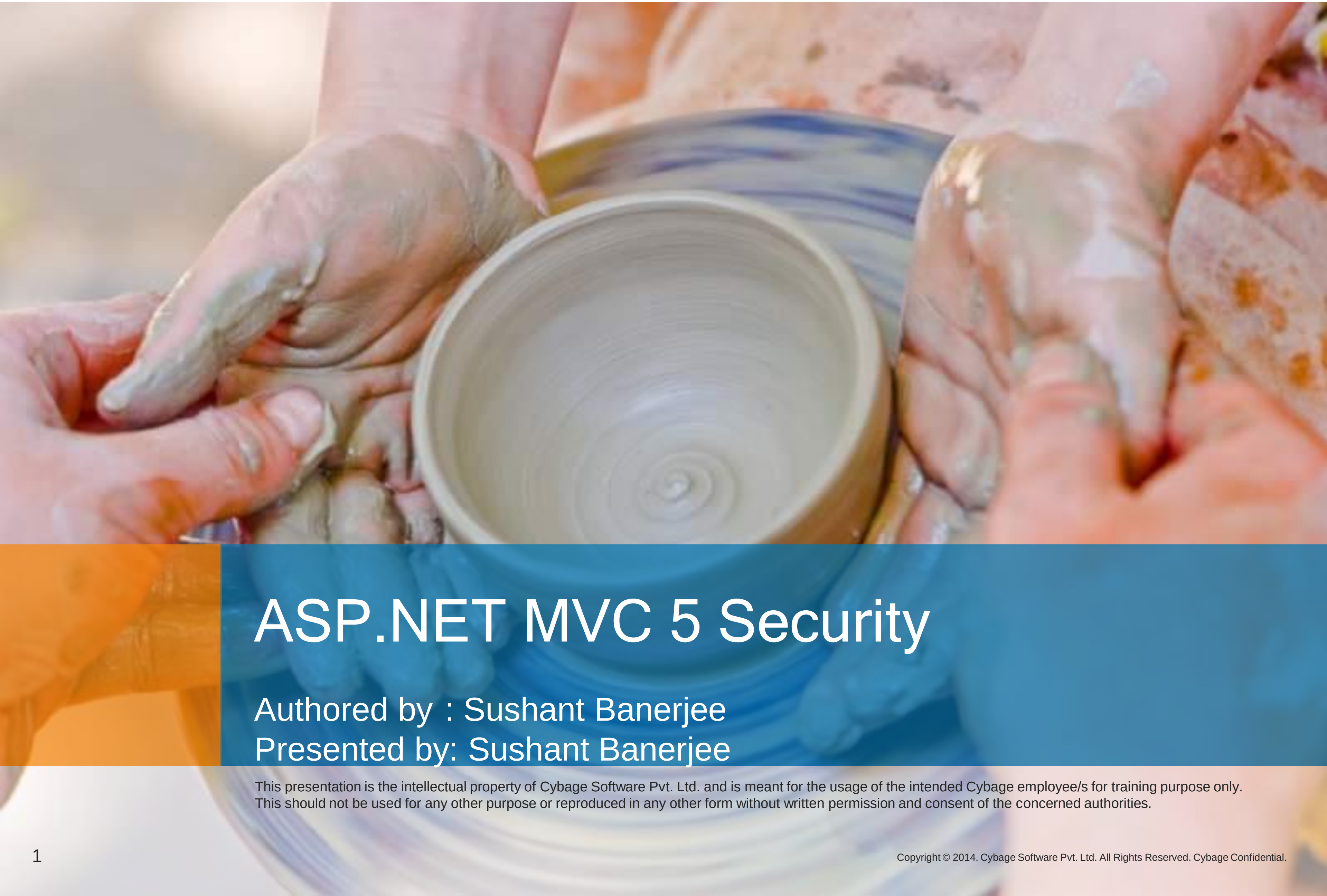


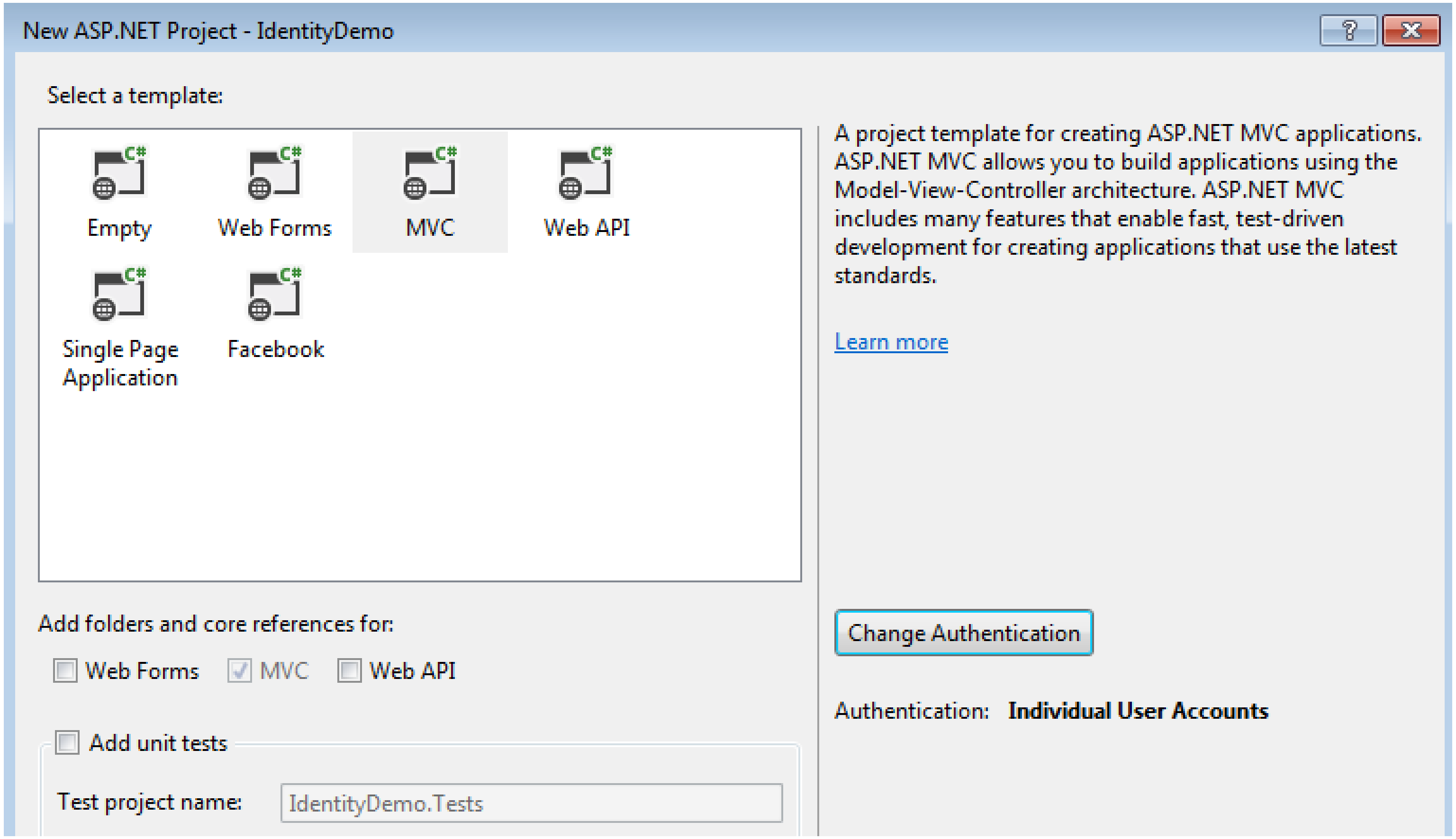
ASP.NET MVC 5 Security

Authored by : Sushant Banerjee
Presented by: Sushant Banerjee

Agenda

- Security in ASP.NET MVC 5
- Identity
- Creating Users and Roles
- XSS AND CSRF
- Using External Authentication

Select Authentication



New ASP.NET Project - IdentityDemo

Select a template:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Add folders and core references for:

□ Web Forms ☑ MVC □ Web API

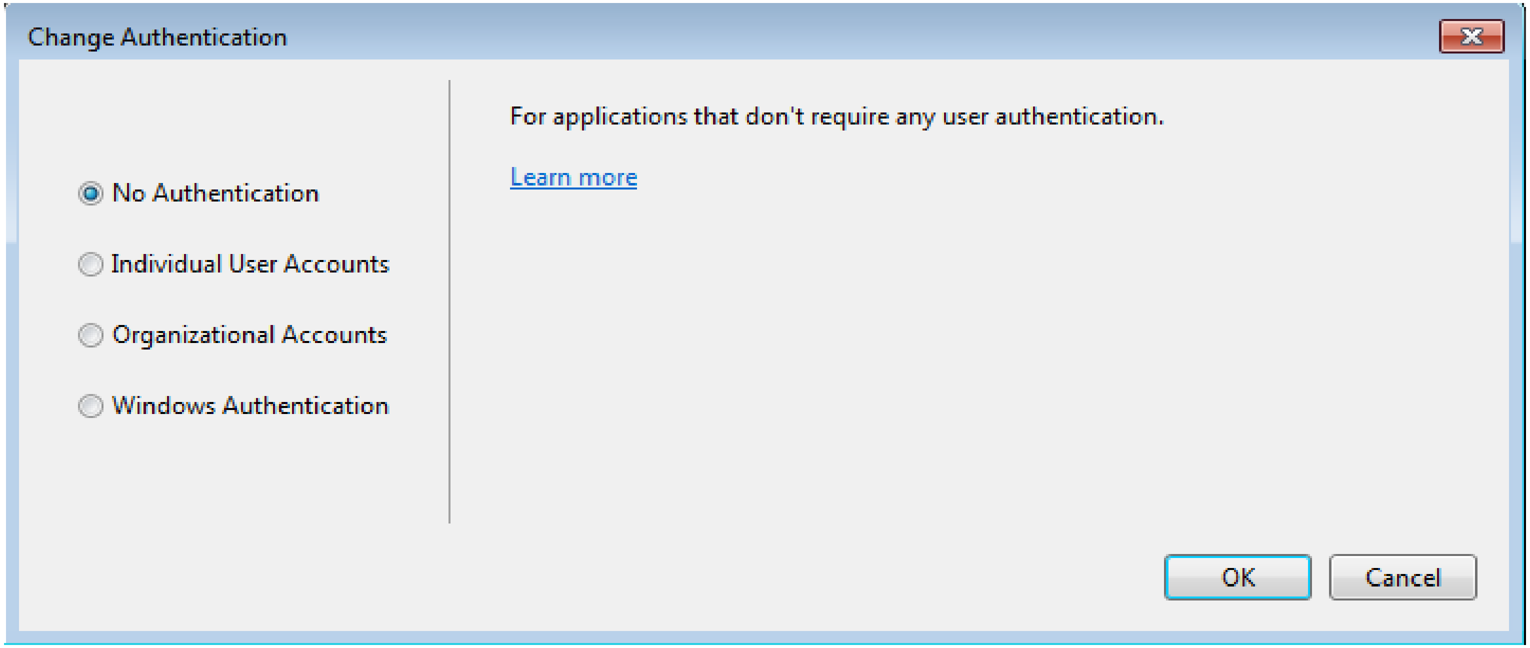
□ Add unit tests

Test project name: IdentityDemo.Tests

Authentication: **Individual User Accounts**

No Authentication

- Allows anonymous users
- Doesn’t identify or authenticate users



Change Authentication

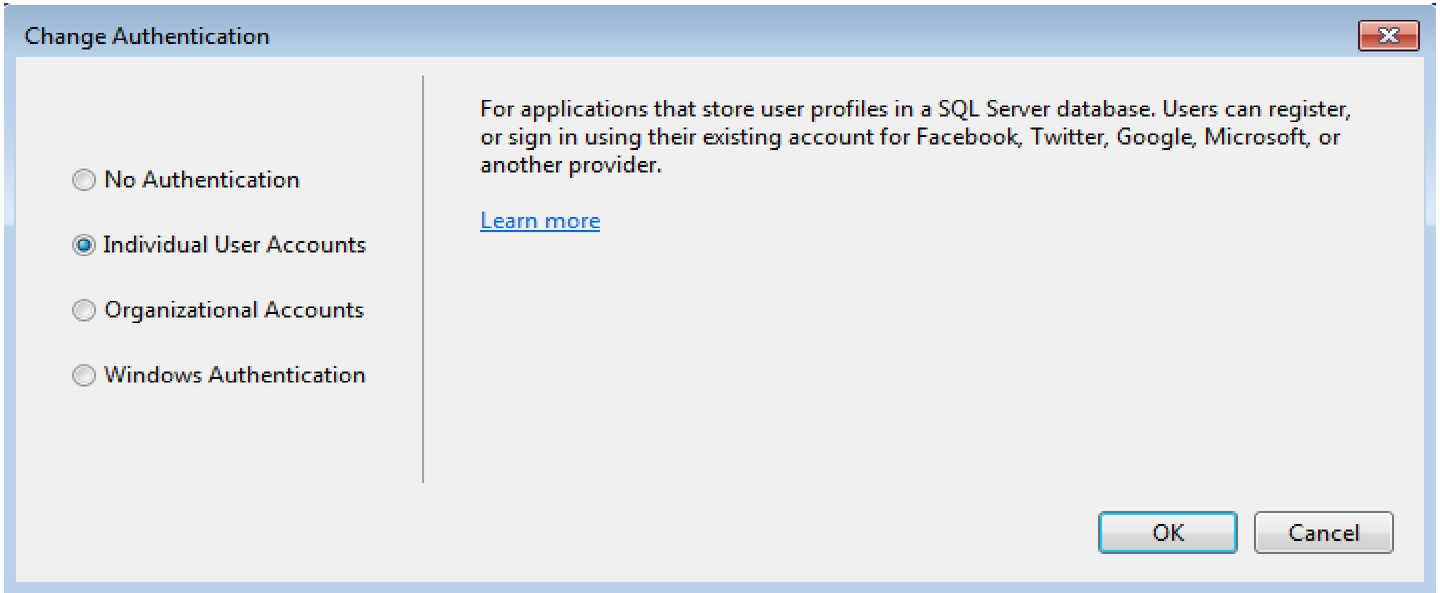
● No Authentication
○ Individual User Accounts
○ Organizational Accounts
○ Windows Authentication

For applications that don't require any user authentication.

Learn more

Individual User Accounts

- Traditional form based authentication
- Uses SQL Server database
- Can implement third party authentication



Change Authentication

○ No Authentication

● Individual User Accounts

○ Organizational Accounts

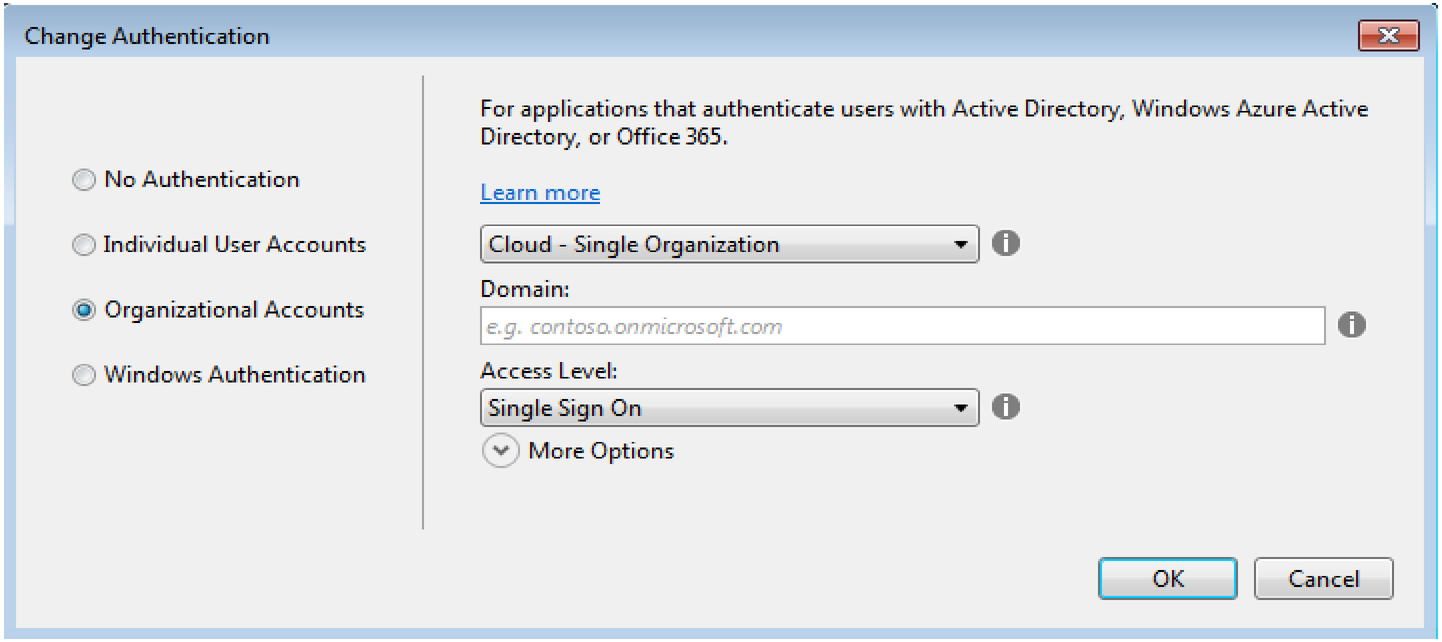
○ Windows Authentication

For applications that store user profiles in a SQL Server database. Users can register, or sign in using their existing account for Facebook, Twitter, Google, Microsoft, or another provider.

Learn more

Organizational Accounts

- Uses Active Directory Authentication with more options
- Single Sign On for Internal and Cloud App



Change Authentication

○ No Authentication
○ Individual User Accounts
● Organizational Accounts
○ Windows Authentication

For applications that authenticate users with Active Directory, Windows Azure Active Directory, or Office 365.

Learn more

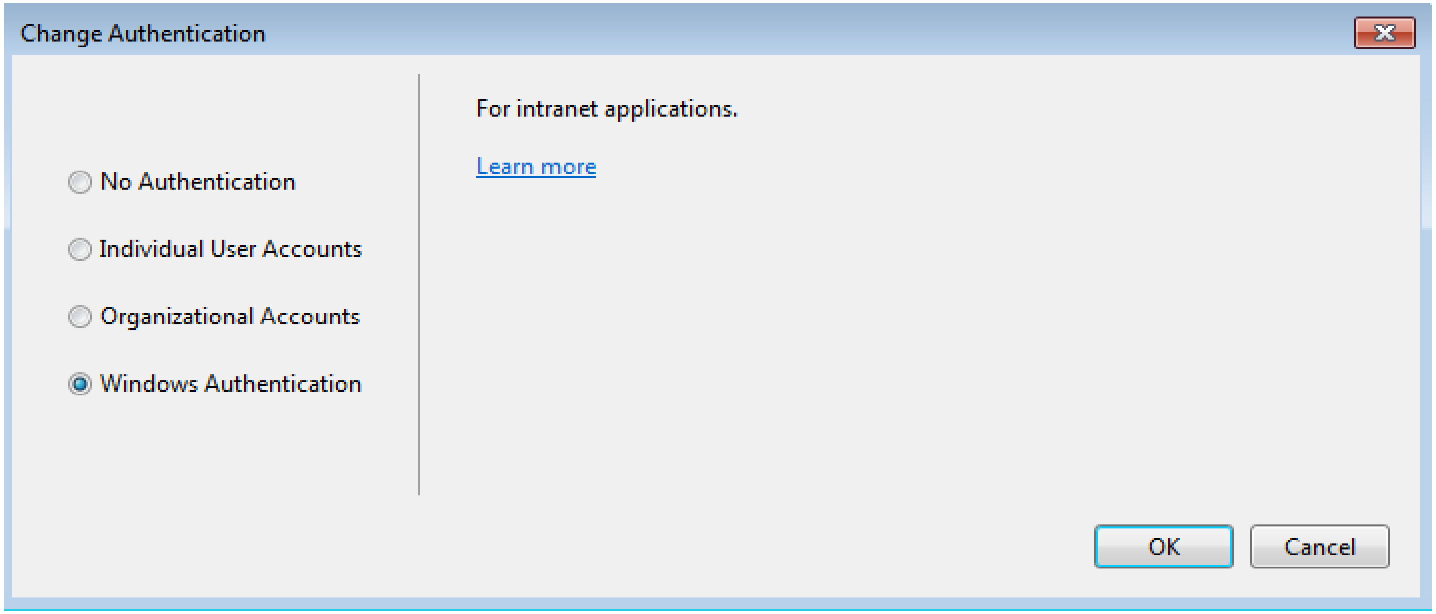
Cloud - Single Organization

Domain: e.g. contoso.onmicrosoft.com

Access Level: Single Sign On

Windows Authentication

- Suitable for Intranet based applications
- Doesn't allow anonymous users
- Uses Active Directory



Change Authentication

○ No Authentication

○ Individual User Accounts

○ Organizational Accounts

● Windows Authentication

For intranet applications.

Learn more

Authorize Attribute

- Where to Apply
  - Apply on Controller level
  - Apply on Action level
- What it does
  - Authenticates Users
  - Implements Roles Based Authentication
- Allow Anonymous Users

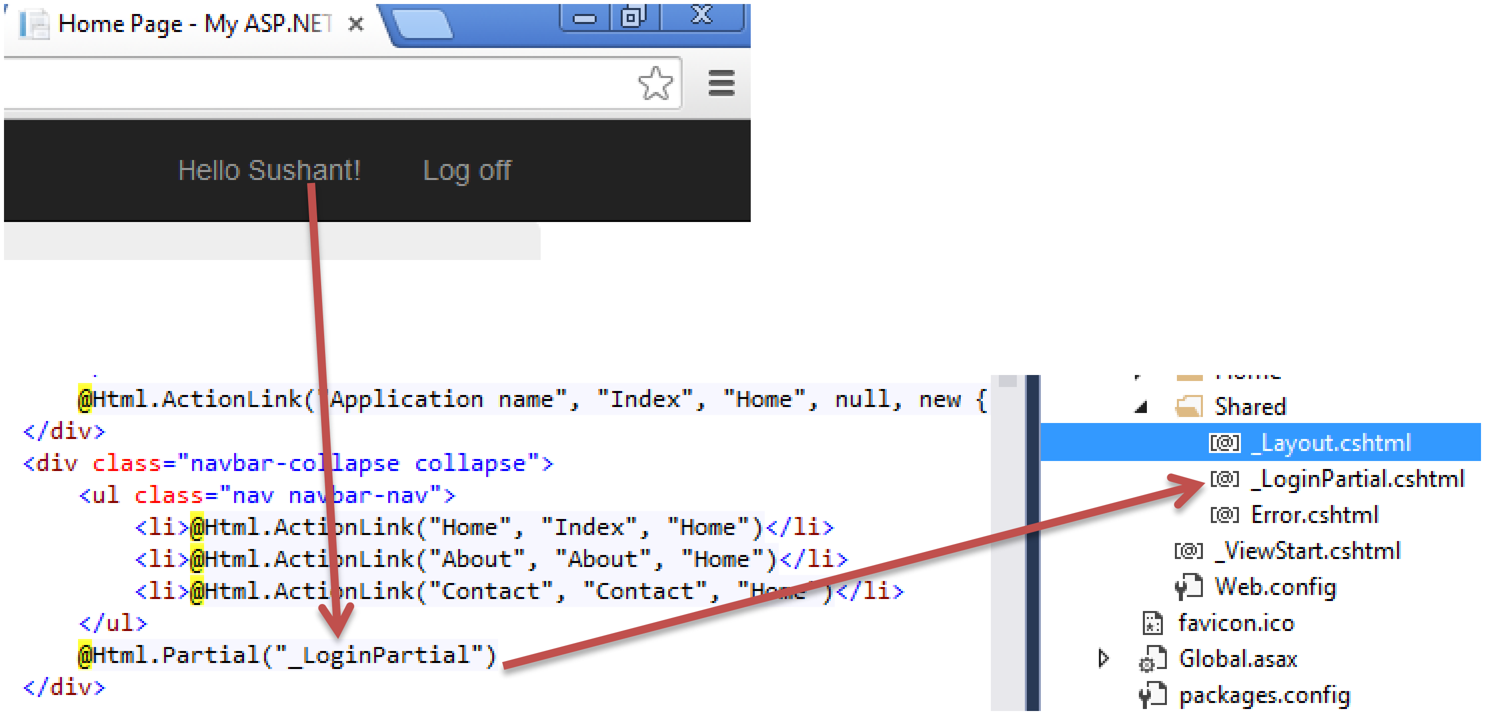
Where is the Data?

- Database created automatically on registration
- Stores information in multiple tables
- By default uses Default Connection
- Uses Entity Framework to build Models

Assemblies

- using Microsoft.AspNet.Identity;
- using Microsoft.AspNet.Identity.EntityFramework;

Register and Identify Users



```
    @Html.ActionLink("Application name", "Index", "Home", null, new {
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
    </ul>
    @Html.Partial("_LoginPartial")
</div>
```

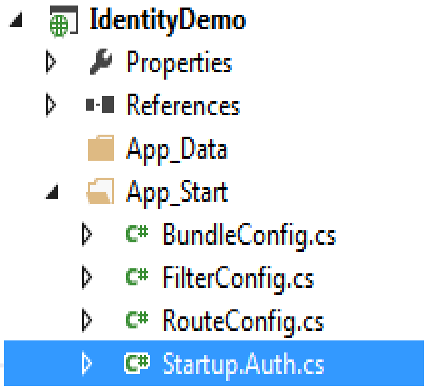
Common Attacks

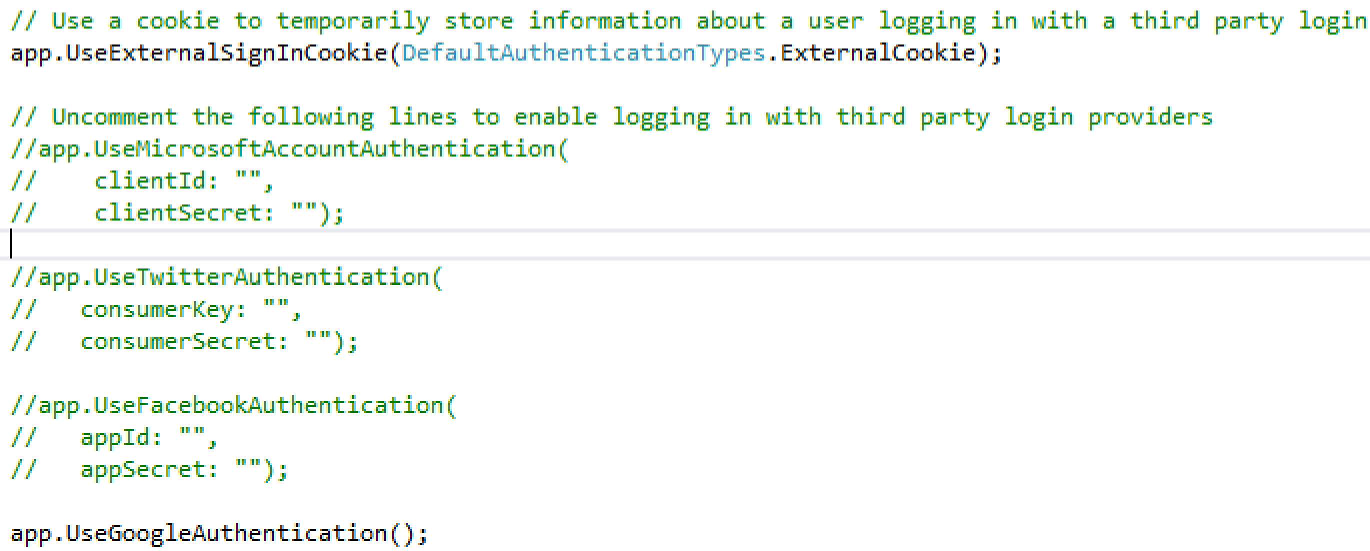
- XSS – Cross-site Scripting
    - Use HTML Encoding for all kind of inputs
- CSRF / XSRF – Cross-site Request Forgery
    - Use ValidateAntiForgeryToken Attribute in Action Methods
    - Use @Html.AntiForgeryToken() method in Views

External Authentication

- Microsoft
- Twitter
- Facebook
- Google

Google Authentication

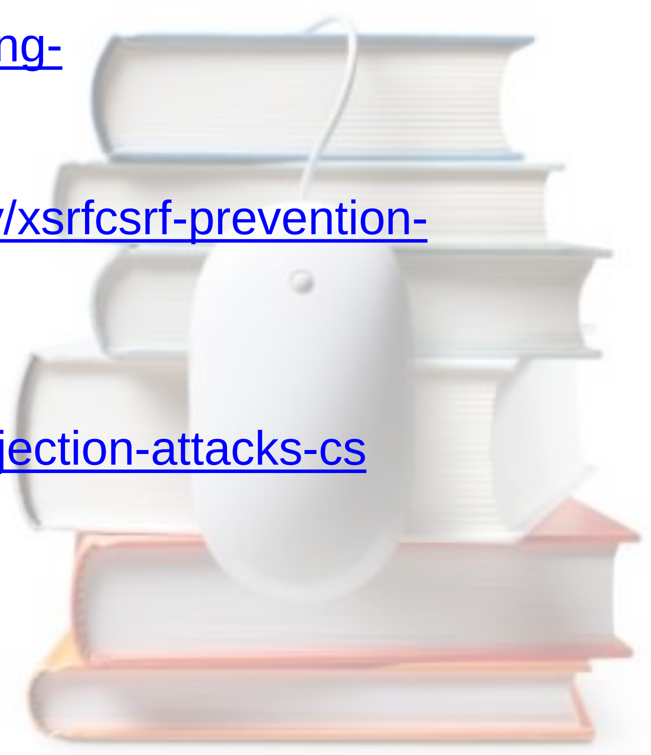




```
// Use a cookie to temporarily store information about a user logging in with a third party login
app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

// Uncomment the following lines to enable logging in with third party login providers
//app.UseMicrosoftAccountAuthentication(
//    clientId: "",
//    clientSecret: "");

//app.UseTwitterAuthentication(
//   consumerKey: "",
//   consumerSecret: "");

//app.UseFacebookAuthentication(
//   appId: "",
//   appSecret: "");

app.UseGoogleAuthentication();
```

Bibliography, Important Links

- http://www.asp.net/identity
- http://www.asp.net/identity/overview/getting-started/introduction-to-aspnet-identity
- http://www.asp.net/mvc/overview/security/xsrfcsrf-prevention-in-aspnet-mvc-and-web-pages
- http://www.asp.net/mvc/tutorials/older-versions/security/preventing-javascript-injection-attacks-cs



Any Questions?

Thank you!