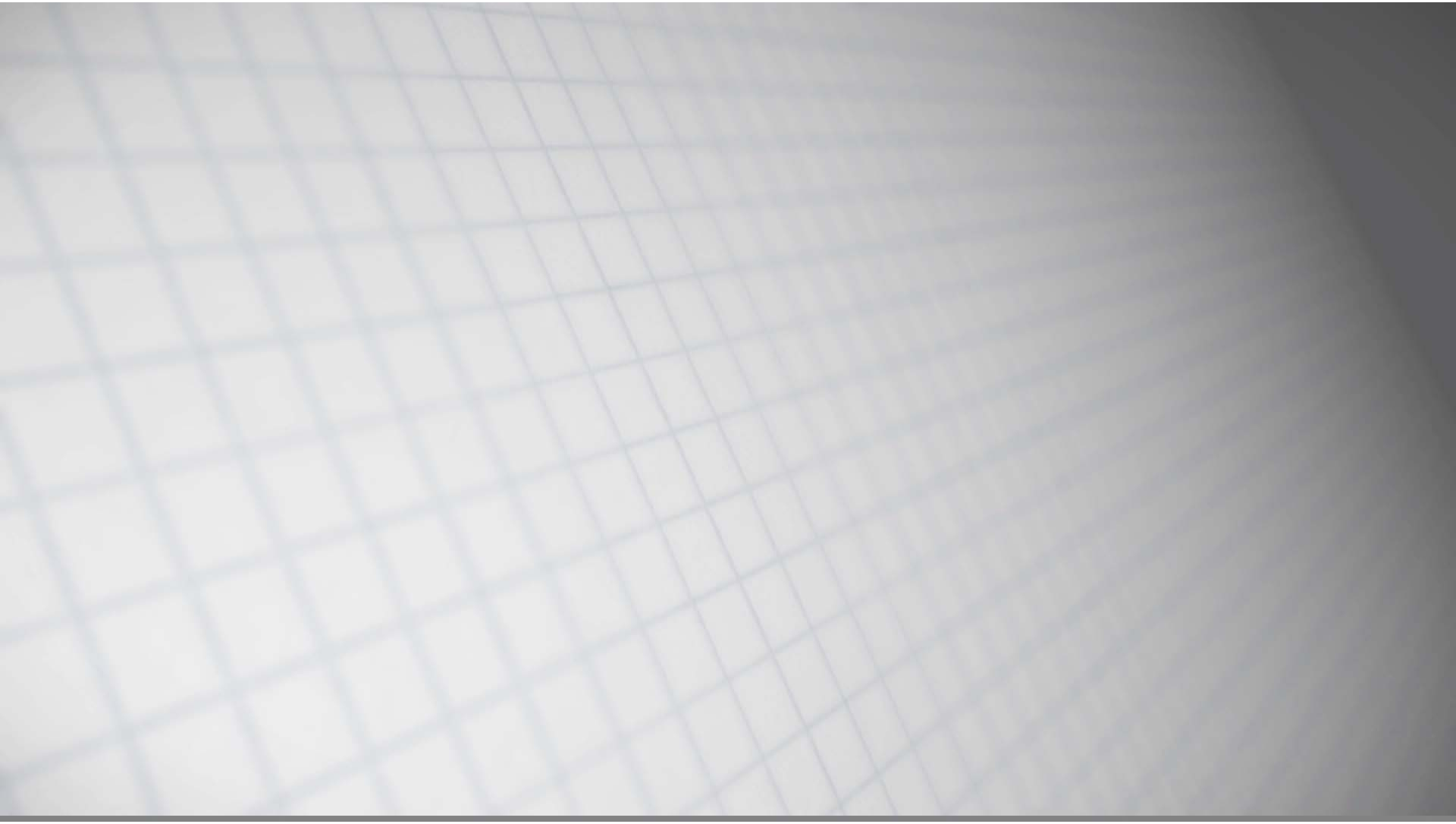
Using Telerik Kendo UI with WebSpeed

Kendo UI
THE ART OF WEB DEVELOPMENT

Telerik®
A PROGRESS COMPANY

PUG CHALLENGE EXCHANGE AMERICAS

Mike Fechner, Director, Consultingwerk Ltd.
mike.fechner@consultingwerk.de



Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany
- Customers in Europe, North America, Australia and South Africa
- Vendor of tools and consulting programs
- 25 years of Progress experience (V5 … OE11)
- Specialized in GUI for .NET, OO, Software Architecture, Application Integration

Progress WebSpeed

- When was the last time, Progress added functionality to WebSpeed?
- Version 8?
- Version 9?
- Certainly nothing new to WebSpeed in OpenEdge?
- WebSpeed is dead!

- WRONG ANSWER!

Progress WebSpeed

- Some OpenEdge 11 releases added little things

- Support for detecting JSON Input (WEB-CONTEXT:IS-JSON)
- Support for LONGCHAR Input

Progress WebSpeed

- OpenEdge 11.5.1 (released May 2015) allows to deploy WebSpeed messenger on PASOE (Tomcat)
- OpenEdge 11.6 will implement ability to execute WebSpeed in PASOE Agent
- https://community.progress.com/community_groups/openedge_general/f/26/t/17250.aspx

Agenda

- **Introduction into Kendo UI**
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

- JQuery and HTML5 based Widgets for browser based and mobile applications
- 70+ Widgets
- Integrates with AngularJS (MVVM Framework) and Bootstrap (responsive UI)
- Optimized for Performance and Resource Usage
- Themable and Localizable
- Server Side Wrappers for ASP.NET, JSP, PHP

Telerik Online Demos

- http://demos.telerik.com/kendo-ui/

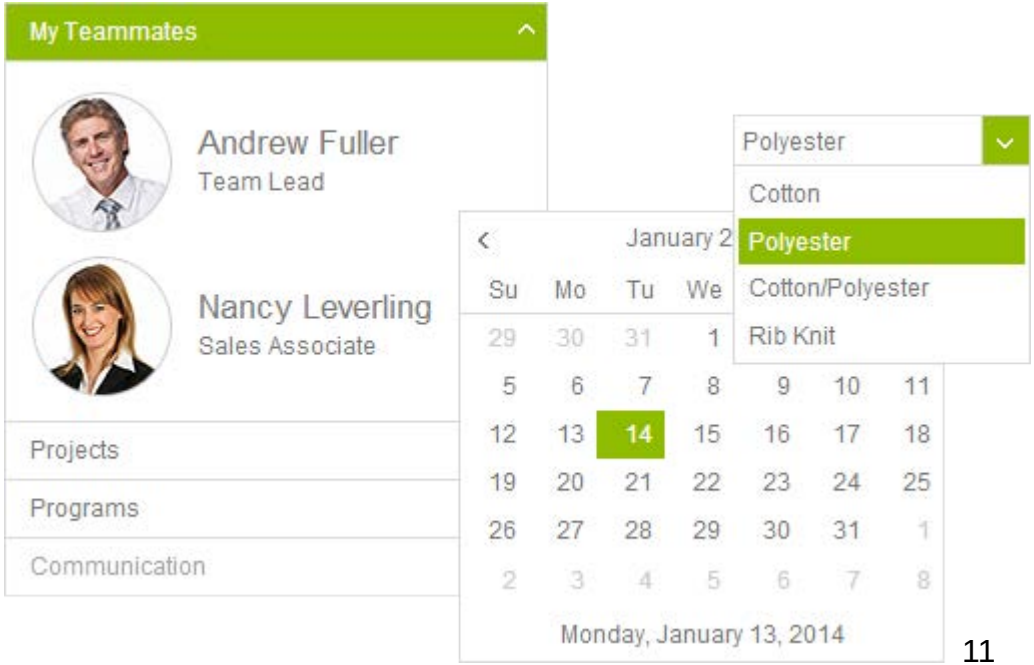
- Access to Documentation (API reference) from demos

JavaScript?

- Kendo UI can be used without deep JavaScript knowledge – as long as you are not starting to write single-page-applications
- Lots of documentation and great forums
- Widgets are initialized using standard code pattern (JQuery widget constructor)
- Widgets are customized using JSON representation of properties
- Data can be provided using properties (arrays)
- Data can be accessed by calling into web server

Kendo UI Core – open source

- Subset of Controls released as open source
- 40+ Controls
- http://www.telerik.com/kendo-ui/open-source-core
- Apache 2.0 license
- No support



Controls not included in open source

- BarCode
- **Charts**
- Editor
- Gauges
- **Grid**
- Map
- QRCode
- Scheduler
- StockChart
- Treeview
- Upload

Agenda

- Introduction into Kendo UI
- **Basic usage – Kendo UI in SpeedScript**
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Basic usage

- Kendo UI is integrated into HTML page by adding two JavaScript Libraries
  - JQuery
  - Kendo UI

```
<script src="/static/js/jquery.min.js"></script>
<script src="/static/js/kendo.all.min.js"></script>
```

- Path must be adjusted to match real location of file
- JavaScript code typically minimized to reduce loading time (and protect IP of Telerik)

Creating Kendo UI Widgets

- Standard HTML Elements placed on page
- JQuery is used to manipulate the page DOM
- JavaScript code block used to add Kendo UI Widget behavior to HTML Element (widget constructor)
- Widget behavior controlled by properties or event handlers (function callbacks) passed to widget constructor
- Widget typically rendered in place of HTML Element

Sample adding Date Picker to Textbox

- HTML page defines standard INPUT tags for text boxes

```
<p>
    <label for="datepicker">Show e-mails from:</label><input id="datepicker" value="10/10/2011" style="width:200px;" />
</p>

<p>
    <label for="monthpicker">Add to archive mail from:</label><input id="monthpicker" value="November 2011" style="width:200
</p>
```

- ($document).ready function adds Kendo UI Widget behaviour

```
<script>
    $(document).ready(function() {
        // create DatePicker from input HTML element
        $("#datepicker").kendoDatePicker();

        $("#monthpicker").kendoDatePicker({
            // defines the start view
            start: "year",

            // defines when the calendar should return date
            depth: "year",

            // display month and year in the input
            format: "MMMM yyyy"
        });
```

Providing values from SpeedScript

- Statement Escape

```
<!--WSS
    DEFINE VARIABLE cDateString AS CHARACTER NO-UNDO .
    SESSION:DATE-FORMAT = "mdy" .

    cDateString = SUBSTITUTE ("&1 &2",
                              Consultingwerk.MonthEnum:FromValue (MONTH (TODAY)):ToString(),
                              YEAR (TODAY)) .
-->
```

- Expression Escape

```
<p>
    <label for="datepicker">Date Picker:</label>
    <input id="datepicker" value="<!--WSE STRING (TODAY, '99/99/9999') -->"
                           style="width:200px;" />
</p>

<p>
    <label for="monthpicker">Month Picker:</label
    ><input id="monthpicker" value="<!--WSE cDateString -->"
                             style="width:200px" />
</p>
```

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- **Extending mapped web objects**
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Mapped web objects

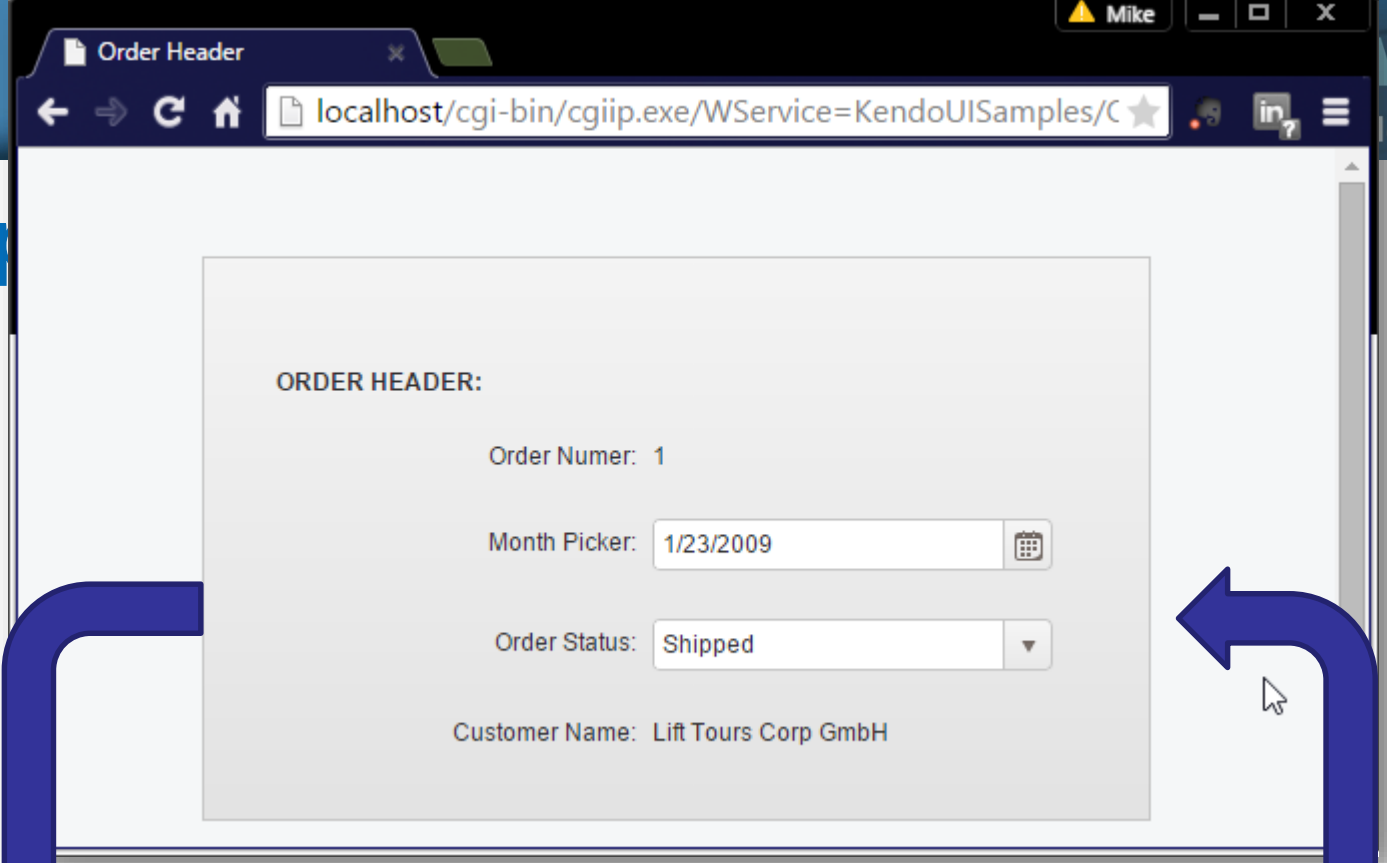
- Mapped web objects is **one of the development models** provided with WebSpeed
- Comprehensive separation of html layout and any ABL code
- Allows change of html layout without recompile
- Maps textbox, checkbox, radio-set, optionset, editor to ABL widgets in text default frame of the web object
- ABL program interacts with these widgets for input and output

Mapped web objects

- Static html page: OrderStatus.html
- Offset file: OrderStatus.off
- ABL web object: OrderStatus.w

- Offset file automatically generated whenever change in .html file is detected
- ABL web object maintained in AppBuilder, section editor
- ABL web object contains frame for fields in html file

Map



INPUT

OUTPUT

```
DEFINE FRAME Web-Frame
     Order.OrderStatus AT ROW 1 COL 1 HELP
          "" NO-LABEL
          VIEW-AS SELECTION-LIST SINGLE NO-DRAG
          SIZE 20 BY 4
     ab_unmap.kendoinit AT ROW 1 COL 1 HELP
          "" NO-LABEL FORMAT "X(256)":U
          VIEW-AS FILL-IN
          SIZE 20 BY 1
     Customer.Name AT ROW 5.52 COL 13 NO-LABEL
          VIEW-AS FILL-IN
          SIZE 20 BY 1
     Order.OrderDate AT ROW 7.91 COL 14 NO-LABEL
          VIEW-AS FILL-IN
          SIZE 20 BY 1
     Order.Ordernum AT ROW 2.43 COL 20 NO-LABEL
```

Extending mapped web objects

- tagmap.dat file in %DLC% or working directory may define additional html elements to map
- allows to define specific input/output handlers, e.g. for date fields or combo box
- Kendo UI requires meta tag for JavaScript widget constructor

tagmap.dat

```
input,,text,fill-in,web/support/webinput.p
input,,date,fill-in,web/support/web-date.p
input,,checkbox,toggle-box,web/support/webtog.p
input,,hidden,fill-in,web/support/webinput.p
input,,password,fill-in,web/support/webinput.p
input,,radio,radio-set,web/support/webradio.p
select,/select,,selection-list,web/support/weblist.p
textarea,/textarea,,editor,web/support/webedit.p

!--KendoConstructor,,,fill-in,web/support/kendoconstructor.p

#Custom Tag that can be used to support HTML Tables and 3rd Party controls
!--WSTAG,,,fill-in,web/support/tagrun.p

#Custom Tag that can be used to notify an application to output messages
#messages that have queued up using the queue-message function
!--WSMSG,,,fill-in,web/support/webmsg.p
```

Order-Header-Mapped.html

```
<div id="example">
    <div class="demo-section k-header">
        <h4>Order Header:</h4>
        <p>
            <label for="ordernum">Order Numer:</label>
            <input type="text" id="ordernum" name="ordernum" style="width:200px;" >
        </p>

        <p>
            <label for="orderdate">Month Picker:</label>
            <input type="date" id="orderdate" name="orderdate" style="width:200px" value="" >
        </p>

        <p>
            <label for="orderstatus">Order Status:</label>
            <select size=1 id="orderstatus" name="orderstatus" style="width:200px" ></select>
        </p>

        <p>
            <label for="customername">Customer Name:</label>
            <input type="text" id="customername" name="customername" style="width:200px" >
        </p>

    </div>
<script>
    $(document).ready(function() {
        <!--KendoConstructor name="kendoinit" -->
    });
</script>
```

kendoconstructor.p

```
25 PROCEDURE web.output:
26   /* Progress field containing the data */
27   DEFINE INPUT PARAMETER hWid     AS HANDLE    NO-UNDO.
28   /* the initial definition as contained in the HTML field definition */
29   DEFINE INPUT PARAMETER fieldDef AS CHARACTER NO-UNDO.
30
31   DEFINE VARIABLE hFrame     AS HANDLE    NO-UNDO .
32   DEFINE VARIABLE hWidget    AS HANDLE    NO-UNDO .
33   DEFINE VARIABLE cName      AS CHARACTER NO-UNDO .
34   DEFINE VARIABLE cFieldName AS CHARACTER NO-UNDO .
35
36   ASSIGN hFrame = hWid:FRAME
37          hWidget = hFrame:FIRST-CHILD:FIRST-CHILD .
38
39   DO WHILE VALID-HANDLE (hWidget):
40       ASSIGN cName = ? .
41
42       IF hWidget:TABLE > "" THEN
43           ASSIGN cFieldName = SUBSTITUTE ("&1.&2", hWidget:TABLE, hWidget:NAME) .
44       ELSE
45           ASSIGN cFieldName = hWidget:NAME .
46
47       ASSIGN cName = DYNAMIC-FUNCTION ("columnHTMLName" IN hWid:INSTANTIATING-PROCEDURE, cFieldName) .
48
49       CASE hWidget:DATA-TYPE:
50          WHEN "DATE" THEN
51              {&out} SUBSTITUTE ("$(~"#&1~").kendoDatePicker(); ", cName) SKIP .
52       END.
53
54       IF hWidget:TYPE = "SELECTION-LIST" THEN
55           {&out} SUBSTITUTE ("$(~"#&1~").kendoComboBox(); ", cName) SKIP .
56
57       hWidget = hWidget:NEXT-SIBLING .
58   END.
59
60 END PROCEDURE. /* web.output*/
```

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- **Auto-complete TextBox**
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Auto-complete TextBox

- TextBox with integrated search
- Search on local data (JavaScript Array) or remote data (http requests)
- Single selection or multi selection

Auto-complete TextBox

```
<script>
    $(document).ready(function () {
        var data = [
                "Albania",
                "Andorra",
                "Armenia",
                "Austria",
                "Azerbaijan",
                "Belarus",
                "Belgium",
                "Bosnia & Herzegovina",
                "Bulgaria",
                "Croatia",
                "Cyprus",
                "Czech Republic",
                "Denmark",
                "Estonia",
                "Finland",
                "France",
                "Georgia",
                "Germany",
                "Greece",
                "Hungary",
                "Iceland",
```

```
<div id="example">

        <div id="shipping">
    <label for="countries" class="info">Choose shipping countries:</label>

    <input id="countries" />

    <div class="hint">Start typing the name of an European country</div>

        </div>
```

```
//create AutoComplete UI component
$("#countries").kendoAutoComplete({
    dataSource: data,
    filter: "startswith",
    placeholder: "Select country...",
    separator: ", "
});
```

Use JSON Parser to build dataSource array

```
<script language="SpeedScript">
DEFINE VARIABLE oItems      AS Progress.Json.ObjectModel.JsonArray NO-UNDO .

oItems = NEW Progress.Json.ObjectModel.JsonArray () .

FOR EACH Item NO-LOCK:
    oItems:Add (Item.ItemName) .
END.
</script>
```

```
<script>
     $(document).ready(function () {
        var data = <!--WSS oItems:WriteStream ("WebStream":U) . -->;

        $("#items").kendoAutoComplete({
            dataSource: data,
            filter: "startswith",
            height: 400,
            seperator: ",",
            highlightFirst: true
        });
    });
</script>
```

Data Picker using remote Data Source

- Remote Data Source allows Kendo UI Widgets to do a separate http request to get data
- Supports remote filtering, sorting, paging, … (later in this talk)

```
$(document).ready(function () {
    $("#items").kendoAutoComplete({
        dataTextField: "ItemName",
        dataSource: {
                "type": "json",
                "transport": {
                    "read": "item-json.w",
                    "dataType": "json"
                        },
            "schema": {
                "data": "eItem"
            },
            "pageSize": 20,
            "serverPaging": false,
            "serverFiltering": false,
            "serverSorting": false
        },
        filter: "startswith",
        height: 400,
        seperator: ",",
        highlightFirst: true
    });
});
```

Simple Remote Data Soruce

```
DEFINE TEMP-TABLE eItem NO-UNDO LIKE Item
    INDEX ItemName IS PRIMARY ItemName.


/* ************************  Main Code Block

FOR EACH Item:
    CREATE eItem.
    BUFFER-COPY Item TO eItem .
END.

/* Process the latest Web event. */
RUN process-web-request.
```

```
PROCEDURE process-web-request :
/*------------------------------------------------------
  Purpose:     Process the web request.
  Parameters:  <none>
  Notes:
------------------------------------------------------*/

  RUN outputHeader.

  TEMP-TABLE eItem:WRITE-JSON ("stream", "webstream")

END PROCEDURE.
```

```
PROCEDURE outputHeader :
/*------------------------------------------
  Purpose:     Output the MIME header, and any "
               by this procedure.
  Parameters:  <none>
  Notes:       In the event that this Web object
               a good place to set the webState a
------------------------------------------*/

  output-content-type ("application/json":U).

END PROCEDURE.
```

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- **Using Kendo UI Grid with WebSpeed**
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Kendo UI Grid with WebSpeed

- Kendo UI Grid can use existing HTML table including existing <tr>, <th>, <td> to generate grid from it

- http://demos.telerik.com/kendo-ui/grid/from-table

```
<head>
    <title></title>
    <link href="/KendoUIExamples/content/shared/styles/examples-offline.css" rel=

    <link rel="stylesheet" href="/KendoUI/styles/kendo.common.min.css" />
    <link rel="stylesheet" href="/KendoUI/styles/kendo.default.min.css" />
    <link rel="stylesheet" href="/KendoUI/styles/kendo.dataviz.min.css" />
    <link rel="stylesheet" href="/KendoUI/styles/kendo.dataviz.default.min.css" /

    <script src="/KendoUI/js/jquery.min.js"></script>
    <script src="/KendoUI/js/kendo.all.min.js"></script>
```

```
<div class="demo-section k-header">
    <h4>HTML Table turned into Grid:</h4>

    <table id="grid">
        <tr>
            <th>Salesrep</th>
            <th>Name</th>
            <th>Region</th>
```

```
<script>
    $(document).ready(function() {
        $("#grid").kendoGrid({
            height: 400,
            sortable: true
        });
    });
</script>
</div>
```

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- **Asynchronous Data Access**
- Implementing custom request handlers
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Kendo UI async Data Access

- Allows to read data in separate http request
- Possibly from a different server, using different caching settings
- Supports for server side filtering, sorting, paging without need to reload page
- Starting point for single-page-application (SPA)

Kendo UI and JSDO

- JSDO:
  - Open-Source JavaScript Library from PSC
  - Mimics ProDatasets in JavaScript

- https://github.com/CloudDataObject/JSDO

- Initially designed for AppServer and REST Adapter
- Open for other data provider – including WebSpeed
- It's just using a http protocol

Demo

- Kendo UI Grid
- JSDO / REST Adapter / AppServer
- Server side filtering, sorting paging
- 200,000 customer records

- Chrome Developer Tools
- dataSource definition (Kendo UI, JSDO dialect)
- JSON Filter Pattern

```
var serviceURI = "http://localhost:8980/SmartJsdoBackendService",
    jsdoSettings = {
        serviceURI: serviceURI,
        catalogURIs: serviceURI + "/rest/SmartJsdoBackendService/Catalog/
    },
```

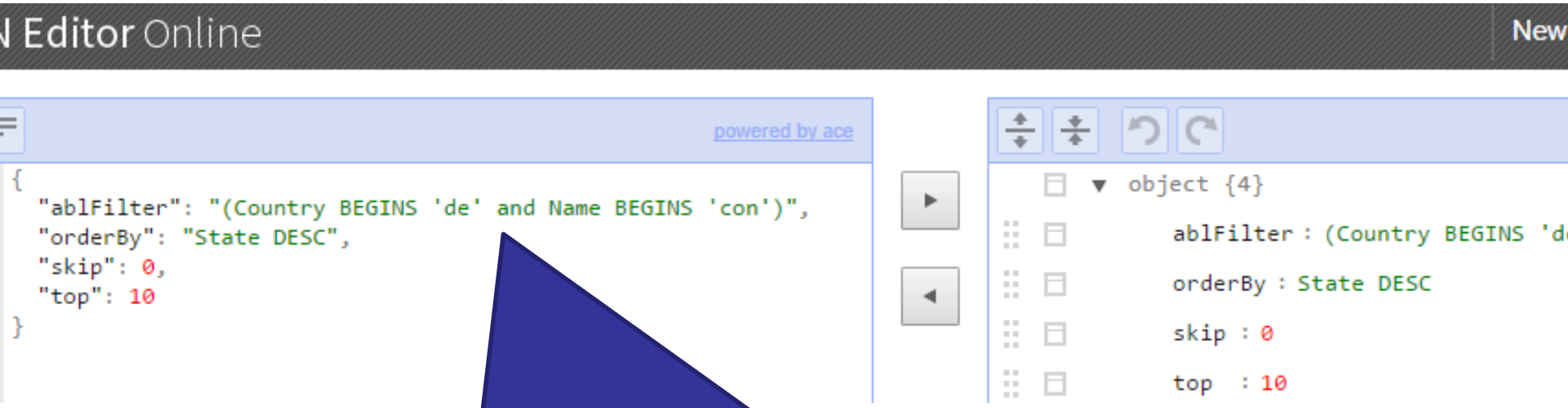
```
dataSource = new kendo.data.DataSource( {
                type: "jsdo",
                serverPaging: true,
                serverFiltering: true,
                filter: { field: "Country", operator: "eq", value: "USA" },
                serverSorting: true,
                sort: { field: "State", dir: "desc" },
                pageSize: 10,
                transport: {
                    jsdo: "Demo.Customer.CustomerBusinessEntity",
                    tableRef: "eCustomer",
                    countFnName: "count"
                },
                error: function(e) {
                    console.log ('Error: ', e);
                }
            }
);
```

```
$("#grid").kendoGrid(

    {
      "dataSource": dataSource
      "height": 600,
      "filterable": {
        "mode": "row, menu"
      },
      "groupable": false,
      "reorderable": true,
      "resizable": true,
      "sortable": true,
      "pageable": {
        "refresh": true,
        "pageSizes": [
          10,
          25,
          100
        ],
        "pageSize": 10,
```

Request URL for a single data page

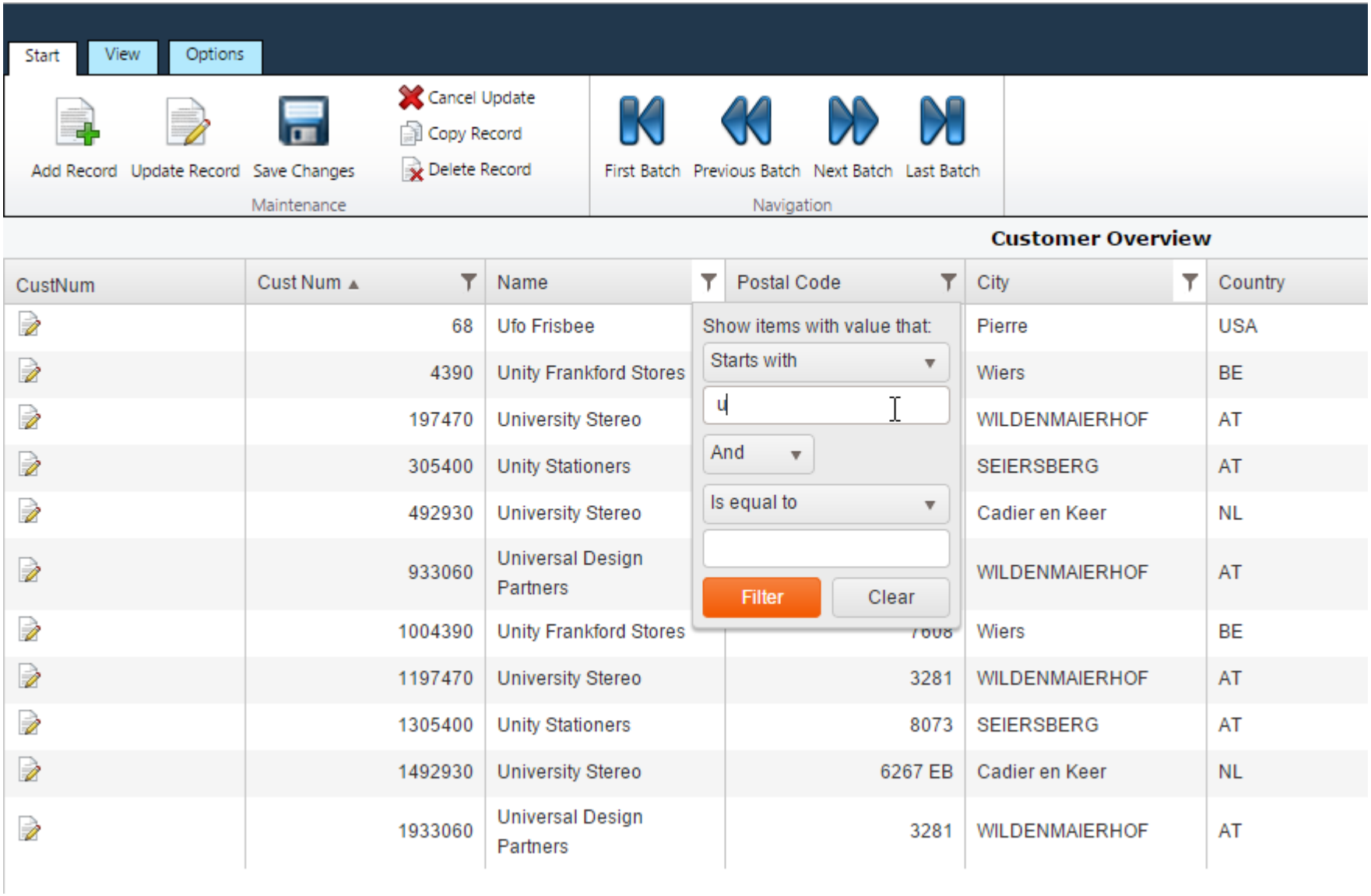
http://localhost:8980/SmartJsdoBackendService/rest/SmartJsdoBackendService/Resource/Consultingwerk.SmartComponentsDemo.OERA.Sports2000.CustomerBusinessEntity/count?**filter={"ablFilter":"(Country BEGINS 'de' and Name BEGINS 'con')","orderBy":"State DESC","skip":0,"top":10}**&_ts=143393199-227099112-23



Kendo UI produces ABL Query String ☺
That‘s the beauty of well integrated Progress products
(Telerik and OpenEdge)

Async Data Access for WebSpeed

- WebSpeed app could be combined with AppServer/REST Adapter
- Possible to implement JSDO protocol using WebSpeed
  - Requires JSON Catalog generation
    - Meta schema of ProDataset response
  - Requires GET
- Alternative: simple JSON Data Source, does also support server side filtering, sorting, paging


Start
View
Options
Add Record
Update Record
Save Changes
Cancel Update
Copy Record
Delete Record
Maintenance
First Batch
Previous Batch
Next Batch
Last Batch
Navigation
Customer Overview
CustNum
Cust Num
Name
Postal Code
City
Country
Show items with value that:
Starts with
u
And
Is equal to
Filter
Clear
68 Ufo Frisbee Pierre USA
4390 Unity Frankford Stores Wiers BE
197470 University Stereo WILDENMAIERHOF AT
305400 Unity Stationers SEIERSBERG AT
492930 University Stereo Cadier en Keer NL
933060 Universal Design Partners WILDENMAIERHOF AT
1004390 Unity Frankford Stores Wiers BE
1197470 University Stereo 3281 WILDENMAIERHOF AT
1305400 Unity Stationers 8073 SEIERSBERG AT
1492930 University Stereo 6267 EB Cadier en Keer NL
1933060 Universal Design Partners 3281 WILDENMAIERHOF AT

Simple JSON Data Source

```
{
  "dataSource": {
    "type": "json",
    "transport": {
      "read": "\/cgi-bin\/cgiip.exe\/WService=SmartWeb\/JsonData\/Consultingwerk\/SmartComponentsDemo\/Web\/Custom
      "dataType": "json"
    },
    "schema": {
      "data": "eCustomer"
    },
    "pageSize": 20,
    "serverPaging": true,
    "serverFiltering": true,
    "serverSorting": true
  },
  "height": 500,
  "filterable": {
    "mode": "menu"
  },
  "reorderable": true,
  "resizable": true,
  "sortable": true,
  "columns": [
```

Request URL to read data

- Sort[0]=CustNum ASC
- Filter[0]=Name startswith U
- Filter[1]=City contains IER

- http://localhost/cgi-bin/cgiip.exe/WService=SmartWeb/JsonData/Consultingwerk/SmartComponentsDemo/Web/CustomerDataSource?take=20&skip=0&page=1&pageSize=20&**sort[0][field]=CustNum&sort[0][dir]=asc**&**filter[filters][0][field]=Name&filter[filters][0][operator]=startswith&filter[filters][0][value]=u**&**filter[filters][1][field]=City&filter[filters][1][operator]=contains**&filter[filters][1][value]=ier&filter[logic]=and

Implementing Backend for Data Source

- Typically based on cgi-wrapper – as it does output JSON, not HTML
- May return TEMP-TABLE or ProDataset in JSON
- Or use JSON Object Model Parser to return custom JSON

- Details follow in next chapter

Updating with WebSpeed

- Kendo UI supports updating of data
- Uses REST verbs PUT, POST, DELETE
- Will post JSON Document to WebSpeed Server

WEB-CONTEXT JSON Support

- In theory, WebSpeed can be used as backend to JSDO (incl. server side paging, filtering, sorting)
- WEB-CONTEXT:IS-JSON attribute
- Returns TRUE when content-type application/json or text/json was posted to server
- WEB-CONTEXT:FORM-LONG-INPUT returns JSON String
- Use Progress.JsonObject.ObjectModelParser or ProDataset:READ-XML for input processing

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- **Implementing custom request handlers**
- Generating JSON definition from ABL
- Using Telerik Chart Controls

Implementing custom request handlers

- web/objects/web-disp.p handles every request to WebSpeed
- Agent startup procedure
- Available in source code
- Can be customized

- Parses URL and determines the web object to run to serve the request

Customizing web-disp.p

- Standard web-disp.p does only support procedural web objects
- We ***need*** ☺ class based web objects
- Class based cgi-wrapper
  - Generating html pages
  - Handling JSON Data requests
- Adds more structure to code than just a bunch of cgi-wrappers calling into class files

web-disp.p

```
109   Consultingwerk.Web.WebContext:Refresh () .
110
111   /* Mike Fechner, Consultingwerk Ltd. 04.04.2012
112      SmartWeb Widgets */
113   IF AppProgram BEGINS "SmartWeb":U THEN DO ON ERROR UNDO, THROW:
114
115       set-user-field ("SmartWeb.Page":U, SUBSTRING (AppProgram, 10)) .
116
117       IF VALID-OBJECT (oRedirectHandler) THEN
118           oRedirectHandler:ProcessWebInput() .
119
120       ASSIGN AppProgram = "Consultingwerk/Web/page-handler.w":U .
121
122       CATCH err AS Progress.Lang.Error:
123             {&out} err:GetMessage (1) .
124             {&out} err:CallStack .
125       END CATCH.
126   END.
```

```
/* Mike Fechner, Consultingwerk Ltd. 09.12.2014
   JSON Data Handler */
ELSE IF AppProgram BEGINS "JsonData":U THEN DO ON ERROR UNDO, THROW:

    set-user-field ("JsonData.Source":U, SUBSTRING (AppProgram, 10)).

    IF VALID-OBJECT (oRedirectHandler) THEN
        oRedirectHandler:ProcessWebInput() .

    ASSIGN AppProgram = "Consultingwerk/Web/data-handler.w":U .

    CATCH err AS Progress.Lang.Error:
          {&out} err:GetMessage (1) .
          {&out} err:CallStack .
    END CATCH.
END.
```

Request URL for data

- http://localhost/cgi-bin/cgiip.exe/WService=SmartWeb/JsonData/Consultingwerk/SmartComponentsDemo/Web/CustomerDataSource

- WebSpeed Messenger URL
- JsonData keyword to redirect request to JSON data-handler.w
- ABL Class Name (Data Source), turn / into period

data-handler.w

ABL_115/Consultingwerk/Web/data-handler.w

```
74 PROCEDURE process-web-request :
75 /*------------------------------------------------------------------------------
76   Purpose:     Process the web request.
77   Parameters:  <none>
78   Notes:
79 ------------------------------------------------------------------------------*/
80
81   DEFINE VARIABLE cDataObject AS CHARACTER      NO-UNDO .
82   DEFINE VARIABLE oDataSource AS JsonDataSource NO-UNDO .
83
84   RUN outputHeader.
85
86   ASSIGN cDataObject =
87     REPLACE (WebUtilities:GetUserField ("JsonData.Source":U),
88              "/":U, ".":U) .
89
90   oDataSource = DYNAMIC-NEW (cDataObject) () .
91
92   IF VALID-OBJECT (oDataSource) THEN DO:
93       oDataSource:FetchData() .
94       oDataSource:OutputTable() .
95   END.
96
97   CATCH err AS Progress.Lang.Error:
```

Query Manipulation

- Review CustomerDataSource.cls in PDSOE

```
ASSIGN cFields      = WebUtilities:GetField (?)
       cQueryString = SUBSTITUTE ("FOR EACH &1":U,
                                  THIS-OBJECT:EntityTable)
       oListQueryExpression = NEW ListQueryExpression () .

ASSIGN i = 0 .

REPEAT:
    ASSIGN cFilterField    = WebUtilities:GetField (SUBSTITUTE ("filter[
           cFilterOperator = WebUtilities:GetField (SUBSTITUTE ("filter[
           cFilterValue    = WebUtilities:GetField (SUBSTITUTE ("filter[
```

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- **Generating JSON definition from ABL**
- Using Telerik Chart Controls

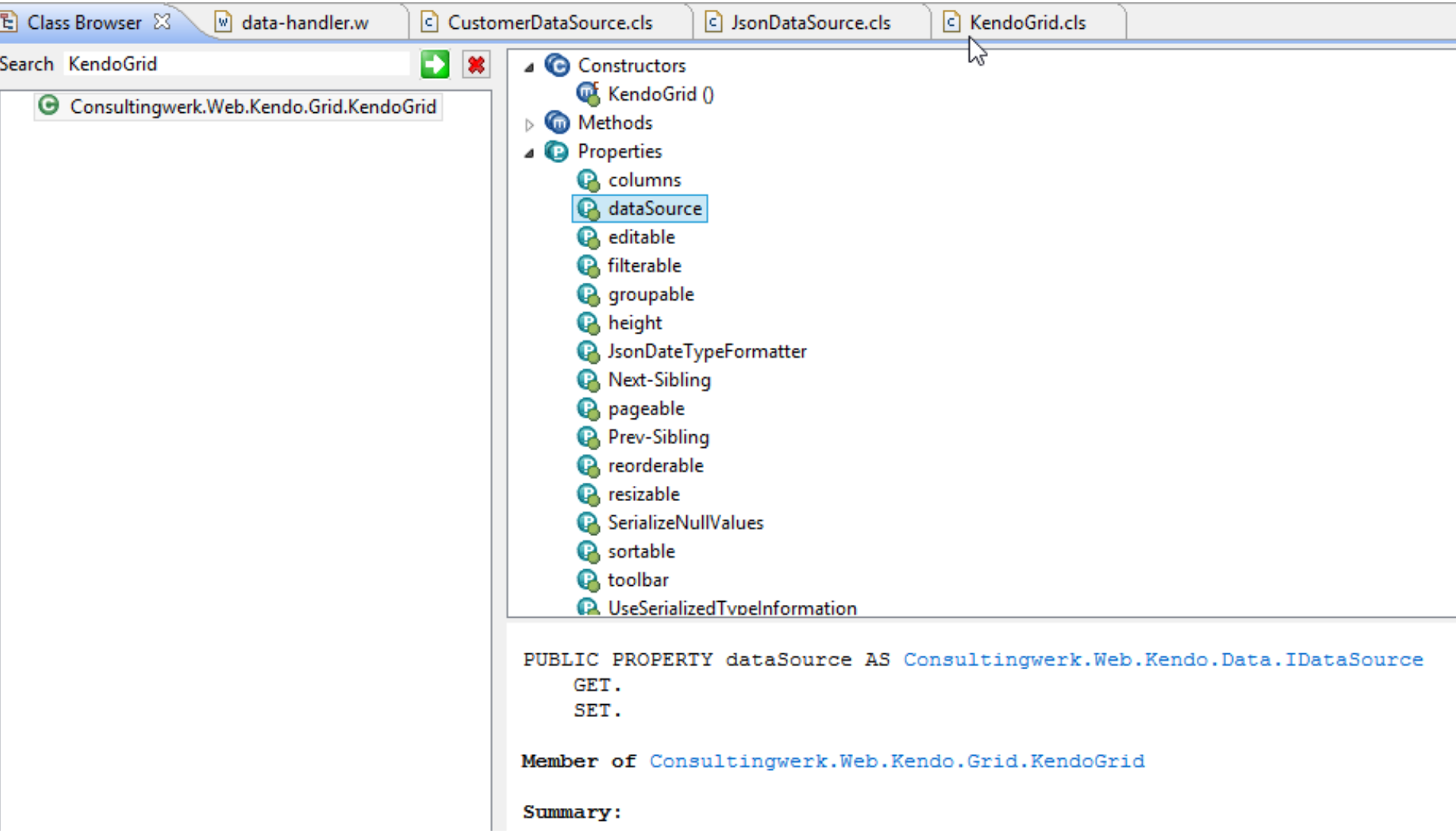
Generating JSON Definition from ABL

- Kendo Widget Properties: JSON Objects
- We can use ABL Object Model Parser to generate those object definition – instead of manually coding them in a designer

- If your ABL framework support JSON Serializable objects and lists … that's even a much simpler task

```
CLASS Consultingwerk.Web.Kendo.Grid.KendoGrid
    INHERITS JsonSerializable:

    {Consultingwerk/JsonSerializableProperty.i dataSource Consultingwerk.
    {Consultingwerk/JsonSerializableProperty.i height INTEGER "INIT ?"} .
    {Consultingwerk/JsonSerializableProperty.i filterable Consultingwerk.
    {Consultingwerk/JsonSerializableProperty.i groupable LOGICAL "INIT ?"
    {Consultingwerk/JsonSerializableProperty.i reorderable LOGICAL "INIT
    {Consultingwerk/JsonSerializableProperty.i resizable LOGICAL "INIT ?"
    {Consultingwerk/JsonSerializableProperty.i sortable LOGICAL "INIT ?"}
    {Consultingwerk/JsonSerializableProperty.i pageable Consultingwerk.Web
    {Consultingwerk/JsonSerializableProperty.i editable Consultingwerk.Web
    {Consultingwerk/JsonSerializableProperty.i toolbar "CHARACTER EXTENT"
    {Consultingwerk/JsonSerializableProperty.i columns Consultingwerk.Web
```


Class Browser
data-handler.w
CustomerDataSource.cls
JsonDataSource.cls
KendoGrid.cls
Search KendoGrid
Consultingwerk.Web.Kendo.Grid.KendoGrid
Constructors
KendoGrid ()
Methods
Properties
columns
dataSource
editable
filterable
groupable
height
JsonDateTypeFormatter
Next-Sibling
pageable
Prev-Sibling
reorderable
resizable
SerializeNullValues
sortable
toolbar
UseSerializedTypeInformation
PUBLIC PROPERTY dataSource AS Consultingwerk.Web.Kendo.Data.IDataSource
GET.
SET.
Member of Consultingwerk.Web.Kendo.Grid.KendoGrid
Summary:

```
CLASS Consultingwerk.Web.Kendo.Grid.Column
    INHERITS JsonSerializable:

    {Consultingwerk/JsonSerializableProperty.i field CHARACTER} .
    {Consultingwerk/JsonSerializableProperty.i command "CHARACTER EXTE
    {Consultingwerk/JsonSerializableProperty.i title CHARACTER} .
    {Consultingwerk/JsonSerializableProperty.i type CHARACTER} .
    {Consultingwerk/JsonSerializableProperty.i filterable LOGICAL "INI
    {Consultingwerk/JsonSerializableProperty.i sortable LOGICAL "INIT
    {Consultingwerk/JsonSerializableProperty.i width CHARACTER} .
    {Consultingwerk/JsonSerializableProperty.i attributes Consultingwe
    {Consultingwerk/JsonSerializableProperty.i template CHARACTER "INI
```

Code Review

- CustomerOverview.cls
- WebSpeed – OO enabled
- Web widgets that hide Kendo UI from developer

Agenda

- Introduction into Kendo UI
- Basic usage – Kendo UI in SpeedScript
- Extending mapped web objects
- Auto-complete TextBox
- Using Kendo UI Grid with WebSpeed
- Asynchronous Data Access
- Implementing custom request handlers
- Generating JSON definition from ABL
- **Using Telerik Chart Controls**

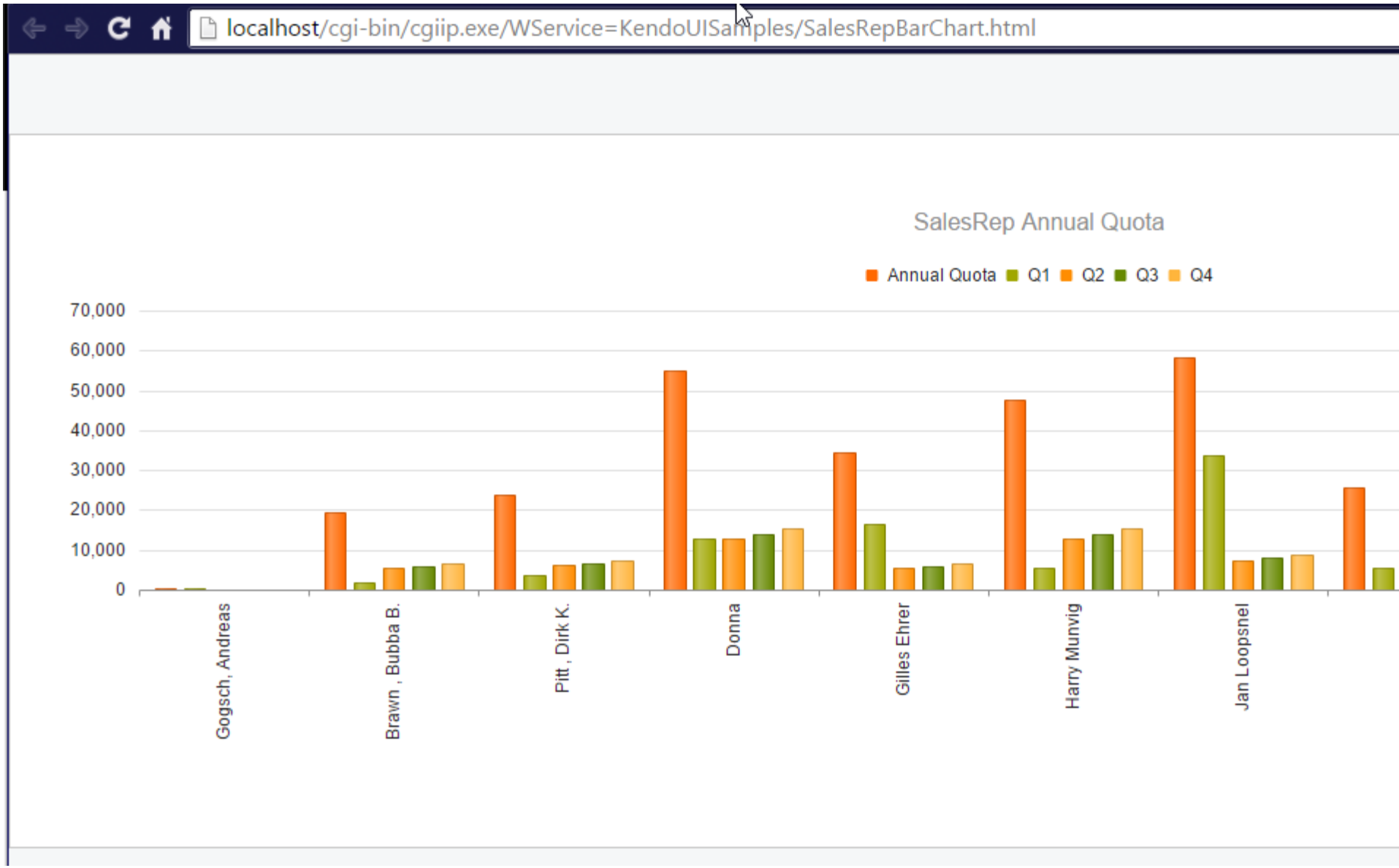
Kendo UI Chart Dojo

- Go to: http://dojo.telerik.com/tutorial/dataviz-chart-basics

Kendo UI Charts

- Data may be contained in the html page (like in the Dojo sample)
- Data may be loaded form remote data source

```
function createChart() {
    $("#chart").kendoChart({
        dataSource: { "type": "json",
                      "transport": { "read": "/cgi-bin/cgiip.exe/WServi
                                     "dataType": "json" },
                       "schema": { "data": "eSalesrep" }
                    },
        sort: {
            field: "Repname",
            dir: "asc"
        },
```


localhost/cgi-bin/cgiip.exe/WService=KendoUISamples/SalesRepBarChart.html
SalesRep Annual Quota
Annual Quota
Q1
Q2
Q3
Q4
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
Gogsch, Andreas
Brawn , Bubba B.
Pitt , Dirk K.
Donna
Gilles Ehrer
Harry Munvig
Jan Loopsnel

Questions

Don‘t miss my other presentations

- Monday 11.00: **Telerik .NET for Infragistics Users**
- Monday 16.45: DIY: **Lists, Enumerators, Enumerations, Serialization**
- Tuesday 11.00: **Modernization – the SmartComponent Library**
- Tuesday 14.15: **Structured Error Handling**
- Wednesday 11.00: **Telerik Kendo UI with WebSpeed**


PUG
CHALLENGE
EXCHANGE
AMERICAS