**XSS On WebAPI**

### **What is XSS?**

XSS(Cross Site Security) is a security attack where the attacker injects malicious code while doing data entry. This code can be a javascript, vbscript or any other scripting code. Once the code is injected in end user’s browser. This code can run and gain access to cookies,sessions, local files and so on.

**1st Approach :**

In MVC by default XSS attack is validated. So if any one tries to post javascript or HTML code he lands with the below error.

[ValidateInput(true)]

public ActionResult PostProduct(Product obj)

{

return View(obj);

}



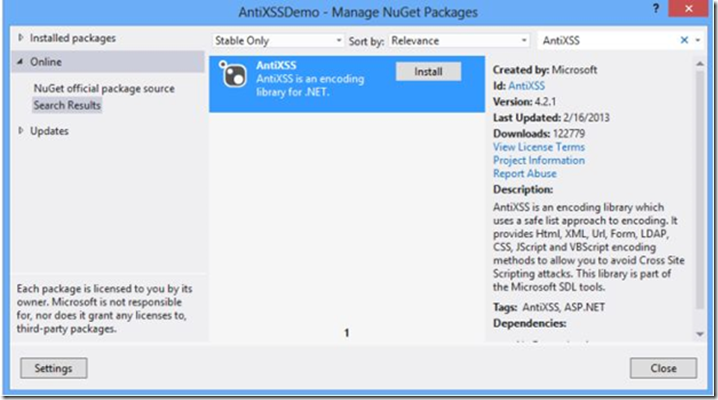
**Solution :**  
Even though its by default is handled in MVC, we will explicitly write attribute on each controller for precaution.

[ValidateInput(true)]

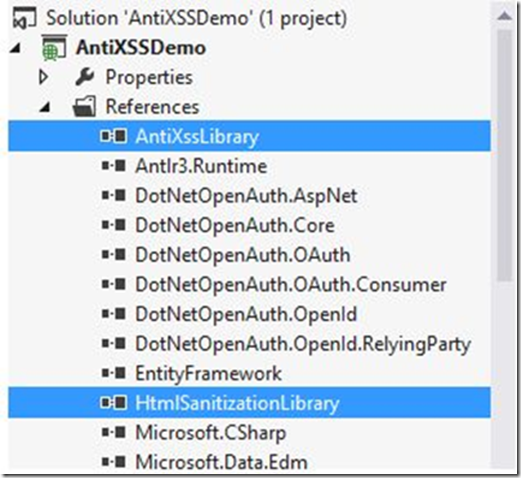
*[Source :* [*https://www.codeproject.com/Articles/995931/Preventing-XSS-Attacks-in-ASP-NET-MVC-using-Valida*](https://www.codeproject.com/Articles/995931/Preventing-XSS-Attacks-in-ASP-NET-MVC-using-Valida) *]*

**2nd Approach : Using AntiXSS Library**

Open NuGet, and then look for the "AntiXSS" package. Once you've found the package, simply install it.

[](http://www.dotnet-programming.com/image.axd?picture=image_11.png)

After the installation, you'll be able to see two new libraries in your projects references folder: AntiXssLibrary and HtmlSantizationLibrary.

[](http://www.dotnet-programming.com/image.axd?picture=image_12.png)

All you need to do is to make a little change in the controller, so as to prevent XSS.

[](http://www.dotnet-programming.com/image.axd?picture=image_13.png)

That's it! Now, in case any user will try inserting a malicious script with the text message, it will automatically be removed.

*[Source :* [*http://www.dotnet-programming.com/post/2015/04/12/How-to-Handle-Cross-Site-Scripting-in-ASPNET-MVC-Application.aspx*](http://www.dotnet-programming.com/post/2015/04/12/How-to-Handle-Cross-Site-Scripting-in-ASPNET-MVC-Application.aspx)*]*