



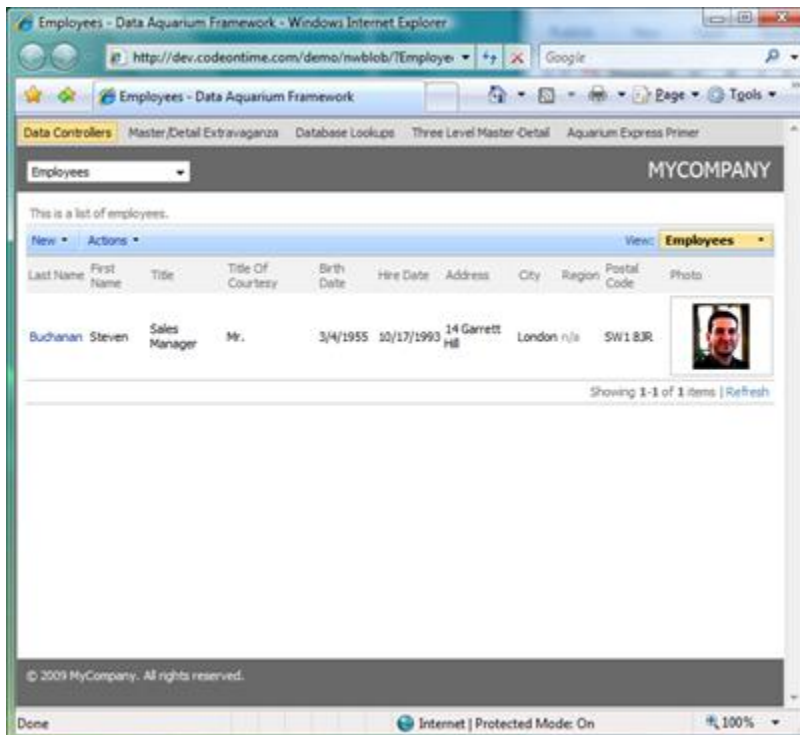
Present What You Want

Data Aquarium Framework offers users of your applications impressive data filtering and reporting capabilities. Nevertheless there will always be a situation when you need to present a specific database record or a group of records on demand.

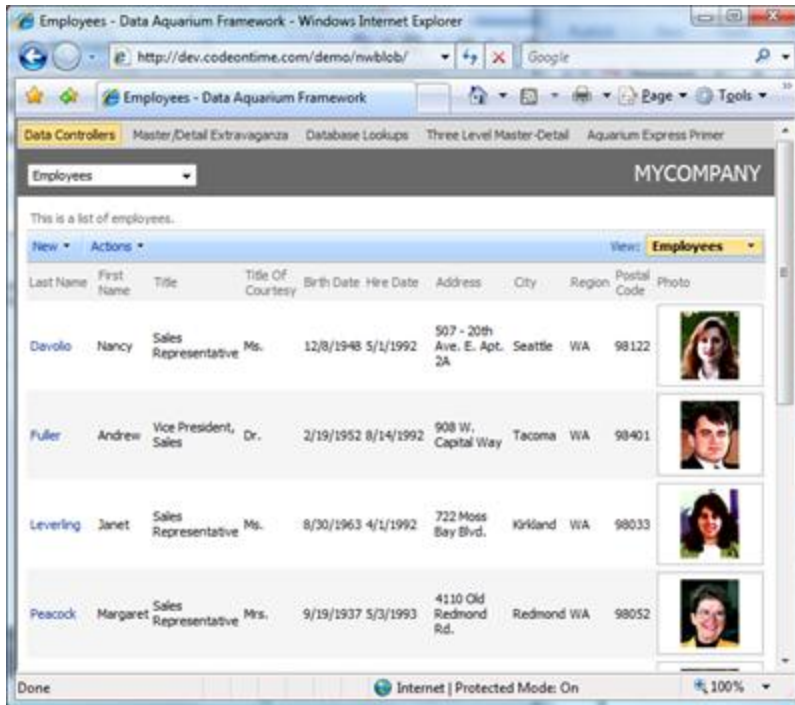
FILTERING VIA URL PARAMETER

The simplest method is to redirect your user to a specific page while supplying a record ID in the URL. For example, navigate to

<http://dev.codeontime.com/demo/nwblob/?EmployeeID=5> and you will see an employee with *EmployeeID* equal to 5 presented in the list.

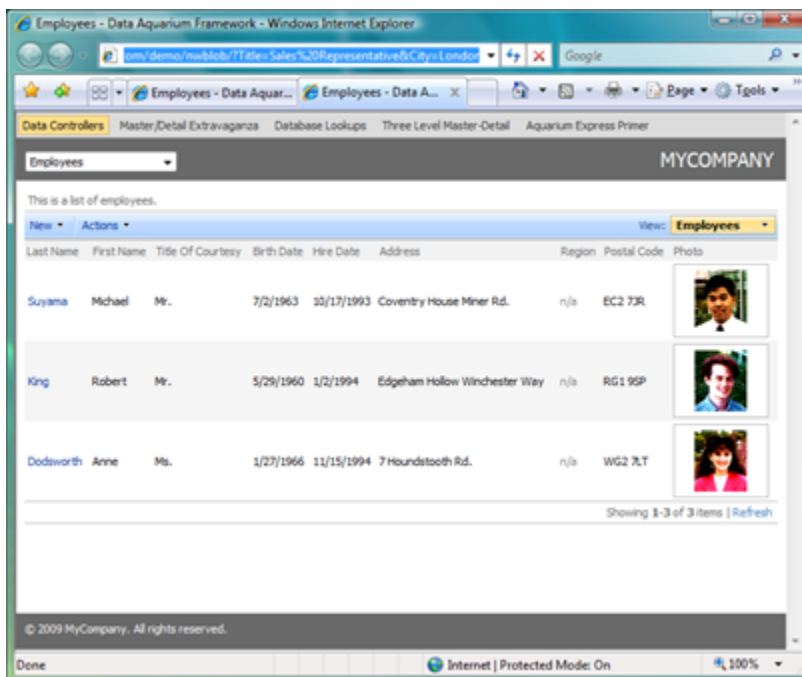


Remove *EmployeeID* from the URL and all employees are displayed.



You are not limited to the record IDs only. For example, the following URL will select all employees with job title “Sales Representative” working in London office:

<http://dev.codeontime.com/demo/nwblob?Title=Sales%20Representative&City=London>



If the filtered field is visible by default then [Data Aquarium Framework](#) will automatically hide the corresponding field column in the grid view since the field value is presumed to be known to the user. There is little value repeating “Sales Representative” and “London” in each row in the picture above. You may want to verify that each employee is indeed working in London and has the title that we have specified.

FILTERING VIA AN ELEMENT ON THE PAGE

You can also point your *GridViewExtender* components to read the filter value from any element on the page.

Add ~/Products.aspx page to the [Data Aquarium](#) project generated from *Northwind* database. Modify this page as shown below.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
    AutoEventWireup="true" CodeFile="Products.aspx.cs"
    Inherits="Products" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">
    <style type="text/css">
        div.Caption
        {
            font-weight: bold;
            padding: 4px 4px 4px 4px;
            background-color: #FDEAB1;
            color: #60302A;
        }
    </style>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="Header1Placeholder"
    runat="Server">
    Products
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="Header2Placeholder"
```

```

        runat="Server">
    Northwind
</asp:Content>
<asp:Content ID="Content4" ContentPlaceHolderID="BodyPlaceholder"
    runat="Server">
    <asp:HiddenField ID="ProductID" runat="server" Value="5" />
    <!-- "edit" mode -->
    <div class="Caption">
        "Edit" Mode</div>
    <div id="EditProduct" runat="server">
</div>
    <aquarium:DataViewExtender ID="DataViewExtender1" runat="server"
        TargetControlID="EditProduct"
        Controller="Products"
        FilterSource="ProductID" FilterFields="ProductID"
        ShowActionBar="false"
        StartCommandName="Edit" StartCommandArgument="editForm1" />
    <!-- "new" mode -->
    <div class="Caption">
        "New" Mode</div>
    <div id="NewProduct" runat="server">
</div>
    <aquarium:DataViewExtender ID="DataViewExtender2" runat="server"
        TargetControlID="NewProduct"
        Controller="Products" ShowActionBar="false"
        StartCommandName="New" StartCommandArgument="createForm1"
        PageSize="1" />
    <!-- "read" mode -->
    <div class="Caption">
        "Read" Mode</div>
    <div id="ViewProduct" runat="server">
</div>
    <aquarium:DataViewExtender ID="ProductListExtender" runat="server"

```

```
        TargetControlID="ViewProduct"
        Controller="Products"
        FilterSource="ProductID" FilterFields="ProductID"
        ShowActionBar="false"
        StartCommandName="Select" StartCommandArgument="editForm1" />
</asp:Content>
```

You can see this page live at <http://dev.codeontime.com/demo/filtering>.

At the top of the page there is *HiddenField* server control *ProductID* with its value set to 5.

```
<asp:HiddenField ID="ProductID" runat="server" Value="5" />
```

Three different views are presented on the page. Let's take a look at each of them.

The first view presents data in *edit* mode.

```
<!-- "edit" mode -->
<div class="Caption">
    "Edit" Mode</div>
<div id="EditProduct" runat="server">
</div>
<aquarium:DataViewExtender ID="DataViewExtender1" runat="server"
    TargetControlID="EditProduct"
    Controller="Products"
    FilterSource="ProductID" FilterFields="ProductID"
    ShowActionBar="false"
    StartCommandName="Edit" StartCommandArgument="editForm1" />
```

It uses hidden field as a filter source and filters by the field *ProductID*. The startup command displays view *editForm1* in *edit* mode.

"Edit" Mode

Please review products information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

* - indicates a required field

OK Delete Cancel

Products
These are the fields of the products record that can be edited.

Product Name *
Chef Anton's Gumbo Mix

Supplier Company Name
New Orleans Cajun Delights

Category Name
Condiments

Quantity Per Unit
36 boxes

Unit Price
\$21.35

Units In Stock
0

Units On Order
0

Reorder Level
0

Discontinued *
Yes

* - indicates a required field

OK Delete Cancel

The second view displays an empty record and does not specify any filter.

```

<!-- "new" mode -->
<div class="Caption">
    "New" Mode</div>
<div id="NewProduct" runat="server">
</div>
<aquarium:DataViewExtender ID="DataViewExtender2" runat="server"
    TargetControlID="NewProduct"
    Controller="Products" ShowActionBar="false"
    StartCommandName="New" StartCommandArgument="createForm1"
    PageSize="1" />

```

The startup command is executed only after the data controller data is transferred to the client browser. This will cause the controller to read the first page of records from the database. We are reducing the number of returned records to one by specifying the corresponding page size. You can eliminate retrieval of any records if you link a business rules filter to the *Products* data controller.

Here is the screen shot of the view.

The screenshot shows a web form titled "New Mode" with a yellow header. Below the header, there is a instruction: "Please fill this form and click OK button to create a new products record. Click Cancel to return to the previous screen." A note indicates that an asterisk (*) denotes a required field. The form contains several input fields: "Product Name" (required), "Supplier Company Name" (select), "Category Name" (select), "Quantity Per Unit", "Unit Price", "Units In Stock", "Units On Order", "Reorder Level", and "Discontinued" (required, dropdown menu with "No" selected). "OK" and "Cancel" buttons are located at the top right and bottom right of the form area.

The third view uses the hidden field as a filter source to display a view of a single record in "read" mode in *editForm1*.

```
<!-- "read" mode -->
<div class="Caption">
    "Read" Mode</div>
<div id="ViewProduct" runat="server">
</div>
<aquarium:DataViewExtender ID="ProductListExtender" runat="server"
    TargetControlID="ViewProduct"
    Controller="Products"
    FilterSource="ProductID" FilterFields="ProductID"
    ShowActionBar="false"
    StartCommandName="Select" StartCommandArgument="editForm1" />
```

Here is how the view looks when rendered in a web browser.

"Read" Mode

Please review products information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Products	Product Name
These are the fields of the products record that can be edited.	Chef Anton's Gumbo Mix
	Supplier Company Name New Orleans Cajun Delights
	Category Name Condiments
	Quantity Per Unit 36 boxes
	Unit Price \$21.35
	Units In Stock 0
	Units On Order 0
	Reorder Level 0
	Discontinued Yes

Generated data controllers are configured by default to switch to *grid1* view when user successfully executes *Insert*, *Update*, *Delete*, or *Cancel* command. You may want to play with configuration of actions in `~/Controls/Products.xml` to make do what you see fit for your purposes. Read about it [here](#).

PREVENTING FILTERING

You may also want to prohibit any filtering at all. This is easy to accomplish if you assign *None* to the *FilterSource* property of the *DataViewExtender* component.

```
<aquarium:DataViewExtender ID="ProductExtender" runat="server"
    TargetControlID="ProductList" Controller="Products"
    FilterSource="None" />
```

CONCLUSION

It is very easy to filter and display records that you want to be presented. URL parameters and elements on the page can be automatically consumed as sources of filter values.

Next we will take a look at server-side filtering via code in business rules.

Code OnTime LLC <http://www.codeontime.com>