Episerver

Setting Up Episerver Sites for Training

January 2018

Document version: 18.01

Episerver CMS Visual Studio Extension version: 11.1.0.326

Episerver CMS version: 11.3.1
Episerver Commerce version: 11.6.1

Contents

Overview 2
- Authentication with Windows or SQL-stored accounts 2
- Minimum development system requirements 2

Setting up your development environment 3
- Installing Microsoft Visual Studio Community 2017 3
- Installing the Episerver CMS Visual Studio Extension 3
- Configuring the Episerver NuGets package source 4

Creating an Alloy (MVC) Episerver web site with sample content 5
- Updating the Episerver NuGet packages 6
- Updating the Episerver database 6
- Testing the Episerver website and registering an administrator account 7
- Windows 7 users 9
- Congratulations! 9

Creating an Empty Episerver web site 10
- Updating the Episerver NuGet packages and updating the Episerver database 10
- Testing the Episerver web site 10
- Congratulations! 11

Creating or resetting a SQL-stored admin account automatically 12

Installing Additional Tools 14
- Installing Microsoft SQL Server Management Studio 14
- Installing the Azure SDK for Visual Studio 2015 or earlier 14

Creating Episerver Commerce websites 15
- Creating the front-end CommerceTraining website 15
- Creating the back-end CommerceManager website 16
- Update the packages and database schema for CommerceTraining 18
- Migrate the CommerceTraining website 19
- Test the Commerce websites 20
- Congratulations! 21

Creating an Episerver Find demo index 22
- Registering a Find account 22
- Creating a developer index 23
- Congratulations! 24

Overview

Using the instructions in this document, you will ensure that your computer is ready for use during an Episerver training course by setting up some Episerver web sites *before* you attend a course.

The projects have different names so that they can all be created in the same Visual Studio solution. We recommend a location of **C:\Episerver** and a solution name of **Training**, but you can choose alternatives, for example, your organization's name.

Before a course	Project name	Project template
Create these sites *before* a training course to ensure your laptop is set up correctly to work with Episerver. Detailed instructions are in this document.	AlloyDemo	Episerver Web Site – Alloy (MVC)
	EmptyDemo	Episerver Web Site – Empty

During courses, you will create additional projects as shown in the following table:

Course		Project name	Project template
CMS Developer Boot Camp	**CMS** Development Fundamentals	AlloyTraining	Episerver Web Site – Empty
	CMS Advanced Development	AlloyAdvanced	Episerver Web Site – Alloy (MVC)
		FindConsole	Console App (.NET Framework)
Commerce Development Fundamentals, **Commerce** Advanced Development, **Commerce** Developer Boot Camp		CommerceTraining	Episerver Web Site – Empty
		CommerceManager	ASP.NET Web Application (.NET Framework) – Empty
Developing for **DXC Service**		AlloyDxc	Episerver Web Site – Alloy (MVC)
Find for Developers		AlloyFind	Episerver Web Site – Alloy (MVC)
		FindConsole	Console App (.NET Framework)

Authentication with Windows or SQL-stored accounts

When you create an *Episerver Web Site – Empty* project, it is configured to allow authentication using either a Windows user account or a SQL-stored user account. However, there won't be any SQL-stored accounts created yet.

If your organization has a security set up that requires you to be logged into your work network to authenticate to the domain, then you will need to create a *local* Windows account and add it to the *local* **Administrators** group on your laptop for use when you are disconnected from your organisation's network.

Alternatively, there are detailed instructions showing how to automatically create or reset a SQL-stored Admin account that is a member of WebAdmins by writing an initialization module.

Minimum development system requirements

http://world.episerver.com/documentation/Items/System-Requirements/System-Requirements---EPiServer/

- Microsoft Windows 8, or later, or Windows Server 2012, or later.
- Microsoft Visual Studio 2015 or 2017 with latest updates.

These instructions will use our most recent recommended development stack: Microsoft Windows 10, Microsoft Visual Studio 2017 including SQL Server LocalDb, and Episerver CMS Visual Studio Extension 11.1.0.326. Older versions may look and behave slightly differently.

Setting up your development environment

Installing Microsoft Visual Studio Community 2017

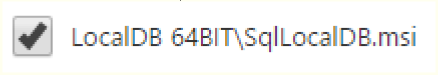
If you have already installed **Microsoft Visual Studio 2017**, or you are using a virtual machine provided by Episerver, then you can skip this section.

1. Download and install **Microsoft Visual Studio Community 2017** from the following link:

   https://www.visualstudio.com/downloads/

2. At a minimum, choose the workloads: **ASP.NET and web development**, **Azure development**, and **.NET Core cross-platform development**.
3. Optionally, add the individual components: **Azure Storage AzCopy**, **Class Designer**, **Git for Windows**, **GitHub extension for Visual Studio**, and **PowerShell tools**.

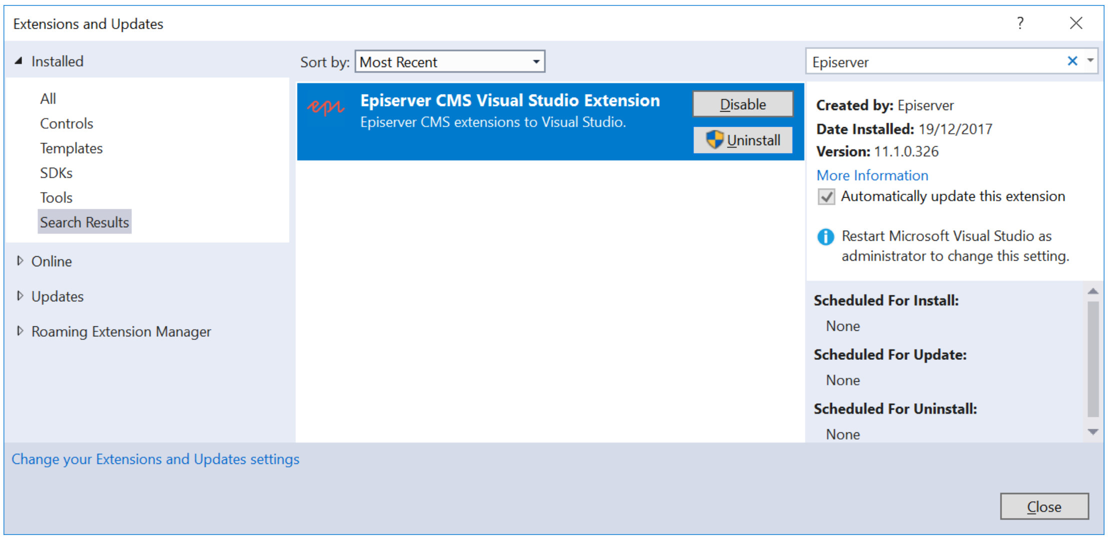
Microsoft SQL Server Express LocalDb should be included with Visual Studio, but you can also install it separately from the following link: https://www.microsoft.com/en-gb/download/details.aspx?id=42299
Click **Download**, check the box for **LocalDB 64BIT\SqlLocalDB.msi**, and click **Next**.


✓ LocalDB 64BIT\SqlLocalDB.msi

Installing the Episerver CMS Visual Studio Extension

If you have already installed the **Episerver CMS Visual Studio Extension**, or you are using a virtual machine provided by Episerver, then you can skip this section.

1. Start Microsoft Visual Studio.
2. Navigate to **Tools** | **Extensions and Updates...**, and in the section on the left, click **Online**, to show the **Visual Studio Marketplace**.
3. Press *Ctrl* + *E*, or click in the **Search** box, and then enter **Episerver**.
4. Select the **Episerver CMS Visual Studio Extension**, click **Download**, as shown in the following screenshot:


Extensions and Updates

Installed: All, Controls, Templates, SDKs, Tools, Search Results
Online
Updates
Roaming Extension Manager

Sort by: Most Recent

Episerver

Episerver CMS Visual Studio Extension
Episerver CMS extensions to Visual Studio.
Disable
Uninstall

Created by: Episerver
Date Installed: 19/12/2017
Version: 11.1.0.326
More Information
✓ Automatically update this extension
Restart Microsoft Visual Studio as administrator to change this setting.

Scheduled For Install:
None
Scheduled For Update:
None
Scheduled For Uninstall:
None

Change your Extensions and Updates settings

Close

Episerver CMS Visual Studio Extension 11.1.0.326 was released on 18th December 2017. We recommend that you automatically update this extension. To install an older version, download from the following link:
https://marketplace.visualstudio.com/items?itemName=EPiServer.EpiserverCMSVisualStudioExtension

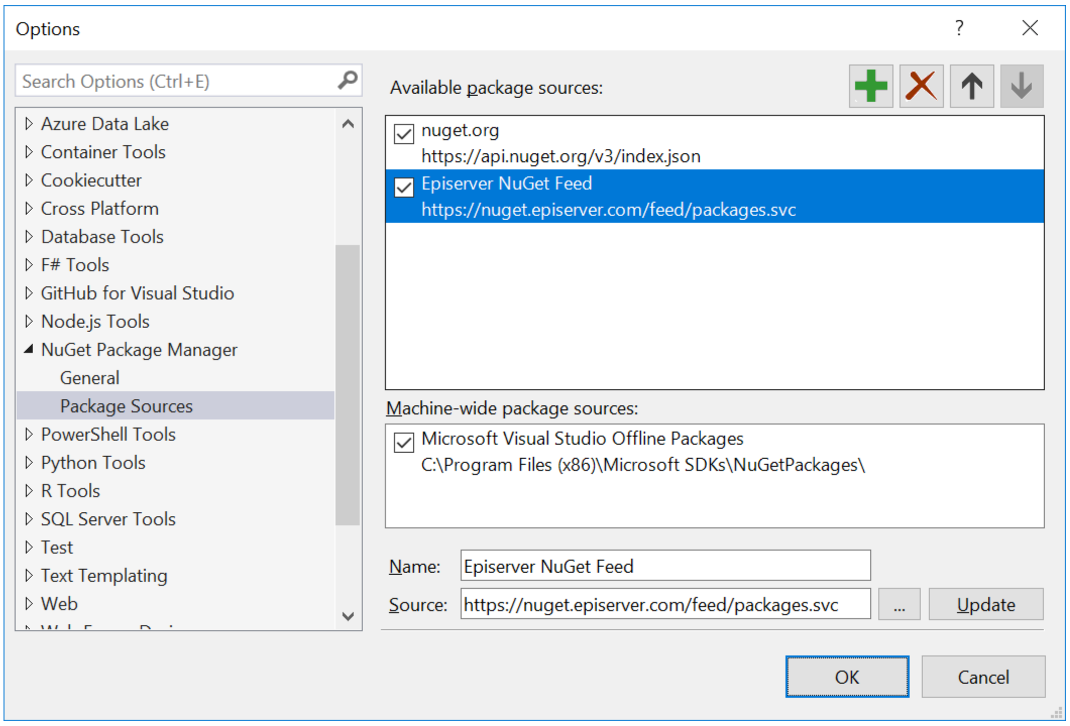
5. Wait for the extension to download, and then click **Close**.
6. Close Visual Studio, and wait for the **VSIX Installer** to start.
7. When the installer has finished, click **Close**.

Configuring the Episerver NuGets package source

If you have already configured the **Episerver NuGets** package source, or you are using a virtual machine provided by Episerver, then you can skip this section.

1. Start Microsoft Visual Studio.
2. Navigate to **Tools** | **NuGet Package Manager** | **Package Manager Settings**.
3. In the **Options** dialog, in the list on the left, click **Package Sources**.
4. If the Episerver NuGet feed doesn't exist as an available package source, as shown in the following screenshot, then click the **green plus** button to add it. The name can be anything, although we recommend using **Episerver NuGets**, and the path must be:

   ```
   https://nuget.episerver.com/feed/packages.svc/
   ```



Available package sources:

- ✓ nuget.org — https://api.nuget.org/v3/index.json
- ✓ Episerver NuGet Feed — https://nuget.episerver.com/feed/packages.svc

Machine-wide package sources:

- ✓ Microsoft Visual Studio Offline Packages — C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\

Name: Episerver NuGet Feed

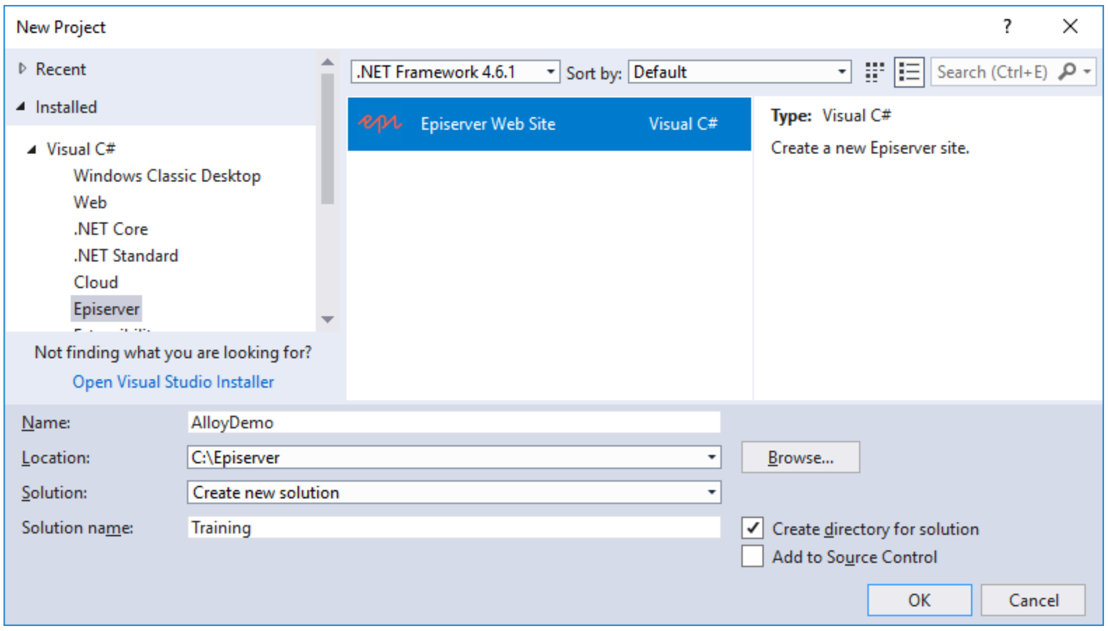
Source: https://nuget.episerver.com/feed/packages.svc

Creating an Alloy (MVC) Episerver web site with sample content

1. In Visual Studio, navigate to **File** | **New** | **Project...**, or press *Ctrl* + *Shift* + *N*.
2. In the left section, navigate to **Installed** | **Templates** | **Visual C#** | **Episerver**.
3. In the middle section, select **Episerver Web Site** project, and enter the following options, as shown in the following screenshot:
   - Target: **.NET Framework 4.6.1** (or a later compatible version).

For compatability with Episerver CMS 11 you must choose a minimum target of .NET Framework 4.6.1 because it requires .NET Standard 2.0.

   - Name: **AlloyDemo**
   - Location: **C:\Episerver** (or some other folder).
   - Solution name: **Training** (or something else unique).
   - Create directory for solution: **Yes**

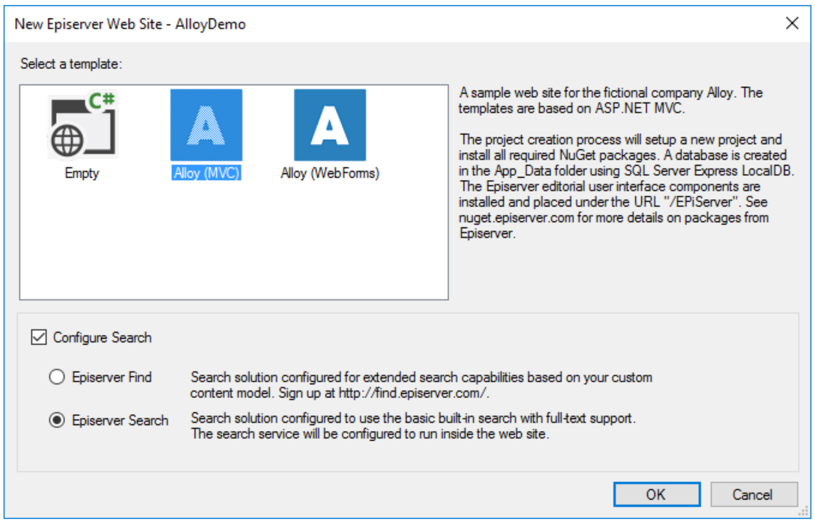


.NET Framework 4.6.1
Sort by: Default
Type: Visual C#
Create a new Episerver site.
Name: AlloyDemo
Location: C:\Episerver
Solution: Create new solution
Solution name: Training
✓ Create directory for solution
☐ Add to Source Control

If you already have a solution named **Training**, you can open the existing solution and choose **Add New Project** instead.

4. Click **OK**.

5. Choose the **Alloy (MVC)** template, and in the **Configure Search** section, select **Episerver Search**, as shown in the following screenshot:



New Episerver Web Site - AlloyDemo

Select a template:

A sample web site for the fictional company Alloy. The templates are based on ASP.NET MVC.

The project creation process will setup a new project and install all required NuGet packages. A database is created in the App_Data folder using SQL Server Express LocalDB. The Episerver editorial user interface components are installed and placed under the URL "/EPiServer". See nuget.episerver.com for more details on packages from Episerver.

[✓] Configure Search

○ Episerver Find — Search solution configured for extended search capabilities based on your custom content model. Sign up at http://find.episerver.com/.

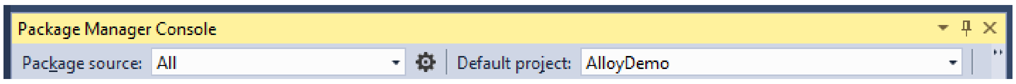
● Episerver Search — Search solution configured to use the basic built-in search with full-text support. The search service will be configured to run inside the web site.

6. Click **OK**, and wait for the project to be created.

Updating the Episerver NuGet packages

Episerver CMS Visual Studio Extension 11.1.0.326 uses NuGet packages from 18th December 2017, i.e. EPiServer.CMS.Core 11.3.0. Recommended practice is to update to the latest NuGet packages for your chosen major version number that are usually released on a weekly schedule.

1. In Visual Studio, navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.
2. In **Package Manager Console**, set these two options, as shown in the following screenshot:
   - Package source: **All**
   - Default project: **AlloyDemo**



Package Manager Console

Package source: All

Default project: AlloyDemo

If you cannot see **All** as a **Package source**, then close Visual Studio, use **File Explorer** to open **%appdata%\NuGet**, rename **NuGet.config** to **NuGet.backup**, and restart Visual Studio. That should recreate **NuGet.config**, including the **All** option. You can redefine the **Episerver NuGets** feed following the instructions above, or copy and paste previous configuration from the backup, if necessary.

3. Enter the following command to update all packages referenced by the project **AlloyDemo** using restrictions defined in packages.config:

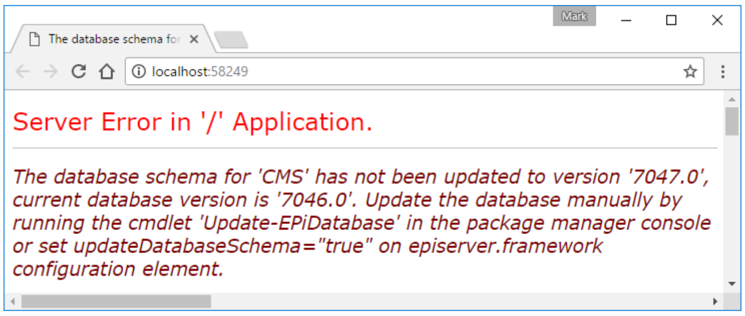
```
Update-Package -ProjectName AlloyDemo -ToHighestMinor
```

Doing updates at the command line allows Episerver packages to be safely installed because it won't upgrade to a higher major version that would have breaking changes, and non-Episerver dependencies will be updated, as well as the EPiServer ones, for example, **Microsoft.Tpl.DataFlow**, but won't accidently install newer but incompatible packages. If you don't have the latest version of NuGet, you can follow instructions here: https://docs.microsoft.com/en-us/nuget/guides/install-nuget

Updating the Episerver database

1. Start the site by navigating to **Debug** | **Start Without Debugging**, or press *Ctrl + F5*.

2. If you see an exception message about the **EPiServerDB** database, as shown in the following screenshot, then a change has been made to the database schema:



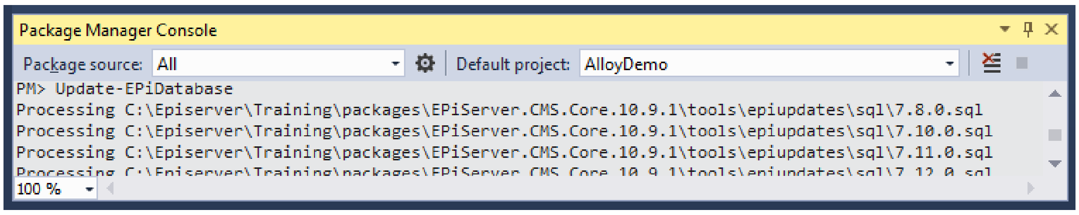
Server Error in '/' Application.

The database schema for 'CMS' has not been updated to version '7047.0', current database version is '7046.0'. Update the database manually by running the cmdlet 'Update-EPiDatabase' in the package manager console or set updateDatabaseSchema="true" on episerver.framework configuration element.

There are two ways to update the Episerver CMS database schema: manually with a console command, or automatically, with a Web.config setting, as explained in the error message. Recommended practice for deployments that you have control over is to perform migrations manually, especially if you have written custom SQL scripts to manipulate the CMS database schema, for example, to improve DDS performance. If you are deploying to DXC Service, we recommend using the Web.config setting.

3. Close the browser.
4. In Visual Studio, navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.
5. Make sure the **Default project** is **AlloyDemo**, as shown in the following screenshot, and at the prompt enter:

```
Update-EPiDatabase
```



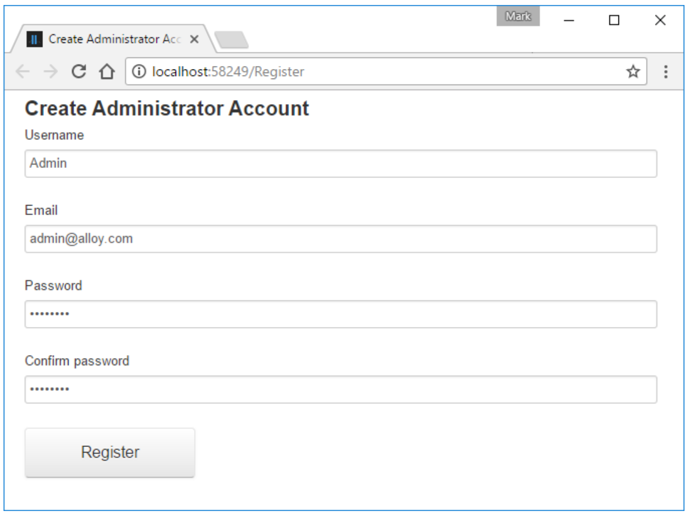
```
PM> Update-EPiDatabase
Processing C:\Episerver\Training\packages\EPiServer.CMS.Core.10.9.1\tools\epiupdates\sql\7.8.0.sql
Processing C:\Episerver\Training\packages\EPiServer.CMS.Core.10.9.1\tools\epiupdates\sql\7.10.0.sql
Processing C:\Episerver\Training\packages\EPiServer.CMS.Core.10.9.1\tools\epiupdates\sql\7.11.0.sql
Processing C:\Episerver\Training\packages\EPiServer.CMS.Core.10.9.1\tools\epiupdates\sql\7.12.0.sql
```

If you get a warning about a missing **packages** folder, exit, and restart Visual Studio, and try again.

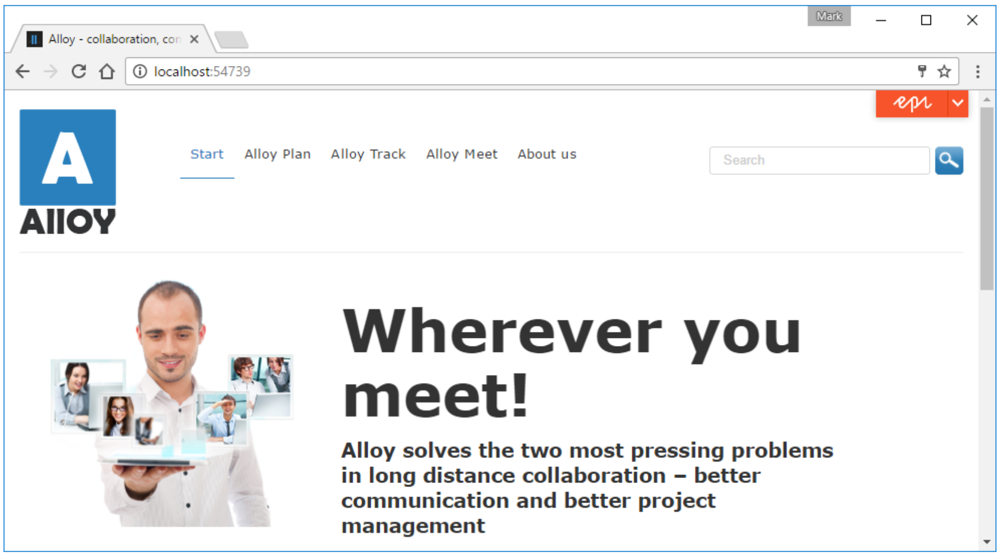
Testing the Episerver website and registering an administrator account

1. Start the site by navigating to **Debug** | **Start Without Debugging**, or press *Ctrl + F5*.
2. You will be prompted to **Create Administrator Account**, as shown in the following screenshot, so enter the following details:

- Username: **Admin** or something else, but make a note of it!
- Email: **admin@alloy.com** or some other e-mail address (it does not need to be real).
- Password: **Pa$$w0rd** or something else, but make a note of it!



3. You should now see the **Alloy** sample site's **Start** page, as shown in the following screenshot:



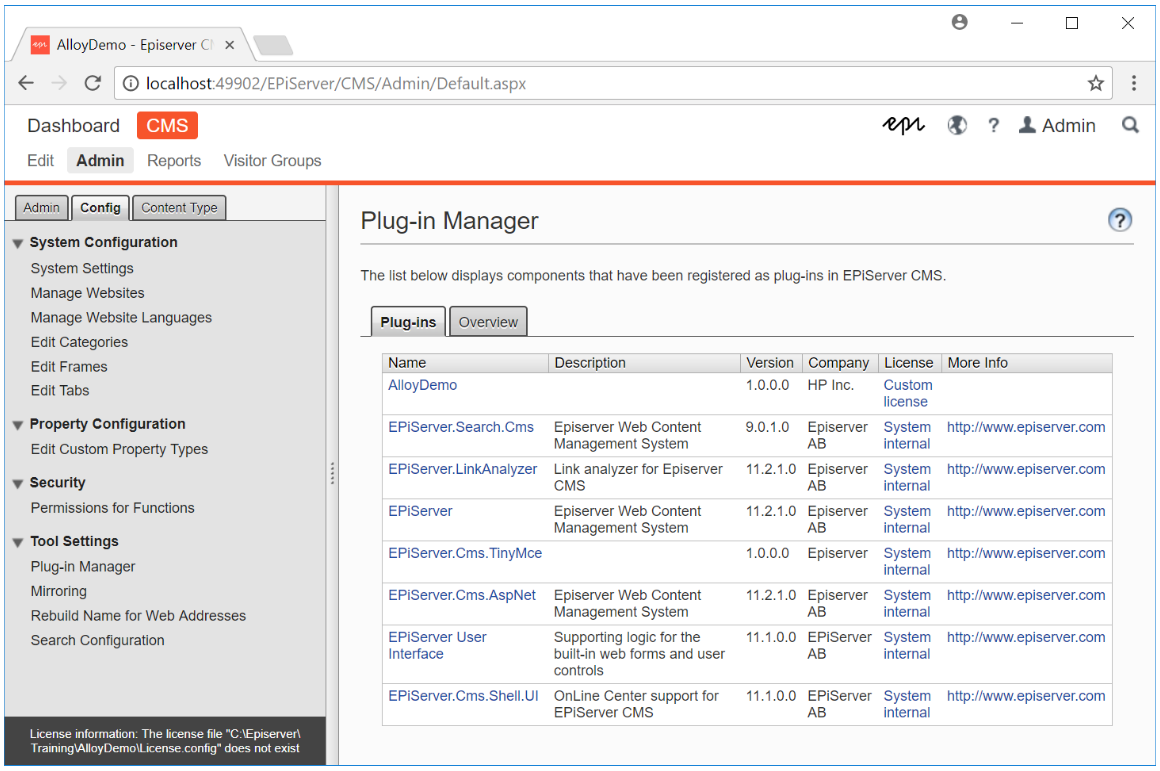
Wherever you meet!

Alloy solves the two most pressing problems in long distance collaboration – better communication and better project management

4. Click the **epi** menu in the top-right corner, as shown in the following screenshot:

If you can't see the **epi** quick access menu, scroll down to the bottom of the page, and in the footer, click **Log In**, and log in as Admin.

5. At the top of the page, pull down the orange **Global** menu, and navigate to **CMS** | **Admin** | **Config** | **Plug-in Manager** to confirm the version of Episerver CMS you are using, as shown in the following screenshot:



License information: The license file "C:\Episerver\Training\AlloyDemo\License.config" does not exist

Plug-in Manager

The list below displays components that have been registered as plug-ins in EPiServer CMS.

Name	Description	Version	Company	License	More Info
AlloyDemo		1.0.0.0	HP Inc.	Custom license	
EPiServer.Search.Cms	Episerver Web Content Management System	9.0.1.0	Episerver AB	System internal	http://www.episerver.com
EPiServer.LinkAnalyzer	Link analyzer for Episerver CMS	11.2.1.0	Episerver AB	System internal	http://www.episerver.com
EPiServer	Episerver Web Content Management System	11.2.1.0	Episerver AB	System internal	http://www.episerver.com
EPiServer.Cms.TinyMce		1.0.0.0	Episerver	System internal	http://www.episerver.com
EPiServer.Cms.AspNet	Episerver Web Content Management System	11.2.1.0	Episerver AB	System internal	http://www.episerver.com
EPiServer User Interface	Supporting logic for the built-in web forms and user controls	11.1.0.0	EPiServer AB	System internal	http://www.episerver.com
EPiServer.Cms.Shell.UI	OnLine Center support for EPiServer CMS	11.1.0.0	EPiServer AB	System internal	http://www.episerver.com

Congratulations! You have now created an Episerver CMS website and updated it to the latest version.

Windows 7 users

If you are using Windows 7, or your operating system has web sockets disabled, then in Edit view you will see warning messages about no support for real-time communication. To disable these warning messages, add the following element to Web.config:

```
<appSettings>
  <add key="Epi.WebSockets.Enabled" value="false"/>
```

Congratulations!

You are now ready to explore developing for the **Episerver CMS** platform.

You do not need to complete any more of the tasks in this document. The following tasks are optional.

Creating an Empty Episerver web site

1. In Visual Studio, navigate to **File** | **Add** | **New Project...**
2. In the left section, navigate to **Installed** | **Templates** | **Visual C#** | **Episerver**. In the middle section, select **Episerver Web Site** project, and enter the following options, as shown in the following screenshot:
   - Target: **.NET Framework 4.6.1** or later compatible version.
   - Name: **EmptyDemo**
   - Location: **C:\Episerver\Training**
3. Click **OK**.
4. Choose the **Empty** template, and click **OK**.

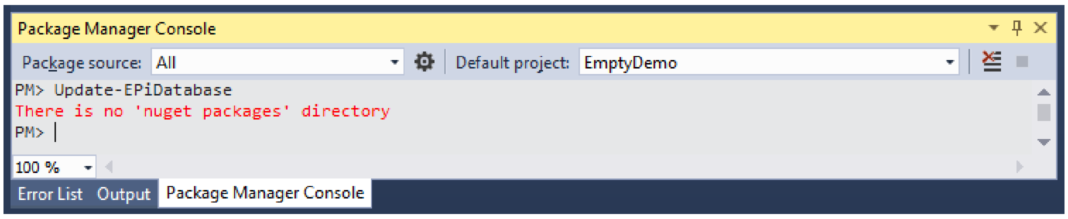
Updating the Episerver NuGet packages and updating the Episerver database

1. In Visual Studio, navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.
2. At the top of the **Package Manager Console**, set **Package source** to **All**.
3. Enter the following commands to install Episerver Search indexing service and its integration with CMS, install Episerver Forms, update all packages using restrictions defined in packages.config, and update the CMS database:

```
Install-Package -ProjectName EmptyDemo EPiServer.Search
Install-Package -ProjectName EmptyDemo EPiServer.Search.Cms
Install-Package -ProjectName EmptyDemo EPiServer.Forms
Update-Package -ProjectName EmptyDemo -ToHighestMinor
Update-EPiDatabase
```

It is worth installing **Episerver Search** now so that you can use the Global search and search boxes in the **Navigation** and **Assets** panes as an editor and administrator.

4. If you see an error message, as shown in the following screenshot, then close Visual Studio, restart, reopen the solution, and try again:

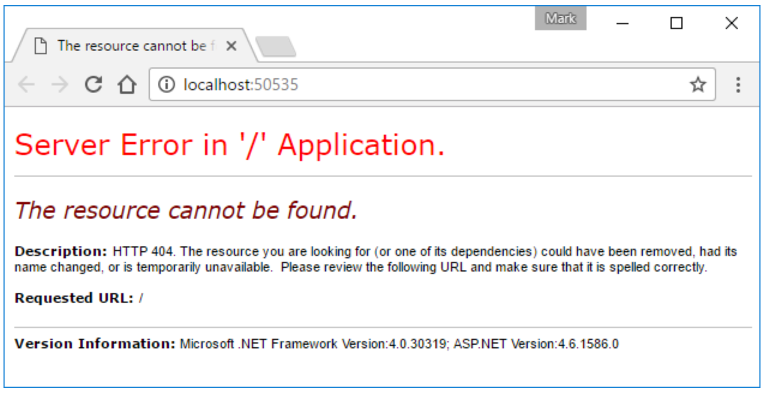


```
PM> Update-EPiDatabase
There is no 'nuget packages' directory
PM> |
```

Testing the Episerver web site

1. Start the site by navigating to **Debug** | **Start Without Debugging**, or press *Ctrl* + *F5*.
2. You will see a **HTTP 404** error message, as shown in the following screenshot:

This is expected, because the site is empty, and does not yet have a start page.



Server Error in '/' Application.

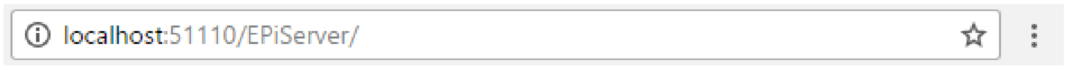
The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.6.1586.0

3. Enter **/EPiServer/** at the end of the address box, as shown in the following screenshot:



The **Episerver Web Site - Empty** project template configures ASP.NET Membership providers to allow a member of the Windows group named Administrators to be recognized as an administrator of the CMS.

4. Log in using a *local* Windows account that is a member of the *local* Windows group named **Administrators**, or follow the instructions in the next section to automatically create or reset a SQL-stored account.

Congratulations!

You are now ready to explore developing for the **Episerver CMS** platform.

Creating or resetting a SQL-stored admin account automatically

WARNING! The password of a SQL-stored account cannot be reset, so this code will **delete** an existing SQL-stored account named **Admin** (if it already exists) and recreate it with a known password. It will create the **WebAdmins** group (if does not already exist) and assign full access rights for the role to the Root page.

1. In an **Episerver Web Site – Empty** project with default configuration, open **Web.config**.
2. Add an entry to `<appSettings>`, as shown in the following markup:

   ```
   <add key="alloy:RegisterAdmin" value="true" />
   ```

3. Find the `<membership>` element and set the **defaultProvider** to **SqlServerMembershipProvider**, as shown in the following markup:

   ```
   <membership defaultProvider="SqlServerMembershipProvider"
   ```

4. Find the `<roleManager>` element and set the **defaultProvider** to **SqlServerRoleProvider**, as shown in the following markup:

   ```
   <roleManager enabled="true" defaultProvider="SqlServerRoleProvider"
   ```

5. Save and close **Web.config**.
6. Right-click **Business**, and add a folder named **Initialization**.
7. In **~\Business\Initialization**, add an **Episerver** | **Initialization Module** named **RegisterAdminInitializationModule.cs**.
8. Modify the statements, as shown in the following code:

```
using EPiServer;
using EPiServer.Core;
using EPiServer.Framework;
using EPiServer.Framework.Initialization;
using EPiServer.Security;
using EPiServer.ServiceLocation;
using System.Linq;
using System.Web.Security;

namespace EmptyDemo.Business.Initialization
{
    [InitializableModule]
    [ModuleDependency(typeof(EPiServer.Web.InitializationModule))]
    public class RegisterAdminInitializationModule : IInitializableModule
    {
        private const string roleName = "WebAdmins";
        private const string userName = "Admin";
        private const string password = "Pa$$w0rd";
        private const string email = "admin@alloy.com";

        public void Initialize(InitializationEngine context)
        {
            #region Use ASP.NET Membership classes to create the role and user

            // if the role does not exist, create it
            if (!Roles.RoleExists(roleName))
            {
                Roles.CreateRole(roleName);
            }

            // if the user already exists, delete it
            MembershipUser user = Membership.GetUser(userName);
            if (user != null)
```

```
            {
                Membership.DeleteUser(userName);
            }

            // create the user with password and add it to role
            user = Membership.CreateUser(userName, password, email);
            Roles.AddUserToRole(userName, roleName);

            #endregion

            #region Use EPiServer classes to give full access to root of content tree

            // get the root page
            var loader = ServiceLocator.Current.GetInstance<IContentLoader>();
            var root = loader.Get<PageData>(ContentReference.RootPage)
                .CreateWritableClone() as PageData;

            // try to get the access control entry for the role
            var ace = root.ACL.Entries.Where(
                e => e.Name == roleName).FirstOrDefault();

            // if the entry exists, remove it if it isn't full access
            if ((ace != null) && (ace.Access == AccessLevel.FullAccess))
            {
                return;
            }
            if (ace != null)
            {
                root.ACL.RemoveEntry(ace);
            }

            // create, add, and save a new access control entry
            ace = new AccessControlEntry(roleName, AccessLevel.FullAccess);
            root.ACL.Add(ace);
            root.ACL.Save();

            #endregion
        }

        public void Uninitialize(InitializationEngine context) { }
    }
}
```

9. Start the site, and log in as **Admin** with **Pa$$w0rd**.
10. Close the browser.
11. In **AlloyTraining**, open **Web.config**.
12. Modify the entry in `<appSettings>` to disable the registration of Admin, as shown in the following markup:

    ```
    <add key="alloy:RegisterAdmin" value="false" />
    ```

13. Save and close Web.config.

You now have full access to CMS administrator features without needing a Windows account.

Installing Additional Tools

Installing Microsoft SQL Server Management Studio

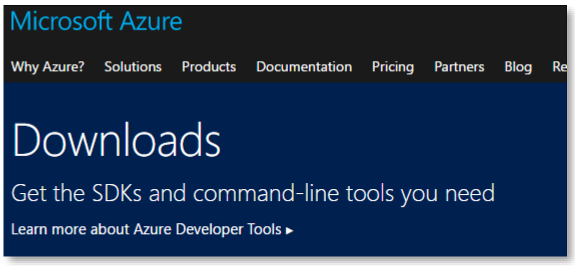
If you are using a virtual machine provided by Episerver, or you are not attending the *Developing for the DXC Service* course, then you can skip this section.

Go to the following link to download and install Microsoft SQL Server Management Studio: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms

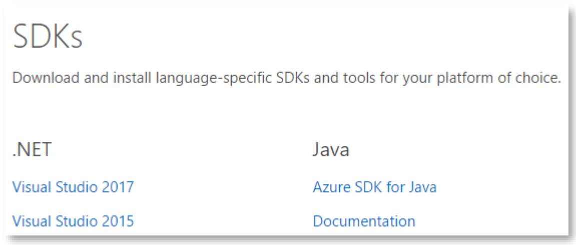
Installing the Azure SDK for Visual Studio 2015 or earlier

If you are using Visual Studio 2017, and you installed the **Azure development** workload, or you are using a virtual machine provided by Episerver, or you are not going to deploy to Microsoft Azure, or you are not attending the Developing for the DXC Service course, then you can skip this section.

1. Search for **Azure SDK** on Google to find the downloads page, as shown in the following screenshot:

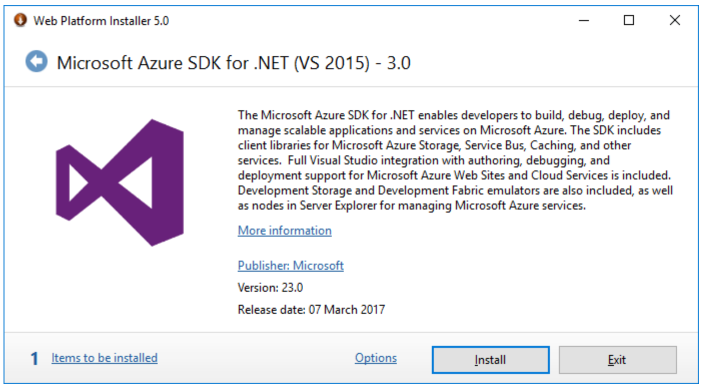

Microsoft Azure
Why Azure? Solutions Products Documentation Pricing Partners Blog Re
Downloads
Get the SDKs and command-line tools you need
Learn more about Azure Developer Tools ▸

2. Click the link for your version of Visual Studio, as shown in the following screenshot:


SDKs

Download and install language-specific SDKs and tools for your platform of choice.

3. In **Web Platform Installer**, click **Install**, as shown in the following screenshot, and wait for the SDK to install:


Web Platform Installer 5.0

Microsoft Azure SDK for .NET (VS 2015) - 3.0

The Microsoft Azure SDK for .NET enables developers to build, debug, deploy, and manage scalable applications and services on Microsoft Azure. The SDK includes client libraries for Microsoft Azure Storage, Service Bus, Caching, and other services. Full Visual Studio integration with authoring, debugging, and deployment support for Microsoft Azure Web Sites and Cloud Services is included. Development Storage and Development Fabric emulators are also included, as well as nodes in Server Explorer for managing Microsoft Azure services.

Publisher: Microsoft
Version: 23.0
Release date: 07 March 2017

Creating Episerver Commerce websites

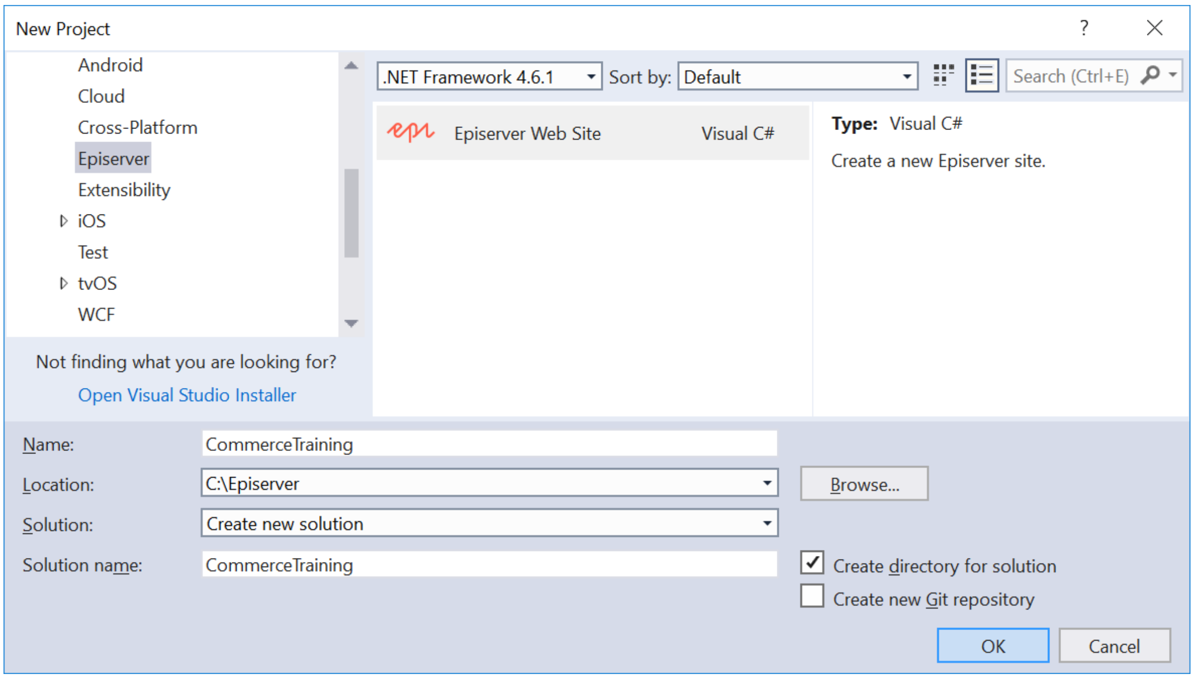
To use Episerver Commerce, you need two websites:

- Front-end website for visitors, editors, merchandisers, and administrators.
- Back-end website for Commerce administrators.

Creating the front-end CommerceTraining website

It is best to initially create the two projects in their own solution. Once they are set up, you can then safely add them to another solution, like the Training solution, if you want.

1. In Visual Studio, navigate to **File** | **New** | **Project...** or press *Ctrl* + *Shift* + *N*.
2. In the left section, navigate to **Installed** | **Visual C#** | **Episerver**.
3. In the middle section, select **Episerver Web Site** project.
4. Enter the following, as shown in the following screenshot:
   - Target: **.NET Framework 4.6.1** (or later compatible version).
   - Name: **CommerceTraining**
   - Location: **C:\Episerver**



.NET Framework 4.6.1 Sort by: Default

Episerver Web Site Visual C#

Type: Visual C#
Create a new Episerver site.

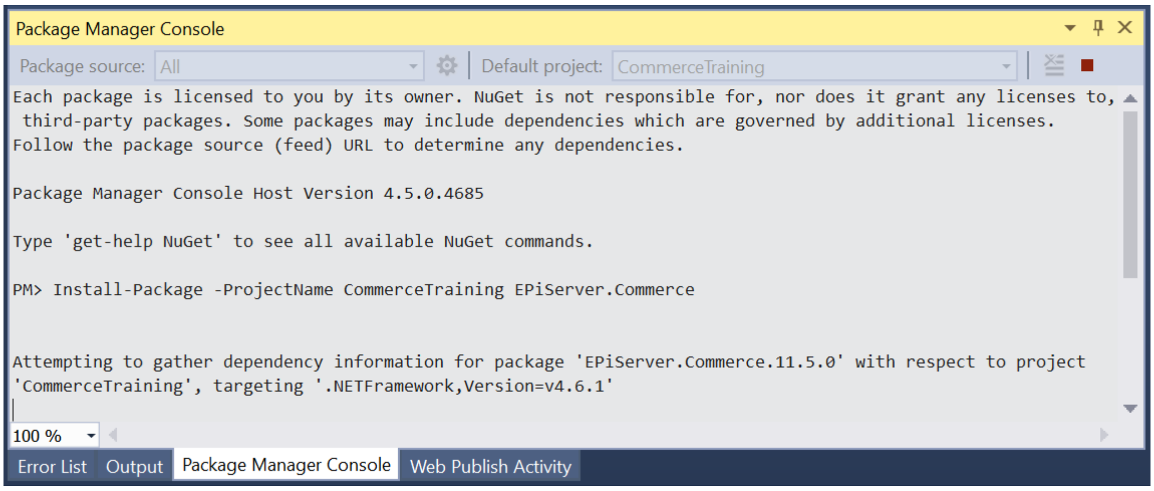
Name: CommerceTraining
Location: C:\Episerver
Solution: Create new solution
Solution name: CommerceTraining

[✓] Create directory for solution
[ ] Create new Git repository

5. Click **OK**.
6. Choose the **Empty** template, and click **OK**.
7. Navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.

8. At the top of the **Package Manager Console**, set:
   a. **Package source** to **All**
   b. **Default project** to **CommerceTraining**
9. Enter the following commands, as shown in the following screenshot, to:
   a. Install a package to add Commerce functionality to the front-end website.
   b. Install a package to integrate the front-end website with the back-end website.
   c. Update all packages using restrictions defined in packages.config.
   d. Update the database schema.

```
Install-Package -ProjectName CommerceTraining EPiServer.Commerce

Install-Package -ProjectName CommerceTraining
    EPiServer.Commerce.UI.ManagerIntegration

Update-Package -ProjectName CommerceTraining -ToHighestMinor

Update-EPiDatabase
```

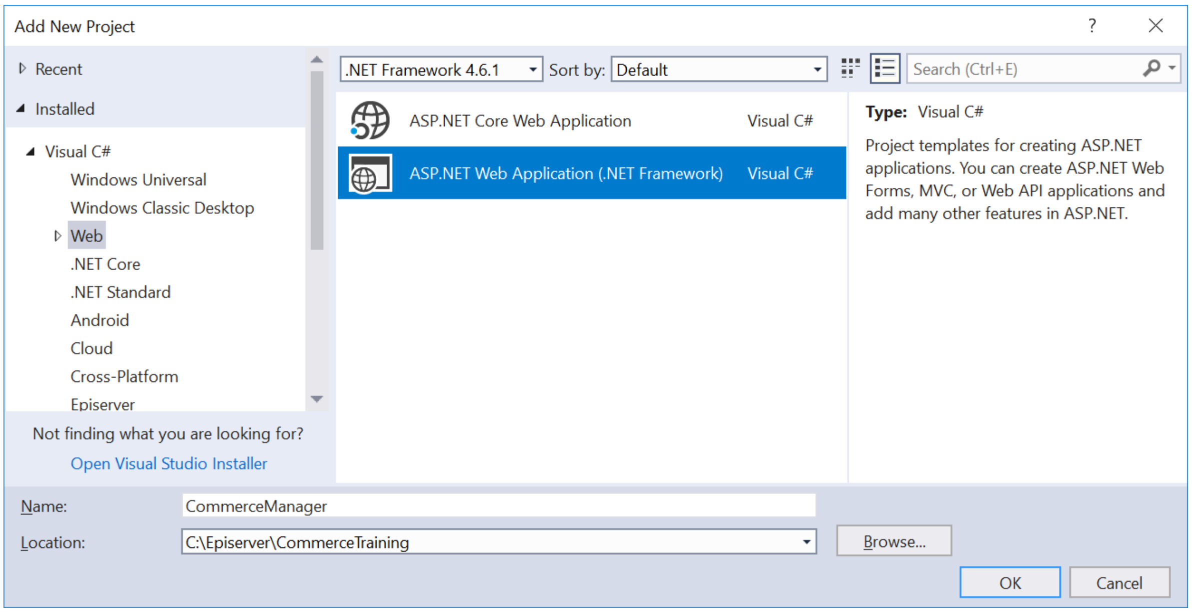


```
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to,
 third-party packages. Some packages may include dependencies which are governed by additional licenses.
Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 4.5.0.4685

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package -ProjectName CommerceTraining EPiServer.Commerce


Attempting to gather dependency information for package 'EPiServer.Commerce.11.5.0' with respect to project
'CommerceTraining', targeting '.NETFramework,Version=v4.6.1'
```

You might need to restart Visual Studio a couple of times to update Entity Framework from 6.1 to 6.2.

Creating the back-end CommerceManager website

1. In Visual Studio, navigate to **File** | **Add** | **New Project...**
2. In the left section, navigate to **Installed** | **Visual C#** | **Web**.
3. In the middle section, select **ASP.NET Web Application (.NET Framework)** project.
4. Enter the following options, as shown in the following screenshot:
   - Target: **.NET Framework 4.6.1** (or later compatible version).
   - Name: **CommerceManager**
   - Location: **C:\Episerver\CommerceTraining**



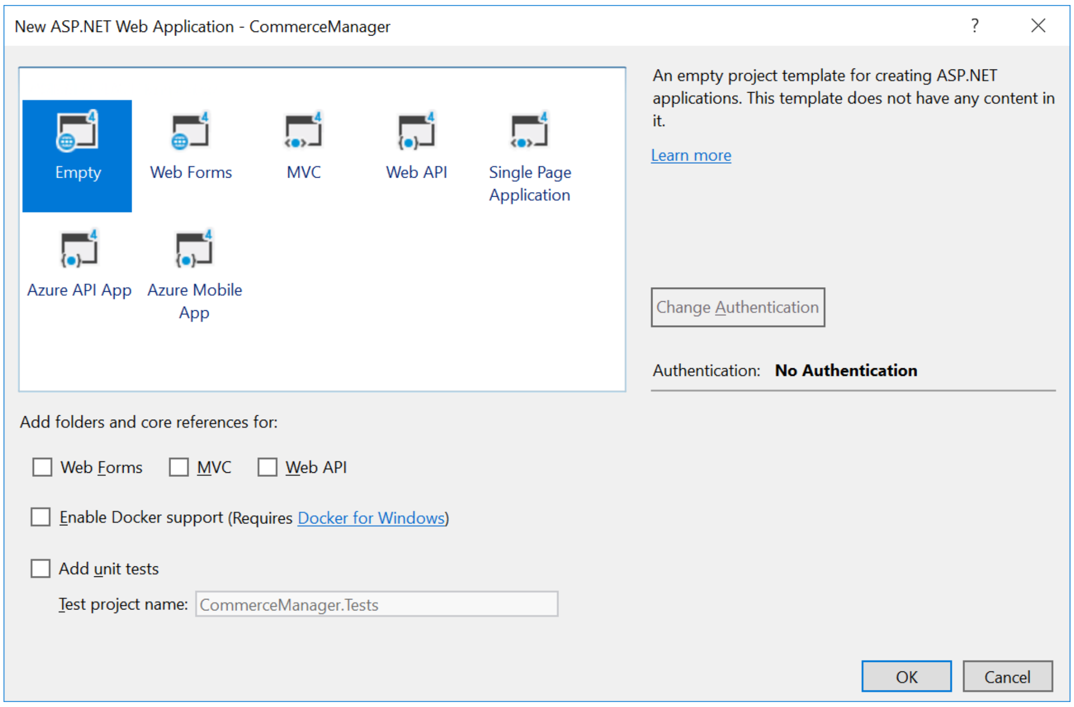
Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: CommerceManager

Location: C:\Episerver\CommerceTraining

5. Click **OK**.
6. In the templates list, select **Empty**, and click **OK**, as shown in the following screenshot:



An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication: **No Authentication**

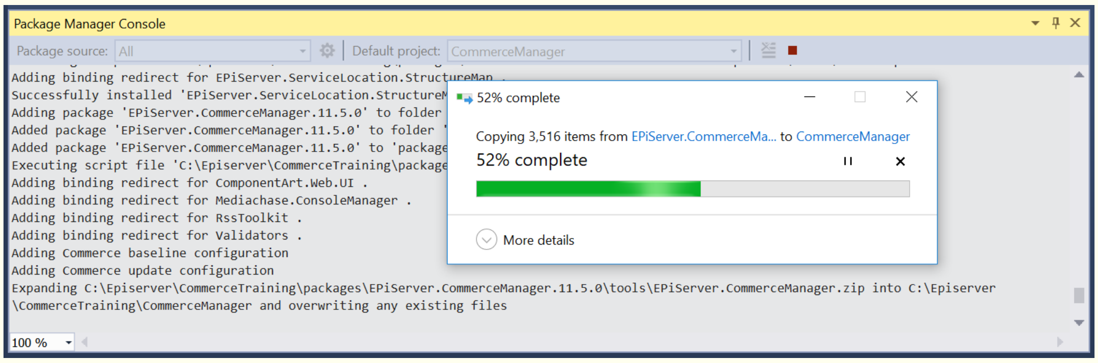
Test project name: CommerceManager.Tests

7. Navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.
8. At the top of the **Package Manager Console**, set **Package source** to **All**, and **Default project** to **CommerceManager**.

9. Enter the following command to add a package:

   ```
   Install-Package -ProjectName CommerceManager EPiServer.CommerceManager
   ```

> This will copy a lot of files, including copy all the pages, controls and configuration needed for Commerce Manager to run, so it can take a few minutes, depending on hardware, so be patient!

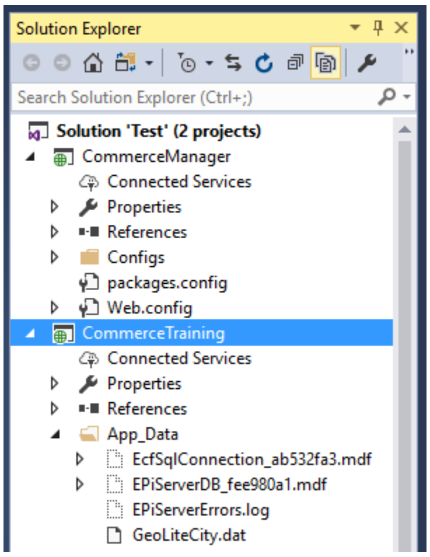


```
Adding binding redirect for EPiServer.ServiceLocation.StructureMap .
Successfully installed 'EPiServer.ServiceLocation.StructureM
Adding package 'EPiServer.CommerceManager.11.5.0' to folder
Added package 'EPiServer.CommerceManager.11.5.0' to folder '
Added package 'EPiServer.CommerceManager.11.5.0' to 'package
Executing script file 'C:\Episerver\CommerceTraining\package
Adding binding redirect for ComponentArt.Web.UI .
Adding binding redirect for Mediachase.ConsoleManager .
Adding binding redirect for RssToolkit .
Adding binding redirect for Validators .
Adding Commerce baseline configuration
Adding Commerce update configuration
Expanding C:\Episerver\CommerceTraining\packages\EPiServer.CommerceManager.11.5.0\tools\EPiServer.CommerceManager.zip into C:\Episerver
\CommerceTraining\CommerceManager and overwriting any existing files
```

Copying 3,516 items from EPiServer.CommerceMa... to CommerceManager
52% complete

Update the packages and database schema for CommerceTraining

1. In the **CommerceTraining** project, expand **App_Data**, and show all files, and note the filenames of the two databases, as shown in the following screenshot:



```
Solution 'Test' (2 projects)
  CommerceManager
    Connected Services
    Properties
    References
    Configs
    packages.config
    Web.config
  CommerceTraining
    Connected Services
    Properties
    References
    App_Data
      EcfSqlConnection_ab532fa3.mdf
      EPiServerDB_fee980a1.mdf
      EPiServerErrors.log
      GeoLiteCity.dat
```

> **EPiServerDB_{guid}.mdf** is the CMS database.
> **EcfSqlConnection_{guid}.mdf** is the Commerce database.

2. Open **Web.config**.
3. In the `<connectionStrings>` element, make sure that the two connection strings are correct for the two databases:
   a. CMS database: **EPiServerDB**
   b. Commerce database: **EcfSqlConnection**

> If you have added the two Commerce websites to an existing solution with other Episerver websites, then the connection strings might be pointing at the wrong projects!

```
<connectionStrings>
  <add name="EPiServerDB"
       connectionString="Data Source=(LocalDb)\MSSQLLocalDB;
```

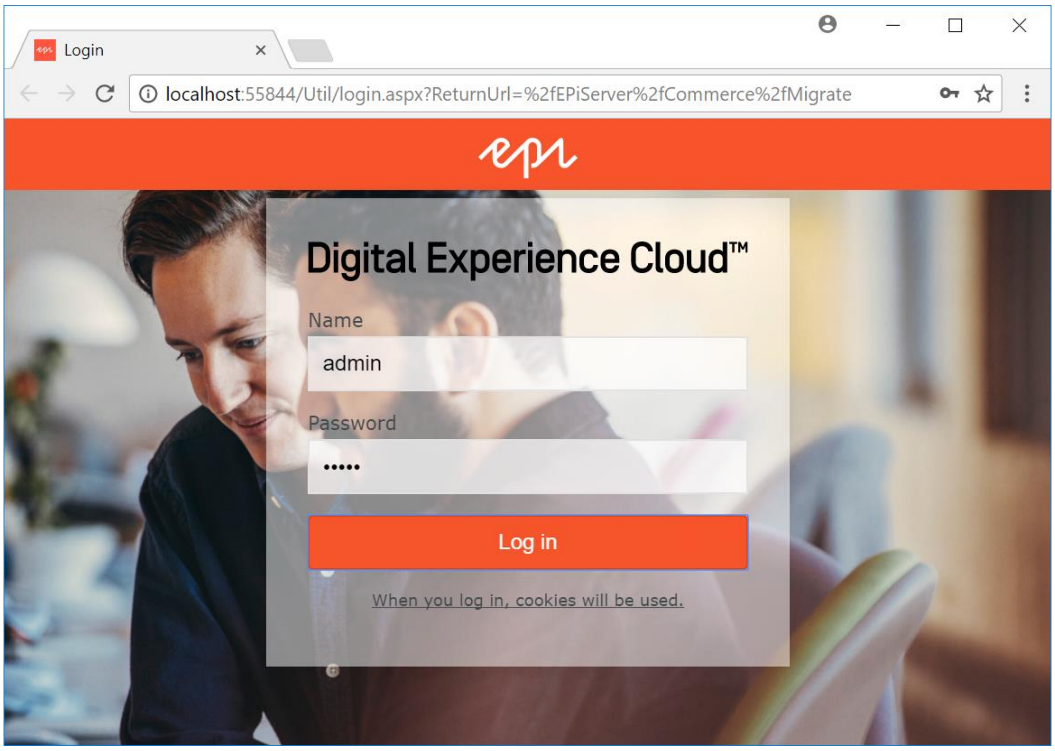
```
           AttachDbFilename=|DataDirectory|EPiServerDB_fee980a1.mdf;
           Initial Catalog=EPiServerDB_fee980a1;
           Connection Timeout=60;
           Integrated Security=True;
           MultipleActiveResultSets=True"
       providerName="System.Data.SqlClient" />

  <add name="EcfSqlConnection"
       connectionString="Data Source=(LocalDb)\MSSQLLocalDB;
           AttachDbFilename=|DataDirectory|EcfSqlConnection_ab532fa3.mdf;
           Initial Catalog=EcfSqlConnection_ab532fa3;
           Connection Timeout=60;
           Integrated Security=True;
           MultipleActiveResultSets=True"
       providerName="System.Data.SqlClient" />
</connectionStrings>
```

4. Navigate to **Tools** | **NuGet Package Manager** | **Package Manager Console**.
5. At the top of the **Package Manager Console**, set **Package source** to **All**, and **Default project** to **CommerceTraining**.
6. Enter the following command to update the packages and database schemas:

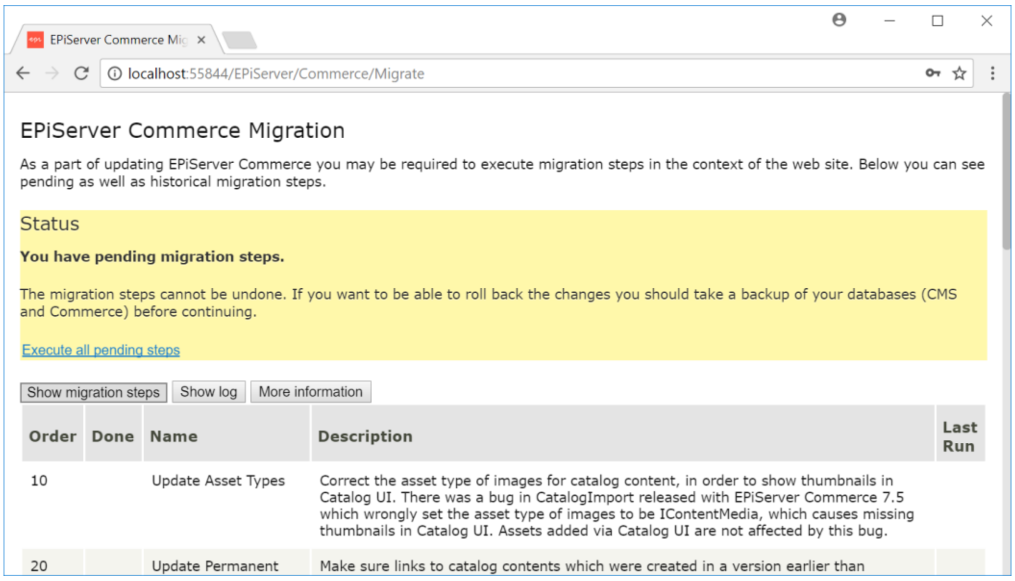
```
Update-Package -ProjectName CommerceTraining -ToHighestMinor
Update-EPiDatabase
```

Migrate the CommerceTraining website

1. Right-click the **CommerceTraining** project and select **Set as StartUp Project**.
2. Navigate to **Debug** | **Start without debugging** or press *Ctrl* + *F5*.
3. Log in with a user name of **admin** and a password of **store**, as shown in the following screenshot


Login
localhost:55844/Util/login.aspx?ReturnUrl=%2fEPiServer%2fCommerce%2fMigrate
Digital Experience Cloud™
Name
admin
Password
•••••
Log in
When you log in, cookies will be used.

4. On the **EPiServer Commerce Migration** page, click the link **Execute all pending steps**, as shown in the following screenshot:



EPiServer Commerce Mi
localhost:55844/EPiServer/Commerce/Migrate

EPiServer Commerce Migration

As a part of updating EPiServer Commerce you may be required to execute migration steps in the context of the web site. Below you can see pending as well as historical migration steps.

Status

You have pending migration steps.

The migration steps cannot be undone. If you want to be able to roll back the changes you should take a backup of your databases (CMS and Commerce) before continuing.

Execute all pending steps

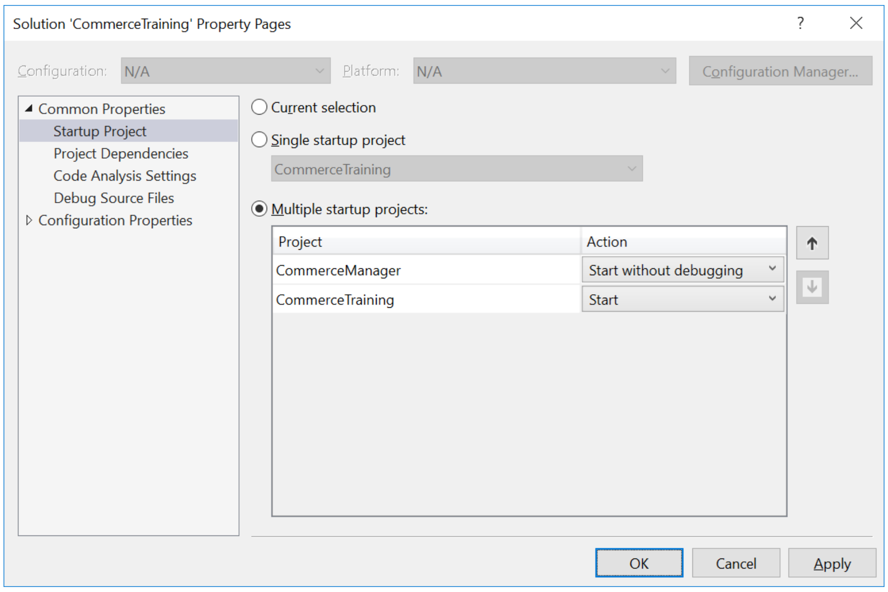
Show migration steps | Show log | More information

Order	Done	Name	Description	Last Run
10		Update Asset Types	Correct the asset type of images for catalog content, in order to show thumbnails in Catalog UI. There was a bug in CatalogImport released with EPiServer Commerce 7.5 which wrongly set the asset type of images to be IContentMedia, which causes missing thumbnails in Catalog UI. Assets added via Catalog UI are not affected by this bug.	
20		Update Permanent	Make sure links to catalog contents which were created in a version earlier than	

5. When the migration has completed, close the browser.

Test the Commerce websites

1. Right-click the solution, and select **Set StartUp Projects...**
2. Set multiple startup projects as follows:
   a. **CommerceManager**: Start without debugging
   b. **CommerceTraining**: Start



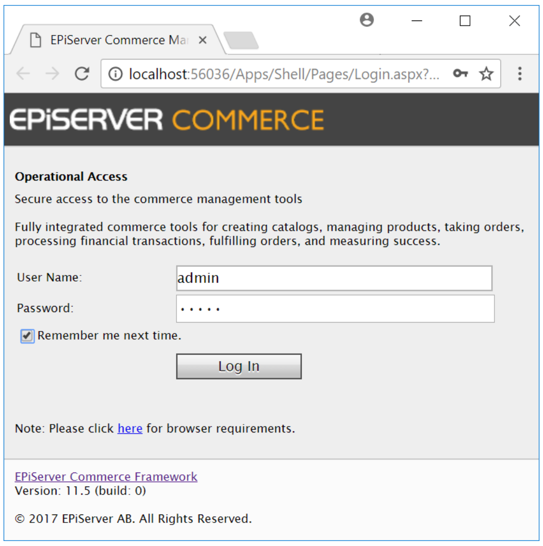
Solution 'CommerceTraining' Property Pages

Configuration: N/A   Platform: N/A   Configuration Manager...

Current selection
Single startup project: CommerceTraining
[✓ Multiple startup projects:]

Project	Action
CommerceManager	Start without debugging
CommerceTraining	Start

3. Navigate to **Debug** | **Start Debugging** or press *F5*, and wait for both websites to start, as shown in the following screenshots:



Operational Access

Secure access to the commerce management tools

Fully integrated commerce tools for creating catalogs, managing products, taking orders, processing financial transactions, fulfilling orders, and measuring success.

User Name: admin

Password: •••••

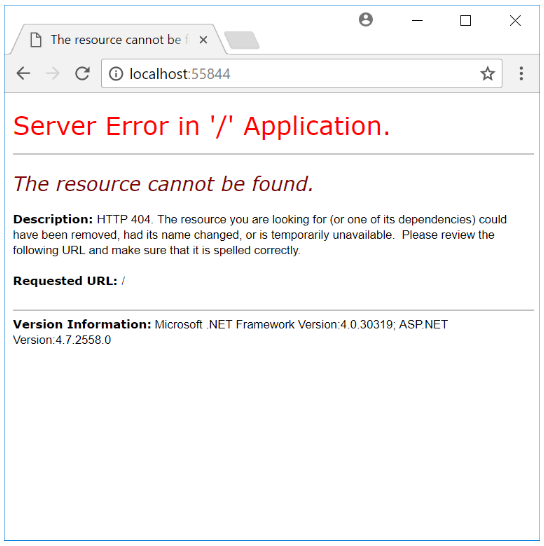
☑ Remember me next time.

Log In

Note: Please click here for browser requirements.

EPiServer Commerce Framework
Version: 11.5 (build: 0)

© 2017 EPiServer AB. All Rights Reserved.



Server Error in '/' Application.

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

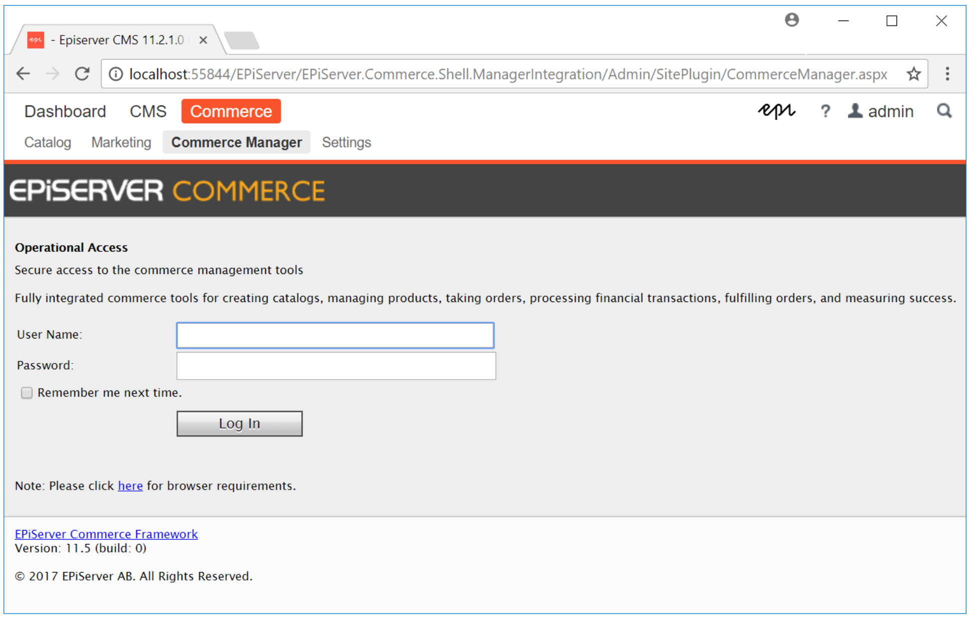
Requested URL: /

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.7.2558.0

4. In **Commerce Manager**, enter a user name of **admin**, enter a password of **store**, select **Remember me next time**, and click **Log In**, as shown in the preceding screenshot.

In a real site, you should use Commerce Manager to navigate to **Customer Management** | **Contacts**, click the **admin** user name, in the **Account** box click **Change Password**, and set a stronger password.

5. In the **CommerceTraining** site, in the browser address box, enter **/EPiServer/** at the end of the URL, and press *Enter*.
6. Enter a user name of **admin** and a password of **store**, and click **Log In**.
7. Navigate to **Commerce** | **Commerce Manager**, and nore that you can access Commerce Manager from the front-end website, as well as access the new CMS-style user interface for **Catalog** and **Marketing**, as shown in the following screenshot:



Operational Access

Secure access to the commerce management tools

Fully integrated commerce tools for creating catalogs, managing products, taking orders, processing financial transactions, fulfilling orders, and measuring success.

User Name:

Password:

☐ Remember me next time.

Log In

Note: Please click here for browser requirements.

EPiServer Commerce Framework
Version: 11.5 (build: 0)

© 2017 EPiServer AB. All Rights Reserved.

Congratulations!

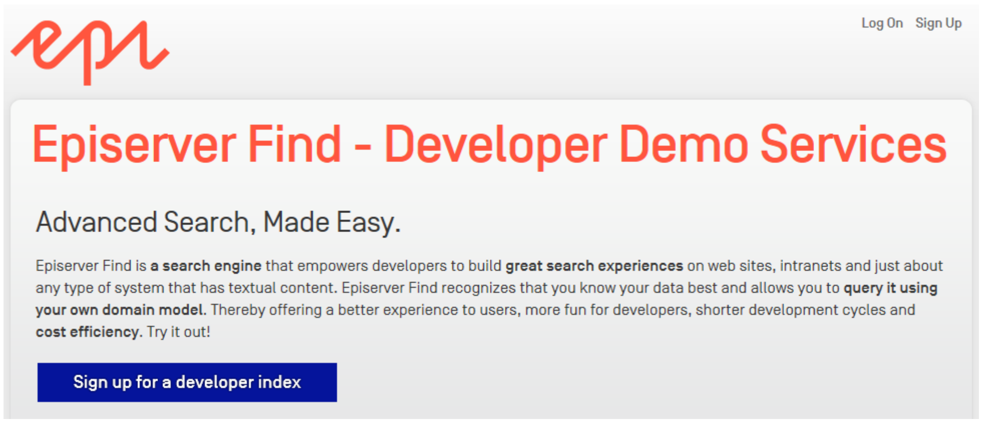
You are now ready to explore developing for the **Episerver Commerce** platform.

Creating an Episerver Find demo index

To use Episerver Find, you need a Find index.

Registering a Find account

1. Open a web browser and navigate to `http://find.episerver.com/`

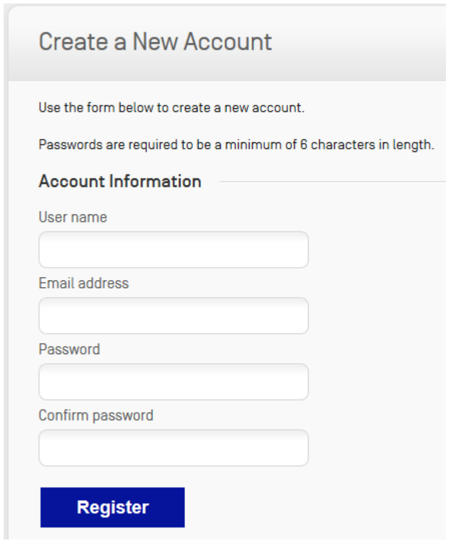


Episerver Find - Developer Demo Services

Advanced Search, Made Easy.

Episerver Find is **a search engine** that empowers developers to build **great search experiences** on web sites, intranets and just about any type of system that has textual content. Episerver Find recognizes that you know your data best and allows you to **query it using your own domain model**. Thereby offering a better experience to users, more fun for developers, shorter development cycles and **cost efficiency**. Try it out!

2. Click **Sign up for a demo index**, or **Sign Up** in the top right corner.



Create a New Account

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

Account Information

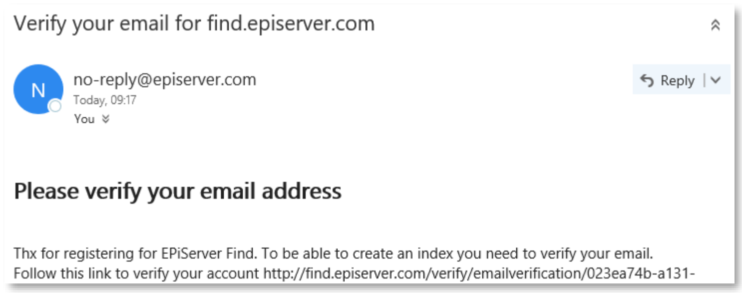
User name

Email address

Password

Confirm password

3. Fill in the required information, and click **Register**.
4. When the verification email arrives, copy and paste the link into your browser's address box and press ENTER to confirm your address and activate the index.



Verify your email for find.episerver.com

no-reply@episerver.com
Today, 09:17
You

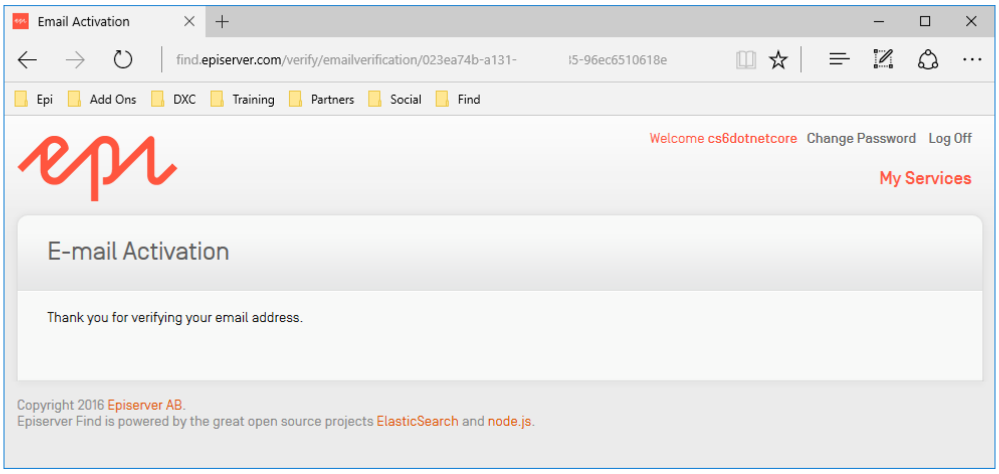
Please verify your email address

Thx for registering for EPiServer Find. To be able to create an index you need to verify your email.
Follow this link to verify your account http://find.episerver.com/verify/emailverification/023ea74b-a131-

If you don't receive your verification email, check your junk mail folder!

Creating a developer index

1. On the **E-mail Activation** verification page, click **My Services**.

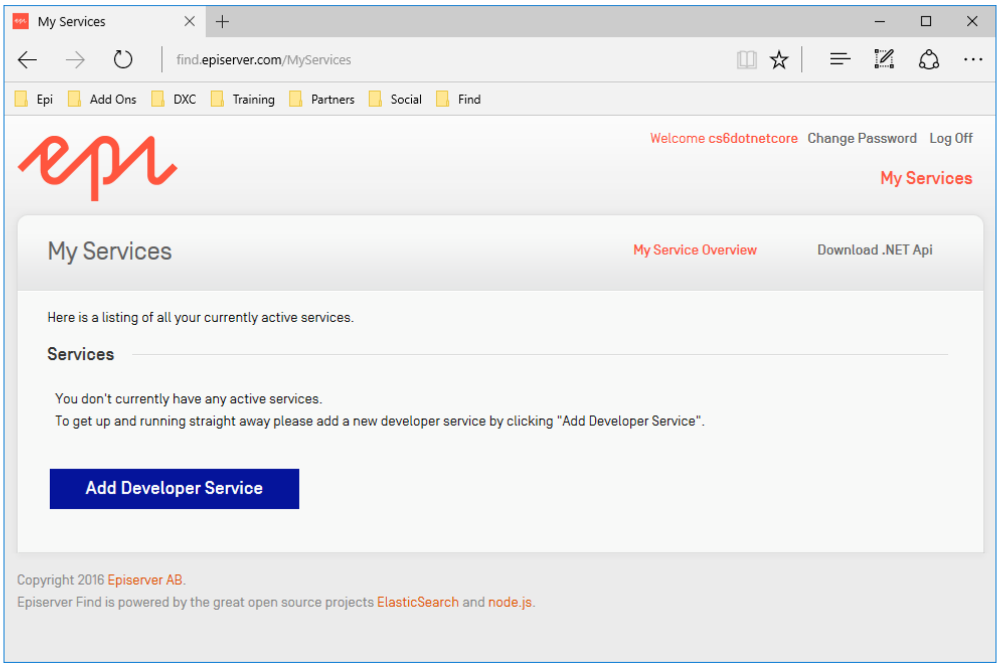


E-mail Activation

Thank you for verifying your email address.

Copyright 2016 Episerver AB.
Episerver Find is powered by the great open source projects ElasticSearch and node.js.

2. In **My Services**, click **Add Developer Service**:



My Services

Here is a listing of all your currently active services.

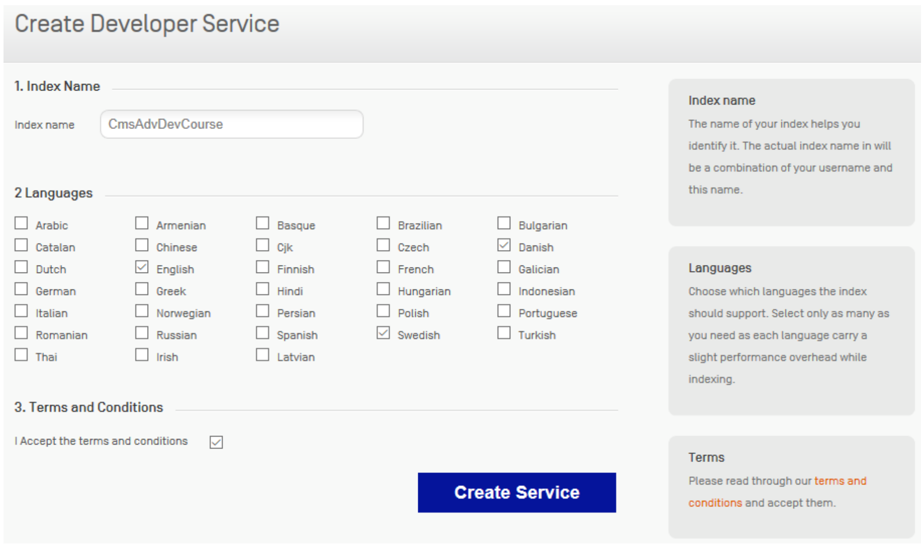
Services

You don't currently have any active services.
To get up and running straight away please add a new developer service by clicking "Add Developer Service".

Copyright 2016 Episerver AB.
Episerver Find is powered by the great open source projects ElasticSearch and node.js.

3. Fill in a name for your index. The name is technically irrelevant but should preferably represent what you're using the index for, such as, ***CustomerName*TestIndex**.
4. Select at least **English** from the list of languages. You may also select another language, preferably one that you know.

5. Check the terms and conditions checkbox, and click **Create Service**.



Create Developer Service

1. Index Name

Index name: CmsAdvDevCourse

2 Languages

☐ Arabic	☐ Armenian	☐ Basque	☐ Brazilian	☐ Bulgarian
☐ Catalan	☐ Chinese	☐ Cjk	☐ Czech	☑ Danish
☐ Dutch	☑ English	☐ Finnish	☐ French	☐ Galician
☐ German	☐ Greek	☐ Hindi	☐ Hungarian	☐ Indonesian
☐ Italian	☐ Norwegian	☐ Persian	☐ Polish	☐ Portuguese
☐ Romanian	☐ Russian	☐ Spanish	☑ Swedish	☐ Turkish
☐ Thai	☐ Irish	☐ Latvian		

3. Terms and Conditions

I Accept the terms and conditions ☑

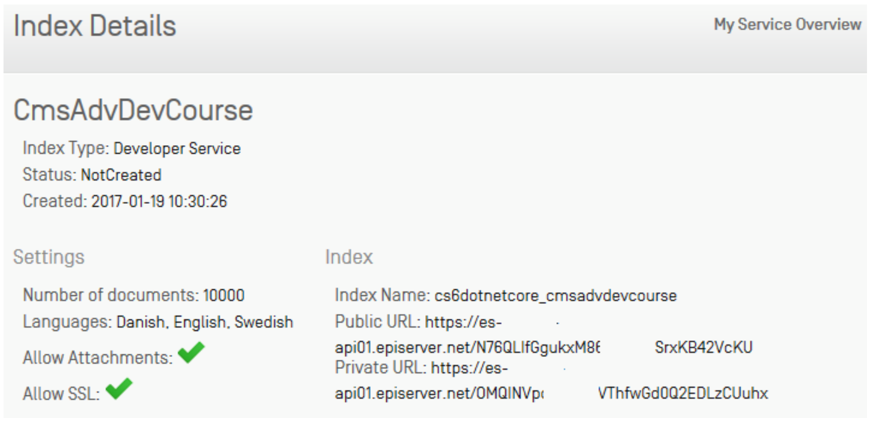
Create Service

Index name
The name of your index helps you identify it. The actual index name in will be a combination of your username and this name.

Languages
Choose which languages the index should support. Select only as many as you need as each language carry a slight performance overhead while indexing.

Terms
Please read through our terms and conditions and accept them.

6. You should now see a list of details about your index. Note the **Status**: it will probably be **NotCreated**, as shown in the following screenshot. If you wait a minute and refresh the page it should say **CreatedWithStats**.



Index Details

My Service Overview

CmsAdvDevCourse

Index Type: Developer Service
Status: NotCreated
Created: 2017-01-19 10:30:26

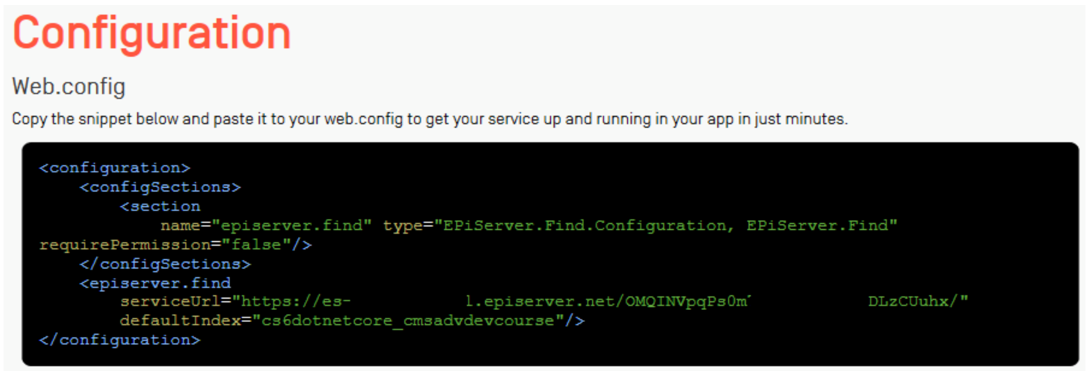
Settings

Number of documents: 10000
Languages: Danish, English, Swedish
Allow Attachments: ✓
Allow SSL: ✓

Index

Index Name: cs6dotnetcore_cmsadvdevcourse
Public URL: https://es-api01.episerver.net/N76QLIfGgukxM8 SrxKB42VcKU
Private URL: https://es-api01.episerver.net/OMQINVp VThfwGd0Q2EDLzCUuhx

7. Note the **Configuration**, as shown in the following screenshot. In the next section, you will copy and paste this into the site's Web.config.



Configuration

Web.config

Copy the snippet below and paste it to your web.config to get your service up and running in your app in just minutes.

```
<configuration>
    <configSections>
        <section
            name="episerver.find" type="EPiServer.Find.Configuration, EPiServer.Find"
requirePermission="false"/>
    </configSections>
    <episerver.find
        serviceUrl="https://es-           l.episerver.net/OMQINVpqPs0m           DLzCUuhx/"
        defaultIndex="cs6dotnetcore_cmsadvdevcourse"/>
</configuration>
```

Congratulations!

You are now ready to explore developing for the **Episerver Find** platform.