INFO3067 Week 5 Class 2

Review

- Case 1
- Midterm

Using a Mail Client

One of the things that websites typically do is send email to their users. We're going to add this capability in a couple of places. The first place we'll use it is for a registration confirmation, the other place we'll send an email is when the customer has ordered a good. Fortunately with the Asp.Net environment this is very easy to do. Before looking at the code to do this we will need to install a smtp server to test this functionality out. I suggest to keep the overhead to a minimum to install a simple smtp driver called **Smtp4Dev**. You can download this software at: http://smtp4dev.codeplex.com/

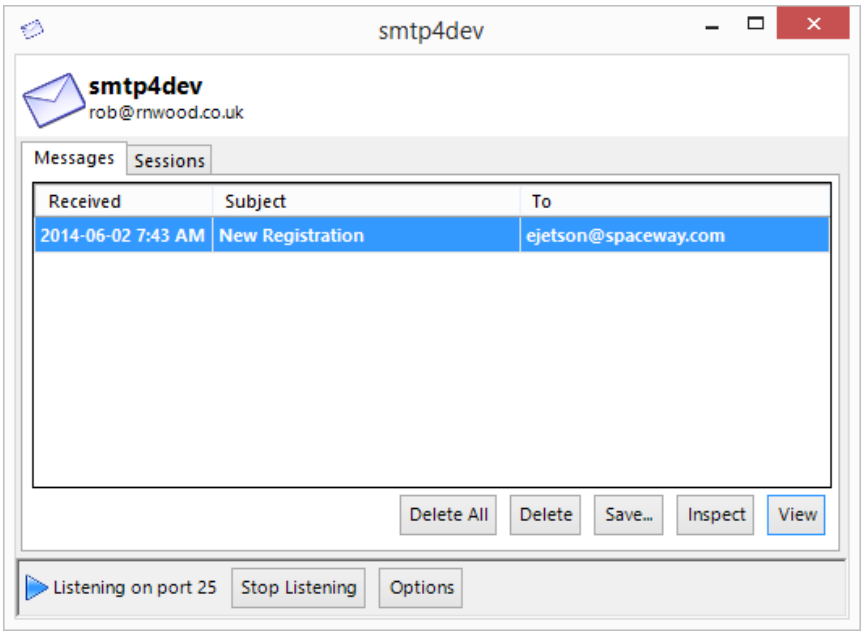
Now to send mail via our register method in the home controller it's simply a matter of using 3 classes; **SmtpClient, MailAddress** and **MailMessage**. To use these objects, add a using to the **Register** page for **System.Net.Mail.** The code to send a mail message is highlighted below:

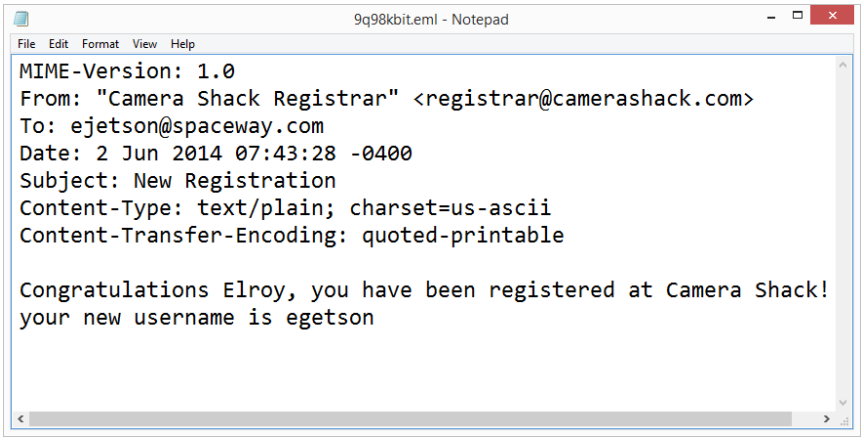
```
var user = new ApplicationUser() { UserName = model.Username };
var result = await UserManager.CreateAsync(user, model.Password);
if (result.Succeeded)
{
    ViewBag.Message = model.Message + ". Please proceed to login";
    MailMessage msg = new MailMessage();
    msg.Subject = "New Registration";
    msg.From = new MailAddress("registrar@camerashack.com", "Camera Shack Registrar");
    msg.To.Add(new MailAddress(model.Email));
    msg.Body = "Congratulations " + model.Firstname +
               ", you have been registered at Camera Shack!" +
              " your new username is " + model.Username;
    using (SmtpClient mailClient = new SmtpClient())
    {
        mailClient.Send(msg);
    }
}
else
```

To hook the mail client to the smtp4dev server we need to make an addition to the site's **Web.config** file like this (starting with <system.net), place right before </configuration>:

```
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="Network">
        <network host="localhost" port="25"/>
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

Start the smtp driver, register a new user and you should see the email show up





You may see some funny characters (eg. =) show up, a normal mail client will decipher those correctly.

Now add the code necessary to send a mail message after ordering a good.

Case Study cont'd – Order Viewing

Now that we generated some orders, it would be nice to allow the user to be able to look at them. This kind of processing can cause a lot of back and forth between the browser and user. To streamline this, we are going to let the client's browser do most of the processing by providing just the raw data in **JSON** format to the client and do the front end coding in JQuery.

An Easier Javascript Programming Environment – JQuery

JQuery is available at http://jquery.com and as we've already seen built right into VS 2013. Its goal is to make cross browser coding less tedious and make coding DOM calls easier. There is a little bit of a learning curve to using this library but once you get it, it is indeed easier. I've provided the JQuery versions for 5 exercises to get you started (note these examples are using **jquery-1.10.2.min**.js so if you're using a different version you'll need to update the scripts accordingly.

To use JQuery we don't have to code the onclick attribute in our html. We can simply code a JQuery event handler like this:

```
var oldRow;

$(document).ready(function() {

    $("td").click(function() {
        if (oldRow != null)
            $(oldRow).toggleClass("itemSelected");
        $(this).parents("tr").toggleClass("itemSelected");
        oldRow = $(this).parents("tr");
    });
});
```

Or as we have seen you can replace the $(document).ready(function() with just:
$(function () {

The **$(document).ready(function. Or $(functiion..** will fire when the page loads . You then embed the event handler you want to keep track of, in the example when **a td cell** is clicked on the page. Here is a link for a quick JQuery tutorial to get you started:

1. http://dotnetslackers.com/articles/ajax/JQuery-Primer-Part-1.aspx

Let's look at some JQuery examples that will help with the concept of building html tables dynamically (for you css purists who don't use html tables you can come up with your own solution or just grin and bear it for the next couple of classes).

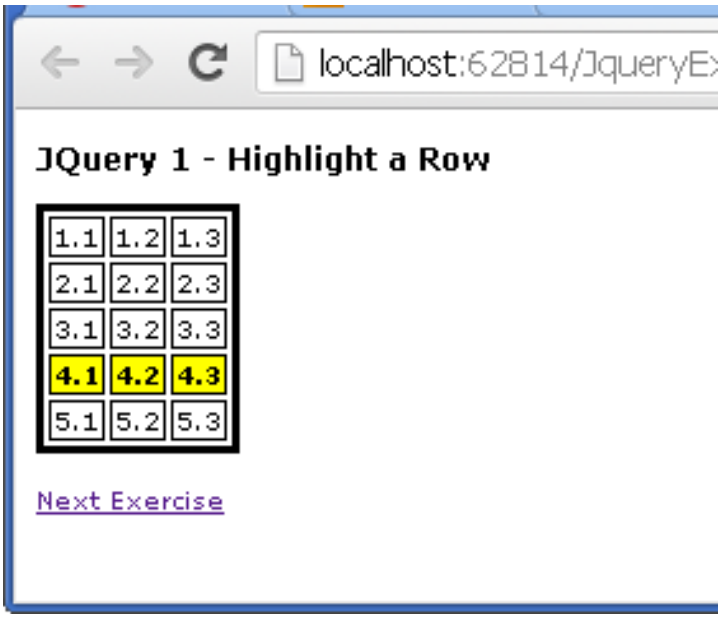
Example 1 - displays an HTML table and highlights the row clicked on.

- **Notes** – here we have a hardcoded table. We have registered the JQuery library with a script tag and we have the default ready function from above. When the user clicks on any table cell (<td>) we execute the following code:

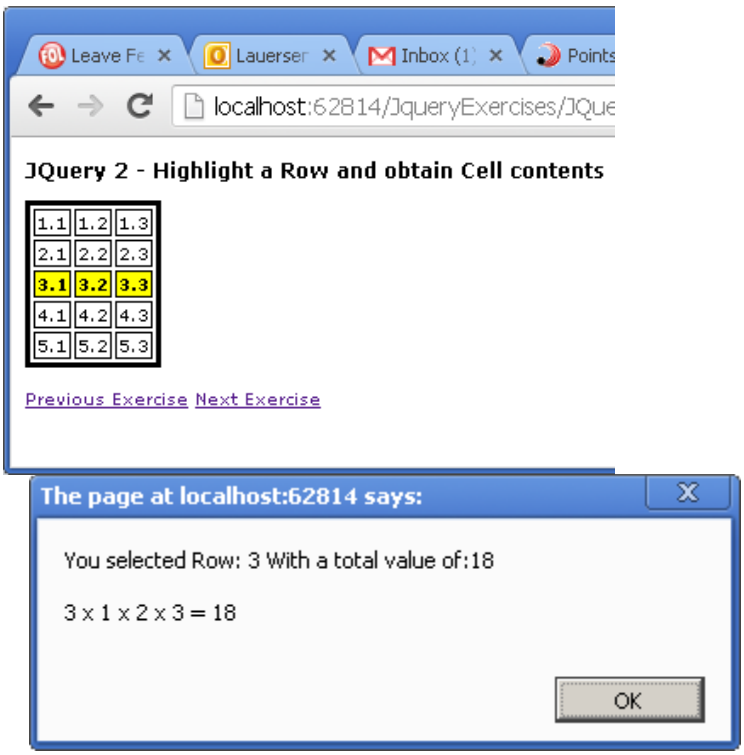
```
$("td").click(function() {
    if (oldRow != null)
        $(oldRow).toggleClass("itemSelected");
    $(this).parents("tr").toggleClass("itemSelected");
    oldRow = $(this).parents("tr");
});
```

  - This code basically toggles a table row's style (itemSelected) by adding and removing the class attribute

- See .zip file ***JQuery1.htm***.



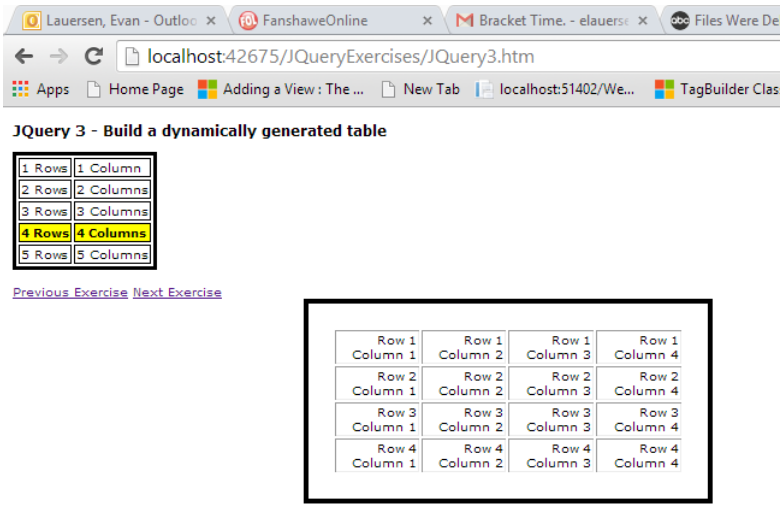
Exercise # 2 - displays a table and obtains data from individual table cells



- **Notes for example 2**: Builds on the first example. Processes the entire selected row cell by cell by taking the value in front of the period and multiplying by the value after the period and sums each cell to a total. Then it displays the total via an alert statement. See .zip file ***JQuery2.htm***.

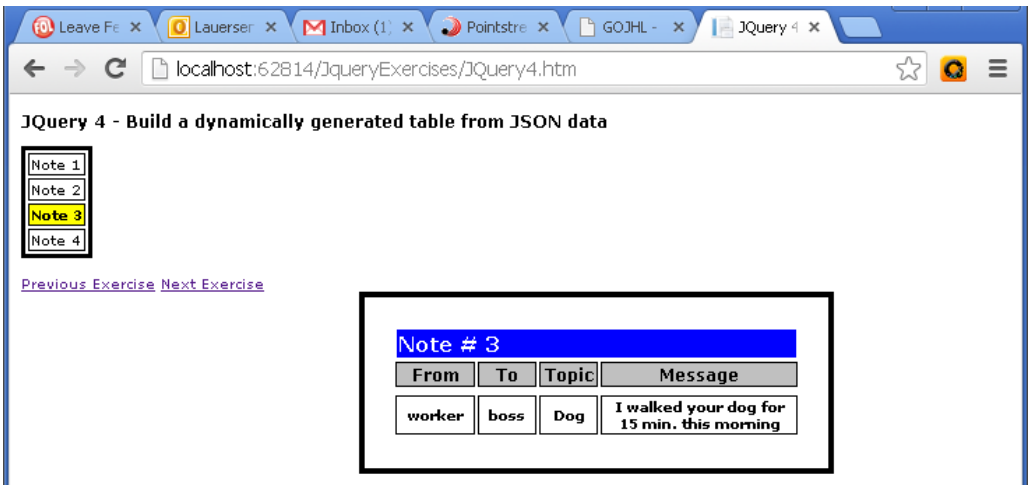
```
        oldRow = $(this).parents("tr");                 // determine row
        oldRow.toggleClass("itemSelected");             // turn off previous row

        $('tr.itemSelected td').each(function() {       // loop through each cell in the row
            cellContents = $(this).html();
            val1 = cellContents.substring(cellContents.indexOf(".") + 1, cellContents.length); // after decimal
            val2 = cellContents.substring(cellContents.indexOf(".") - 1, 1);                   // before decimal
            sumOfTdValues = sumOfTdValues + (val1 * val2);
        });
        alert("You selected Row: " + val2 + " With a total value of:" +
            sumOfTdValues + "\n\n" + val2 + " x 1 x 2 x 3 = " + sumOfTdValues);
        sumOfTdValues = 0;
    });
});
```

Exercise 3 - generate a new table from data obtained from an existing table



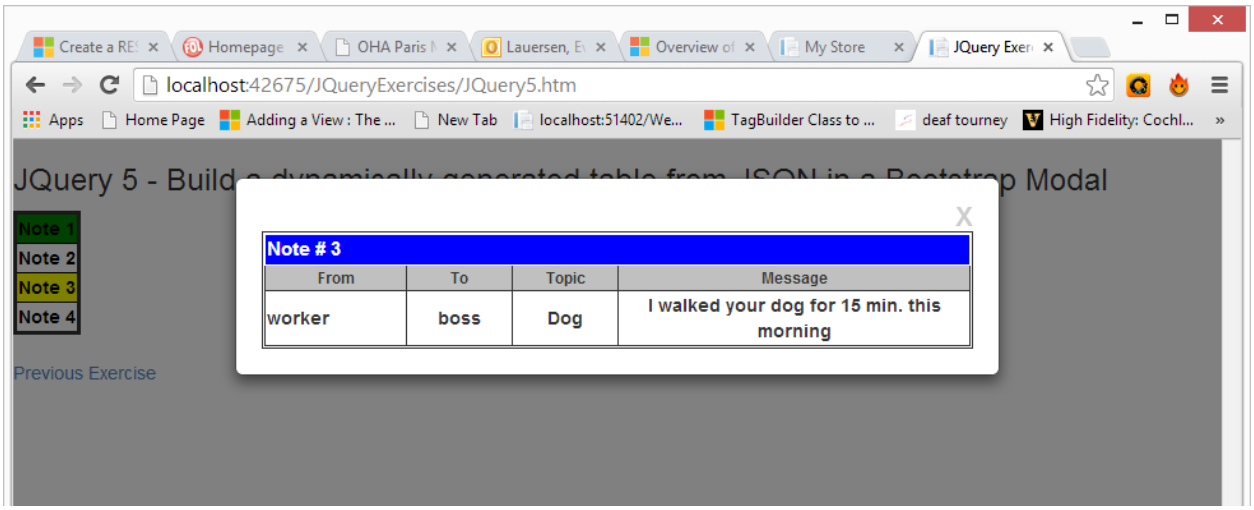
- Notes for example 3: Builds a secondary table based on the selection made from the first table. This is what we'll be doing a little later on (selecting an order row in one table, and then generating the order details in another table). Deletes any rows in an existing table first. Adds cells to row in inner loop. Adds rows to table in outer loop. See .zip file ***JQuery3.htm***.

Exercise 4 - generates an html table from JSON object array



- Headings for the lower table are built before any json data is processed.
- After the headings are generated, the JSON collection is searched until a matching array element = the note #. When the desired note is found the details are obtained and the results displayed.
- See .zip file ***JQuery4.htm***.

Example 5 - display JSON information in a Bootstrap modal



- Pretty much the same as the prevous example, except here we place the output in a bootstrap modal window (a number of libraries were added to facilitate this)

See .zip file ***JQuery5.htm***.

Lab 8

- Update the register process **and** order process to include sending an email
- Study the code and try the JQuery 1-5 examples (feel free to make changes if you like)
- Create a 6th exercise that modifies exercise 5 and adds a new property to the JSON note object called **date** (just fill in a date for the data) and then modify the code to display the new data (make sure heading markup and table format correctly):
- Submit screen shots of:
    1. Successful registration generated email
    2. Successful order generated email
    3. JQuery 6 results.