

Project 5

T-2013-VEFF, Vefforritun, 2014-1

Assignment

Note: you are allowed (and encouraged!) to work on this assignment in groups up to 3.

In this assignment you are supposed to create a comment application that uses Ajax. You will start with [this application](#), which currently allows the user to view a video, and to add comments to it. However, the current implementation is flawed, because when a comment is added, the entire page is reloaded, and the video starts from the beginning. In your implementation, you will fix this.

There are two operations the user should be able to perform, and it is your task to implement them:

- Add a new comment to the comment thread
- "Like" a previous comment

Adding a comment

Instead of implementing the first requirement using a standard HTML <form> like the application currently does, you are supposed to remove the form and use Ajax instead, meaning that if the user is watching the video and wants to comment, he can do so without re-submitting the page, and therefore starting the video again.

When a new comment is added, the application should check if other comments have been added by other users since the page was loaded (or since the last time the page retrieved a list of updates), and display them in the proper order.

You can test this by:

1. start your application from Visual Studio
2. open the url to the application in two different browsers
3. add a new comment in the first browser
4. add a new comment in the second browser
5. the comment added in step 3 should appear when the comment from step 4 is added, along with the newly added comment from step 4 itself.

Note that this is only required when a comment is added, i.e. the page does not have to update itself under other circumstances.

Since we haven't covered authentication in ASP.NET MVC, we will simply assume that the user types his username in a separate input field, and we will not validate that username (we assume he will always type a correct username). You could also simply hardcode the username.

Adding a "like"

Each user should be able to "like" each comment exactly once. Comments that have been "liked" should display a list of the users that liked them, for instance in a similar way this is done in Facebook.

FAQ

This section will contain frequently asked questions.

Q.: how can I change this silly Date format that is returned in the JSON response:

\Date(1292962456255)\

A.: There are at least two ways to do this:

1. Do the conversion on the server, i.e. don't let the server return an object that contains a DateTime property, convert it to a string instead:

```
var result = repository.GetAllComments( );
var newResult = from c in result
                select new {
                    CommentDate=c.CommentDate.ToString( ),
                    ID = c.ID,
                    CommentText=c.CommentText,
                    UserName=c.UserName };

return Json( newResult, ... );
```

2. Do the conversion on the client. When you receive a reply after having issues an Ajax query, you can convert the strings to a JavaScript date:

```
success : function( data ){
    for ( var i = 0; i < data.length; i++ ){
        data[i].CommentDate = ConvertStringToJSDate( data[i].CommentDate );
    }
    // ... etc., convert the data into HTML by using the template as
    before...
```

where the function ConvertStringToJSDate() could be defined like this:

```
function ConvertStringToJSDate( dt )
{
    var dtE = /^\/Date\((-?[0-9]+\))\/$/ .exec(dt);
    if (dtE) {
        var dt = new Date(parseInt(dtE[1], 10));
        return dt;
    }
    return null;
}
```

Hand in

Please hand in a single archive file (.zip, .rar, 7z), containing the entire solution.

It should be possible to unzip the archive, and run it instantaneously without any changes to the code.

Again: you may work on this in groups up to 3. (and this is encouraged).