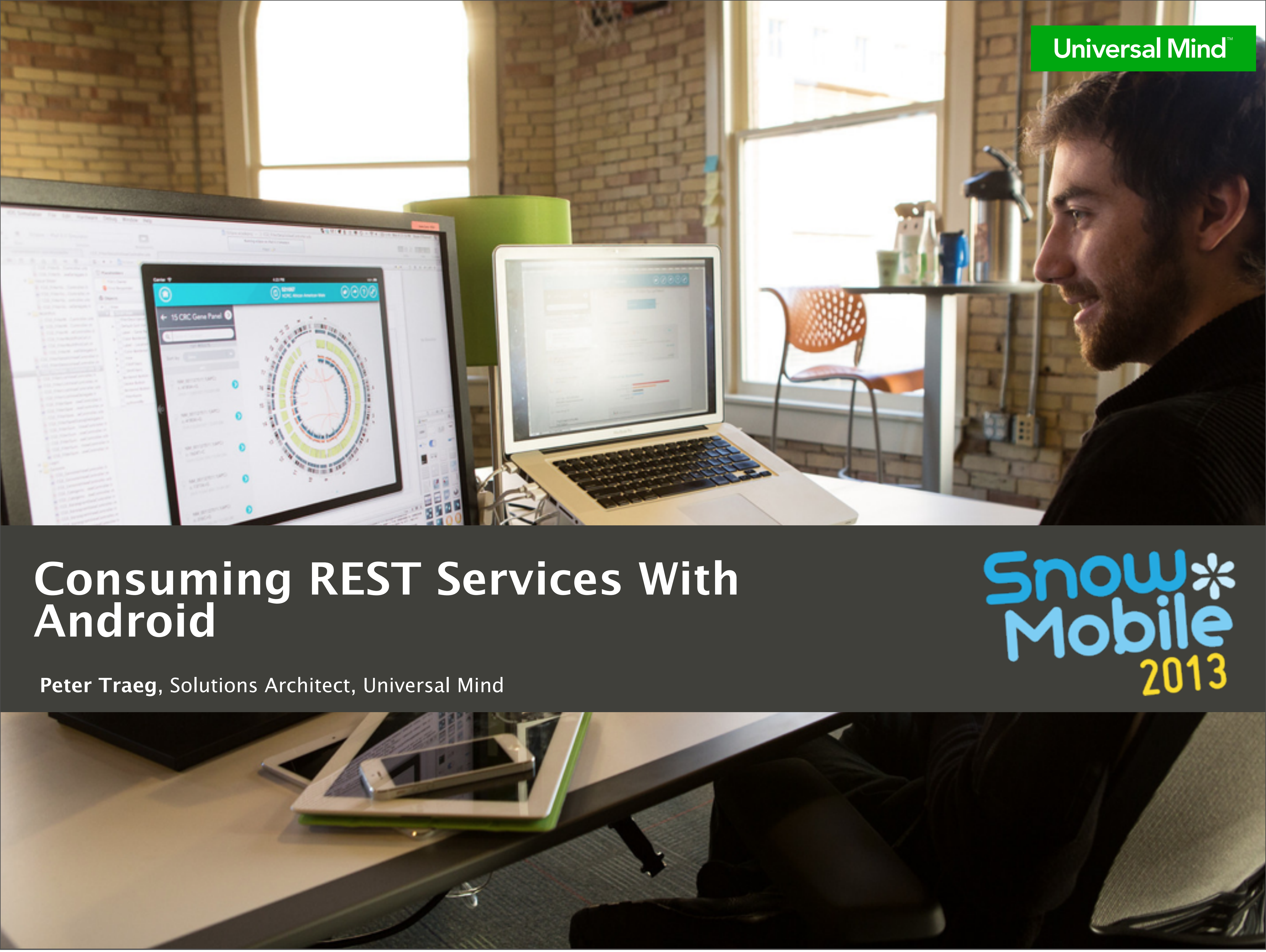

Universal Mind™

Consuming REST Services With Android

Peter Traeg, Solutions Architect, Universal Mind

Snow Mobile 2013

Universal Mind™

About Me:

Peter Traeg

Solutions Architect, Universal Mind

iOS, Android, and HTML 5 Developer

peter.traeg@universalmind.com

@ptraeg

Snow Mobile 2013

dish NETWORK T · ·Mobile· FedEx HOUGHTON MIFFLIN HARCOURT SONY MUSIC

Amway PEARSON Education Kodak Gallery NASDAQ ORACLE

TODO 1 ENERNOC Principal Adobe CISCO

RIM SCHOLASTIC verizon Johnson Controls NetApp

ADP AOL

Universal Mind™

cde ChannelWeb

Children's Hospital Boston CONCERTO SOFTWARE CVS daltile Dealer Tire

Dorado eDiets.com YOUR DIET. YOUR WAY. eshots Interactive Event Marketing ESRI EXAM BUILDER

hp iCongo INTERSCOPE GEFFEN A&M RECORDS IHG InterContinental Hotels Group interwise

intuit. JIBJAB Johnson & Johnson Jonas JWT

Agenda:

Demonstration of how to consume REST based services in a native Android application.

- Application Demonstration
- An Example Service
- Structuring your Application
- Using Goggle GSON
- Performance Considerations
- Common Pitfalls

Application Demonstration

Universal Mind™

REST Calls to:

Action	HTTP verb
Create	POST
Read	GET
Update	PUT
Delete	DELETE

Snow Mobile 2013

The REST Service for our demo

MongoLab features:

- Hosted MongoDB Instance
- Free Accounts
- REST Based API
- Simple authentication - good for prototypes

An example REST call

HTTP GET to:

```
https://api.mongolab.com/api/1/
databases/mobilejournal/collections/
entries?apiKey=<your-api-key>
```

Example GET reponse:

```
[
   {
      "_id":{
         "$oid":"51131229e4b068616ee6f2d4"
      },
      "updatedDate":{
         "$date":"2013-02-07T02:34:27.000Z"
      },
      "notes":"My test2",
      "title":"New list entry for Pete",
      "categoryId":0
   },
  {
      "_id":{
         "$oid":"510eae7fe4b040e25c29a865"
      },
      "updatedDate":{
         "$date":"2013-02-06T04:07:08.000Z"
      },
      "notes":"More testing.  Here is more text.  ",
      "title":"Testing stuff ",
      "categoryId":0
   }
]
```

Code Tour

Getting the list of entries

Universal Mind™

Getting the list of journal entries:

1. AsyncTask - to fetch the content in a background thread
2. HttpURLConnection to make the network call
3. Google GSON to parse the JSON into an ArrayList of Java objects
4. android.widget.BaseAdapter to bind the ArrayList to an Android ListView

Snow Mobile 2013

google-gson
A Java library to convert JSON to Java objects and vice-versa

Key Features:

1. Serialization / Deserialization
2. Supports nested classes
3. Custom Serializaton / Deserialization via TypeAdapters
4. Versioned fields support - include/exclude fields based on a version number passed to GSON
5. Field naming support via @SerializedName

Type Adapters:

Needed in our example to support MongoDB OIDs and DataTime values.

Used to adjust JSON data to your Java objects.

Consider for non-standard JSON types like Date objects.

Also use if your Java object does not support a no-arguments constructor.

Code Tour

POSTing and PUTing data

Performance Considerations

Techniques demonstrated here are for moderate size result sets

For large results sets consider using GSON Streaming so as to not load the entire object graph into the parser at once.

It's also possible to use a JsonReader and the JsonObject to walk the JSON response manually if you prefer. Look at the getAsJsonObject() and getAsJsonArray() methods on JsonObject.

Gotchas - Device rotation

Activities restart on device orientation change. You loose the request if someone rotates a device while your HTTP request is pending....

1. Prevent device orientation changes while processing requests: `activity.setRequestedOrientation()`

2. Use fragments. Fragments can be retained on orientation changes - `setRetainInstance()` on the fragment

3. Use an AsyncTaskLoader which supports activity restarts, but won't survive if the activity is pushed into the back stack.

4. Move HTTP requests to an Android service - eg: IntentService

Gotchas - GSON in the classpath

Some HTC devices have placed `com.google.gson` in the public class path - this is an old version of GSON and will conflict with newer versions. Use jarjar to assign another namespace to the GSON library to avoid conflict.

Gotchas - HTTP Keep Alive

HTTP keep-alive with HttpURLConnection is broken on some devices. Closed connections are sometimes returned from the connection pool.

- Consider shutting off keep alive via:
```
System.setProperty("http.keepAlive", "false");
```
- Explicitly close connections by passing HTTP headers:
```
urlConnection.setRequestProperty("Connection", "close");
```
- Google recommends against going back to the older HttpClient even though that is stable.

Development Tip - Charles Proxy

Charles proxy is useful in debugging your REST calls. Here's the process for setting up the emulator to use it:

- In the Eclipse run/launch configuration "target" tab set additional emulator command line options to:
  `-http-proxy http://<ip-addr-of-charles-machine>:8888`

- If connecting over SSL the Android AVD must trust the Charles SSL cert:
  - Open the Android browser and hit http://www.charlesproxy.com/charles.crt
  - Accept the cert. Might be necessary for you to set a PIN on the AVD to store the certificate.

Universal Mind™

Thank You !

peter.traeg@universalmind.com

Snow Mobile 2013