

## Join the Stack Overflow Community

Stack Overflow is a community of 7.1 million programmers, just like you, helping each other.

Join them; it only takes a minute:

[Sign up](#)

## MOQ - Mocking MVC Controller's Response.Cookies.Clear()

```

36 if (dev.isBored() || job.sucks()) {
37     searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 // A career site that's by developers, for developers.

```



I am new to MOQ, but am using it with NUnit for unit testing.

I have all parts of my controller mocked, except the following line which throws an 'Object not set to an instance of an object' error message.

```
Response.Cookies.Clear();
```

I have the following extension method to mock the controller context which works for everything else I have come accross so far (very much thanks to the good people on this forum).

```

public static int SetUpForTest(this System.Web.Mvc.Controller ctl, string username,
TestingUtility.Roles role)
{
    var routes = new RouteCollection();
    MvcApplication.RegisterRoutes(routes);

    var request = new Mock<HttpRequestBase>(MockBehavior.Strict);
    request.SetupGet(x => x.ApplicationPath).Returns("/");
    request.SetupGet(x => x.Url).Returns(new Uri("http://localhost/a",
UriKind.Absolute));
    request.SetupGet(x => x.ServerVariables).Returns(new
System.Collections.Specialized.NameValueCollection());

    var response = new Mock<HttpResponseBase>(MockBehavior.Strict);
    response.Setup(x => x.ApplyAppPathModifier(Moq.It.IsAny<String>
())).Returns((String url) => url);
    // response.SetupGet(x => x.Cookies).Returns(new HttpCookieCollection());
    // This also failed to work

    var context = new Mock<HttpContextBase>(MockBehavior.Strict);
    context.SetupGet(x => x.Request).Returns(request.Object);
    context.SetupGet(x => x.Response).Returns(response.Object);
    context.SetupGet(x => x.Response.Cookies).Returns(new
HttpCookieCollection()); // still can't call the Clear() method

    //
    // Mock the controller context (using the objects mocked above)
    //
    var moqCtx = new Mock<ControllerContext>(context.Object, new RouteData(),
ctl);
    moqCtx.SetupGet(p => p.HttpContext.User.Identity.Name).Returns(username);
    moqCtx.SetupGet(p =>
p.HttpContext.User.Identity.IsAuthenticated).Returns(true);
    if (!string.IsNullOrEmpty(role.ToString()))
        moqCtx.Setup(p =>
p.HttpContext.User.IsInRole(role.ToString())).Returns(true);

    //
    // Pass the mocked ControllerContext and create UrlHelper for the
controller and return
    //
    ctl.ControllerContext = moqCtx.Object;
    ctl.Url = new UrlHelper(new RequestContext(context.Object, new
RouteData(), routes));
    return 1;
}

```

As you can see above I have tried to mock the 'get' of the cookies collection, which does not help.

Also, the actual Clear() method cannot be mocked as it is not a virtual method.

Obviously I don't want to test that the cookies are being cleared, I just want to be able to ignore it in testing.

Thanks,

Greg

c# asp.net-mvc nunit moq

edited Aug 7 '13 at 10:08



tereško

44.3k

16

69

127

asked Aug 7 '13 at 8:38



Greg

47

1

5

Have you seen this question: [stackoverflow.com/questions/1228179/...](http://stackoverflow.com/questions/1228179/...) – Halvard Aug 7 '13 at 9:08

Thanks for the response, but I have looked through the examples in the link and I do not see anything that I haven't done in the mocking of the controller context above. If I am missing something then please point me in the right direction! – Greg Aug 7 '13 at 9:38

Are you sure you are getting the mocked context and not a HttpContext.Current (for example) in that part of the code? Just an idea of what might be wrong. – JuhaKangas Aug 7 '13 at 11:07

## 2 Answers

This works for me when I do cookies.Clear()

```
var request = new Mock<HttpRequestBase>(MockBehavior.Strict);
request.SetupGet(x => x.ApplicationPath).Returns("/");
request.SetupGet(x => x.Url).Returns(new Uri("http://localhost/a",
UriKind.Absolute));
request.SetupGet(x => x.ServerVariables).Returns(new
System.Collections.Specialized.NameValueCollection());

var response = new Mock<HttpResponseBase>(MockBehavior.Strict);
response.Setup(x => x.ApplyAppPathModifier(Moq.It.IsAny<String>
())).Returns((String url) => url);
// response.SetupGet(x => x.Cookies).Returns(new HttpCookieCollection());
// This also failed to work

var context = new Mock<HttpContextBase>(MockBehavior.Strict);
context.SetupGet(x => x.Request).Returns(request.Object);
context.SetupGet(x => x.Response).Returns(response.Object);
context.SetupGet(x => x.Response.Cookies).Returns(new
HttpCookieCollection()); // still can't call the Clear() method
context.SetupGet(p => p.User.Identity.Name).Returns("blah");
context.SetupGet(p => p.User.Identity.IsAuthenticated).Returns(true);

var rc = new RequestContext(context.Object, new RouteData());

controller.ControllerContext = new ControllerContext(rc, controller);
```

answered Aug 7 '13 at 11:10



Luc Bos

1,336

9

22

Thank you - the solution was to only mock the HttpContextBase and not the ControllerContext as you showed in your example. Also thanks to Halvard for his answer pointing me in the right direction – Greg Aug 7 '13 at 13:46



Free sandbox.  
Hassle free.

Create a web app

No credit card required

(This is only half an answer, but was too big for the comment field ...)

Your mocking of new HttpCookieCollection() is correct. This code works in separation:

```
var request = new Mock<HttpRequestBase>(MockBehavior.Strict);
request.SetupGet(x => x.ApplicationPath).Returns("/");
request.SetupGet(x => x.Url).Returns(new Uri("http://localhost/a",
UriKind.Absolute));
request.SetupGet(x => x.ServerVariables).Returns(new
System.Collections.Specialized.NameValueCollection());
```

```
var response = new Mock<HttpServletResponse>(MockBehavior.Strict);
response.Setup(x => x.ApplyAppPathModifier(Moq.It.IsAny<String>())).Returns((String
url) => url);
// response.SetupGet(x => x.Cookies).Returns(new HttpCookieCollection()); // This
also failed to work

var context = new Mock<HttpContextBase>(MockBehavior.Strict);
context.SetupGet(x => x.Request).Returns(request.Object);
context.SetupGet(x => x.Response).Returns(response.Object);
context.SetupGet(x => x.Response.Cookies).Returns(new HttpCookieCollection()); //
still can't call the Clear() method

// Here clearing the cookies works just fine:
var instance = context.Object;
instance.Response.Cookies.Clear();
```

So the error is not there, but somewhere else. What happens if you comment out the line with `Response.Cookies.Clear()` from your code? Is everything else mocked correctly then? When you debug the test, can you see that the rest of the mock is as expected? I would be surprised if it was (but I have been surprised before...).

answered Aug 7 '13 at 11:03



Halvard

1,800 5 23 45