

ID dependency in post and comment

Q. Why deleting comment id from the comments array of the post is also essential, and why does it have to be done before deleting a comment?

Sol. Deleting comments will be slightly different from deleting posts because comments can be made on another user's posts and also on the user's own post.

```
module.exports.destroy = function(req, res){
  Comment.findById(req.params.id, function(err, comment){
    if(comment.user == req.user.id){
      let postId = comment.post;
      comment.remove();
      Post.findByIdAndUpdate(postId, {
        $pull: {
          comments: req.params.id
        }
      }, function (err, post) {
        return res.redirect('back');
      })
    }else{
      return res.redirect('back');
    }
  });
}
```

Every comment has an attribute 'post' which refers to the id of the post on which comment is made. Also, posts have an array of comments inside them (i.e id of comments) for preloading purposes (fetching all the comments alongside the post). When a post is rendered, alongside its comments are also rendered using the comments array.

So before deleting the comment, we need to fetch the post id of that comment we want to delete because we need to go inside that post and find the comment and delete it but If we directly delete the comment, this post id has been lost.

That's why we need to save the post id in a variable and then be able to find the comment id, with the help of the post id inside the comment of this post and remove it from here also.

In the end, we have to update the post because deleting one of the comment reference id.

Common error/bug faced in this lecture:

For e.g.

- a) if you got some error like `req.params.id` → undefined, then also page gets refreshed and comment is not getting deleted.
- b) Unable to pulling the comment id from the comments array.

Possible Solution:

1. Be careful while putting brackets for \$pull, don't enclose the function in the same brackets.

```
Post.findByIdAndUpdate(postId, { $pull: { comments: req.params.id
} }, function(err, post) {
    return res.redirect('back');
});
```