

Boost your Notes database with SQL-like features

Stanislav Marszalek
TCL DigiTrade

SUTOL
Conference2015

November 11, Barceló Hotel Praha

Thanks to our sponsors!

Ytria



save secure automate



Stanislav Marszalek

- TCL DigiTrade
- 20 years on Domino
- smarszalek@tcl-digitrade.com
- www.tcl-digitrade.com

Agenda

- Demo application
- Overview of the components
- JSON in Notes
- Vue.js
- Google Charts
- AlaSQL
- Demo application

How data flow?

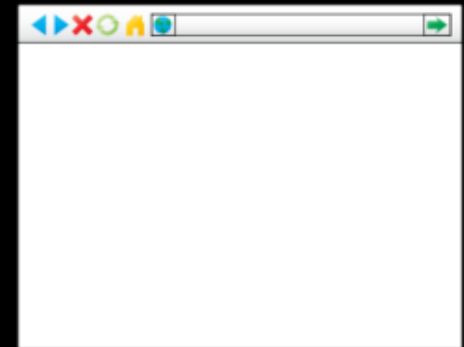
1. Notes
database



2. JSON

```
{
  "name": "Product",
  "properties": {
    "id": {
      "type": "number",
      "description": "Product identifier",
      "required": true
    },
    "name": {
      "type": "string",
      "description": "Name of the product",
      "required": true
    },
    "price": {
      "type": "number",
      "minimum": 0,
      "required": true
    }
  }
}
```

3. WEB
browser



AlaSQL
Vue.js

1. Notes database

- Documents in views
- Not incredibly fast database



2. JSON

- Xpages application
- Grab the data using Java
 - db.Search, db.FTSearch
 - Walk the view – (view entry – the fastest)

```
//walk the view
while(entry!=null ){

    detail=new HashMap<String,Object>();
    Name n=au.getSession().createName((String) entry.getColumnValues().get(0));
    Object user=n.getCommon();

    if (!entry.getColumnValues().get(2).toString().equals("") ){ //date has to be there
        DateTime date_ass= (DateTime) entry.getColumnValues().get(2);
        JDateTime jdt=new JDateTime(date_ass.toJavaDate());

        Object month=jdt.getMonth();
        Object year=jdt.getYear();

        Double hours=0.0;
        if (!entry.getColumnValues().get(3).toString().equals("") ){
            hours=(Double) entry.getColumnValues().get(3);
        }
    }
}
```

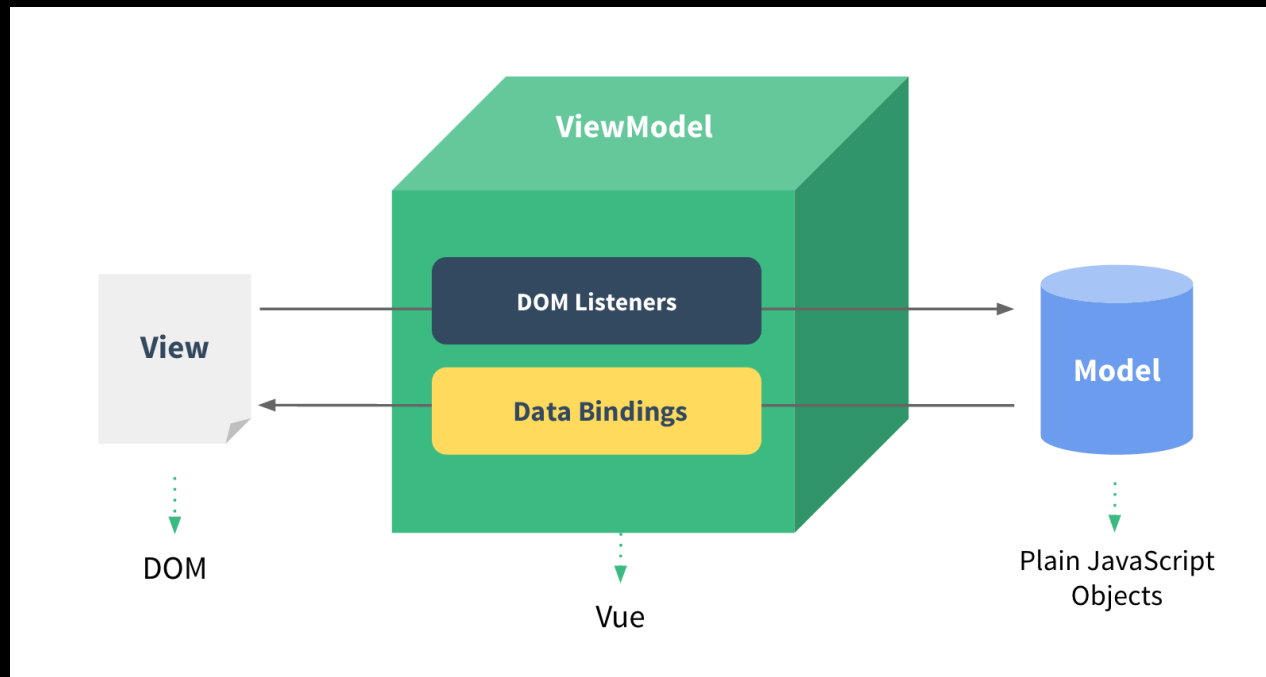
3. Web browser

- Get data from JSON
- Fill in array
- Run SQL commands using AlaSQL
- Display results in chart

```
46 GRAPH.setChartData = function(groupby, where, label){
47
48     var chart_data = [];
49     var grp="";
50     if (groupby!=""){
51         grp="GROUP BY " + groupby;
52         ord=" ORDER BY " + groupby + " ASC";
53     }
54
55     chart_data = new google.visualization.DataTable();
56     chart_data.addColumn('string', label);
57     chart_data.addColumn('number', 'Hours');
58     // chart_data.addColumn('number', 'No.'):
59
60     var row_counter = 0;
61     var select="SELECT status,year, month, name,activity,company, SUM(hours) AS hours, COUNT(name) as qty FROM ?
62     alasql(select,[GRAPH.data],
63         function (xlData) {
64             var items = [];
65             chart_data.addRows(xlData.length);
66
67             xlData.forEach(function (val) {
68                 chart_data.setCell(row_counter, 0, val[groupby].toString());
69                 chart_data.setCell(row_counter, 1, val.hours.toFixed(2));
70                 // chart_data.setCell(row_counter, 2, val.qty);
71                 row_counter = row_counter + 1;
72             });
73
74         });
75
76     GRAPH.chart_data=chart_data;
77     return select;
78 }
```


Vue.js

- Reactive data-binding system that keeps your data and the DOM in sync.



Vue.js

HTML

HTML

```
<ul id="example-1">
  <li v-for="item in items">
    {{ item.message }}
  </li>
</ul>
```

Javascript

JS

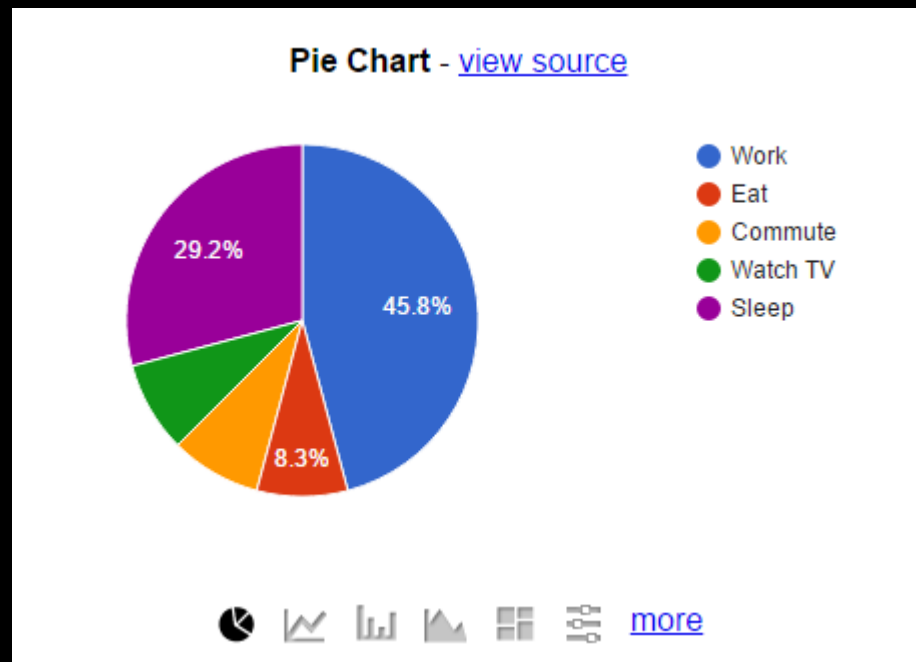
```
var example1 = new Vue({
  el: '#example-1',
  data: {
    items: [
      { message: 'Foo' },
      { message: 'Bar' }
    ]
  }
})
```

Result

- Foo
- Bar

Google Charts

- Display live data on your site
- Rich gallery of interactive charts and data tools



AlaSQL



alasql.org

What is AlaSQL?

- **AlaSQL - is a lightweight client-side in-memory SQL database.**
- **AlaSQL is written with JavaScript and does not use browser WebSQL database.**
- **AlaSQL is fully functional compact sql server with JOINS, GROUPs, UNIONs, ANY, ALL, IN, subqueries and very limited transactions support.**
- **AlaSQL supports ROLLUP(), CUBE() and GROUPING SETS() functions**
- **AlaSQL works with all modern versions of browsers and mobile iOS and Android.**
- **AlaSQL is fast, because it uses some optimization methods.**

AlaSQL – easy use

Javascript

```
var data = [{ dep: 'A', qt: 10, price: 5},
            { dep: 'A', qt: 5,  price: 2.30 },
            { dep: 'B', qt: 3,  price: 2.20 },
            { dep: 'C', qt: 1,  price: 4  },
            { dep: 'C', qt: 4,  price: 10 }];

var res = alasql('SELECT dep, SUM(qt) AS qt, SUM(qt*price) AS
amt, AGGR(amt/qt) AS price
FROM ?
GROUP BY dep',[data]);
```

Result

```
[{"dep":"A","qt":15,"amt":61.5},
{"dep":"B","qt":3,"amt":6.6000000000000005},
{"dep":"C","qt":5,"amt":44}]
```

AlaSQL functions

- Data manipulation - array filtering, grouping, ordering
- Data import and export - TXT, CSV, XLS, HTML, JSON
- Search in JSON arrays and objects
- SQL database - in-memory database

Data import from the file

```
Var res=alasql('SELECT * FROM CSV("my.csv". {headers:true})');
```

```
Var res=alasql('select * from  
xlsx("cities.xlsx",{sheetid:"Sheet2"},,, [],function(data){});
```


Data manipulation

Filtering

```
var res = alasql('SELECT * FROM ? WHERE value = 55',[data]);
```

Grouping

```
var res = alasql('SELECT [date], name FROM ? GROUP BY [date], name', [data]);
```

Joining

```
var res = alasql('SELECT id, FIRST(Name) AS Name, COUNT(*) AS [Count] FROM ? facilities \ JOIN ? hotels ON facilities.id IN hotels.WebFacilities \ GROUP BY id', [facilities,hotels]);
```

Complex

```
var res = alasql('SELECT category, sum(price) AS price, sum(qty) as qty \ FROM ? \ GROUP BY category \ ORDER BY price DESC',[data]);
```

Tie it all together

- Xpage to generate JSON data
- Xpage to display data
 - Alasql.js
 - Vue.js
 - googleCharts.js

Questions ?