DB2JS STEP BY STEP

初始化环境

1. 设置 tomcat server.xml 中 connector 的 URIEncoding=”utf-8”。
2. 在 postgres 中建立数据库，名为 bookstore, 从 sample 所给数据库备份文件 bookstore.backup 恢复该数据库。
3. 在 web/META-INF/context.xml 中修改数据库连接。

最简单的查询

建立一个名为 book.dbjs 的文件，输入：

```
dbjs.fetch = function(params){
	sql{.
		select * from book
	.}
	return this.query(sql, params);
}
```

打开浏览器，输入网址 http://localhost:8080/db2js/bookstore/book.dbjs?_m=fetch （应根据自己实际路径调整网址），即得到：

增加查询条件

修改 book.dbjs，扩充为如下代码：

```
dbjs.fetch = function(params){
    sql{.
        select * from book where 1=1
        code{.
            if(params.title){
                sql{. and strpos(title, :title) > 0    .}
            }
        .}
    .}
    return this.query(sql, params);
}
```

现在，输入网址 http://localhost:8080/db2js/bookstore/book.dbjs?_m=fetch&title=天，即得到：

```
{
 - columns: [
    - {
          name: "id",
          type: "INT"
      },
    - {
          name: "title",
          type: "STRING"
      },
    - {
          name: "kind",
          type: "STRING"
      },
    - {
          name: "publish_date",
          type: "DATE"
      },
    - {
          name: "isbn",
          type: "STRING"
      },
    - {
          name: "remarks",
          type: "STRING"
      },
    - {
          name: "author",
          type: "INT"
      }
   ],
 - rows: [
    - {
          id: 6,
          title: "天龙八部",
          kind: "k",
          publish_date: "1983-02-01T16:00:00.000Z",
          isbn: "23231",
          remarks: null,
          author: 2
      }
   ]
}
```


可见，参数已经映射为 params 对象参数。

db2js 中可以使用 request, response, session, out, application 等几个活动页面对象，其中 request['param'] 效果相当于 request.getParameter('param')，session['attr'] 相当于 session.getAttribute('attr') 和 session.setAttribute('attr', value)，同样可以写为 request.param,session.attr。

经典方式 CRUD

经典方式是指完全靠表单提交，页面跳转的古老办法展现内容。

列表页面

列表页面使用 JSSP 技术实现。JSSP 是 JavaScript Server Page 的简称，采用 js 语言编写服务器端脚本，与 jsp 类似。不同之处主要有：

1. JSSP 用[% %] 作为代码块括号
2. JSSP 有[%= %]，[%~ %] [%- %] 等三种输出方式
3. JSSP 是 DB2JS 的变种。JSSP 页面转换后生成的代码为

```
db2js.jssp = function(params){
    out.print('<html>\n');
    out.print('<head>\n');
    ...
}
```

因此，jssp 也可以使用所有 db2js 技术。

在 book.dbjs 文件夹增加一个名为 book.jssp 的文件，输入：

```
<!DOCTYPE html>
<html>

<head>
    <title>书</title>
    <meta charset="utf-8">
    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>
</head>
<body>

[% var r = this.callDbjs('book.dbjs', 'fetch', params); %]

<div class="container">
    <table class="table table-striped">
    <caption>book</caption>
    <thead>
        <tr>
    [%  r.columns.forEach(function(col){ %]
        <th>[%= col.name %]</th>
    [%  }); %]
        </tr>
    </thead>
```

```
        [%    r.rows.forEach(function(row){ %]
        <tr>
            [%    r.columns.forEach(function(col){ %]
                <td>[%= row[col.name] %]</td>
            [%    }); %]
        </tr>

        [%    }); %]

        </table>
</div>
</body>

</html>
```

打开 http://localhost:8080/db2js/bookstore/book.jssp 即得到：

book

id	title	kind	publish_date	isbn	remarks	author
1	纯粹理性批判	p	Tue Jul 28 2015 00:00:00 GMT+0800 (CST)	1112233	null	4
7	红楼梦	f	Thu Feb 03 1977 00:00:00 GMT+0800 (CST)	2323121	null	1
2	判断力批判	p	Wed Sep 08 1993 00:00:00 GMT+0800 (CST)	2323231	null	4
3	飘	f	Thu May 06 1982 00:00:00 GMT+0800 (CST)	22313	null	3
4	神雕侠侣	k	Sun Feb 02 1992 00:00:00 GMT+0800 (CST)	242313	null	2
5	书剑恩仇录	k	Mon Apr 05 1982 00:00:00 GMT+0800 (CST)	3131	null	2
6	天龙八部	k	Wed Feb 02 1983 00:00:00 GMT+0800 (CST)	23231	null	2

打开 http://localhost:8080/db2js/bookstore/book.jssp?title=天 即得到：

book

id	title	kind	publish_date	isbn	remarks	author
6	天龙八部	k	Wed Feb 02 1983 00:00:00 GMT+0800 (CST)	23231	null	2

由于经典方式不是 db2js 主流模式，本示例不包含如何解决词典转换、日期转换等问题。

新增表单

在列表页面增加一个超链接：

```
<a class="btn btn-primary" href="book-add.html">+</a>
```

在同一文件夹新建一个 html，命名为 book-add.html，输入内容如下：

```
<!DOCTYPE html>
<html>

<head>
```

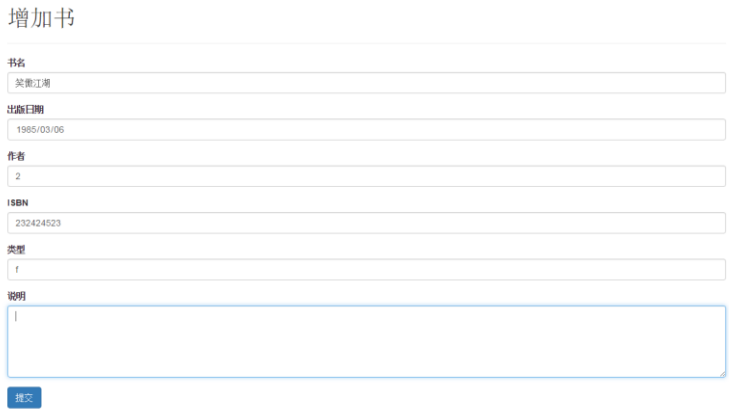
```
    <title>增加书</title>
    <meta charset="utf-8">
    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <h1>增加书</h1>
        <hr>
        <form action="book.dbjs?_m=create" method="post">
          <div class="form-group">
            <label>书名</label>
            <input name="title" type="text" class="form-control">
          </div>
          <div class="form-group">
            <label>出版日期</label>
            <input name="publish_date" type="date" class="form-control">
          </div>
          <div class="form-group">
            <label>作者</label>
            <input name="author" type="text" class="form-control">
          </div>
          <div class="form-group">
            <label>ISBN</label>
            <input name="isbn" type="text" class="form-control">
          </div>
          <div class="form-group">
            <label>类型</label>
            <input name="kind" type="text" class="form-control">
          </div>
          <div class="form-group">
            <label>说明</label>
            <textarea name="remarks" rows="5" cols="12" class="form-control"></textarea>
          </div>
          <input type="submit" class="btn btn-primary">
        </form>
    </div>
</body>

</html>
```

由于经典方式不是 db2js 主流模式，本示例不包含如何解决下拉作者列表、下拉词典列表、输入校验等问题。

在 book.dbjs 增加如下接收代码：

```
dbjs.create = function(rcd){
    logger.info('add book ' + JSON.stringify(rcd));
    rcd.publish_date = parseDate(rcd.publish_date);
    rcd.author *= 1;
    this.insertRow('book', rcd, ['title', 'author', 'publish_date', 'kind', 'isbn', 'remarks'])
}
```

现在点击列表页面的 + 按钮，输入一本新书：



点击提交按钮，得到如下返回：

```
{
    success: true
}
```

当 dbjs 执行后没有返回结果，即自动输出该 {success:true} 的 JSON 返回。

在 dbjs.create 末尾增加 response.sendRedirect(“book.jssp”) 即可回到列表页面，即：

```
dbjs.create = function(rcd){
    logger.info('add book ' + JSON.stringify(rcd));
    rcd.publish_date = parseDate(rcd.publish_date);
    rcd.author *= 1;
    this.insertRow('book', rcd, ['title', 'author', 'publish_date', 'kind', 'isbn', 'remarks'])
    response.sendRedirect('book.jssp');
}
```

book

id	title	kind	publish_date	isbn	remarks	author
1	纯粹理性批判	p	Tue Jul 28 2015 00:00:00 GMT+0800 (CST)	1112233	null	4
7	红楼梦	f	Thu Feb 03 1977 00:00:00 GMT+0800 (CST)	2323121	null	1
2	判断力批判	p	Wed Sep 08 1993 00:00:00 GMT+0800 (CST)	2323231	null	4
3	飘	f	Thu May 06 1982 00:00:00 GMT+0800 (CST)	22313	null	3
4	神雕侠侣	k	Sun Feb 02 1992 00:00:00 GMT+0800 (CST)	242313	null	2
5	书剑恩仇录	k	Mon Apr 05 1982 00:00:00 GMT+0800 (CST)	3131	null	2
6	天龙八部	k	Wed Feb 02 1983 00:00:00 GMT+0800 (CST)	23231	null	2
9	笑傲江湖	f	null	232424523	null	2

+

可见，出版日期没有添加成功，查看日志，可以看到如下输出：

bookstore\book.dbjs:17 INFO - add book {"_m":"create","title":"笑傲江湖","publish_date":"1985-03-06","author":"2","isbn":"232424523","kind":"f","remarks":""}

这里接收到的日期格式为 yyyy-MM-dd 格式字符串，并非日期类型要求的 JSON 格式，所以本框架的 parseDate 函数无法处理。js 也没有相应处理函数，由于经典方式不是 db2js 推荐的方式，如何增加类库进行日期处理此处不赘述。

可以看到，在本套做法中，form 表单提交的参数已经映射为接收函数的 params 参数，但是由于参数值均为 string 类型，需要在 dbjs 文件中自己进行转换。

db2js

DataTable 基础类

新建一个页面，加入如下代码：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
```

```
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>

</body>
<script>

    var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /*  默认  10 */});

    books.load({ _m : 'fetch'});

</script>
</html>
```

在浏览器输入该网址后，打开开发者工具，在 Console 中输入 books.monitor()。即可看到内存中的数据如下：

book

id(undefined)	title(undefined)	kind(undefined)	publish_date(undefined)	isbn(undefined)	remarks(undefined)	author(undefined)	author_name(undefined)	row._state	row._error
1	纯粹理性批判	p	Tue Jul 28 2015 00:00:00 GMT+0800 (中国标准时间)	1112233	null	4	康德	none	
7	红楼梦	f	Thu Feb 03 1977 00:00:00 GMT+0800 (中国标准时间)	2323121	null	1	曹雪芹	none	
2	判断力批判	p	Wed Sep 08 1993 00:00:00 GMT+0800 (中国标准时间)	2323231	null	4	康德	none	
3	飘	f	Thu May 06 1982 00:00:00 GMT+0800 (中国标准时间)	22313	null	3	玛格丽特 米切尔	none	
4	神雕侠侣	k	Sun Feb 02 1992 00:00:00 GMT+0800 (中国标准时间)	242313	null	2	金庸	none	

这说明数据已经成功加载。

下面将表示在 DataTable 结构中的数据渲染到页面元素。

列表

修改 book.dbjs，使之支持排序、分页，并查出作者姓名：

```
dbjs.fetch = function(params){
    sql{.
        select b.*, a.name author_name from book b, author a where b.author = a.id
        code{.
            if(params.title){
                sql{. and strpos(b.title, :title) > 0    .}
            }
        .}
    .}
    sql = this.orderBy(sql, params._sorts, {'b.title' : 'asc'});
    return this.query(sql, params, params._page);
}
```

新建一个名为 book.html 的页面：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>
    <div class="container">
        <h1>书</h1>
        <div data="#book,error" renderer="stderr"></div>
        <table class="table table-striped" data="#book" renderer="table">
            <thead>
                <tr>
                    <th data-t="rows,N,title" renderer="std">
                        Title
                    </th>
                    <th data-t="rows,N,kind" renderer="dict({f:'小说', k:'武侠小说', p:'哲学
'})|std">
                        Kind
                    </th>
                    <th data-t="rows,N,author_name" renderer="std">
                        Author
                    </th>
                    <th data-t="rows,N,publish_date" format="yyyy-MM-dd"
renderer="date|std">
                        Publish Date
                    </th>
                </tr>
            </thead>
```

```
            <tfoot>
                <tr>
                    <td width="*" align="center">
                        <nav>
                            <ul class="pagination" data="#book" renderer="pagination">
                            </ul>
                        </nav>
                    </td>
                </tr>
            </tfoot>
        </table>

    </div>
</body>
<script>

    var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /*  默认  10 */});
    books.load({ _m : 'fetch'}, function(){db2js.render();});

</script>
</html>
```

这样，就完成了一个初级的列表页面。

下面为它增加搜索功能。

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
```

```
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>
    <div class="container">
        <h1>书</h1>
        <div data="#book,error" renderer="stderr"></div>
        <div class="text-right">
            <form class="form-inline">
              <div class="form-group">
                <label>Title</label>
                <input type="text" class="form-control" placeholder=""
data="#book,search,params,title" collector="c|s">
              </div>
              <button type="button" class="btn btn-default"
onclick="search($(this).parent('form'))">Search</button>
            </form>
        </div>
        <table class="table table-striped" data="#book" renderer="table">
            <thead>
                <tr>
                    <th data-t="rows,N,title" renderer="std">
                        Title
                    </th>
                    <th data-t="rows,N,kind" renderer="dict({f:'小说', k:'武侠小说', p:'哲学
'})|std">
                        Kind
                    </th>
                    <th data-t="rows,N,author_name" renderer="std">
                        Author
                    </th>
                    <th data-t="rows,N,publish_date" format="yyyy-MM-dd"
renderer="date|std">
                        Publish Date
                    </th>
```

```
                </tr>
            </thead>
            <tfoot>
                <tr>
                    <td width="*" align="center">
                        <nav>
                            <ul class="pagination" data="#book" renderer="pagination">
                            </ul>
                        </nav>
                    </td>
                </tr>
            </tfoot>
        </table>

    </div>
</body>
<script>

    var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /*  默认  10 */});
    books.on('load', function(){db2js.render();});

    books.load({ _m : 'fetch'});

    function search(form){
        db2js.collect(form[0]);
        books.load();
    }


</script>
</html>
```

下面是运行结果：

这里用户输入的参数被收集到数据路径为 #book,search,params 的对象，当再次 load 时，即以此为搜索条件进行检索。

排序

将列头修改为超链接，并增加超链接 render 函数 sortLink，得到页面如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>
    <div class="container">
        <h1>书</h1>
        <div data="#book,error" renderer="stderr"></div>
        <div class="text-right">
            <form class="form-inline">
              <div class="form-group">
                <label>Title</label>
                <input type="text" class="form-control" placeholder=""
data="#book,search,params,title" collector="c|s">
              </div>
              <button type="button" class="btn btn-default"
onclick="search($(this).parent('form'))">Search</button>
            </form>
        </div>
        <table class="table table-striped" data="#book" renderer="table">
            <thead>
                <tr>
```

```
                    <th data-t="rows,N,title" renderer="std">
                        <a data="#book" renderer="sortLink('title', 'Title')"/>
                    </th>
                    <th data-t="rows,N,kind" renderer="dict({f:'小说', k:'武侠小说', p:'哲学
'})|std">
                        <a data="#book" renderer="sortLink('kind', 'Kind')"/>
                    </th>
                    <th data-t="rows,N,author_name" renderer="std">
                        <a data="#book" renderer="sortLink('author_name', 'Author
Name')"/>
                    </th>
                    <th data-t="rows,N,publish_date" format="yyyy-MM-dd"
renderer="date|std">
                        <a data="#book" renderer="sortLink('publish_date', 'Publish
Date')"/>
                    </th>
                </tr>
            </thead>
            <tfoot>
                <tr>
                    <td width="*" align="center">
                        <nav>
                            <ul class="pagination" data="#book" renderer="pagination">
                            </ul>
                        </nav>
                    </td>
                </tr>
            </tfoot>
        </table>

    </div>
</body>
<script>

    db2js.Renderers.sortLink = function(colName, text){
        return function(element, table){
            var e = $(element);
            e.attr('href', '###');
            element.onclick =  function(){
                var sort = {};
                var old = table.search._sorts && table.search._sorts[colName];
                if(old == 'asc'){
                    sort[colName] = 'desc';
                } else {
```

```
                    sort[colName] = 'asc';
                }
                table.search._sorts = sort;
                table.load();
            };
            var icon =    null;
            var sorts = table.search._sorts || {};
            switch(sorts[colName]){
            case 'asc' : icon = 'glyphicon glyphicon-arrow-up'; break;
            case 'desc' : icon = 'glyphicon glyphicon-arrow-down'; break;
            }
            e.html(text + (icon ? '<span class="'    + icon + '">' : ''));
        }
    }

    var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /*  默认  10 */});
    books.on('load', function(){db2js.render();});

    books.load({ _m : 'fetch'});

    function search(form){
        db2js.collect(form[0]);
        books.load();
    }


</script>
</html>
```

刷新页面，即已支持排序动作。

增加和修改

修改 book.dbjs, 调整 create 函数，增加 modify 和 destroy 函数如下：

```
dbjs.create = function(rcd){
    logger.info('add book ' + JSON.stringify(rcd));
    this.insertRow('book', rcd, ['title', 'author', 'publish_date', 'kind', 'isbn', 'remarks'])
}

dbjs.modify = function(rcd){
    logger.info('edit book ' + JSON.stringify(rcd));
    this.updateRow('book', rcd, ['id', 'title', 'author', 'publish_date', 'kind', 'isbn', 'remarks'])
}

dbjs.destroy = function(rcd){
    logger.info('delete book ' + JSON.stringify(rcd));
    this.deleteRow('book', rcd)
}
```

现在新的 book.html 代码为：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>
    <div class="container">
```

```
<h1>书</h1>
<div data="#book,error" renderer="stderr"></div>
<div class="text-right">
    <form class="form-inline">
        <div class="form-group">
            <label>Title</label>
            <input type="text" class="form-control" placeholder=""
data="#book,search,params,title" collector="c|s">
        </div>
        <button type="button" class="btn btn-default"
onclick="search($(this).parent('form'))">Search</button>
        <button type="button" class="btn btn-default"
onclick="editRow(books.addRow())">Add</button>
    </form>
</div>
<table class="table table-striped" data="#book" renderer="table">
    <thead>
        <tr>
            <th data-t="rows,N,title" renderer="std">
                <a data="#book" renderer="sortLink('title', 'Title')"/>
            </th>
            <th data-t="rows,N,kind" renderer="dict({f:'小说', k:'武侠小说', p:'哲学
'})|std">
                <a data="#book" renderer="sortLink('kind', 'Kind')"/>
            </th>
            <th data-t="rows,N,author_name" renderer="std">
                <a data="#book" renderer="sortLink('author_name', 'Author
Name')"/>
            </th>
            <th data-t="rows,N,publish_date" format="yyyy-MM-dd"
renderer="date|std">
                <a data="#book" renderer="sortLink('publish_date', 'Publish
Date')"/>
            </th>
            <th data-t="rows,N,id" renderer="editLink"></th>
        </tr>
    </thead>
    <tfoot>
        <tr>
            <td width="*" align="center">
                <nav>
                    <ul class="pagination" data="#book" renderer="pagination">
                    </ul>
                </nav>
```

```
                    </td>
                </tr>
            </tfoot>
        </table>


        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title" id="myModalLabel">Edit <span data="first_name" renderer="std"/></h4>
                    </div>
                    <div class="modal-body">
                        <form>

                            <div class="form-group" data="#book,curr,_error,title" renderer="flderr">
                                <label>Title</label>
                                <input type="text" class="form-control" data="#book,curr,title" renderer="std" collector="c|s">
                            </div>

                            <div class="form-group" data="#book,curr,_error,kind" renderer="flderr">
                                <label>Kind</label>
                                <div data="#book" renderer="dictToList({f:'小说', k:'武侠小说', p:'哲学'})|options('name', 'id')">
                                    <select class="form-control" data="#book,curr,kind" renderer="std" collector="c|s"></select>
                                </div>
                            </div>

                            <div class="form-group" data="#book,curr,_error,author" renderer="flderr">
                                <label>Author</label>
                                <div data="#author,rows" renderer="options('name', 'id')">
                                    <select class="form-control" data="#book,curr,author" renderer="std" collector="c|n|s"></select>
                                </div>
                            </div>

                            <div class="form-group" data="#book,curr,_error,isbn" renderer="flderr">
```

```
                        <label>ISBN</label>
                        <input type="text" class="form-control" data="#book,curr,isbn" renderer="std" collector="c|s">
                    </div>

                    <div class="form-group" data="#book,curr,_error,publish_date" renderer="flderr">
                        <label>Publish Date</label>
                        <input type="date" class="form-control" data="#book,curr,publish_date" format="yyyy-MM-dd" renderer="date|std" collector="c|d|s">

                    </div>

                    <div class="form-group" data="#book,curr,_error,remarks" renderer="flderr">
                        <label>Remarks</label>
                        <textarea data="#book,curr,remarks" class="form-control" cols="20" rows="5" renderer="std" collector="c|s"></textarea>
                    </div>


                </form>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary" onclick="save()">Save changes</button>
            </div>
        </div>
    </div>
</div>
</body>
<script>
    new db2js.DataTable('author', 'author.dbjs').load({_m : 'listAll'}, function(){ db2js.render(); });          // 加载所有作者，作为选择列表

    function editRow(row){
        books.curr = row;
        db2js.render($('#myModal')[0], books);
        $('#myModal').modal('show');
    }
```

```
$('#myModal').on('hide.bs.modal', function() {
	books.reject();
	books.clearError();
	db2js.render();
});

function save() {
	db2js.collect();

	books.submit({callback : function(error){
		db2js.render();
		if(!error){
			$('#myModal').modal('hide');
			this.reload();
		}
	}});
}

var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /* 默认 10 */});
books.on('load', function(){db2js.render();});

books.load({ _m : 'fetch'});

function search(form){
	db2js.collect(form[0]);
	books.load();
}

db2js.Renderers.editLink = function(element, v, table){
	var e = $(element);
	e.html('');
	var a = $(document.createElement('a')).appendTo(e);
	a.html('EDIT');
	a.attr('href', '###');
	a.data('id', v);
	a.on('click', function(){
		editRow(table.find('id', $(this).data('id')));
	});
}

db2js.Renderers.sortLink = function(colName, text){
	return function(element, table){
		var e = $(element);
		e.attr('href', '###');
```

```
                element.onclick =    function(){
                    var sort = {};
                    var old = table.search._sorts && table.search._sorts[colName];
                    if(old == 'asc'){
                        sort[colName] = 'desc';
                    } else {
                        sort[colName] = 'asc';
                    }
                    table.search._sorts = sort;
                    table.load();
                };
                var icon =    null;
                var sorts = table.search._sorts || {};
                switch(sorts[colName]){
                case 'asc' : icon = 'glyphicon glyphicon-arrow-up'; break;
                case 'desc' : icon = 'glyphicon glyphicon-arrow-down'; break;
                }
                e.html(text + (icon ? '<span class="'    + icon + '">' : ''));
            }
        }
```

```
</script>
</html>
```

原理

DataTable 只有一个 submit 方法，其可以将表中所有有变动的数据行(包括新增、修改、删除)，打包在一起发送到服务器。服务器根据行状态，逐行调用相应函数，新增调用 create 函数、修改调用 modify 函数，删除调用 destroy 函数。

所以 DataTable 支持多行提交。

每次提交都处于同一个事务，一旦出错则整体回滚。

此外，DataTable 支持数据关系，指定数据关系后，主表提交时，可以连带提交子表。此时提交到的目标 dbjs 是主表 url，但在更新子表数据行时，将换用子表的 dbjs 程序处理。

Repeater

新建一个名为 book-list.html 的页面，输入以下内容：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>书</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="../jslib/bootstrap-3.3.4/css/bootstrap.css" rel="stylesheet" media="screen">
    <script src="../jslib/jquery-1.10.2.js"></script>
    <script src="../jslib/bootstrap-3.3.4/js/bootstrap.min.js"></script>

    <script src="../jslib/date.js/Date.js"></script>

    <script src="../jslib/db2js/dataset.js"></script>
    <script src="../jslib/db2js/render.js"></script>
    <script src="../jslib/db2js/renderers.js"></script>
    <script src="../jslib/db2js/pipelines.js"></script>
    <script src="../jslib/db2js/collector.js"></script>

</head>
<body>
    <div class="container">
        <div data="#book,rows" renderer="repeater">
            <div data="#book,error" renderer="stderr"></div>
            <h1>Books</h1>
            <hr>
            <div class="row">
              <div class="col-xs-6 col-md-3" repeater="true" data="this" renderer="expr">
                <h3>{{title}}<small>{{author_name}}</small></h3>
                <p>{{remarks || '(没有详细说明)'}}</p>
                <section    class="text-right">{{publish_date.format('yyyy-MM-dd')}}  出  版
</section>
                <button class="btn btn-primary">购买</button>
              </div>
            </div>

            <nav class="text-center">
                <ul class="pagination" data="#book" renderer="pagination">
                </ul>
            </nav>
        </div>
    </div>
</body>
<script>
```

```
    var books = new db2js.DataTable('book', 'book.dbjs', {pageSize : 5 /* 默认 10 */});
    books.load({_m : 'fetch'}, function(){db2js.render();})


</script>
</html>
```

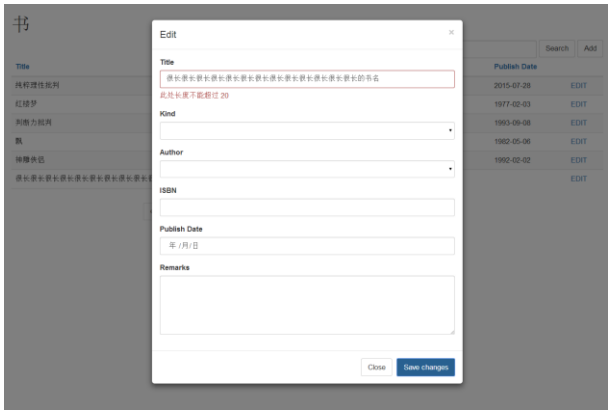
打开该网址，即得到：

校验

Db2js 一般采用服务器端校验，校验过程放在 create, modify, destroy 函数体第一行。
校验器定义于 WEB-INF/jslib/db2js/validation.js，使用方法如：

```
dbjs.create = function(rcd){

    $V(rcd, {
        title : [V.notNull, V.longest(20)],
        author : [V.notNull],
        publish_date : [V.between(Date.parse('1900-01-01'), new Date()),
        remarks : [V.longest(500)]
    });

    logger.info('add book ' + JSON.stringify(rcd));
    this.insertRow('book', rcd, ['title', 'author', 'publish_date', 'kind', 'isbn', 'remarks'])
}
```

当用户输入不正确时，相应 DataRow 会标记为错误，通过 db2js.render，可以将它渲染在页面上。



molecule 组件化

直接看 molecule_test/index.html 吧