# Client server communications

Communications between client and server is done by the use of bson objects.

There are two kind of objects: the TOKEN object and the DATA.

TOKEN is used to request to the server a news.

While the DATA is the real news from the server.

## Retrieving the TOKEN

The first thing the application shall do is ask for a bson token to be sent to the news provider.

Tokens are retrieved using the following jni function:

public static native byte[] getToken(String imei,String cks,String baseDir);

Parameters are :

* imei: smartphone identifier.
* cks: the last checksum result returned by the serialize jni function, 0 if this is the first time application calls getToken.
* baseDir: path to the applications owned directory, where to store bson translated objects. Usually it is a path like: /data/data/<packageName>/

Once bson token has been obtained it can be send to the server as the below example :

InputStream is = socket.getInputStream();

BufferedOutputStream out = new BufferedOutputStream(socket.getOutputStream());

/\* write to the server \*/

out.write(**token**);

out.flush();

## Translate BSON to HASH

Note: the serialize return an HashMap<String,String> with name,value tuples. The most important part is HASH\_FIELD\_CHECKSUM and HASH\_FIELD\_PATH. A valid news has CHECKSUM equal to 0 and a valid PATH to the decoded news payload.

## Bson organization server side

Because bson is just a binarized json, here is shown the input json:

DATA ==> {

"ts": "1059492039482",

"type": "1",

"sha1": "530ecd668c4aa089c880a87840394816ce35ee26",

"frag": "1",

"title": "terrible news",

"headline": "from Africa to Malaysia",

"content": "no way to avoid assassin bees",

"payload": "sfkjjo434hvn3o5gnovtnvt4ngnbfnbv43ebnvldjsf",

}

## Server settings

The application backend is hosted on 46.38.48.178

ssh benews@beesrv

user benews

pwd unknown

copy here the server directory : Benews/server

Inside Benews/server/ is present the python script which act as server (benews-srv.py) and a shell script used to start the server (server.sh)

Once started, the server keeps listening on the 8080 port for clients.

The object contains the following fields:

#define HASH\_FIELDS 7

#define HASH\_FIELD\_TYPE "type"

#define HASH\_FIELD\_CHECKSUM "checksum" note: this field contains the information of the success or fails of the reception and translation of the object and its payload, 0 means ok , otherwise fail.

#define HASH\_FIELD\_SHA1 "sha1"

#define HASH\_FIELD\_PATH "path"

#define HASH\_FIELD\_TITLE "title"

#define HASH\_FIELD\_DATE "date"

#define HASH\_FIELD\_HEADLINE "headline"

#define HASH\_FIELD\_CONTENT "content"

#define HASH\_FIELD\_FRAGMENT "frag"

#define HASH\_FIELD\_PAYLOAD "payload"