

# MVC

## Custom HTML Helpers and Displaying Time Values

A common problem when displaying time values is that date-time information is saved differently depending on the time zone of the server. When updating the time locally, the information will be influenced by the user’s computer’s settings. However, a published website resides on a different server, which will save data according to the server’s time zone.

One approach to solving this problem is to store the user’s time zone and take that into account when displaying time values.

### Storing the User’s Time Zone

1. Add another property to the IdentityModel:

public string TimeZone { get; set; }

1. Users may be asked for their time zone when registering and may have the option to change their time zone under Manage Account. The ViewModel for changing the time zone will require the following:

public SelectList TimeZone { get; set; }

1. In the controller actions, the TimeZone property can be set by looping through the time zones associated with the TimeZoneInfo class.

var timezones = TimeZoneInfo.GetSystemTimeZones();

model.TimeZone = new SelectList(timezones, "Id", "Id");

1. A dropdown list of time zones can then be passed into the views.

@Html.DropDownList("TimeZone", Model.TimeZone, new { @class = "searchable-select"})

1. In the controller’s Post action, pass in the time zone as a string and update the user’s information.

### Creating a Custom HTML Helper

1. In our views, we have been using Razor’s built-in HTML helpers. HtmlHelper is an extension class with methods that return html elements using the data type IHtmlString. We can build our own HTML helpers to use in our views.
2. Create the custom HTML Helper:

namespace CustomHelpers

{

public static class CustomHTMLHelpers

{

public static IHtmlString ToUserTime(this HtmlHelper helper, DateTimeOffset ModelTime, string timezone)

{

var timezoneId = TimeZoneInfo.FindSystemTimeZoneById(timezone);

var newTime = TimeZoneInfo.ConvertTime(ModelTime, timezoneId);

string htmlString = newTime.ToString();

return new HtmlString(htmlString);

}

public static IHtmlString ToUserTime(this HtmlHelper helper, DateTimeOffset ModelTime, string timezone, string ToStringFormat)

{

var timezoneId = TimeZoneInfo.FindSystemTimeZoneById(timezone);

var newTime = TimeZoneInfo.ConvertTime(ModelTime, timezoneId);

string htmlString = newTime.ToString(ToStringFormat);

return new HtmlString(htmlString);

}

}

}

1. In your Web.Config file in your Views folder, add <add namespace="CustomHelpers" /> to your list of namespaces between the <pages pageBaseType="System.Web.Mvc.WebViewPage"> </pages> tag.
2. The HTML helper can now be utilized in the views.

@Html.ToUserTime(Model.Created, Model.User.TimeZone)

1. To take the time zone into account when displaying date-time information, save the current DateTimeOffset as DateTimeOffset.UtcNow. Saving the time as coordinated universal time makes the time simple to convert to any time zone.