Building data driven mobile websites with MVC & jQuery Mobile

RACHEL APPEL

HTTP://RACHELAPPEL.COM

RACHEL@RACHELAPPEL.COM

Agenda

- Overview of ASP.NET MVC & jQuery Mobile
- Building the data model
- Accessing data with MVC
- Mobilizing your website with jQuery Mobile

ASP.NET MVC

- Models
- Views
- Controllers
- ViewModels

ASP.NET MVC Overview

- ASP.NET implementation of MVC
- MVC Pattern
  - What about other patterns?
  - MVVM Pattern, MVW, or MV* Patterns
- Routing
- RESTful

Models

- The application’s data
- Expressed in code as classes and their members
- Contains relationships
- Mapped to database objects

Models

```
namespace Bakery.Models
{
    public class Category
    {
        [Key]
        public int Id { get; set; }
        public string Name { get; set; }

        public virtual ICollection<Product> Products { get; set; }
    }
}
```

Models

```
namespace Bakery.Models
{
    public class Product
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        public string Image { get; set; }
        public decimal Price { get; set; }
        public DateTime ExpirationDate { get; set; }
        public int QuantityOnHand { get; set; }
    }
}
```

Entity Framework

- Entity Framework
  - Code First
  - Database First
  - Model First
- DbSet<T>
- Database Initializer (Code first)
- DBContext

Entity Framework

```
public class BakeryContext : DbContext
{
    public DbSet<CartItem> CartItem { get; set; }
    public DbSet<Order> Order { get; set; }
    public DbSet<OrderDetail> OrderDetails { get; set; }
    public DbSet<ShoppingCart> ShoppingCart { get; set; }
    public DbSet<Category> Category { get; set; }
    public DbSet<Product> Products { get; set; }
}
```

Entity Framework

In the global.asax.cs file

```
System.Data.Entity.Database.SetInitializer(
      new jQueryMobile.Models.BakeryContextInitializer());
```

Views

- The UI/Presentation layer
- Renders a model or ViewModel
- Does not contain business logic
- A visualization of data
- Expects data from one source: a model or ViewModel
- Use HTML Helpers or custom HTML

Views

- Helpers
- Links
- Controls

Views

```
@model IEnumerable<Bakery.Models.Product>
@foreach (var item in Model) {
    <tr>
        <td>@Html.DisplayFor(modelItem => item.Name) </td>
        <td>@Html.DisplayFor(modelItem => item.Description) </td>
        <td>         @
         </td>
        <td>@Html.DisplayFor(modelItem => item.Price) </td>
        <td>@Html.DisplayFor(modelItem => item.QuantityOnHand) </td>
        <td>@Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
            @Html.ActionLink("Details", "Details", new { id=item.Id }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.Id })
        </td>
    </tr>
}
```

Controllers

- Traffic Cop
- Connect models and views
- Perform routing
- Accepts HTTP requests
- Front door of application
- Security

Controllers

```
namespace Bakery.Controllers
{
    public class ProductsController : Controller
    {
        private BakeryContext db = new BakeryContext();

        public ActionResult Index()
        {
            return View(db.Products.ToList());
        }
     }
}
```

Controllers

- HTTP POST Data & HTTP Verbs

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit(Product product)
{
    if (ModelState.IsValid)
    {
        db.Entry(product).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(product);
}
```

ViewModels

- A representation of one or more models
- Formatted & polished data
- UI logic code to format data
- One single ViewModel object per view
- Promotes SOC (Separation of Concerns)

ViewModels

```
public class CustomerViewModel
{
    public Customer Customer { get; set; }
    public StatesDictionary States { get; set;
}
    public CustomerViewModel(Customer customer)
    {
        Customer = customer;
        States = new StatesDictionary();
    }
}
```

Mobilization with jQuery Mobile

- Built on top of jQuery
- HTML5
- Optimized for mobile displays and touch
- Cross platform
- Included in Visual Studio MVC 4 Mobile Project Template

Mobilization with jQuery Mobile

- Visual Studio 2012 Project Template
  - ASP.NET MVC Mobile Internet Template
- Versioned CDN (Content Delivery Network)
- Source at 100-140k minified 250-350 standard depending on version

- Small and streamlined; CDN: jquery.mobile.structure-*.js

Mobilization with jQuery Mobile

- Themes
- Theme Roller

Mobilization with jQuery Mobile

data-role attributes

http://api.jquerymobile.com/data-attribute/

```
<ul data-role="listview" data-inset="true">
    <li data-role="list-divider">Navigation</li>
    <li>@Html.ActionLink("View Products", "Index", "Products")</li>
    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
```

Mobilized Razor Layout Pages, roles, and themes

```
<body>
    <div id="layoutpage" data-role="page" data-theme="b">
    <div data-role="header">
            @if (IsSectionDefined("Header")) {
                @RenderSection("Header")
            } else {
        <h1>@ViewBag.Title</h1>
                @Html.Partial("_LogOnPartial")
            }
    </div>
    <div data-role="content">
    @RenderBody()
    </div>
    </div>
</body>
```

Mobilized Controls: Lists

```
<ul data-role="listview">
    @foreach (var item in Model)
    {
    <li>
        <img src=@Url.Content("~/Content/Images/Products/Thumbnails/")@item.ImageName
alt="Image"  />
        @Html.ActionLink((string)item.Name, "Details", "Products", new { id = item.Id
}, null)</li>
    }
</ul>
```

Page Mobilization and SPA architecture

- Page Anatomy
- http://jquerymobile.com/demos/1.2.0/docs/pages/page-anatomy.html
- SPA style

="page"

Sumary

- Models, Views, Controllers
- jQuery Mobilization