

REST / JQUERY

The [Product Browser](#) example explains how to create a user control that takes advantage of the application server built in a web app created with [Code On Time](#). The scope of this example is to demonstrate reading the app data with [REST](#) requests with a custom *JavaScript*. The example also uses *jQuery* library integrated in the generated web apps.

Most custom user interface extensions will need to update, insert, and delete data. *Create, Read, Update, and Delete* operations are also known as [CRUD](#). A complete example of a standalone *Product Catalog Manager* is implementing a [Cross-Domain CRUD](#).

If a [custom user control](#) is created in a web app, then there is no need for cross-domain *AJAX* requests. Standard *JSON* requests can be executed with explicitly defined HTTP methods POST, GET, PUT, and DELETE to indicate the purpose of an *AJAX* web request.

ENABLING ACCESS TO PRODUCTS, SUPPLIERS, AND CATEGORIES

Three data controllers are required to implement the *Product Catalog Manager*. The data controllers *Products*, *Categories*, and *Suppliers* must [explicitly allow REST requests](#). The application server will deny any web requests if data controllers are not configured accordingly.

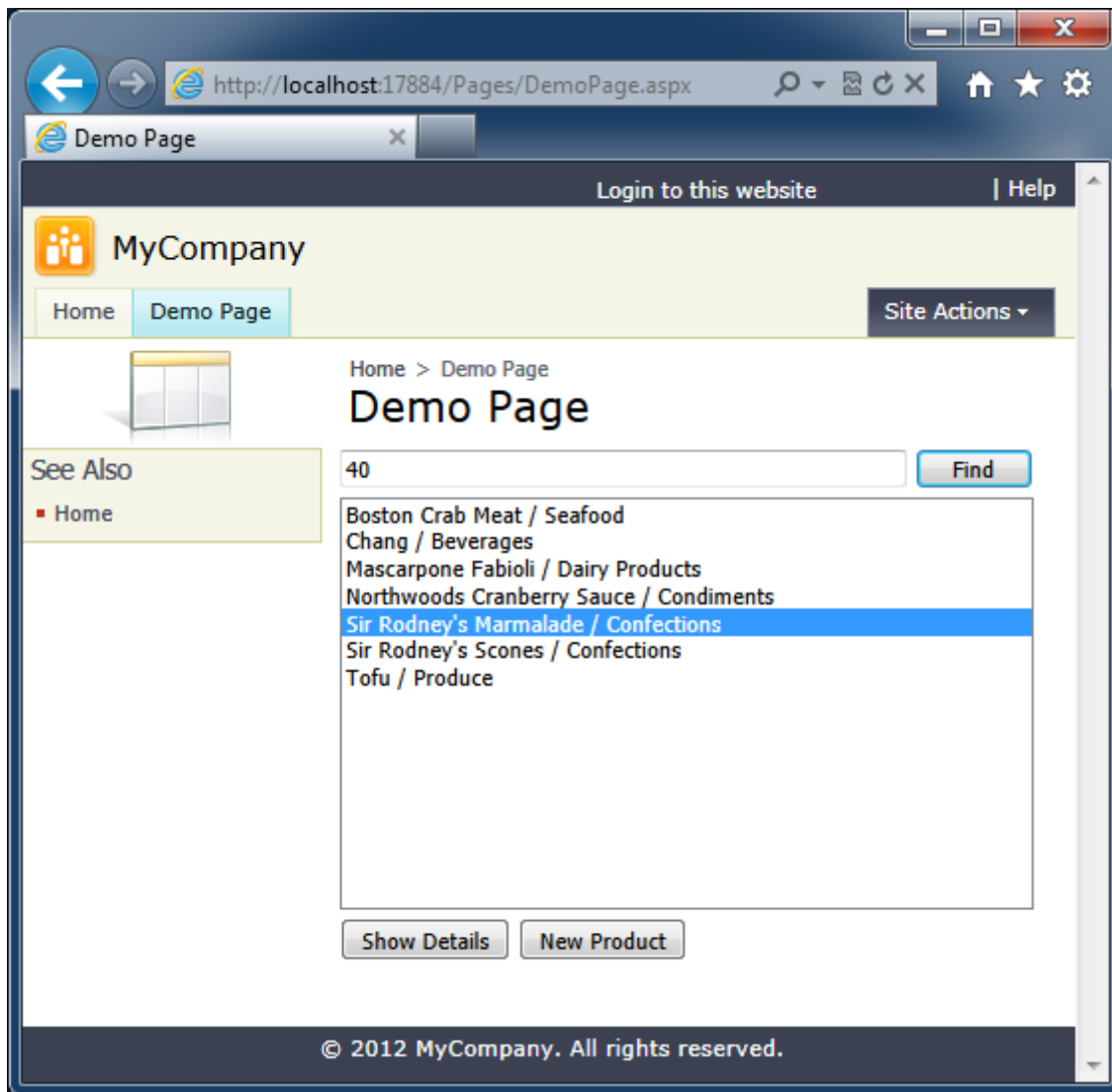
Select *Products*, *Suppliers*, and *Categories* data controllers in [Project Designer](#) and enter the following in its *Representational State Transfer (REST) Configuration* property for each controller:

Property	Value
Representation State Transfer (REST) Configuration	Uri: . Users: *

CREATING A USER CONTROL

Create a custom user control following instructions from the [Product Browser](#) example and replace the markup of a the user control with the example from the *Source Code* section below.

Browse to the page of a running web app that contains the user control. You will see the following:



Note that if you are accessing the page without signing in, then you will see a browser prompt for user name and password. The static link to the list of suppliers will force the application server to issue an identity challenge.

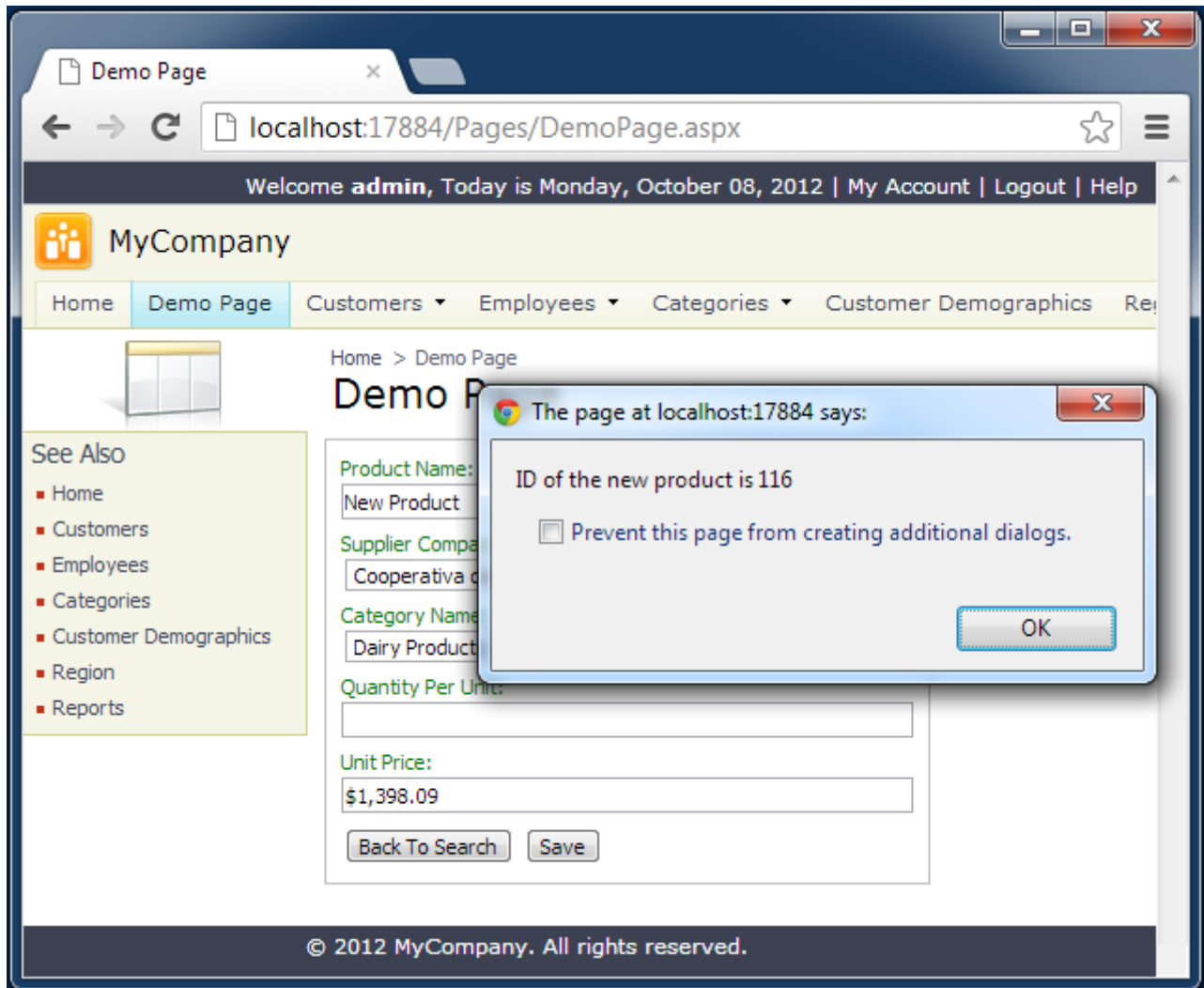
```
<script type="text/javascript"
  src="../appservices/Suppliers?_sortExpression=CompanyName&_instance=SupplierData">
</script>
```

If you do not wish to have a static link to suppliers then dynamically populate *SupplierID* list following the pattern shown in the implementation of *loadCategories* method.

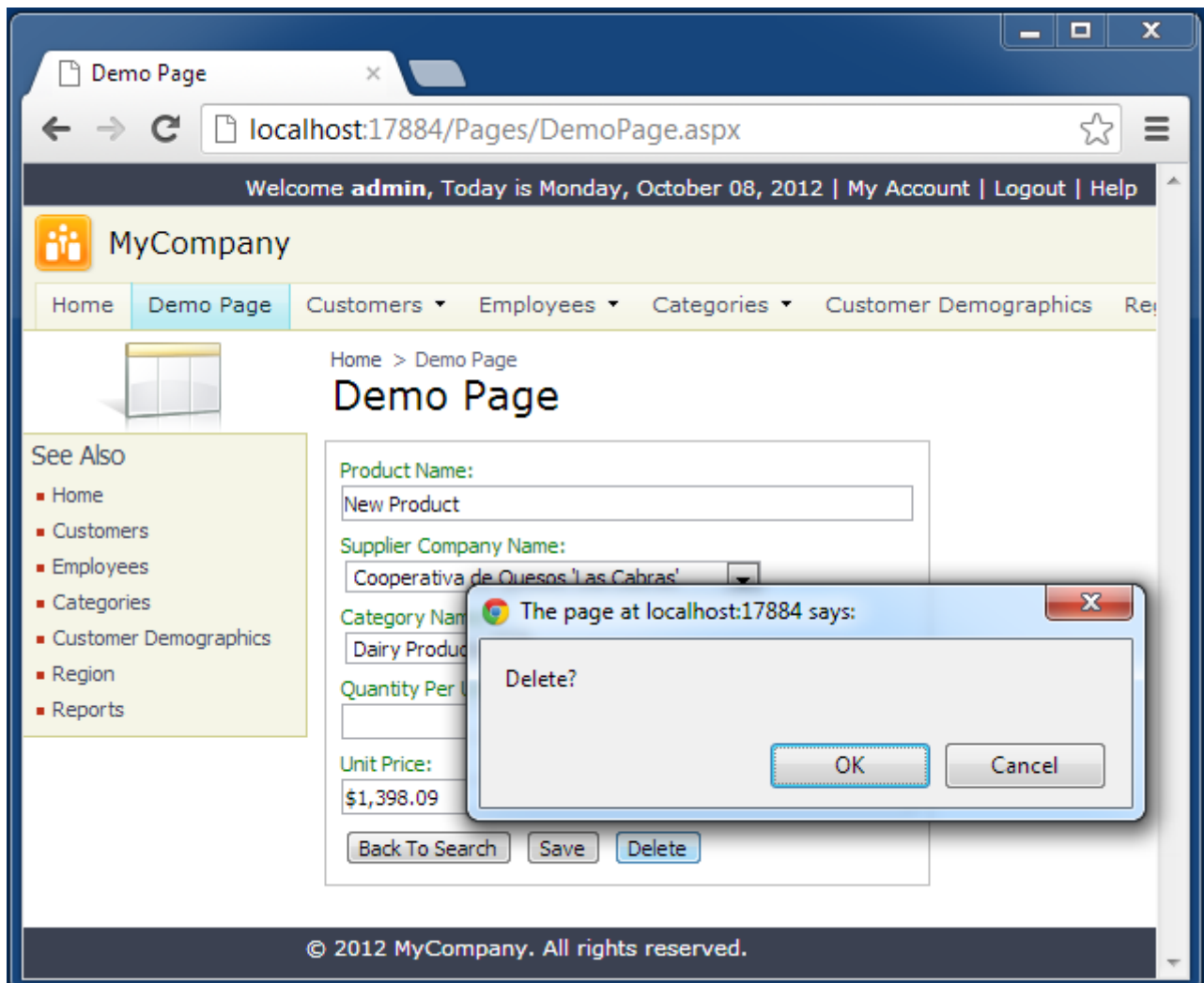
```
loadCategories: function () {
  $.ajax({
    url: this.basePath() + '/Categories?_sortExpression=CategoryName',
    dataType: 'json',
    cache: false,
    success: function (data) {
      $.each(data.Categories, function (index, category) {
        $('<option>').text(category.CategoryName)
          .attr('value', category.CategoryID).appendTo($('#CategoryID'));
      })
    }
  });
}
```

Try updating, inserting, and deleting data.

A new product is created by issuing an *PUT* request to the application server of a web app.



An existing product is deleted by issuing a *DELETE* request to the application server of a web app.



SOURCE CODE

The *ajax* requests are using a relative path to access the application server. Method *ProductManager.basePath()* return the path.

Notice the use of "json" in the "dataType" parameter of *ajax* requests. Also the nature of the CRUD operations is explicitly expressed by the "type" of the requests.

```
<%@ Control AutoEventWireup="true" %>
<!-- this tag is needed to enable jQuery IntelliSense only -->
<script src="../../Scripts/_System.js" type="text/javascript"></script>
<!-- the user interface of the control -->
<script type="text/javascript"
    src="../../appservices/Suppliers?_sortExpression=CompanyName&_instance=SupplierData">
</script>
<script type="text/javascript">
var ProductManager = {
    // Returns the url of the application server of a demo web app.
    basePath: function () { return '../../appservices'; },
    // Loads the list of categories requested from the app server via ajax request.
    loadCategories: function () {
        $.ajax({
            url: this.basePath() + '/Categories?_sortExpression=CategoryName',
            dataType: 'json',
            cache: false,
            success: function (data) {
                $.each(data.Categories, function (index, category) {
                    $('<option>').text(category.CategoryName)
                        .attr('value', category.CategoryID).appendTo($('#CategoryID'));
                });
            }
        });
    }
};
```

```

    })
  }
});
},
// Populate suppliers from the list retrieved by the second <script> element.
loadSuppliers: function () {
  $.each(MyCompany.SupplierData.Suppliers, function (index, supplier) {
    $('<option>').text(supplier.CompanyName)
      .attr('value', supplier.SupplierID).appendTo($('#SupplierID'));
  })
},
// Shows a list of products in <select> element
showProductList: function () {
  var query = '?_sortExpression=ProductName&q=' + encodeURIComponent($('#QuickFind').val());
  $.ajax({
    url: this.basePath() + '/Products' + query,
    cache: false,
    dataType: 'json',
    success: function (data) {
      var selectedValue = $('#ProductList').val();
      $('#ProductList option').remove();
      $.each(data.Products, function (index, product) {
        $('<option>')
          .text(product.ProductName + ' / ' + product.CategoryCategoryName)
          .attr('value', product.ProductID).appendTo($('#ProductList'));
      });
      $('#ProductList').val(selectedValue);
      $('#ProductListPanel').show();
    }
  });
},
// Shows a form with product details,
showProductDetails: function (productId) {
  if (productId == null) return;
  $('#ProductSearchPanel').hide();
  $('#DeleteButton').show();
  $.ajax({
    url: this.basePath() + '/Products/editForm1/' + productId,
    cache: false,
    dataType: 'json',
    success: function (product) {
      $('#ProductDetailsPanel').show();
      $('#ProductID').attr('value', product.ProductID);
      $('#ProductName').attr('value', product.ProductName).focus();
      $('#SupplierID').attr('value', product.SupplierID);
      $('#CategoryID').attr('value', product.CategoryID);
      $('#QuantityPerUnit').attr('value', product.QuantityPerUnit);
      $('#UnitPrice').attr('value', product.UnitPrice);
    }
  });
},
// Returns back to the search mode.
backToSearch: function () {
  $('#ProductDetailsPanel').hide();
  $('#ProductSearchPanel').show();
  $('#ProductList').focus();
},
// Refreshes the search result.
refreshSearch: function () {
  this.backToSearch();
  this.showProductList();
},
// Formats a URI of a product
createProductUrl: function (requestType) {
  if (requestType == 'POST')

```

```

        return this.basePath() + '/Products/createForm1';
    return this.basePath() + '/Products/editForm1/' + encodeURIComponent($('#ProductID').val());
},
// Creates an object with the properties retrieved from the input fields
// of the "ProductDetails" form.
collectFieldValues: function () {
    return {
        ProductID: $('#ProductID').val(),
        ProductName: $('#ProductName').val(),
        SupplierID: $('#SupplierID').val(),
        CategoryID: $('#CategoryID').val(),
        QuantityPerUnit: $('#QuantityPerUnit').val(),
        UnitPrice: $('#UnitPrice').val()
    };
},
// Deletes a products.
deleteProduct: function () {
    if (!confirm('Delete?')) return;
    $.ajax({
        url: this.createProductUrl('DELETE'),
        dataType: 'json',
        type: 'DELETE',
        data: ProductManager.collectFieldValues(),
        success: function (result) {
            if (result.errors)
                alert(result.errors[0].message);
            else
                ProductManager.refreshSearch();
        }
    });
},
// Shows the product form with blank values.
newProduct: function () {
    $('#ProductSearchPanel').hide();
    $('#DeleteButton').hide();
    $('#ProductDetailsPanel').show();
    $('#ProductID').attr('value', null);
    $('#ProductName').attr('value', 'New Product').focus().select();
    $('#SupplierID').attr('value', null);
    $('#CategoryID').attr('value', null);
    $('#QuantityPerUnit').attr('value', null);
    $('#UnitPrice').attr('value', null);
},
// Saves a product. If there is no value in the #ProductID hidden field then
// a new product is created by "POST" request. Otherwise an existing product
// is updated with "PUT" request.
saveProduct: function () {
    if (!confirm('Save?')) return;
    var requestType = ($('#ProductID').val() != '' ? 'PUT' : 'POST');
    $.ajax({
        url: this.createProductUrl(requestType),
        dataType: 'json',
        type: requestType,
        data: ProductManager.collectFieldValues(),
        success: function (result) {
            if (result.errors)
                alert(result.errors[0].message);
            else {
                if (requestType == 'POST')
                    alert('ID of the new product is ' + result.ProductID);
                ProductManager.refreshSearch();
            }
        }
    });
},

```

```

});
$(document).ready(function () {
    // pre-populate the drop down lists of categories and suppliers
    ProductManager.loadCategories();
    ProductManager.loadSuppliers();
    // attach event handlers to buttons
    $('#ProductListPanel,#ProductDetailsPanel').hide();
    $('#FindButton').click(function (e) {
        e.preventDefault();
        ProductManager.showProductList();
    });
    $('#ShowDetailsButton').click(function (e) {
        e.preventDefault();
        ProductManager.showProductDetails($('#ProductList').val());
    });
    $('#BackToSearchButton').click(function (e) {
        e.preventDefault();
        ProductManager.backToSearch();
    });
    $('#NewButton').click(function (e) {
        e.preventDefault();
        ProductManager.newProduct();
    });
    $('#SaveButton').click(function (e) {
        e.preventDefault();
        ProductManager.saveProduct();
    });
    $('#DeleteButton').click(function (e) {
        e.preventDefault();
        ProductManager.deleteProduct();
    });
    $('#QuickFind').focus();
});
</script>
<style type="text/css">
    body, button, input, select
    {
        font-family: Tahoma;
        font-size: 8.5pt;
    }

    #QuickFind
    {
        width: 280px;
    }

    #ProductList
    {
        width: 350px;
        height: 208px;
        margin-bottom: 4px;
    }

    #FindButton
    {
        width: 60px;
    }

    #ProductListPanel
    {
        margin-top: 4px;
    }

    .Field
    {

```

```

padding-bottom: 4px;
}

#ProductDetailsPanel
{
margin-top: 4px;
padding: 8px;
border: solid 1px silver;
display: inline-block;
}

.Field label
{
display: block;
color: green;
}

.Field input
{
width: 300px;
}
</style>
<div>
<div id="ProductSearchPanel">
<input id="QuickFind" type="text" />
<button id="FindButton">
Find</button>
<div id="ProductListPanel">
<select id="ProductList" size="5">
</select>
<div>
<button id="ShowDetailsButton">
Show Details</button>
<button id="NewButton">
New Product</button>
</div>
</div>
</div>
<div id="ProductDetailsPanel">
<input id="ProductID" type="hidden" />
<div class="Field">
<label for="ProductName">
Product Name:</label>
<input id="ProductName" />
</div>
<div class="Field">
<label for="SupplierID">
Supplier Company Name:</label>
<select id="SupplierID">
</select>
</div>
<div class="Field">
<label for="CategoryID">
Category Name:</label>
<select id="CategoryID">
</select>
</div>
<div class="Field">
<label for="QuantityPerUnit">
Quantity Per Unit:</label>
<input id="QuantityPerUnit" />
</div>
<div class="Field">
<label for="UnitPrice">
Unit Price:</label>

```



```
        <input id="UnitPrice" />
    </div>
    <div>
        <button id="BackToSearchButton">
            Back To Search</button>
        <button id="SaveButton">
            Save</button>
        <button id="DeleteButton">
            Delete</button>
    </div>
</div>
</div>
```