
Subject: Keep-Alive in HttpRequest

Posted by [Zbych](#) on Sun, 20 Oct 2013 10:02:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

Can we add KeepAlive function to HttpRequest to change content of "connection" header?

Inet.h:

line 342:

```
bool keep_alive;
```

Line 459:

```
HttpRequest& KeepAlive(bool ka = true)          { keep_alive = ka; return *this;}
```

Http.cpp:

```
void HttpRequest::Init()
```

```
{
    port = 0;
    proxy_port = 0;
    ssl_proxy_port = 0;
    max_header_size = 1000000;
    max_content_size = 10000000;
    max_redirects = 10;
    max_retries = 3;
    force_digest = false;
    std_headers = true;
    keep_alive = false; //<-----
    hasurlvar = false;
    method = METHOD_GET;
    phase = BEGIN;
    redirect_count = 0;
    retry_count = 0;
    gzip = false;
    WhenContent = callback(this, &HttpRequest::ContentOut);
    chunk = 4096;
    timeout = 120000;
    ssl = false;
}
```

```
void HttpRequest::StartRequest()
```

```
{
    StartPhase(REQUEST);
    count = 0;
    String ctype = contenttype;
    if((method == METHOD_POST || method == METHOD_PUT) && !IsNull(ctype))
        ctype = "application/x-www-form-urlencoded";
}
```

```

static const char *smethod[] = {
    "GET", "POST", "HEAD", "PUT", "DELETE", "TRACE", "OPTIONS", "CONNECT", "PATCH",
};
ASSERT(method >= 0 && method <= 8);
data = Nvl(custom_method, smethod[method]);
data << ' ';
String host_port = host;
if(port)
    host_port << ':' << port;
String url;
url << "http://" << host_port << Nvl(path, "/");
if(!IsNull(proxy_host) && !ssl)
    data << url;
else {
    if(*path != '/')
        data << '/';
    data << path;
}
data << " HTTP/1.1\r\n";
String pd = postdata;
if(!IsNull(multipart))
    pd << "--" << multipart << "--\r\n";
if(method == METHOD_GET || method == METHOD_HEAD)
    pd.Clear();
if(std_headers) {
    data << "URL: " << url << "\r\n"
        << "Host: " << host_port << "\r\n"
        << "Connection: " << (keep_alive ? "keep-alive\r\n" : "close\r\n") //<-----
        << "Accept: " << Nvl(accept, "*/") << "\r\n"
        << "Accept-Encoding: gzip\r\n"
        << "User-Agent: " << Nvl(agent, "U++ HTTP request") << "\r\n";
    if(pd.GetCount() || method == METHOD_POST || method == METHOD_PUT)
        data << "Content-Length: " << pd.GetCount() << "\r\n";
    if(ctype.GetCount())
        data << "Content-Type: " << ctype << "\r\n";
}
VectorMap<String, Tuple2<String, int> > cms;
for(int i = 0; i < cookies.GetCount(); i++) {
    const HttpCookie& c = cookies[i];
    if(host.EndsWith(c.domain) && path.StartsWith(c.path)) {
        Tuple2<String, int>& m = cms.GetAdd(c.id, MakeTuple(String(), -1));
        if(c.path.GetLength() > m.b) {
            m.a = c.value;
            m.b = c.path.GetLength();
        }
    }
}
String cs;

```

```

for(int i = 0; i < cms.GetCount(); i++) {
    if(i)
        cs << "; ";
    cs << cms.GetKey(i) << '=' << cms[i].a;
}
if(cs.GetCount())
    data << "Cookie: " << cs << "\r\n";
if(!IsNull(proxy_host) && !IsNull(proxy_username))
    data << "Proxy-Authorization: Basic " << Base64Encode(proxy_username + ':' +
proxy_password) << "\r\n";
if(!IsNull(digest))
    data << "Authorization: Digest " << digest << "\r\n";
else
if(!force_digest && (!IsNull(username) || !IsNull(password)))
    data << "Authorization: Basic " << Base64Encode(username + ":" + password) << "\r\n";
data << request_headers;
LLOG("HTTP REQUEST " << host << ":" << port);
LLOG("HTTP request:\n" << data);
data << "\r\n" << pd;
LLOGB("HTTP request body:\n" << pd);
}

```
