Subject: Re: Is it possible to use Core/Rpc in non blocking mode?
Posted by steffen on Tue, 23 Apr 2013 16:57:16 GMT
View Forum Message <> Reply to Message

Hi nlneilson,

It is not the protocol or data transfers that is causing me problems. It is general socket handling.

My application communicates over GSM a modem, which are automaticly behind NAT routers from all ISP's I have used.

To let the server send notifications to the client, the client either has to poll the server continuously, raising the traffic cost. Or it can use a technique called long polling, where the server delays the response until there is something to send or a timeout occurs.
In most cases we have found that a 4 minute timeout keeps the modems online, at a minimal traffic cost. The response is simply a 204 No data available.

Now with the current JsonRpcRequest the json requests work fine, but it blocks the calling thread for up to 4 minutes.

Simultaneously I have another JsonRpcRequest handling the communication going from the client to the server. This request gets a response immediately, but in case of errors there could also be some sort of timeout here. So it also gets a thread of its own.

So instead of the Execute method used today I would rather if I could send the requests to the SocketWaitEvent and loop through a list to handle the sockets, just like the GuiWebCrawler example does it.

Another thing is that the json-rpc specifies array requests, where multiple requests can be put into a json array and send in the same transmission. This adds more payload to the HTTP/TCP/IP packets, so it is not always sending 90% headers.
This would require an outgoing queue for the json requests.

I have looked into the RpcRequest::Execute method and I think I could make a derived class with a couple of methods to use it in non blocking mode.

At last I think if first I have the non blocking mode running, it will not be so big a step to let it run over a Html5 websocket connection. Without knowing the details, I think it is just a bidirectional socket connection.

Regards,
Steffen