Building Blocks – pt 1

Our Vision

“Though it is the smallest of all seeds, yet when it grows, it is the largest of garden plants and becomes a tree, so that the birds come and perch in its branches.”
– Jesus

Prerequisites

- Visual Studio or Visual Studio Express
- SQL Server or SQL Server Express

QuickStart Tutorials

From 0 to 60 in about 60 minutes

Rockit SDK

- rockrms.com/Rock/Developer/Rockit

My Church

mychurch.org

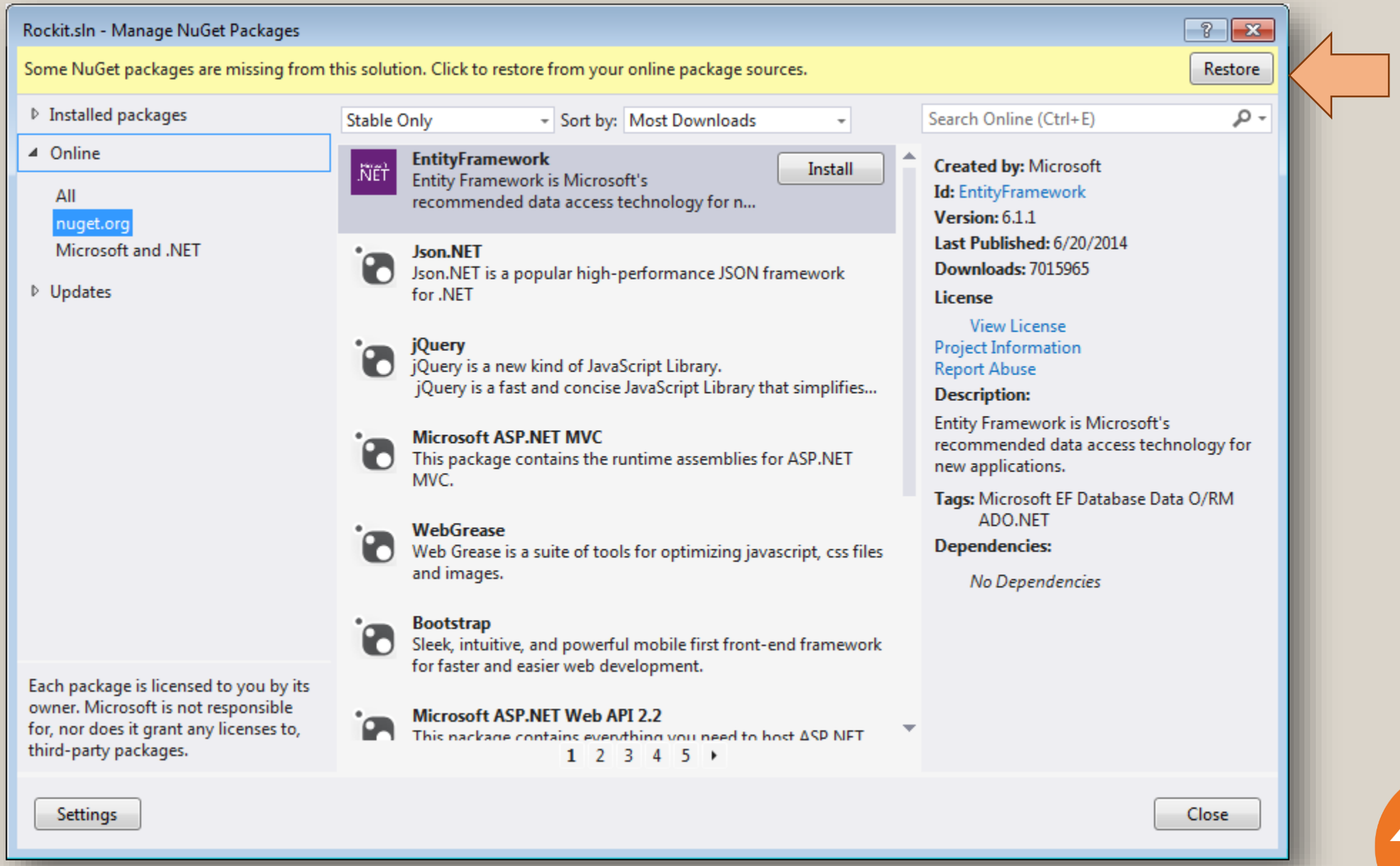
Unzip to a folder

- Open the solution (.sln) file
- Edit web.ConnectionStrings.config and replace with your database name

NOTE

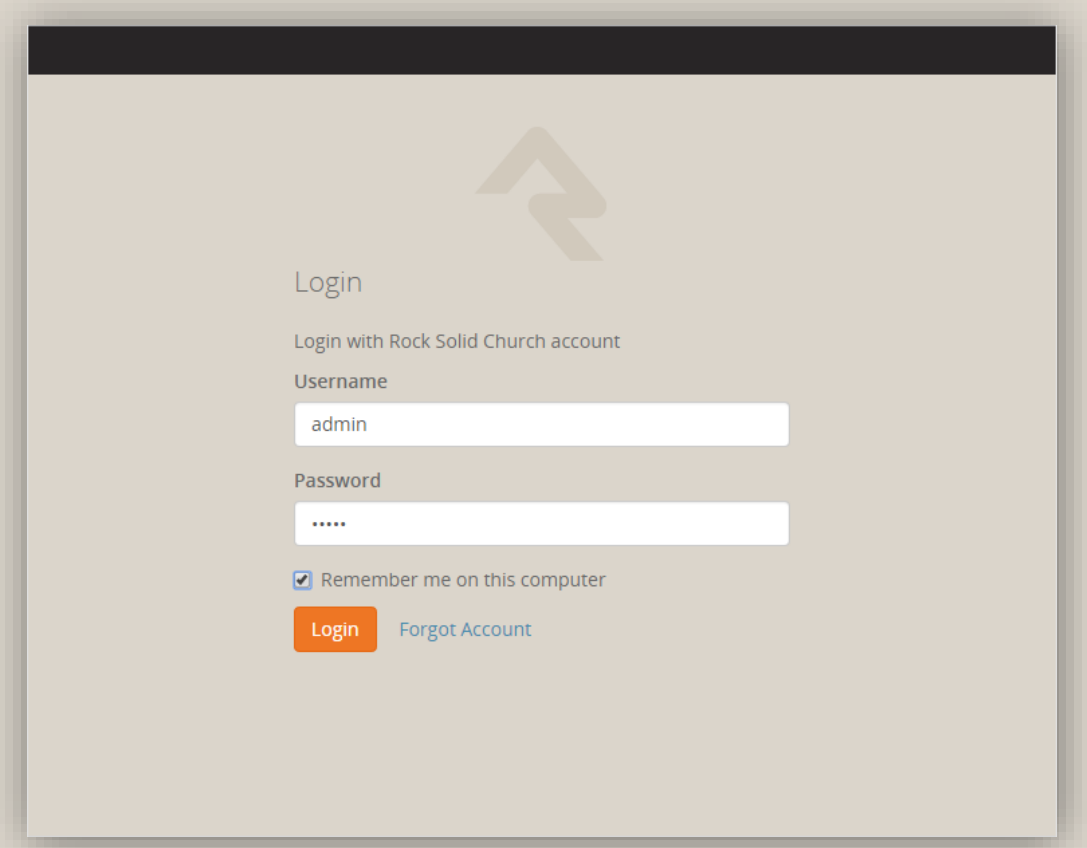
If your project won't compile, check your NuGet Managed Packages.

Tools > NuGet Package Manager >
Manage NuGet Packages for Solution...


Rockit.sln - Manage NuGet Packages
Some NuGet packages are missing from this solution. Click to restore from your online package sources.
Restore
Installed packages
Online
All
nuget.org
Microsoft and .NET
Updates
Stable Only
Sort by: Most Downloads
Search Online (Ctrl+E)
EntityFramework
Entity Framework is Microsoft's recommended data access technology for n...
Install
Json.NET
Json.NET is a popular high-performance JSON framework for .NET
jQuery
jQuery is a new kind of JavaScript Library.
jQuery is a fast and concise JavaScript Library that simplifies...
Microsoft ASP.NET MVC
This package contains the runtime assemblies for ASP.NET MVC.
WebGrease
Web Grease is a suite of tools for optimizing javascript, css files and images.
Bootstrap
Sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.
Microsoft ASP.NET Web API 2.2
1 2 3 4 5
Created by: Microsoft
Id: EntityFramework
Version: 6.1.1
Last Published: 6/20/2014
Downloads: 7015965
License
View License
Project Information
Report Abuse
Description:
Entity Framework is Microsoft's recommended data access technology for new applications.
Tags: Microsoft EF Database Data O/RM ADO.NET
Dependencies:
No Dependencies
Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Settings
Close

F5

- Because of the App_Data\Run.Migration file
  - A database is created
  - Schema is added
  - Data is added



See Rocksolidchurch.org?

- When you see it or any variation in this tutorial:
  - org_rocksolidchurch
  - org.rocksolidchurch
  - _org.rocksolidchurch

- Replace with your own organization name

Hello World

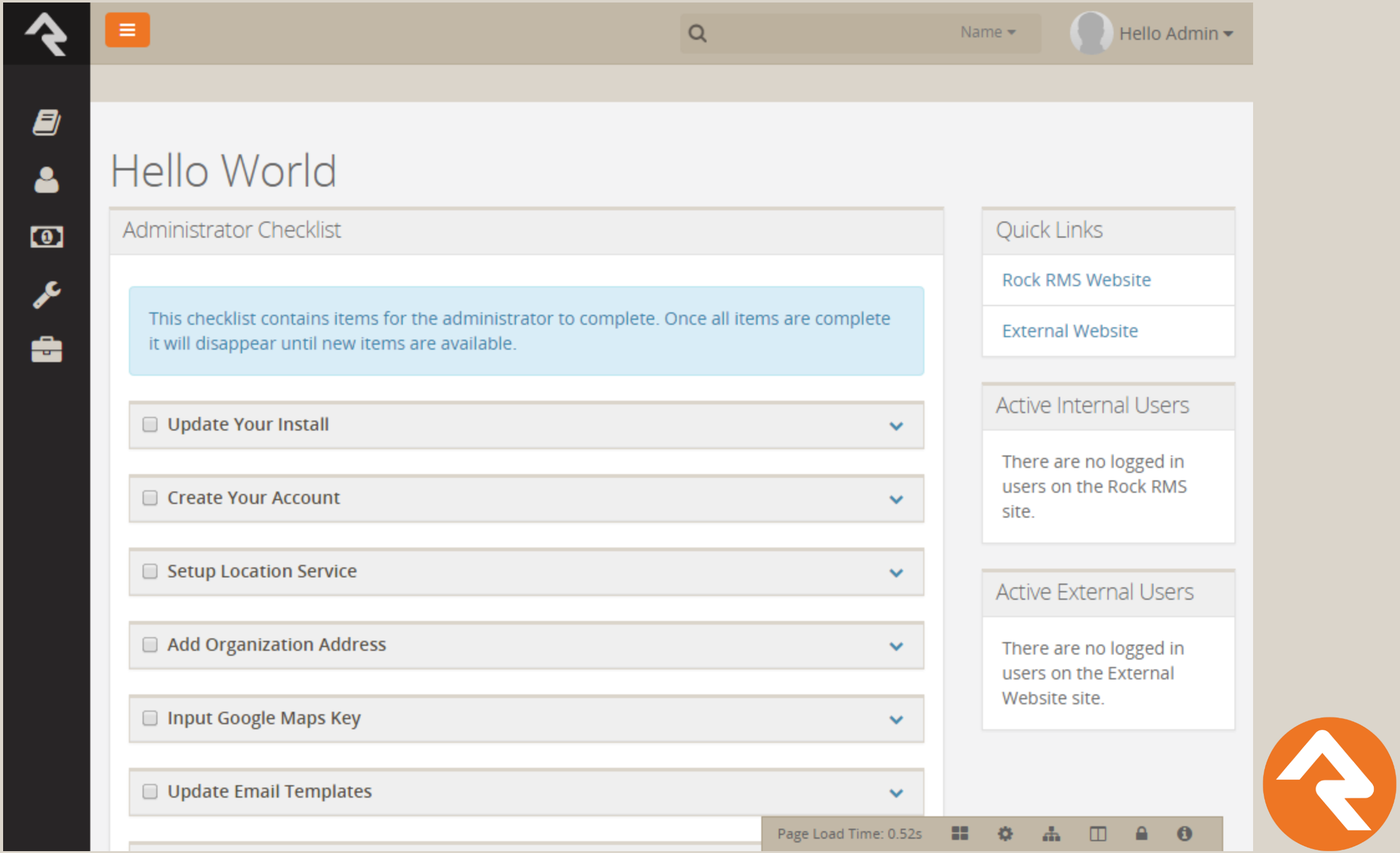
Getting comfortable in Rock

HelloWorld.ascx

- Step 1 – Add New Item
  - Right-click your RockWeb\Plugins\org_rocksolidchurch\
  - Add New Item → Web User Control
- Call it HelloWorld
- Add “Hello World” to the markup (in the .ascx)
- Change from inheritance from System.Web.UI.UserControl to Rock.Web.UI.RockBlock

Register & Add to Page

1. Click Zone Editor
2. Click the fly-out-toolbar in the Feature Zone
3. Click the Add button
4. Add your new block (bottom of list; uncategorized)
5. Refresh page


Name
Hello Admin
Hello World
Administrator Checklist
This checklist contains items for the administrator to complete. Once all items are complete it will disappear until new items are available.
Update Your Install
Create Your Account
Setup Location Service
Add Organization Address
Input Google Maps Key
Update Email Templates
Quick Links
Rock RMS Website
External Website
Active Internal Users
There are no logged in users on the Rock RMS site.
Active External Users
There are no logged in users on the External Website site.
Page Load Time: 0.52s

Fetching Data

Getting familiar with the service layers

StarkList.ascx

- Copy Blocks/Utility/Stark.ascx into your folder & rename it PeopleList.ascx(.cs)
- Change Inherits="RockWeb.Blocks.Utility.StarkList“ to Inherits="RockWeb.Plugins.org_rocksolidchurch.PeopleLIst“
- Change the corresponding namespace and class name too
- Change the block attributes

```
[DisplayName( "People List" )]
[Category( "org_rocksolidchurch > Tutorials" )]
[Description( "Fetch a bunch of people" )]
```

Markup – Rock:Grid

```
 <Rock:Grid ID="gList" runat="server" AllowSorting="true">
    <Columns>
        <asp:BoundField DataField="FullName" HeaderText="Name" SortExpression="FullName" />
        <asp:BoundField DataField="Gender" HeaderText="Gender" SortExpression="Gender" />
        <asp:BoundField DataField="Email" HeaderText="Email" SortExpression="Email" />
    </Columns>
</Rock:Grid>
```

BindGrid()

```
RockContext rockContext = new RockContext();
PersonService personService = new PersonService( rockContext );

var qry = personService.Queryable()
        .Where( p => p.Gender == Gender.Male)
        .Take(10);

gList.DataSource = qry.ToList();
gList.DataBind();
```

People

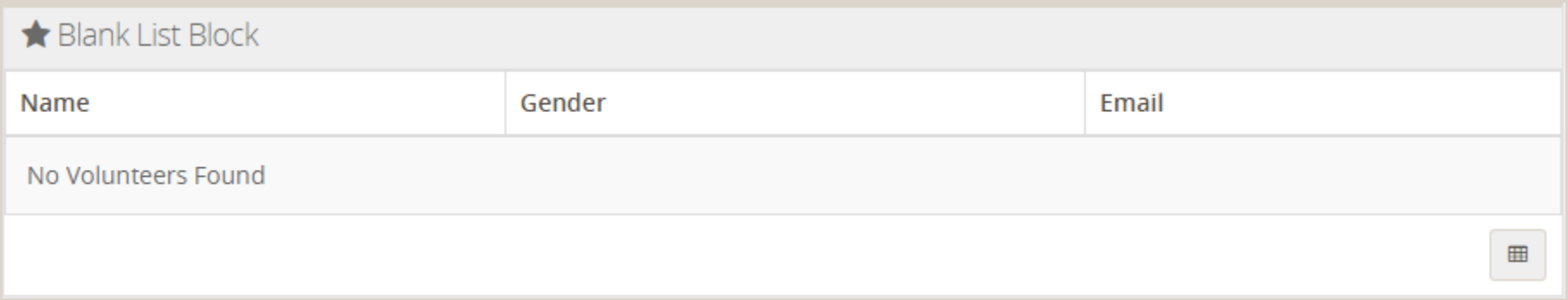
★ Blank List Block

Name	Gender	Email
No People Found		

Volunteers

```
 <Rock:Grid ID="gList" runat="server" AllowSorting="true“ RowItemText="Volunteer">
    <Columns>
        <asp:BoundField DataField="FullName" HeaderText="Name" SortExpression="FullName" />
        <asp:BoundField DataField="Gender" HeaderText="Gender" SortExpression="Gender" />
        <asp:BoundField DataField="Email" HeaderText="Email" SortExpression="Email" />
    </Columns>
</Rock:Grid>
```

Pluralized



Sample Data?

- Admin Tools > Power Tools

Configurable

Providing options via Block Attributes

Block Attribute Fields

```
[DisplayName( "People List" )]
[Category( "org_rocksolidchurch > Tutorials" )]
[Description( "Fetch a bunch of people" )]
[CustomRadioListField( "Gender Filter", "Select to list only records for that gender",
     "Male,Female", required: false )]
public partial class PeopleList : Rock.Web.UI.RockBlock
{
```

Use the Attribute Value

```
private void BindGrid()
{
      RockContext rockContext = new RockContext();
      PersonService personService = new PersonService( rockContext );

       var qry = personService.Queryable();

       var genderValue = GetAttributeValue( "GenderFilter" );
       if ( !string.IsNullOrEmpty( genderValue ) )
       {
           var gender = genderValue.ConvertToEnum<Gender>();
           qry = qry.Where( p => p.Gender == gender );
       }

       gList.DataSource = qry.ToList();
       gList.DataBind();
}
```

- AccountField
- AccountsField
- AttributeCategoryField
- AttributeField
- BinaryFileTypeField
- BooleanField
- CampusesField
- CampusField
- CategoryField
- CodeEditorField
- CommunicationTemplateField
- ComparisonField
- ComponentField
- ComponentsField
- ContentChannelField
- CurrencyField
- CustomCheckboxListField
- CustomDropdownListField
- CustomRadioListField
- DateField
- DateRangeField
- DayOfWeekField
- DaysOfWeekField
- DecimalField
- DecimalRangeField
- DefinedTypeField
- DefinedValueField
- EmailField
- EmailTemplateField
- EntityField
- EntityTypeField
- EnumField
- EnumsField
- GroupField
- GroupLocationTypeField
- GroupRoleField
- GroupTypeField
- GroupTypeGroupField
- GroupTypesField
- IntegerField
- IntegerRangeField
- LinkedPage
- LocationField
- MemoField
- MetricCategoriesField
- MetricEntityField
- PersonBadgesField
- PersonField
- RemoteAuthsField
- SchedulesField
- SecurityRoleField
- SiteField
- SlidingDateRangeField
- TextField
- TimeField
- UrlLinkField
- WorkflowActivityType
- WorkflowAttribute
- WorkflowTextOrAttribute
- WorkflowTypeField

OnRowSelected to Known Route

- Markup

```
<Rock:Grid ID="gList" runat="server" AllowSorting="true" RowItemText="Volunteer"
                      DataKeyNames="Id" OnRowSelected="gList_RowSelected">
```

- Code

```
protected void gList_RowSelected( object sender, RowEventArgs e )
{
    int personId = (int)e.RowKeyValue;
    Response.Redirect( string.Format( "~/Person/{0}", personId ), false );
}
```

To a LinkedPage

- Block Attribute

```
[LinkedPage( "Related Page" )]
```

- Code

```
protected void gList_RowSelected( object sender, RowEventArgs e )
{
    NavigateToLinkedPage( "RelatedPage", "PersonId", e.RowKeyId );
}
```

Access Control

Enabling features based on authorization

Rock.Security.Authorization

- VIEW
- EDIT
- APPROVE
- ADMINISTRATE

IsUserAuthorized()

- Show Add Button

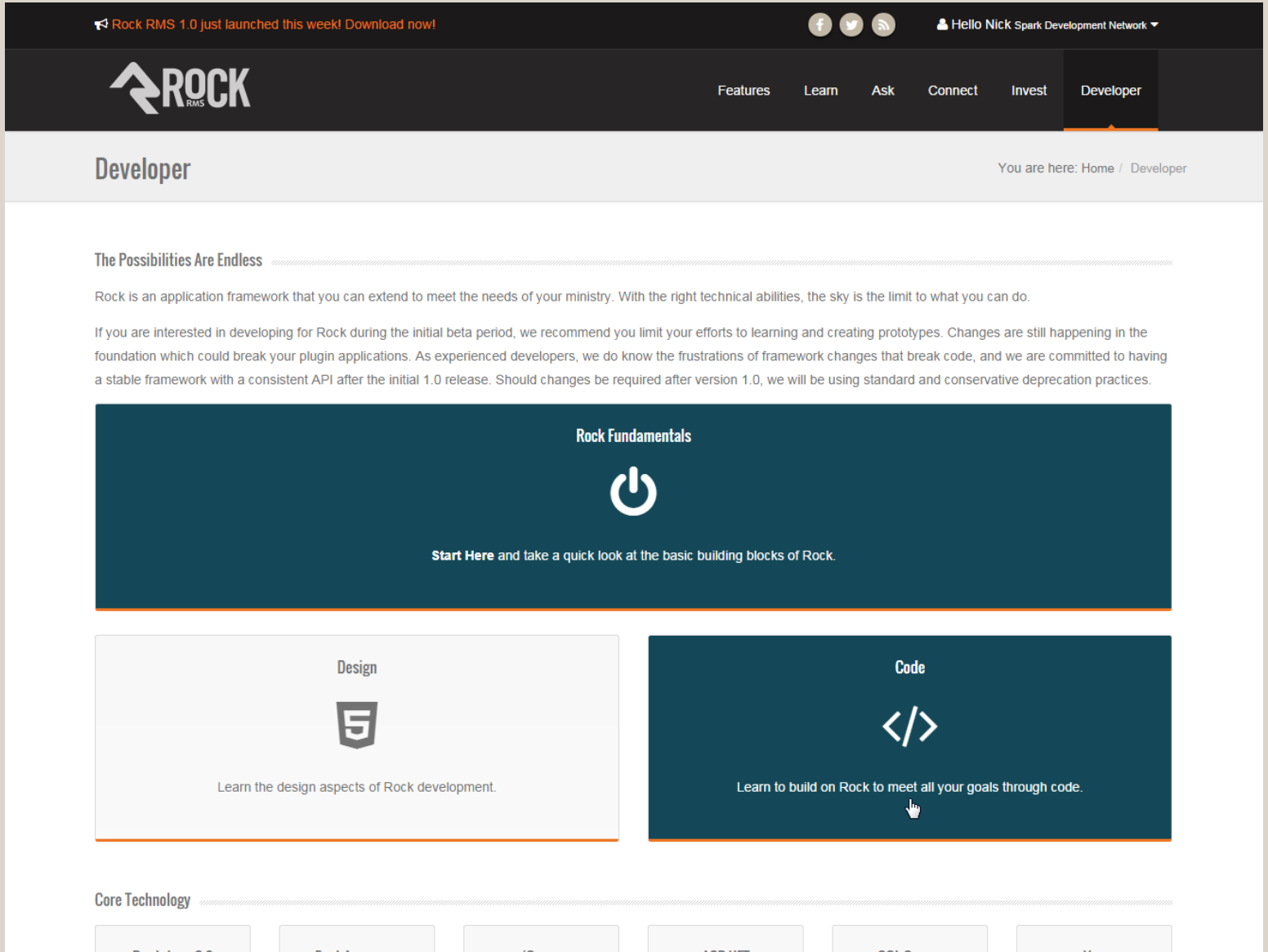
```
if ( IsUserAuthorized( Rock.Security.Authorization.EDIT ) )
{
    gList.Actions.ShowAdd = true;
}
```

- Define and add click handler

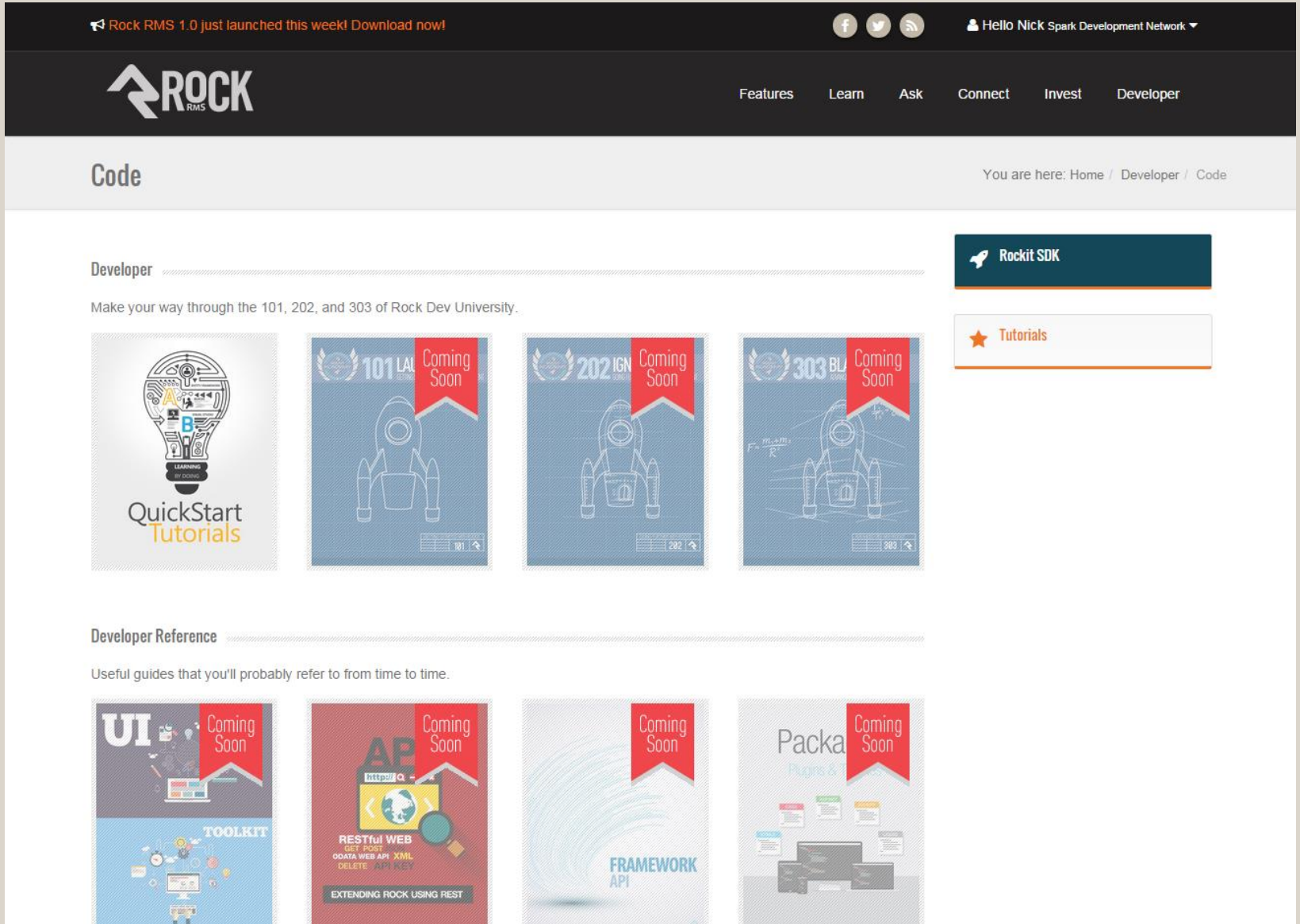
```
gList.Actions.AddClick += Actions_AddClick;
```

```
void Actions_AddClick( object sender, EventArgs e )
{
    Response.Redirect( "~/NewFamily/" );
}
```

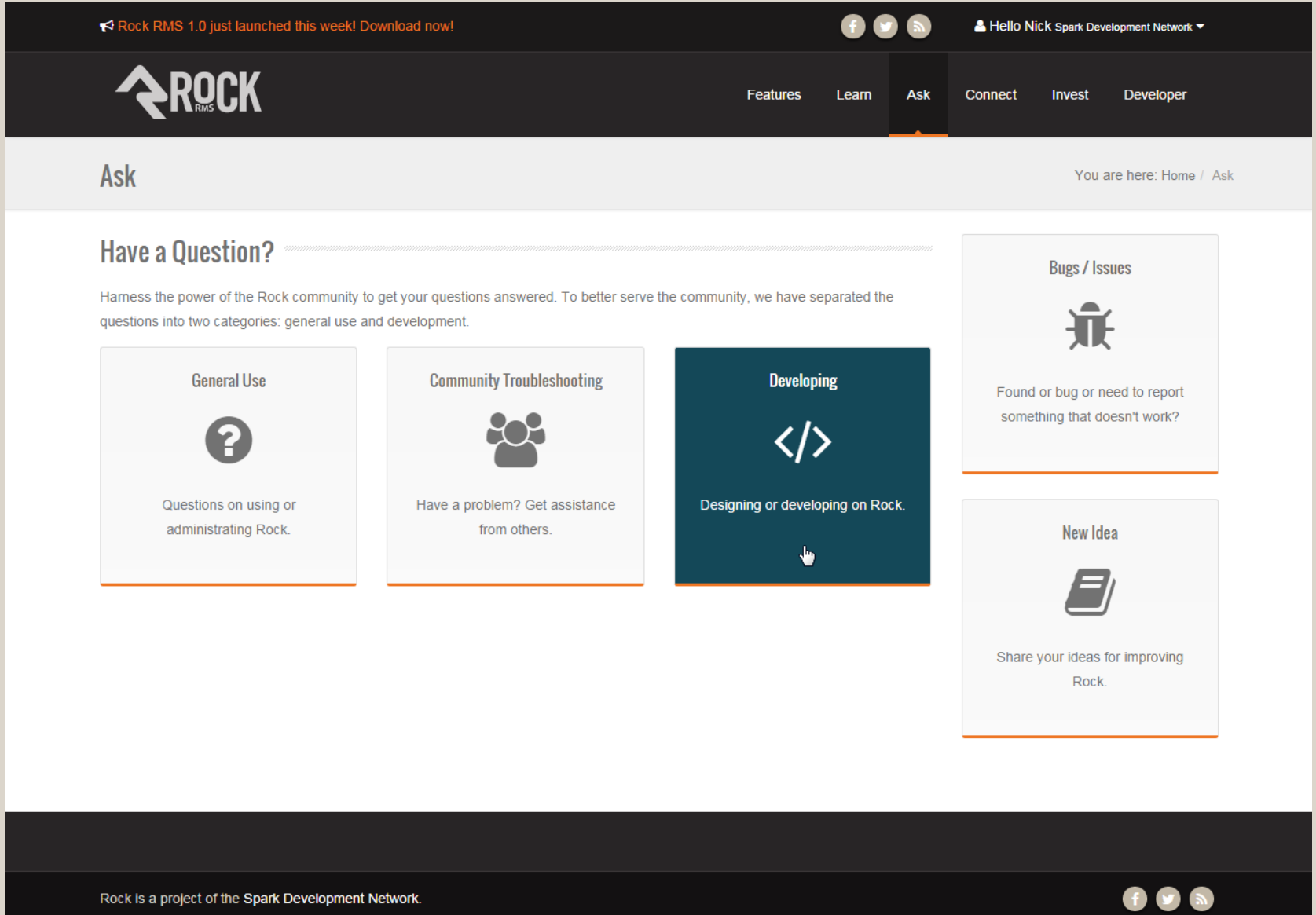
rockrms.com/Rock/Developer



rockrms.com/Rock/Developer/Code



rockrms.com/Rock/Ask



CITRT Sessions

Wednesday

- Rock Liquid (1:15 pm)
- Odata & REST (2:15 pm)
- Intro to Rock (3:30 pm)
- Rock Extensibility (4:00 pm)
- Building Blocks I (4:30 pm)
- Cascading Dataviews (5:00 pm)
- Rock Dev Env Setup (7:00 pm)
- Building Blocks II (7:30 pm)
- Importing Data (8:30 pm)

Thursday

- Bootstrap & LESS (1:15 pm – Room B)
- Rock CMS (3:30 pm)
- Rock Security (4:30 pm)
- Building Workflows I (5:00 pm)
- Dynamic Dataviews (9:00 pm)
- Custom WF Actions (9:30 pm)

All classes are in Room A unless otherwise noted