Preliminares a la comunicación Android con GLAMPs - N

Créditos

#### Abstract.

Lo directo, cuando el desarrollador Android quiere acceder a la red Elgg es usar oAuth [1] y Account [2]. Una vez que el cliente puede manejar los tokens que el servidor reparte [2.1] puede convertir sus clases Java nativas a JSON [3] y enviar peticiones síncronas [4.1] dentro del Adaptador de sincronización [5] o asíncronas [4.2] en cualquier punto de la aplicación. El **plugin webservices** [7] junto con algún otro plugin que extienda métodos responderán a esas peticiones enviando arrays de objetos JSON [8].

<sup>[1]</sup> Write your own Android Authenticator: http://blog.udinic.com/2013/04/24/write-your-own-androidauthenticator/

<sup>[2]</sup> Android.Account dev pages: https://developer.android.com/reference/android/accounts/Account.html

<sup>[2.1]</sup> Never your asked? http://stackoverflow.com/questions/7030694/why-do-access-tokens-expire [2.2] Enhanced plugin ApiAdmi: https://elgg.org/plugins/1105480

<sup>[3]</sup> A Java serialization/deserialization library that can convert Java Objects into JSON and back. https://github.com/google/gson

<sup>[4.1]</sup> Go salmon: http://stackoverflow.com/questions/16904741/can-i-do-a-synchronous-request-withvolley

<sup>[4.2]</sup> Asynchronous HTTP Requests in Android Using Volley <a href="http://arnab.ch/blog/2013/08/asynchronous-">http://arnab.ch/blog/2013/08/asynchronous-</a> http-requests-in-android-using-volley/

<sup>[5]</sup> Write your own Android Sync Adapter: http://blog.udinic.com/2013/07/24/write-your-own-androidsync-adapter/

<sup>[7]</sup> Docs on Webservices plugin: http://learn.elgq.org/es/stable/quides/web-services.html

<sup>[8]</sup> Ver lista de métodos que expone un sitio elgg visitando su url: /services/api/rest/json/? method=system.api.list

<sup>1</sup> No tiene valor técnico, es pieza artística. – nadievstodos.tumblr.com – Verano 2016 – Iberia

#### Forward 1.

De acuerdo, en este punto, la **conexión de Adaptadores** [1] que c**arguen cursores a listas**, quizás alguna con *swang* a borbotones [2] que tenga opciones en botonera cargadas al vuelo completa una posible vía de interacción desde Android.

- [1] Crear ListView con cursor: <a href="http://www.sgoliver.net/foro/viewtopic.php?t=233">http://www.sgoliver.net/foro/viewtopic.php?t=233</a>
- [2] A swipe menu for ListView: <a href="https://github.com/baoyongzhang/SwipeMenuListView">https://github.com/baoyongzhang/SwipeMenuListView</a>

#### Forward 2.

Pero, aunque lo anterior tiene todo el jugo y juego: el título de este micro ensayo sugiere algo, digamos, más allá.

Ensayando, se sabe que no se habla libre de error sino liberando el error del hablar, tomando notas y tal  $(\underline{\dots})$ , sin querer interrumpir en el buen progreso de lo no ensayado sino planeado, etc.

Decimos: MicroEnsayo Más Elgg-UX-browsers y, porque todas y cada una de las palabras del título han sido escogidas atendiendo a criterios de peso específico semántico, tenemos gran parte del microensayo relatado.

(continuará) <a href="http://aleph1888.github.io/ataraxia">http://aleph1888.github.io/ataraxia</a> archive/

### Schema

# ELGG (ataraxia.server)

MySQL (ataraxia.db)

- Engine
- Modules

LAMP SERVER HTTPS, HTTP

- (ataraxia.modules)
- Webservices: api key api\_auth\_admin: token
- api\_points

ANDROID (ataraxia.app)

SQLite (ataraxia.db)

## (ataraxia.database)

- Contracts
- Provider

- (ataraxia.sync)
- Adapter
- Directory (Java.class [GSON] Volley.Requests)

## (ataraxia.system)

- Network
- Notifications
- DateHelper

# (ataraxia.auth)

- · Auth (signer): api key
- User: api key + token
- · Entities: api key + token

### (ataraxia.api)

- Launcher
- Async/Sync request

## (ataraxia.ui)

- Splash
- Settings
- Signer (Auth)
- Profile (Form + widgets)
- Entities (SwipeList)