Subject: Multi-UI REST applications with U++
Posted by lectus on Thu, 25 Jul 2013 20:07:02 GMT
View Forum Message <> Reply to Message

Hello!

Recently I've been studying more of the web stuff so I got a few ideas.

Correctly me if I'm wrong but it should be possible to write a REST API using Skylark, returning JSON for HTTP requests and then write a GUI client with HttpRequest class and also a web UI (Skylark).

This makes for a very modular architecture. Quite interesting for large applications. We could have a bulky server that does all the processing and thin clients powered by U++ GUIs.

Is this correct or am I dreaming?

Subject: Re: Multi-UI REST applications with U++
Posted by dolik.rce on Sat, 03 Aug 2013 06:59:27 GMT
View Forum Message <> Reply to Message

I think this idea is quite possible to implement. Actually pretty much everything is possible with U++

For bigger apps, it could be quite a lot of work, depending on how much processing would be done on server. If you just used the server for some CPU heavy processing, it might require only few specialized methods, but if you want to use it for data sharing and synchronization, than you'd probably have to write specialized code for each method that touches the data both on server and on client, effectively writing twice as much code Of course, in some cases this might be the right solution, it really depends on your needs.

Best regards,
Honza

Subject: Re: Multi-UI REST applications with U++
Posted by NilaT on Fri, 12 Feb 2016 10:07:08 GMT
View Forum Message <> Reply to Message

Hello and sorry for picking up this old Thread.
I just wanted to ask if there is a REST implementation in UPP already?
Because I searched here and this thread was the only "useful" info I found.

I need to program a webserver which uses REST communication and JSON.
For JSON I already found the Json class but for REST I'm not sure.

So, any information about this is very appreciated.
Thanks in advance.

PS: Or maybe lectus, have you done an implementation by yourself and want to share it?

Subject: Re: Multi-UI REST applications with U++
Posted by dolik.rce on Fri, 12 Feb 2016 11:45:10 GMT
View Forum Message <> Reply to Message

Hi NilaT,

NilaT wrote on Fri, 12 February 2016 11:07Hello and sorry for picking up this old Thread.
I just wanted to ask if there is a REST implementation in UPP already?
Because I searched here and this thread was the only "useful" info I found.
Don't be afraid to create a new thread if you don't find what you need.

NilaT wrote on Fri, 12 February 2016 11:07I need to program a webserver which uses REST communication and JSON.
For JSON I already found the Json class but for REST I'm not sure.
There is not much to implement for this in U++. REST is not a technology, it is just a "style" of API. It actually doesn't even have to be implemented using HTTP, the only constraints that REST imposses on API are client-server architecture and statelessness.

I believe Skylark already supports all the necessary HTTP methods, so if you need to implement REST API on your webserver, you just need to write the skylark methods in stateless way, that is they always get all the necessary input data within the request. SkylarkPack might help you a bit with the organization of the code, see chapter 12 of skylark tutorial.

Best regards,
Honza

Subject: Re: Multi-UI REST applications with U++
Posted by mirek on Fri, 12 Feb 2016 11:48:23 GMT
View Forum Message <> Reply to Message

NilaT wrote on Fri, 12 February 2016 11:07Hello and sorry for picking up this old Thread.
I just wanted to ask if there is a REST implementation in UPP already?
Because I searched here and this thread was the only "useful" info I found.

I need to program a webserver which uses REST communication and JSON.
For JSON I already found the Json class but for REST I'm not sure.

So, any information about this is very appreciated.
Thanks in advance.

PS: Or maybe lectus, have you done an implementation by yourself and want to share it?

There is nothing magical about REST. It is just regular HTTP server, easy to do without any special support. Skeleton code can look e.g. like this:

```
void RESTProcess(TcpSocket& r)
{
 HttpHeader hdr;
 if(hdr.Read(r)) {
  VectorMap<String, String> urlvar;

  String uri = hdr.GetURI();
  int q = uri.Find('?');
   if(q >= 0) {
    ParseUrlVars(urlvar, ~uri + q + 1);
    uri.Trim(q);
   }

   String req, n;
   q = uri.Find('/');
   if(q >= 0) {
    n = uri.Mid(q + 1);
    req = Filter(n, CharFilterAlphaToLower);
   }

  if(hdr.GetMethod() == "GET") {
   if(req == "order") {
   }
  }
  if(hdr.GetMethod() == "POST") {
   if(req == "order") {
   }
  }
  if(hdr.GetMethod() == "PUT") {
   if(req == "order") {
   }
  }
  if(hdr.GetMethod() == "DELETE") {
   if(req == "order") {
   }
  }
 }
 HttpResponse(r, false, 400, "INVALID REQUEST");
}
```

```
TcpSocket server;

void Server()
{
 for(;;) {
  TcpSocket socket;
  LOG("Waiting...");
  bool b = socket.Accept(server);
  if(b) RESTProcess(socket);
 }
}

CONSOLE_APP_MAIN
{
 if(!server.Listen(4000, 10)) {
  LOG("Cannot open server port for listening\r\n");
  return;
 }
}
```