In this video, we will discuss **preventing XSS while allowing only the HTML that we want to accept**. For example, we only want to accept <b> and <u> tags.

To achieve this let's filter the user input, and accept only <b></b> and <u></u> tags. The following code,  
**1.** Disables input validation  
**2.** Encodes all the input that is coming from the user  
**3.** Finally we selectively replace, the encoded html with the HTML elements that we want to allow.

[HttpPost]  
// Input validation is disabled, so the users can submit HTML  
[ValidateInput(false)]  
public ActionResult Create(Comment comment)  
{  
    StringBuilder sbComments = new StringBuilder();  
      
    // Encode the text that is coming from comments textbox  
    sbComments.Append(HttpUtility.HtmlEncode(comment.Comments));  
      
    // Only decode bold and underline tags  
    sbComments.Replace("&lt;b&gt;", "<b>");  
    sbComments.Replace("&lt;/b&gt;", "</b>");  
    sbComments.Replace("&lt;u&gt;", "<u>");  
    sbComments.Replace("&lt;/u&gt;", "</u>");  
    comment.Comments = sbComments.ToString();  
  
    // HTML encode the text that is coming from name textbox  
    string strEncodedName = HttpUtility.HtmlEncode(comment.Name);  
    comment.Name = strEncodedName;  
  
    if (ModelState.IsValid)  
    {  
        db.Comments.AddObject(comment);  
        db.SaveChanges();  
        return RedirectToAction("Index");  
    }  
  
    return View(comment);  
}  
  
**Warning:** Relying on just filtering the user input, cannot guarantee XSS elimination. XSS can happen in different ways and forms. This is just one example. Please read MSDN documentation on XSS and it's counter measures.