

Re: Calling Web Service method and GPRS

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.compactframework/2007-11/msg002>

- *From:* im.djmatty@xxxxxxxxxxxxxxx
 - *Date:* Thu, 15 Nov 2007 02:11:01 -0800 (PST)
-

Hi Simon

Thanks for the info, our push mechanism uses the same idea as yours, in that it will only work when the network is configured to provide the devices with routable ip addresses. I did start to look at using STUN or some other NAT traversal method to get the packets to the device, but haven't got too far with that at present.

I did some research into AUTD and how it works, it seems that it makes an HTTP request to the server with a long timeout (the default i think is 20 mins), and if anything on the exchange server changes for that user the request returns with a response containing which mail folders have changed, then activesync is kicked off and syncs only those folders... if the request times out the device sends another request. The push mechanism doesn't actually need to use activesync, it's just a notification that the device needs to do a sync.

It's an interesting concept as it does mean that the devices don't need any of that expensive network configuration, e.g. leased lines, ipsec tunnels, private ip address pools etc, and it does work over the public apns.

Matt

On Nov 14, 9:42 pm, Simon Hart [MVP] <srhart...@xxxxxxxxxx> wrote:

Hi, Matt, Scott,

When you say develop a server on your device to listen for push messages, I am interested to know how you get around the issues of IP addressing, and the mobile networks use of NAT and PAT on their standard mobile internet connections?

This is an interesting topic and very real world. My client has a private APN which works via a leased line between the GGSN and the CPN (Converged Packet Network) to our clients LAN, so when our devices connect to the private

Re: Calling Web Service method and GPRS

APN, the GGSN then releases a dynamic IP address from a pool of pre-configured reusable addresses. The GGSN talks directly to the CPN in fact it bridges the gap between the GPRS network and the CPN. The CPN then connects via a leased line channel to the corporate LAN. You can use IPSec here instead or use a combination of both (failover).

Using this infrastructure allows us to have a two communication from and to the devices in the field from the back office running on the LAN.

Our client chose to use dynamic IP addresses because static addresses per SIM are very expensive. Of course this lead to an interesting problem – how do we track the IP address. The solution is simple, the device monitors the GPRS network adapter and when the IP address changes the device sends a message to one of our back end gateway servers (any one of 16 scaled servers) running on the LAN. The only overhead here is the size of the message that needs to be sent on IP address change. But we find in reality the IP's don't change that often.

Or are you proposing that the device uses a private APN or VPN connection and as such each device has a unique IP address and can be addressed from another machine that wants to talk to the listening process on the device?

Yes. As you know using a public APN won't work because of the NAT(Network Address Translation) that has to happen. As you mentioned we have had working a connection to a public APN (doesn't matter which) then connect via a VPN. This works but is not very reliable. Your code will need to be very robust for this.

The "direct push" technology that is implemented in MSFT and Exchange SP2 gets around a lot of these issues by having the connection initiated by the device, and just waiting for a response.

This solution also uses ActiveSync. We've not really looked into the direct push via exchange in great detail, but I don't like the look of it.

Simon.

Matt

On Nov 12, 2:00 pm, Simon Hart [MVP] <srhart...@xxxxxxxxxx> wrote:

Hi Scott,

Re: Calling Web Service method and GPRS

It seems that connecting when required, sending the data, then disconnecting is ideal for when the client is generating data and sending it to the server.

However, if the server is generating data and sending it to the client (for example push email), it seems to me that this approach basically becomes polling the server, and it's much harder to know when to connect. In particular, if your goal is to receive messages from the server as quickly as possible, you're stuck with frequent polling (which uses more power), or longer delays.

You are right about this if you haven't implemented true push technology.

However, if you have implemented true push technology, ie: a server process listens on the device waiting for the back office to connect to it, once connected, pushes out the message then disconnects, then you will remove the need for polling.

AUTD takes advantage of the GPRS radio waking up the device, and so it can basically initiate the connection, then go into suspend mode until and unless notifications are available.

This is true in a true push type environment as well. Of course you would need to connect to your APN via GPRS/UMTS when your application starts so that the back office can see the device.

Re: Calling Web Service method and GPRS

Of course, reality is more complicated (for example keepalive messages must be sent to keep the connection up), but I think the basic tradeoffs remain.

This will be a much more expensive approach to using true push technology.

What are your thoughts on a system in which a Windows Mobile client must receive messages from a server as quickly as possible?

As I said above, you will always need a GPRS connection to your APN, but with true push technology the only data going up and down the wire is the data you want to send receive. No polling or keep-alives or checking for data needs to occur.

The overhead of implementing true push is developing a server to listen on the device.

—
Simon Hart
Visual Developer – Device Application Development
MVP <http://simonhart.blogspot.com>