**SQL Injection:**

To test enter below string in search box. It will drop UserData table. Below is given query to re-create table

Live';drop table UserData;--

CREATE TABLE [dbo].[UserData](

[UserId] [int] NOT NULL,

[Name] [varchar](50) NULL,

[Address] [varchar](500) NULL,

[EmailId] [varchar](100) NULL

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

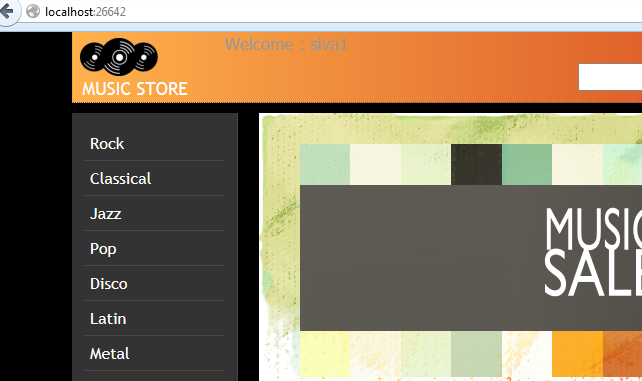
GO

INSERT [dbo].[UserData] ([UserId], [Name], [Address], [EmailId]) VALUES (1, N'John', N'Banglore', N'john@gmail.com')

INSERT [dbo].[UserData] ([UserId], [Name], [Address], [EmailId]) VALUES (2, N'Martin', N'Pune', N'martin@gmail.com')

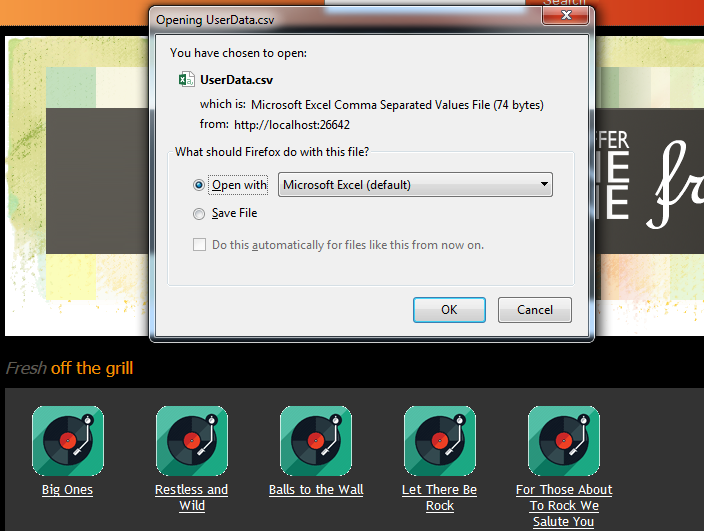
**HttpForbiddenHandler:**

1:Go to Music store.

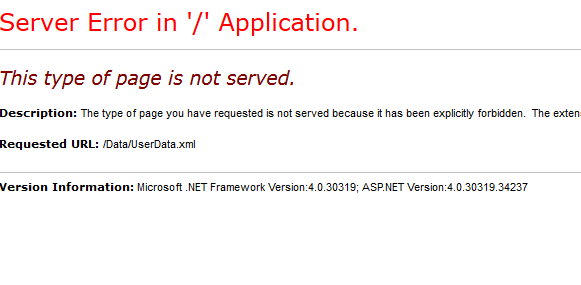


2:Try to change url to <http://localhost:26641/Data/UserData.csv>

3:Now you able to open csv file.



4:If you try to access <http://localhost:26641/Data/UserData.xml> ,it will Not Allow.



This configuration need to be specified for all file extension required to be blocked.

<system.web>

<httpHandlers>

<add verb="\*" path="\*.xml" type="System.Web.HttpForbiddenHandler" />

</httpHandlers>

</system.web>

<system.webServer>

<handlers>

<add verb="\*" path="\*.xml" type="System.Web.HttpForbiddenHandler" name="XML"/>

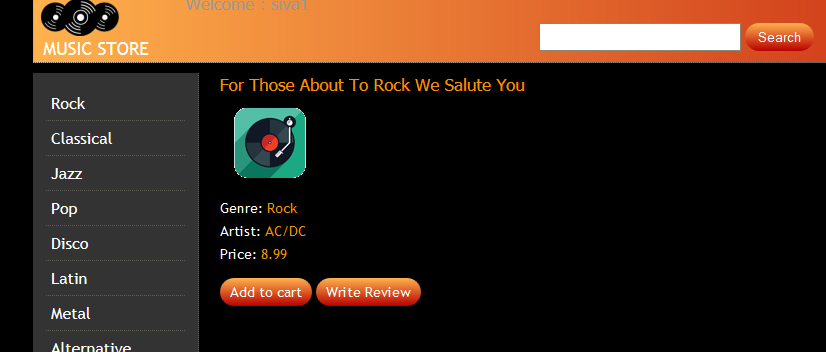
</handlers>

</system.webServer>

The httpHandlers element is used on sites running on IIS 5 – 6, or IIS 7.x in Classic mode (IIS 6 compatibility mode). The handlers element is used on sites running on IIS 7.x (Integrated mode).

**Cross Site Scripting:**

1:Click on Music Image.



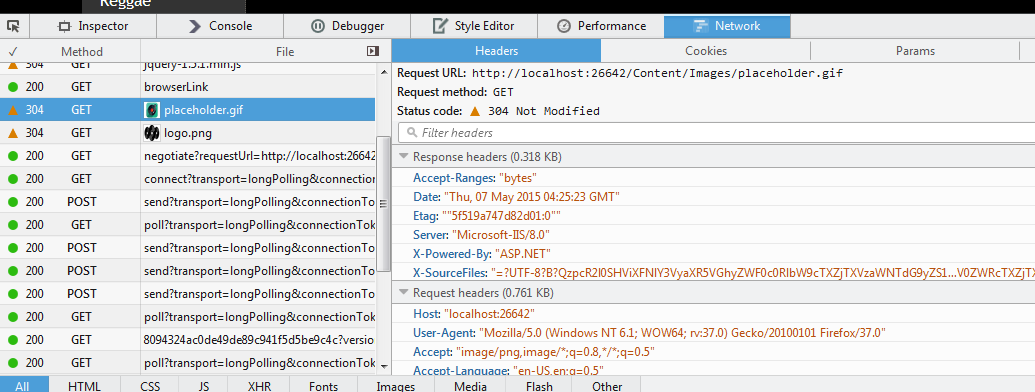
2:In write Review text area enter below text and save it. After that whenever page get loaded it will execute script in below text and open new window.

This product is good.Checkout some more<script>window.open('http://www.w3schools.com/', '\_blank');</script>

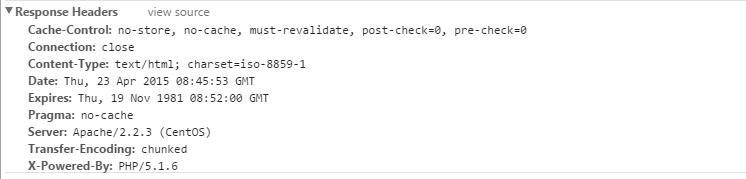
**Information Disclosure (HTTP Response Headers)**

1:Go to music store and press F12.

2:you will get below screen.



3:click on Network Tab.



Above data can be seen in network tab of browser which shows the server details which can help user to predict some of setting and get malicious information.

Below piece of code to be added to Global.aspx to remove some information from header.

protected void Application\_PreSendRequestHeaders(object sender, EventArgs e)

{

// Remove the "Server" HTTP Header from response

var app = sender as HttpApplication;

if (null != app && null != app.Context && null != app.Context.Response)

{

NameValueCollection headers = app.Context.Response.Headers;

if (null != headers)

{

headers.Remove("Server");

headers.Remove("X-AspNet-Version");

headers.Remove("X-AspNetMvc-Version");

headers.Remove("X-Powered-By");

}

}

}

Business Logic Bypassing (Client Side):

Relying only on client side validation can cause malicious users can easily bypass logic (Validation)

Steps To Reproduce:

1. Select an Item from Album.
2. Add item to Cart.
3. Click on **checkout.**
4. Enter Promo code as “FREE”.
5. Click on “**Submit**” button without filling any other details other than Promo code. You will see validation messages.
6. Disable JavaScript (How to disable and enable JavaScript [Click Here](http://enable-javascript.com/)).
7. Refresh the Checkout page.
8. Now click on “**Submit**” button without filling any other details other than Promo code. You can submit data without any validation messages.

Application errors with stack trace:

Steps To Reproduce:

1. Select an Item from Album.
2. When you go to Item Details page you will see the URL something Like <http://localhost:26641/Store/Details/6>
3. Change the Integer value appended at the end of the URL to some characters. (In above example from 6 to “some Characters”).
4. Press Enter key you will get the error message with Stack trace.

Always set a custom error page:

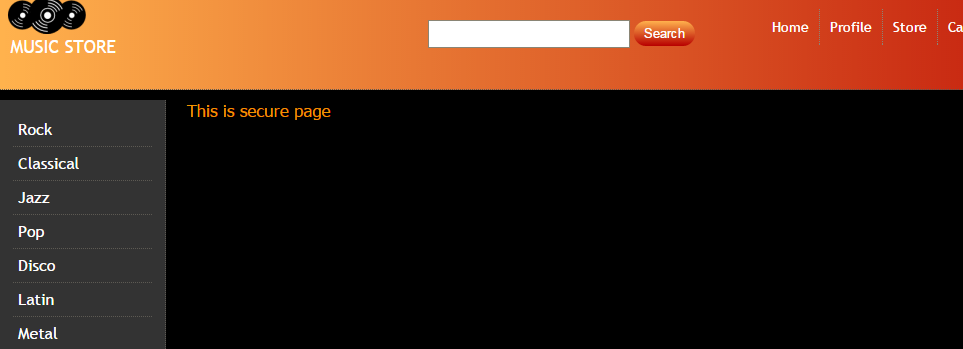
Steps To Reproduce:

1. Select an Item from Album.
2. When you go to Item Details page you will see the URL something Like <http://localhost:26641/Store/Details/6>
3. Change the Integer value appended at the end of the URL to 0 (Zero). (In above example from 6 to 0 (Zero))
4. Press Enter key you will get the source level error message.

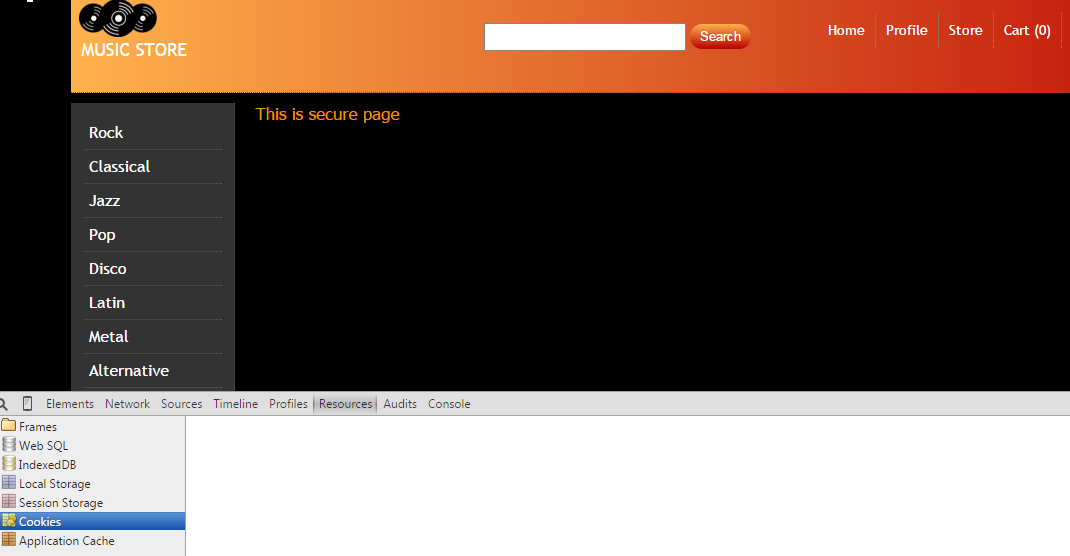
Session Fixation

Steps to Reproduce:

1. Click on **admin** Link.
2. You will get login screen, enter credentials and login.
3. Now click on **Profile** Link you will see the screen as below



1. Press F12 button on key board to open developer toll as below

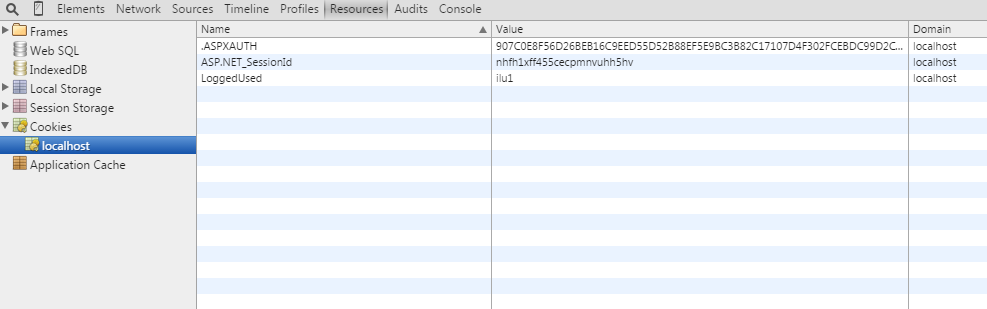


1. Expand Cookies tab as shown below

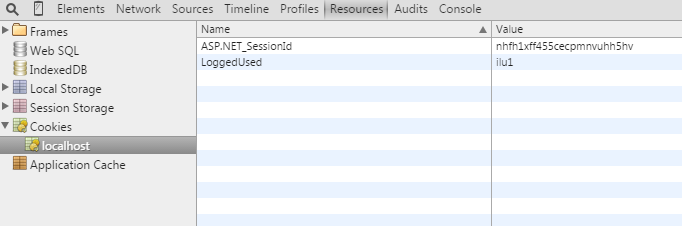
You will see two cookies

1) .ASPXAUTH (Authentication cookie)

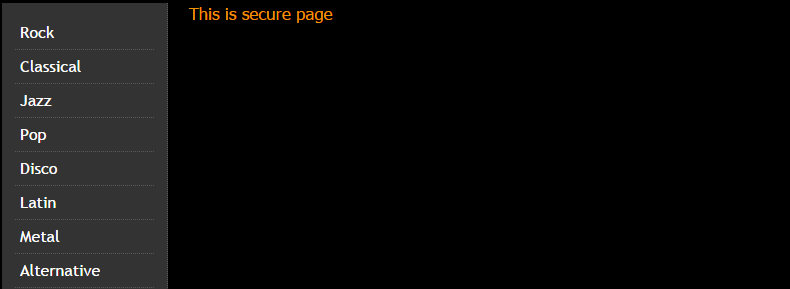
2) ASP.NET\_SessionId(Default Session cookie)



1. Now Click on **Sign out** Menu.
2. Repeat the steps from 4 to 5.
3. You can still see the **ASP.NET\_SessionId** as shown below.



1. If you again click on **Profile** Link. Still can access this page because, in **profile** page only default session id is being checked.



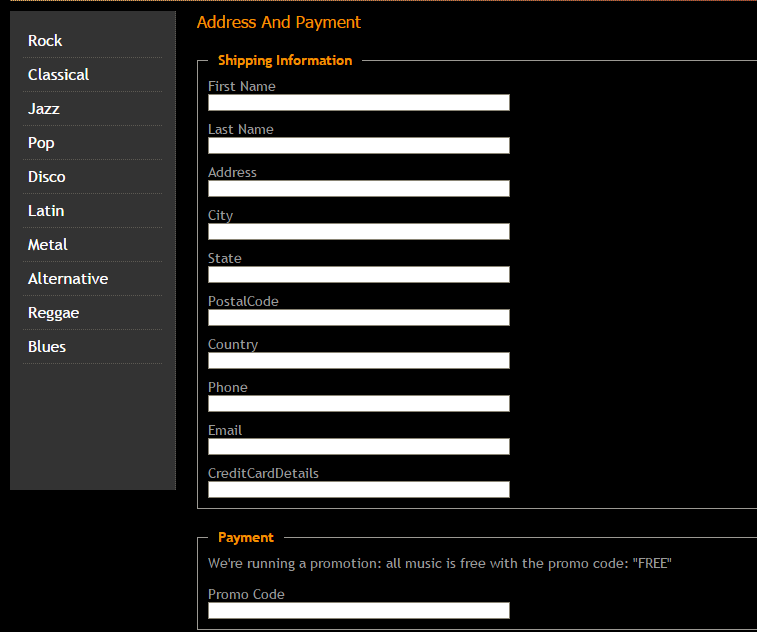
1. To avoid this problem, need to generate an .ASPXAUTH (Authentication cookie (token)) using forms authentication.
2. For example if you go to Checkout page by selecting and adding an item to cart, you cannot access it because it is secured (Implemented with Forms authentication).

**Non-secured cookie**

A non-secured cookie is the one which does not include “HttpOnly” and “Secure” attributes.

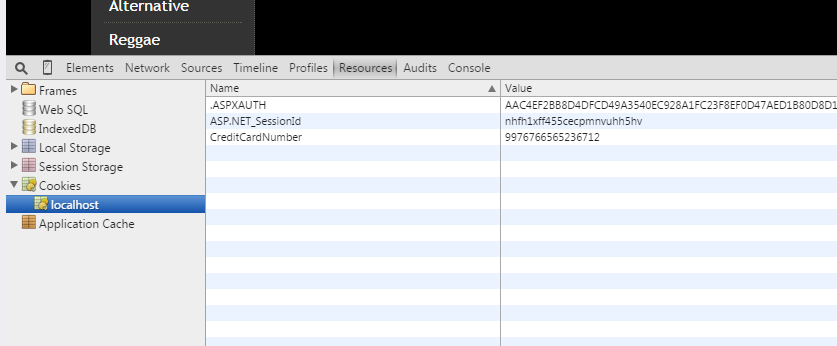
Steps to reproduce -

1. Add music to cart and checkoutit.
2. Now You able to see Payment Gateway as below.



3:Enter credit card Details , promo code and submit it.

4:Click F12 go to resource and expand cookies.



5:Now you able to see Creditcard Numbers.

**Fixing the Problem**

To make the cookie secured then set the “HttpOnly” and “Secure” attributes as shown below.

HttpCookie loggedUser = new HttpCookie("CreditCardNumber", model.Credit);

loggedUser.HttpOnly = true;

loggedUser.Secure = true;

Response.Cookies.Add(loggedUser);

**Authorization Bypassing:-**

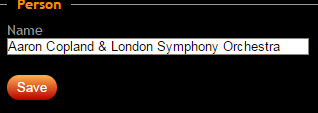
1:Run MVCMusic store application and click on Admin Page.

2:Try to login as administrator.

3:Then click on Album Page and u can able to see album Grid details.



4:Try to Edit Album Details.

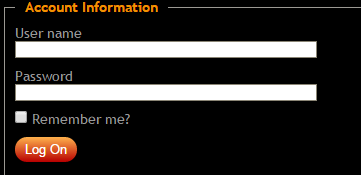


5:You can able to edit album details.

6:Go to admin page and try to login as Non Admin.

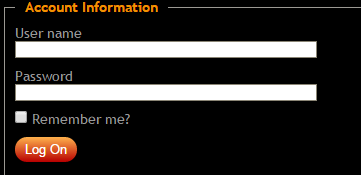
7:Click on album tab and try to click on edit button.

8:Now U unable to edit Album details and it always redirect to login page.



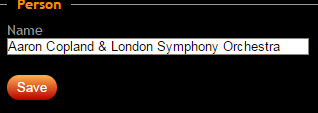
**Applied Authorization Bypass:**

1:Try to login as Non Admin and click on Album Tab.



2:Then click on Edit Button.

3:Now you able to change Album details.



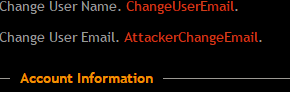
**Cross Site Request Forgery (CSRF):**

Cross Site Request forgery is a type of a hack where the hacker exploits the trust of a website on the user. In other words, the site trusts the user (because they have authenticated themselves) and accepts data that turns out to be malicious.

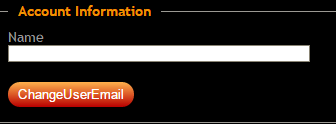
Attacker try to manipulate user details in CSRF.

**Generate scenario:**

1:click on Admin tab and click on Attacker Change Email.



2:You will get one text box.



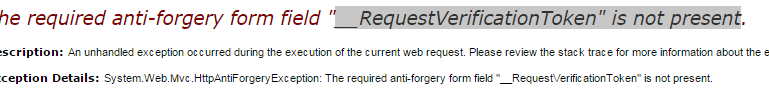
3:If you add some value in that text box and submit it will reflect on Artist Table on DB.

**Fixing the Problem**

1: First, we must add the unique token to the form to change the user's email when we display it.

2:Then add Html.AntiForgeryToken() it Generates a hidden form field (anti-forgery token) that is validated when the form is submitted.

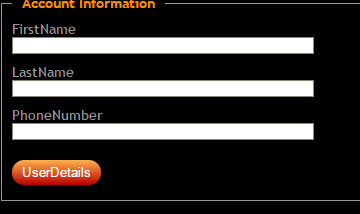
3: Click on ChangeUserEmail link and try to change user mail. It will redirect to requestverificationtoken is not present error.



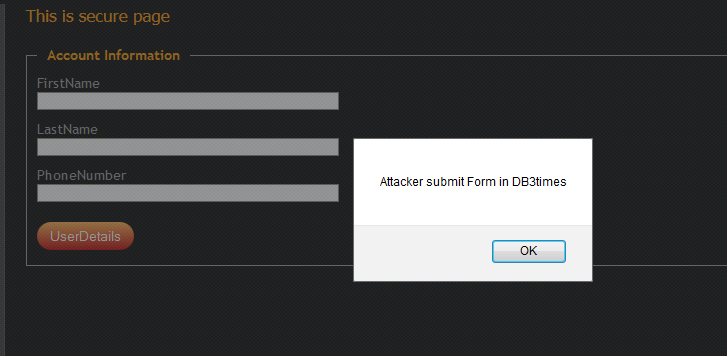
**Missing Anti-Scripting Mechanism:**

1:Try to click on Demo Tab.

2:Now you able to see one sereen like below.



3:If you wait few second You able to see one alert message and feel that form submit happen automatically.



4:Alert message will come 10 times.

5:If you will go to database and check dbo.UserDetails You will find 10 new record added automatically.

This is called BruteForceAttacks.

Prevention:

1:Applied CAPTA in public form so that machine code can’t submit it.

**Information Disclosure (HTTP Response Headers)**

1:Add MvcHandler.DisableMvcResponseHeader = true; in Application\_Start() global.asax

2:Add in global.asax

protected void Application\_PreSendRequestHeaders()

{

//Response.Headers.Remove("Server");

Response.Headers.Set("Server", "AntiHackerServer");

Response.Headers.Remove("X-AspNet-Version"); //alternative to above solution

Response.Headers.Remove("X-AspNetMvc-Version"); //alternative to above solution

}

3:Add in web.config

<!--<httpProtocol>

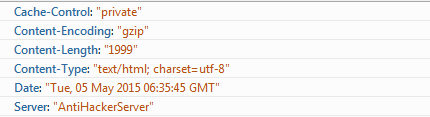
<customHeaders>

<remove name="X-Powered-By" />

</customHeaders>

</httpProtocol>-->

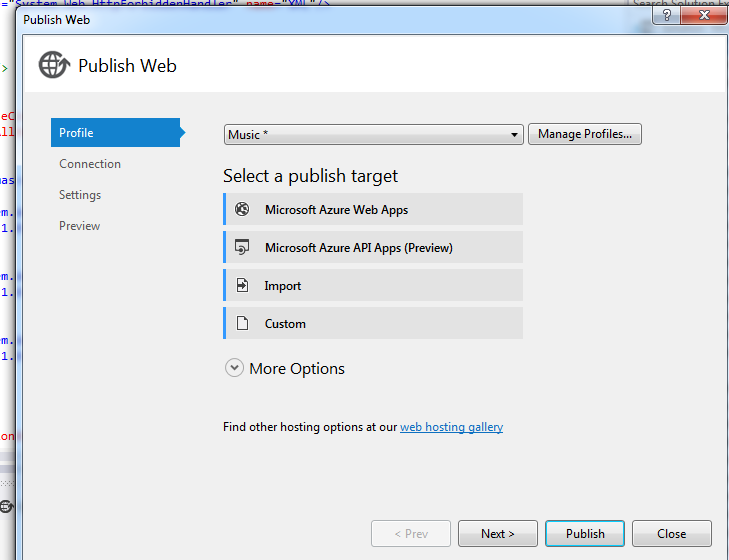
4:Add in web.config <httpRuntime enableVersionHeader="false" />



Deployments:

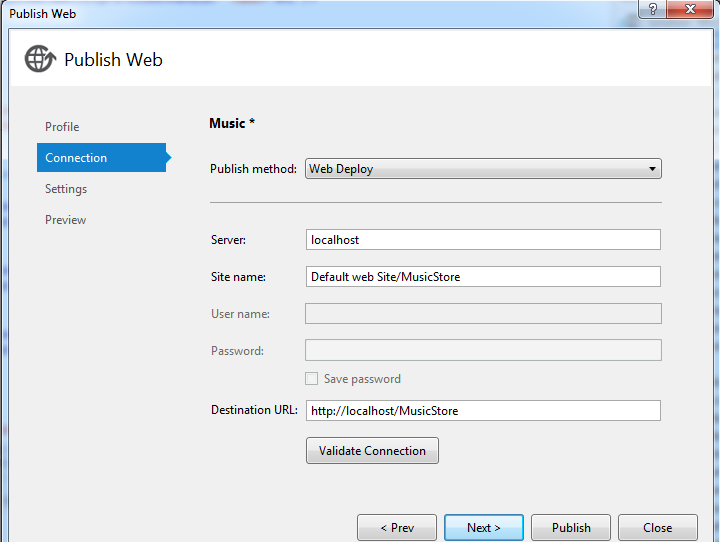
1:Go to Mvc Music store project and click on publish button.

2:Click on custom link as given below.

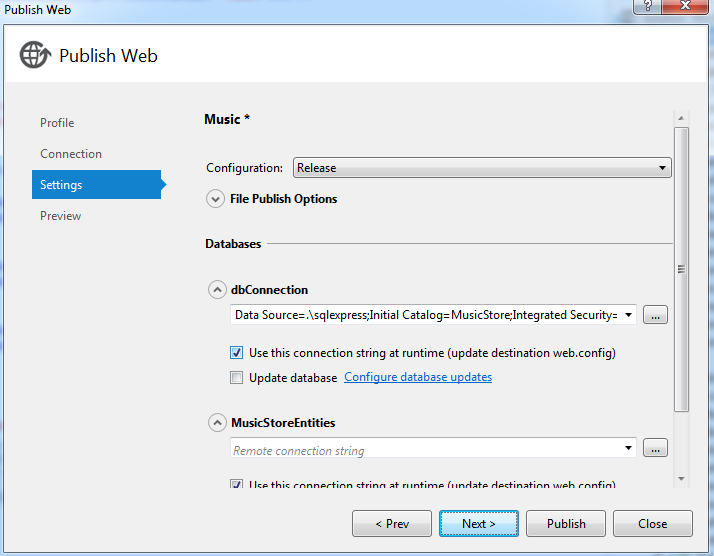
****

3:Give one name in Textbox. Then go to Next.

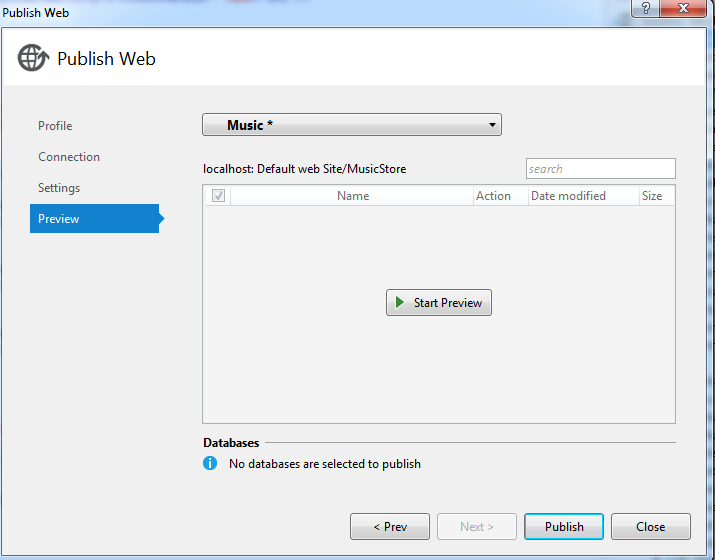
4:Fill detail as given below.



5:Then set db connection and test connection.



6:click on Next and start preview.



8:Click on publish.

Issue:

**You may face below error.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | I tweaked the application settings a lot, changing the application pool's identity (in Windows 8.1's IIS) to LocalSystem, LocalService, NetworkService, and ApplicationPoolIdentity. However, all of them failed to solve the problem I had logging into my database.  **Finally** I set the pool identity on LocalSystem and thought why it might be preventing "NT AUTHRITY\SYSTEM" from opening a connection to my database. I opened up SQL Server Management Studio as the user "Administrator" and checked out the **Server Roles** for NT AUTHORITY\SYSTEM under "logins" section. The default server role for this user was *public* by default. **I also checked sysadmin** and refreshed my web application form. This time it was a success. Great job! |