

Re: changing Request.InputStream

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-03/1331.html>

From: Karahan Celikel (*NOkarahan_celikelSPAM_at_hotmail.com*)

Date: 03/04/04

Date: Thu, 4 Mar 2004 11:24:46 -0600

Hi Steven,

Actually Request.InputStream is readonly. As far as I understand it is the Filter that can be writable. If there is a filter of the request then Application uses this filter instead of Request.InputStream. Yesterday, I found a workaround to the problem. But it is not an elegant solution.

Before the following Read method of the filter is called

```
public override int Read(byte[] buffer, int offset, int count)
{
    int c = _sink.Read(buffer, offset, count);
    Here I want to change the buffer
    return c;
}
```

size of the buffer array and count parameters are determined by the Length property of the Filter itself. So we can deceive the caller here by manipulating the Length property. Here we can generate the new request string and return the length of it

```
public override long Length
{
    get
    {
        if(this.newRequestStr!="")
            return this.newRequestStr.Length;
        else
        {
            if(_sink.Length!=0)
            {
                this.newRequestStr = ProcessRequest();
                return this.newRequestStr.Length;
            }
            else
                return 0;
        }
    }
}
```

```
}  
}  
}
```

and we need to change the Read method

```
public override int Read(byte[] buffer, int offset, int count)  
{  
    byte[] newBuffer =  
    System.Text.Encoding.UTF8.GetBytes(this.newRequestStr);  
  
    for(int i=0;i<count;i++)  
        buffer[i] = newBuffer[i];  
  
    return newRequestStr.Length;  
}
```

But as you can see, generating the new request string in the getter of the Length property is something like a hack rather than an elegant solution.

I would appreciate if you can investigate it further. There is not much documentation about this topic.

Thanks

Karahan Celikel