  
},
```



## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

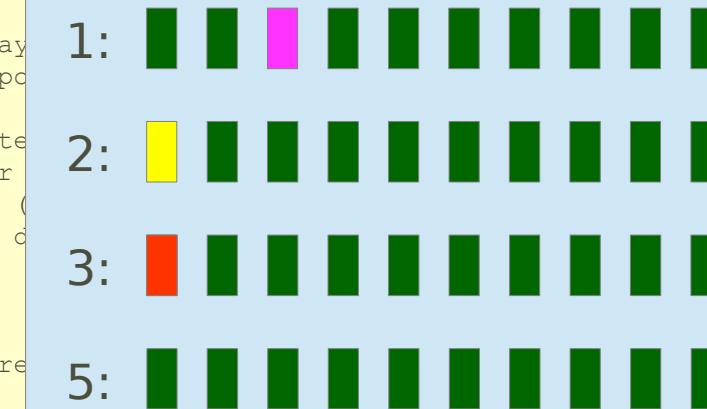
```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var i = 0; i < page.length; i++) {  
        for (var j = 0; j < page[i].length; j++) {  
            if (page[i][j] === record) {  
                page[i][j] = null;  
            }  
        }  
    }  
  
    return re  
},
```



## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

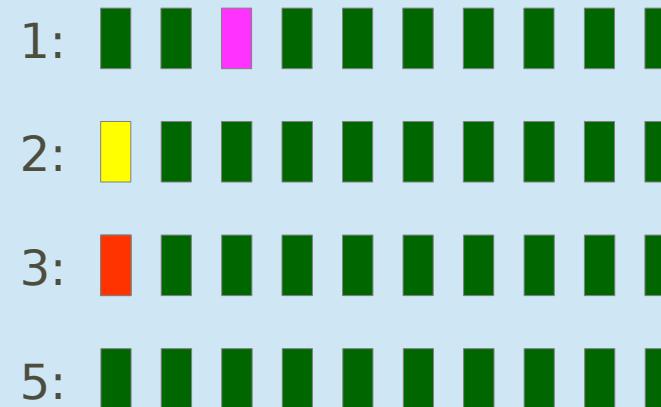
```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var i = 0; i < page.length; i++) {  
        for (var j = 0; j < page[i].length; j++) {  
            page[i][j] = record;  
        }  
    }  
  
    return re  
},
```



## map:

```
1 : {values : [...]},  
2 : {values : [...]},  
3 : {values : [...]},  
5 : {values : [...]}
```

```
addSorted : function(record) {  
    var me          = this,  
        data         = me.getData(),  
        index        = me.findInsertIndexInPagesForRecord(re  
        storeIndex = 0,  
        page, pos, values;  
  
    if (index === null) {  
        return null;  
    }  
  
    page   = index[0].  
    pos    =  
    values =  
    Ext.Array  
    values.po  
  
    // Update  
    for (var i = 0; i < page.length; i++) {  
        for (var j = 0; j < page[i].length; j++) {  
            if (page[i][j] === record) {  
                page[i][j] = null;  
            }  
        }  
    }  
  
    return re  
},
```



poof



add



remove

update

Boeing  
Forever New Frontiers

CISCO

TIAA

KLM

Emails

Inbox 3784

Inbox

Re: Ihre Buchung in der Unterkunft

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

MTB News 01.07.2015 17:09

Vielen Dank für Ihre Bestellung

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

MTB News 30.06.2015 15:46

Vielen Dank für Ihre Bestellung

Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ae

conjoonadmin 30.06.2015 10:59

Achtung! ➔ Antennen sind bald nutzlos, Thorsten Suckow-Homber

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

Amazon 30.06.2015 04:19

Vielen Dank für Ihre Bestellung

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

MTB News 27.06.2015 14:08

**Verbindliche Bestellung Banshee Headbadge**

Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ae

Otto GmbH 26.06.2015 23:16

MTB News 01.07.2015 17:09

Vielen Dank für Ihre Bestellung

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

conjoonadmin 30.06.2015 10:59

Achtung! ➔ Antennen sind bald nutzlos, Thorsten Suckow-Homber

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

Amazon 30.06.2015 04:19

Vielen Dank für Ihre Bestellung

Following news! Lorem ipsum dolor sit amet, consectetur adipiscing e

MTB News 27.06.2015 14:08

**Verbindliche Bestellung Banshee Headbadge**

Blindtext - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ae

Otto GmbH 26.06.2015 23:16

Profile picture of MTB News

MTB News

Vielen Dank für Ihre Bestellung

01.07.2015 17:09

Following news!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa **strong**. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In em enim

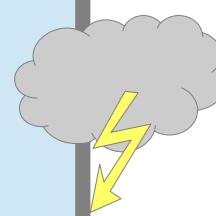
documents.pdf

REPOSITORYPARTSTUF

images.zip

## Requirements

- Grid should immediately show new data while keeping BufferedStore functionality 
- Grid should immediately show changes made to data 
- Grid should immediately update its View properly when data was removed





## LruCache shifting problem

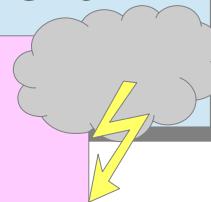
- Pages 1-12 are available, page 1 is shown
- Removing data in page 1 - solution: shift data down
- Load data from server and append at the end





## LruCache shifting problem

- Pages 1-12 are available, page 1 is shown
  - Removing data in page 1 - solution: shift data down
  - Load data from server and append at the end
- Computing pages to reload
  - Does LRU discard or keep existing pages (add new, remove lru)
  - Backend Interaction
  - Asynchronous processes
  - Waiting for server response
  - User scrolls through data while request is active
  - Blocking input
- 





## LruCache shifting problem

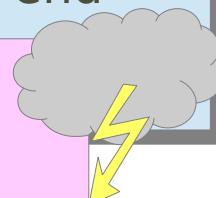
- P
- R
- L

### Fragmentation!

LruCache-map may look like

|     |
|-----|
| 1   |
| 2   |
| 3   |
| •   |
| 11  |
| 12  |
| 21  |
| 22  |
| 23  |
| 24  |
| ... |

vn  
t data  
e end





## L **Improvements**

- - after removing data, already buffered pages need to be purged **once a new page loads**
- d



```
1 : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 [VISIBLE]  
2 : 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
```

```
DEL      Page 1, id 3  
SCROLL  Page 3 (Request Parameters: Start 21, Limit 10)
```

```
1 : 1, 2, [3], 4, 5, 6, 7, 8, 9, 10  
2 : 11, 12, 13, 14, 15, 16, 17, 18, 19, 20  
3 : 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 [VISIBLE]
```

-> Record 21 missing since data shifted in the backend

\*[n] = marked as deleted, i.e. visible in the UI, but n/a from backend

add



remove



update



documents.pdf



REPOSITORYPARTSTUF



images.zip



Sencha Touch Email Application Mockup

The application interface includes a top navigation bar with icons for back, forward, search, and file operations. Below is a header bar with a 'Emails' tab. On the left is a sidebar with links for Inbox (3784), Sent, Junk, Drafts, and Trash. The main area shows an 'Inbox' list with several messages:

- MTB News - Vielen Dank für Ihre Bestellung (01.07.2015 17:09)
- MTB News - Vielen Dank für Ihre Bestellung (30.06.2015 15:46)
- conjoonadmin - Achtung! Antennen sind bald nutzlos, Thorsten Suckow-Homberg (30.06.2015 10:59)
- Amazon - Vielen Dank für Ihre Bestellung (30.06.2015 04:19)
- MTB News - Verbindliche Bestellung Banshee Headbadge (27.06.2015 14:08)
- Otto GmbH - (26.06.2015 23:16)

Each message item has a small edit icon. To the right of the inbox is a detailed view of the first message from MTB News, showing the recipient's profile picture, the subject, date, and body text. The body text contains placeholder text and a strong tag. Below the message are download links for 'documents.pdf' and 'images.zip'. A large green smiley face icon is overlaid on the application window.