

Forum:[Hack Forums](#)

Thread: [Ion Server - Server & Client libraries with Plugins](#)

phrootoo - January 8, 2015, 8:42 am

When I first started programming, one of the concepts that took me a long time to understand was networking. To me, networking was something that was so foreign it just didn't seem to want to sink.

I'm now around 5 years in to my programming journey, and whilst there's still a lot more to learn, I feel like there's lots of things that can be made easier.

Networking is a hassle, that's the bottom line. For beginners, it's tedious, unforgiving, and a bloody nightmare to debug. This is where Ion Server comes in.

Ion Server is a fully functional TCP server that features a rich plugin interface, stable and concise networking, and a great ability for expansion. Ion Server comes with a client based library that you can use to link up to your applications, so you can interface with an Ion Server.

Let me explain the concepts:

Communication via the Ion Network is achieved through so called 'Requests'. A request consists of the following information:

method - The remote method that we are wanting to call. For example: When a user logs in, they should call the 'login' method.

parameters - A request has a unlimited amount of parameters. In the login example, there would be a parameter for the username and one for the password.

The requests work exactly the same on both the client and the server, so there's consistency within the project.

All networking is handled by the library itself. Infact, all you need to do is subscribe to a 'Request' event, connect to the server, then you're good to go. Once the events are triggered, you'll be able to process it on your end.

Plugins

Customizability is a must, since your applications will all be entirely different. Whilst there is no access to the source code for the server, this is where the plugins come in to play. Plugins are without a doubt the most excited feature I have created. Plugins allow you to add functionality to the server. Without the plugins, the server technically wouldn't do anything (Other than accept connections). Plugins have the following events:

OnConnected

OnConnected is called when a new client connects to the system. From this, the plugins have direct access to the newly connected client.

OnUpdate

OnUpdate is called by a separate thread and gives you access to the Ion core, which in turn gives you access to the client pool. This is where you perform your active tasks.

OnRequest

OnRequest is called when a client sends a request to the server. Every plugin will be sent a copy of each request. From this, you can access both the sender information and the request information. This is where you will process your main requests.

OnDisconnected

OnDisconnected is called when a client disconnects from the server. This will give you access to the client, but will not allow you to send information to the client.

Whilst this is a quick overview of the plugins, this should hopefully entice you to read onwards.

Plugin Store

The plugin store is a feature which allows you to access community plugins. I currently have my own plugins added there, and will continue to add the plugins free of charge. Any developers wanting to join the party I will be more than happy to let join on.

Screenshots

This is the code needed to implement it on the client side

Login

Dynamic setting management for plugins

So, anybody interested?

Only3 - January 8, 2015, 9:01 am

we still waiting <https://hackforums.net/showthread.php?tid=4581216>

phrootoo - January 8, 2015, 9:03 am

(Jan 8, 2015 10:01 AM)Only3 Wrote: [\[CITE\]](#) we still waiting
<https://hackforums.net/showthread.php?tid=4581216>

Same code behind it, different approach :-)

Only3 - January 8, 2015, 9:11 am

(Jan 8, 2015 10:03 AM)phrootoo Wrote: [\[CITE\]](#) Same code behind it, different approach :-)
? So, anybody interested? > we still waiting ^^

Skz. - April 14, 2015, 9:45 am

Some news about this awesome work ? :p

Snoweyes - April 14, 2015, 11:43 am

Alright looks like a cool project!, it may pretty useful for exchanging some data quick.

May i ask why you using the full namespace everytime you use something from your framework?
Why aren't you just importing the namespace?