Angus McLeod <angus.p.mcleod@gmail.com>

ObjectiveDDP -> iPhone App talking directly to meteor

7 messages

Martin Ceperley <martin@knote.com> Thu, Dec 5, 2013 at 6:13 PM
To: "yan.make" <yan.make@gmail.com>, Amol Sarva <a@sarva.co>, Lars Kluge <l@larskluge.com>, Long Liangyou <forallandiyou@gmail.com>, Angus McLeod <angus.p.mcleod@gmail.com>

I've discovered a way for the iOS app to directly talk to the meteor server the same way the browser clients do, through DDP websockets. There's a pretty decent Obj-C library someone is working on called ObjectiveDDP - https://github.com/boundsj/ObjectiveDDP

This is pretty exciting as it can allow for real-time data updates sent right to the iPhone app. The app just subscribes to the same queries that the server publishes, and can call methods published in the server. I've already easily implemented Login with it, so the app will use the same password encryption scheme (SRP). This could supplement the app's MongoDB code or possibly replace it in the future, and reduce the need for a full-blown API.

For now the code is in the ddp branch of knotable-ios. The libraries are imported with CocoaPod. With CocoaPods the .xcworkspace file has to be opened rather than the .xcodeproj.

By the way it looks like there is no authentication enforced for DDP, I'm sure you guys are aware of this.

-Martin

Angus McLeod <angus.p.mcleod@gmail.com> Thu, Dec 5, 2013 at 6:34 PM
To: Martin Ceperley <martin@knote.com>

Well done! This sounds really promising.

[Quoted text hidden]

Long Liangyou <forallandiyou@gmail.com> Thu, Dec 5, 2013 at 8:58 PM
To: Martin Ceperley <martin@knote.com>
Cc: "yan.make" <yan.make@gmail.com>, Amol Sarva <a@sarva.co>, Lars Kluge <l@larskluge.com>, Angus McLeod <angus.p.mcleod@gmail.com>

Great!

The publish code of meteor server is here: https://github.com/Knotable/knotable/blob/master/server/publication.coffee
There are lots of method published: https://github.com/Knotable/knotable/tree/master/server/methods

Feel free to contact me if you want to know more about data published in meteor server.

Thanks.
Long

[Quoted text hidden]

yan.make <yan.make@gmail.com> Thu, Dec 5, 2013 at 10:15 PM
To: Martin Ceperley <martin@knote.com>, Amol Sarva <a@sarva.co>, Lars Kluge <l@larskluge.com>, Long Liangyou <forallandiyou@gmail.com>, Angus McLeod <angus.p.mcleod@gmail.com>

Hi Martin:

Meteor's real time engine does lots of cache in RAM, this could be an issue for phone, need to test it.

I am worried about another behaviour, meteor polls data from db and cache it, update cache every 10s or after write operations. This may cause big data transfer between the phone and server. I am not clear about detailed mechanism.

So pls play with it and figure it out.

At least we can pull out SRP implemention from it for now.

A simply sync solution I guess is adding an API in the server to help, because server's current behaviour is "polls mongodb, diff data with in the RAM, generate changes and notify affected clients", we need a way to add the phone as a normal client in the server.

As the production in the future, I perfer the normal API way, this mechanism maybe heavy for the phone and cause banlance problems with cellular network.

Thanks.

yan.make

发件人： Martin Ceperley
发送时间： 2013-12-06 07:13:53
收件人： yan.make; Amol Sarva; Lars Kluge; Long Liangyou; Angus McLeod
抄送：
主题： ObjectiveDDP -> iPhone App talking directly to meteor

I've discovered a way for the iOS app to directly talk to the meteor server the same way the browser clients do, through DDP websockets. There's a pretty decent Obj-C library someone is working on called ObjectiveDDP - https://github.com/boundsj/ObjectiveDDP

This is pretty exciting as it can allow for real-time data updates sent right to the iPhone app. The app just subscribes to the same queries that the server publishes, and can call methods published in the server. I've already easily implemented Login with it, so the app will use the same password encryption scheme (SRP). This could supplement the app's MongoDB code or possibly replace it in the future, and reduce the need for a full-blown API.

For now the code is in the ddp branch of knotable-ios. The libraries are imported with CocoaPod. With CocoaPods the .xcworkspace file has to be opened rather than the .xcodeproj.

By the way it looks like there is no authentication enforced for DDP, I'm sure you guys are aware of this.

-Martin

Lars Kluge <l@larskluge.com> Fri, Dec 6, 2013 at 12:41 AM
To: Martin Ceperley <martin@knote.com>
Cc: "yan.make" <yan.make@gmail.com>, Amol Sarva <a@sarva.co>, Long Liangyou <forallandiyou@gmail.com>, Angus McLeod <angus.p.mcleod@gmail.com>

Martin, that's a cool way for now. We should mark the methods you're calling in the Meteor project somehow, so the engineers in this app are aware that a change there might impact the iOS application.

On Thu, Dec 5, 2013 at 6:13 PM, Martin Ceperley <martin@knote.com> wrote:
[Quoted text hidden]

Martin Ceperley <martin@knote.com> Fri, Dec 6, 2013 at 9:27 AM
To: "yan.make" <yan.make@gmail.com>
Cc: Amol Sarva <a@sarva.co>, Lars Kluge <l@larskluge.com>, Long Liangyou <forallandiyou@gmail.com>, Angus McLeod <angus.p.mcleod@gmail.com>

Yes I agree Yan, in the future a good JSON REST API for this app and other knotable apps is the goal, probably with standard auth like Oauth2 for authentication.

I will try to pull out the SRP Login stuff from the lib so we can directly compare the hashed password in Mongo.

Using websockets would mainly help us with real-time updates of the current screen, to avoid frequently polling the DB for changes. A socket may not stay open long on a spotty Cellular connection, maybe it would only be activated on wi-fi. Minimizing the amount of redundant data sent across the wire is crucial.

Right now in my build the Websockets only last for 30 seconds, and then they restart, and the server sends you the data you are subscribed to over and over. Examining the web-app with Chrome it looks like the websockets stay open for much longer, several minutes. I don't know if this is caused by the iOS library or the server.

Long, do you know if there is any websocket timeout on the server? Or if the client sends a heartbeat or something to stay open?

-Martin
[Quoted text hidden]

Lars Kluge <l@larskluge.com> Fri, Dec 6, 2013 at 11:01 AM
To: Martin Ceperley <martin@knote.com>
Cc: Amol Sarva <a@sarva.co>, "yan.make" <yan.make@gmail.com>, Long Liangyou <forallandiyou@gmail.com>, Angus McLeod <angus.p.mcleod@gmail.com>

Meteor sends a heartbeat.
[Quoted text hidden]