

Test conexión con red de publicidad externa

La idea este test es realizar una conexión a una red de publicidad, pasarle los parámetros y recibir la respuesta, para ello te proporciono una KEY para que puedas hacer peticiones. Se ha de realizar una clase/app que lea un fichero de texto (con un formato que estará definido más abajo, en formato json y con la petición/datos), lo interprete, haga una petición a la red y procese el resultado. (el resultado puede ser positivo [he recibido un anuncio], negativo [no he recibido anuncio], erróneo [se ha producido un error, y ¿qué error?], test [he recibido un anuncio de TEST] o Timeout [se ha sobrepasado el tiempo de espera]).

Como constantes tendrás:

- La key con la que se realiza la petición [Tappx_3134_TriviaCrack_BM_Android]
- Fichero de entrada (request.txt) [es un json que se ha de procesar, ver final del documento]

La documentación de la red está aquí:

<https://confluence.inner-active.com/display/DevWiki/Server+API+integration#ServerAPIintegration-ResponseHeaders>

Test:

- El código ha de estar en JAVA
- Habrá una clase inicial que se llamará *Launcher*:
 - Parámetros de entrada:
 - Ruta completa del fichero a procesar
 - Nº de procesos paralelos a ejecutar [1..n]
 - Timeout (máximo tiempo de espera hasta la respuesta)
 - El proceso ha de:
 - Leer el contenido del fichero (no hace falta controlar si existe o no existe, ya dará error, esta parte no es importante)
 - Realizar llamadas en paralelo (en este caso a la misma red tantas veces como se haya indicado en el launcher) y esperará a que alguna red devuelva un resultado o se produzca un timeout (forzar el timeout en la espera, por si se quedan colgados los subprocesos)
 - Habrá una clase que se llamará *InnerActive* que será la de la red que se va a implementar (al ser posible heredaré de una clase base llamada *Network*)
 - Parámetros:
 - JSON o Array o Clase con el contenido del Fichero
 - Timeout (máximo tiempo de espera a la respuesta de la red)
 - La clase "*Network*" o "*InnerActive*" deberá de generar la URL a la que se va a llamar, realizar la llamada y esperar la respuesta y esa respuesta/error/timeout deberá de ser informado al Launcher.
 - Finalmente el Launcher,
 - si tiene una respuesta positiva deberá de detener el resto de procesos y mostrar por pantalla el resultado
 - Si no hay respuesta positiva, deberá de esperar al timeout o que finalicen los procesos (lo que se produzca antes) y mostrar por pantalla un resumen de cada uno de los subprocesos (#1 Error XXXXX \n #2 Timeout XXXms, etc..)

Guia/Ayuda

El JSON de entrada (request) sigue el estándar openRTB que hay en el enlace que se encuentra a continuación [**NO TE MIRES EL DOCUMENTO, NO PIERDAS EL TIEMPO!!**], te adjunto el documento tan sólo para que puedas saber qué significa cada parámetro de la petición (id, at, allimps, tmax, imp, bidfloorcur, etc..)!

<https://github.com/openrtb/OpenRTB/blob/master/OpenRTB-API-Specification-Version-2-3-1-FINAL.pdf>

Y aquí tienes el JSON para la petición: (request.txt)

```
{
  "id": "58a32ea76badd",
  "at": 1,
  "allimps": 0,
  "test": 0,
  "tmax": 1200,
  "imp": [
    {
      "id": "c4ca4238a0b923820dcc509a6f75849b",
      "bidfloorcur": "USD",
      "secure": 0,
      "banner": {
        "w": 320,
        "h": 480,
        "pos": 1,
        "battr": [
          1,
          3,
          8,
          9,
          10,
          11
        ],
        "api": [
          3,
          5
        ],
        "id": "banner_1"
      },
      "tagid": "2456280990",
      "bidfloor": 0.1,
      "instl": 1,
    }
  ],
  "user": {
    "id": "fc7d629ba40c2abd40a5fe47941edf37",
    "yob": 1980,
    "gender": "M"
  },
  "app": {
    "id": "13377",
    "name": "TriviaCrack",
    "domain": "https://www.triviacrack.com/",
    "bundle": "com.etermax.preguntados.lite",
    "storeurl": "https://play.google.com/store/apps/details?id=com.etermax.preguntados.lite&hl=en",
    "cat": [
      "IAB9-7"
    ],
    "paid": 0
  },
  "device": {
    "lmt": 0,
    "geo": {
```

```

        "country": "NLD",
        "city": "amsterdam",
        "zip": "1011",
        "lat": 52.370998382568,
        "lon": 4.9040999412537,
        "type": 1,
        "regionfips104": "NL07"
    },
    "ua": "Mozilla/5.0 (Linux; Android 4.4.3; HTC One Build/KTU84L) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.102 Mobile Safari/537.36",
    "ip": "213.127.223.58",
    "make": "HTC",
    "model": "One",
    "os": "Android",
    "osv": "4.4.3",
    "js": 1,
    "carrier": "chello",
    "language": "en",
    "connectiontype": 0,
    "dpidsha1": "4dfaa92388699ac6539885aef1719293879985bf",
    "dpidmd5": "5756ae9022b2ea1e47d84fead75220c8",
    "ifa": "38400000-8cf0-11bd-b23e-10b96e40000d"
},
"bcat": [
    "IAB11",
    "IAB23",
    "IAB24",
    "IAB25",
    "IAB26"
]
}

```

El lenguaje de programación a utilizar ha de ser Java.

Sobra decir que no puedes recibir ayuda de terceros.

Cualquier duda puedes ponerte en contacto con Tony a través de su mail: ahervas@tappx.com o skype : tonyhv80

En cuanto tengas listo el código, envíalo por e-mail (valoraremos la calidad del desarrollo, el nivel de pruebas realizadas, así como el tiempo de resolución) y con estos criterios tomaremos una decision.

Gracias y mucha suerte!