Seminar 4 report

Internet Applications, ID1354

Niclas Fölster - nfolster@kth.se

Wed 13 December 2017

1 Introduction

Use JavaScript and AJAX for, at least, reading and writing and deleting recipe comments.

2 Literature Study

I studied many examples of AJAX and javascript, and also used W3Schools.

3 Method

I kept using my framework and implemented javascript in it.

4 Result

I wasn't sure on how to do this, but in the end I found a way. I used AJAX to connect through my route to my CommentController in the read and write parts.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
function deleteComment(id)
{
$.ajaxSetup({
  headers: {
    'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
  }
});

$.ajax({
            type: 'post',
            url: '/delete',
            data: {id : id},
            success: function(){
                $('#comments').load(document.URL +  ' #comments');
            }
        })
}
</script>
```

Illustration 1: delete script

For the delete, I basically added an onclick($commentId) to the deletebutton. So I sent in the

comment id so it knew which comment to delete. In both scripts I had to ad a csrf header to make sure that the cross-site request forgery security kept working.

```
<script>
$(function()
{
$('#write').on('submit',function(e){
    $.ajaxSetup({
      headers: {
        'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
      }
    });

e.preventDefault(e);
$.ajax({
            type: 'post',
            url: '/write',
            data:$(this).serialize(),
            success: function(data){
                $('#comments').load(document.URL + ' #comments');
            }
        })
});
});
</script>
```

Illustration 2: write script

In the write script I serialized and sent the form data to the controller so it could write the new comment to the database.

In both scripts I **reloaded the comment-div**. It is there where the comments are gotten and added. This made it really simple once I figured that out. So both scripts have two parts. Send info to the controller, and once that succeeds, reload the comment-div.

```
<div class="comment">
    <h3>Comments</h3>
        <ul id="comments" class="comment">
            @foreach ($comments as $comment)
                <li id={{$comment->id}}><!--
                -->{{$comment->body}}
                    @if(Auth::check())
                        @if($comment->user_id==Auth::user()->id)
                            <form class="delete" id="delete" method="post" action="">
                                {{ csrf_field() }}
                                {{--<input name="delete" type="image" src="img/delete.png">--}}
                                <input id="commentId" type="hidden" name="recipe_id" value="{{$comment->id}}">
                                <img  id="deleteButton" src="img/delete.png" onclick="deleteComment({{$comment->id}})">
                            </form>
                        @endif
                    @endif
                    <hr><p>{{$comment->created_at}}
                 by <b>{{$name = DB::table('users')->where('id', $comment->user_id)->value('name')}}
                </b></p></li>
            @endforeach
            <div id="writeComment"></div>
```

Illustration 3: Comment code. I reload this part.

Internet Applications, ID1354 Seminar 4 report

Git link: https://github.com/nfolster/ID1354_Projekt

5 Discussion

I have completed mandatory task 1. It was quite challenging, but once I understood how and ajax worked, and how it interacted with laravel, It worked out! The real breakthrough was when I learned how to reload only the required div.

6 Comments About the Course

I will write this in Swedish. Denna del var ganska utmanande, men jag tror att det var mest för att jag inte visste hur ajax interagerade med laravel. Men resultatet var häftigt. Jag la ner ca 15-30h på detta.