Subject: MultipartForm Class for use w/HttpClient
Posted by jeremy_c on Sat, 14 Aug 2010 15:18:36 GMT
View Forum Message <> Reply to Message

I created a class named MultipartForm that can be used w/Http(s)Client. It allows one to do file uploads. I know this will be of use to others as the question has come up on the forum a few times. What I don't know is the best way to share it or get feedback on it. Here's a sample use:

```
MultipartForm form;
form.Set("name", "John Doe");
form.Set("age", "18");
form.SetFile("bio", "mybio.txt", LoadFile("mybio.txt"));
form.SetFile("avatar", "me.jpg", Base64Encode(LoadFile("me.jpg")), "base64");

HttpClient http("http://localhost/upload.cgi");
http.ContentType(form.GetContentType());
http.Post(form.ToString());
```

That's basically it. In the end it creates a properly formatted multipart/form-data submission.

Jeremy

Subject: Re: MultipartForm Class for use w/HttpClient
Posted by koldo on Sat, 14 Aug 2010 19:18:20 GMT
View Forum Message <> Reply to Message

Hello Jeremy

Could you explain more details about your class?
It seems interesting.

Subject: Re: MultipartForm Class for use w/HttpClient
Posted by jeremy_c on Sat, 14 Aug 2010 19:46:17 GMT
View Forum Message <> Reply to Message

Simply put the existing HttpClient does not provide the facilities for file uploads via HTTP. It does not support Multipart Form data. It supports the POST format just fine which is suitable for most form data, but when a file upload is required (or even present) then the HttpClient falls short. That's where this class comes into play. It builds the post data for HttpClient.

The example I gave above would produce post data that looks like:

form.GetContentType() returns:

Content-Type:multipart/form-data; boundary=XgBtCfDkTmZlYqMtPcLgFgY

form.ToString() returns:

--XgBtCfDkTmZlYqMtPcLgFgY
Content-Disposition: form-data; name="name"

John Doe
--XgBtCfDkTmZlYqMtPcLgFgY
Content-Disposition: form-data; name="age"

18
--XgBtCfDkTmZlYqMtPcLgFgY
Content-Disposition: form-data; name="bio"; filename="mybio.txt"
Content-Transfer-Encoding: text/plain

Hello World!
I am a person who does not exist but is always used in examples.
--XgBtCfDkTmZlYqMtPcLgFgY
Content-Disposition: form-data; name="avatar"; filename="me.jpg"
Content-Transfer-Encoding: base64

...bunch of data snipped for the posts sake...
--XgBtCfDkTmZlYqMtPcLgFgY--

---

Subject: Re: MultipartForm Class for use w/HttpClient
Posted by koldo on Sat, 14 Aug 2010 20:59:27 GMT
View Forum Message <> Reply to Message

Hello Jeremy

So more than a new class would it be an improvement to existing HttpClient?

---

Subject: Re: MultipartForm Class for use w/HttpClient
Posted by jeremy_c on Sun, 15 Aug 2010 01:52:48 GMT
View Forum Message <> Reply to Message

I was originally thinking of doing that, however the existing HttpClient doesn't provide a way to

generate the url encoded post data either. Right now that's all done externally. For example, say I want to submit a form with name, email and password. Currently you would:

```
String postData = "name=" + UrlEncode(name) + "&email=" + UrlEncode(email) + "&password=" + UrlEncode(password);

HttpClient http("http://localhost/post.cgi");
http.Post(postData);
http.Execute();
```

So, I kept w/the same type of synario:

```
MultiformPost form;
form.Set("name", name);
form.Set("email", email);
form.Set("password", password);

HttpClient http("http://localhost/post.cgi");
http.Post(form.ToString());
http.ContentType(form.GetContentType());
http.Execute();
```

It's just that a Multipart Form is a more complex post structure than a url encoded post structure. Thus a helper class is just about essential. I did do multipart form uploads before w/o my helper class, but the helper class is a big help

Jeremy

---