

Capítulo 5: Content Provider y Notificaciones

Capítulo 6: Networking para Aplicaciones Android

Capítulo 7: Integración con Redes Sociales



División de Alta Tecnología

# 6

## Networking para Aplicaciones Android

Android Application Developer – Nivel Avanzado

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.




### Objetivos

Al finalizar el capítulo, el alumno logrará:

- Conocer las capacidades de Networking Android
- Consumir Servicios Rest

6 - 2

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Agenda

- Capacidades de Networking Android.
- Consumiendo Servicios Rest.
  - Send Data Rest Services.
  - Receive Data Rest Services.
- Uso de Frameworks.
  - Spring Framework
  - Volley

6 - 3

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Capacidades de Networking Android

- Muchas aplicaciones Android conectadas a la red usan HTTP para enviar y recibir data.
- Android incluye 2 clientes HTTP:
  - HttpURLConnection
  - Apache HttpClient
- Ambos clientes soportan HTTPS, streaming uploads and downloads, timeouts configurables.

6 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Capacidades de Networking Android

Package	Description
java.net	Provides networking-related classes, including stream and datagram sockets, Internet Protocol, and generic HTTP handling. This is the multipurpose networking resource. Experienced Java developers can create applications right away with this familiar package.
java.io	Though not explicitly networking, it's very important. Classes in this package are used by sockets and connections provided in other Java packages. They're also used for interacting with local files (a frequent occurrence when interacting with the network).
java.nio	Contains classes that represent buffers of specific data types. Handy for network communications between two Java language-based end points.
org.apache.*	Represents a number of packages that provide fine control and functions for HTTP communications. You might recognize Apache as the popular open source Web server.
android.net	Contains additional network access sockets beyond the core java.net.* classes. This package includes the URI class, which is used frequently in Android application development beyond traditional networking.
android.net.http	Contains classes for manipulating SSL certificates.
android.net.wifi	Contains classes for managing all aspects of WiFi (802.11 wireless Ethernet) on the Android platform. Not all devices are equipped with WiFi capability, particularly as Android makes headway in the "flip-phone" strata of cell phones from manufacturers like Motorola and LG.
android.telephony.gsm	Contains classes required for managing and sending SMS (text) messages. Over time, an additional package will likely be introduced to provide similar functions on non-GSM networks, such as CDMA, or something like android.telephony.cdma.

6 - 5

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Consumiendo Servicio Rest

- Permisos requeridos
  - android.permission.INTERNET
  - android.permission.ACCESS\_NETWORK\_STATE.
- Evitar realizar operaciones de larga duración en el hilo de la interface de usuario (main thread).
- StrictMode permite configurar políticas para evitar hacer cosas incorrectas, como acceder a la red en el hilo principal.

```
FATAL EXCEPTION: main
java.lang.RuntimeException: Unable to start activity ComponentInfo[com.cyan.rssample/com.cyan.rssample.MainActivity]: android.os.NetworkOnMainThreadException
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2150)
```

6 - 6

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Consumiendo Servicio Rest

- Send Data Rest Service

GET:

```

@Override
protected Respuesta doInBackground(String... strings) {
    boolean resul = true;
    Respuesta respuesta= new Respuesta();
    HttpClient httpClient = new DefaultHttpClient();
    //HttpGet get = new HttpGet("http://10.0.2.2:8080/servicere/rest/seguridad");
    HttpGet get = new HttpGet("http://10.0.2.2:8080/servicere/rest/seguridad?parametro1=valorparametro1");
    get.setHeader("content-type", "text/plain");

    try {
        HttpResponse httpResponse = httpClient.execute(get);
    }
}

```

Diagram illustrating the GET request process:

- 1: Initialize HttpClient and create HttpGet object.
- 2: Set the URL for the GET request.
- 3: Set the content-type header to "text/plain".
- 4: Execute the GET request using httpClient.execute(get).

6 - 7

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Consumiendo Servicio Rest

- Send Data Rest Service

POST:

```

@Override
protected Respuesta doInBackground(String... strings) {
    boolean resul = true;
    Respuesta respuesta= new Respuesta();
    HttpClient httpClient = new DefaultHttpClient();
    //HttpGet get = new HttpGet("http://10.0.2.2:8080/servicere/rest/seguridad");
    HttpPost post = new HttpPost("http://10.0.2.2:8080/servicere/rest/seguridad");

    //get.setHeader("content-type", "text/plain");
    post.addHeader("content-type", "application/x-www-form-urlencoded");

    try {
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
        nameValuePairs.add(new BasicNameValuePair("id", "12345"));
        nameValuePairs.add(new BasicNameValuePair("stringdata", "AndDev is Cool!"));
        post.setEntity(new UrlEncodedFormEntity(nameValuePairs));

        HttpResponse httpResponse = httpClient.execute(post);
    }
}

```

Diagram illustrating the POST request process:

- 1: Initialize HttpClient and create HttpPost object.
- 2: Set the URL for the POST request.
- 3: Set the content-type header to "application/x-www-form-urlencoded".
- 4: Execute the POST request using httpClient.execute(post).

6 - 8

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Consumiendo Servicio Rest

- Receive Data Rest Service

```

2      String resultado = EntityUtils.toString(httpResponse.getEntity());
      respuesta.setRespuesta(resultado);
      Log.d("RESPONSE::", "respuesta: " + resultado);
    } catch (IOException e) {
3      resul=false;
      e.printStackTrace();
    }
    respuesta.setResultado(resul);
    return respuesta;
  }

  @Override
  protected void onPostExecute(Respuesta respuesta) {
    if(respuesta.isResultado()){
4      lblResultado.setText(respuesta.getRespuesta());
    }else{
      lblResultado.setText("Fallo en la conexion");
    }
  }

```

6 - 9

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Uso de Frameworks

- Spring Android
  - Características:
    - Un Cliente Rest para Android
    - Soporte Auth para seguridad
  - Obtener dependencias: En el archivo “build.gradle”, realizar lo siguiente:

```

build.gradle (Module: app)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile 'org.springframework.android:spring-android-rest-template:2.0.0.M1'
}

repositories {
    maven {
        url 'http://repo.spring.io/milestone'
    }
}

```

6 - 10

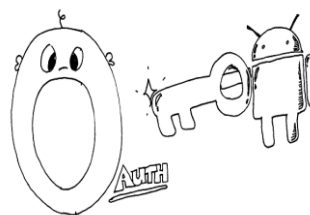
Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Uso de Frameworks

- Spring Android

- El objetivo del módulo de spring-android-auth es solucionar la necesidad de las aplicaciones Android de tener autorización de consumir un servicio web.
- OAuth es un protocolo abierto que permite a los usuarios dar permiso para una aplicación de terceros o sitio web para acceder a recursos restringidos en otro sitio web o servicio. La aplicación de terceros recibe un token de acceso con la que puede hacer peticiones al servicio de protección. Al utilizar esta estrategia de acceso, las credenciales de inicio de sesión de un usuario nunca se almacenan dentro de una aplicación, y solo son necesarios cuando se autentica con el servicio.



6 - 11

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Uso de Frameworks

- Volley

- Volley es una librería desarrollada por Google para optimizar el envío de peticiones HTTP desde las aplicaciones Android hacia servidores externos.
- Encapsula el uso de hilos y de parsing para ser más fácil la publicación del resultado en el hilo principal.
- Procesamiento concurrente de peticiones.
- Cancelación de peticiones.
- Implementación de caché en disco y memoria.

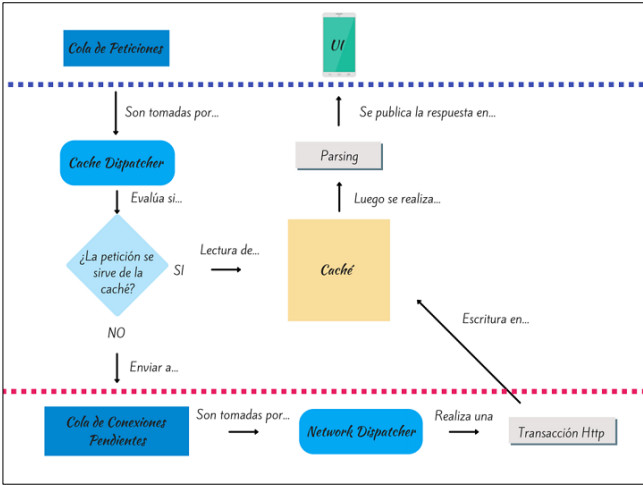
6 - 12

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Uso de Frameworks

- Volley- Procesamiento



6 - 13

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Ejercicio Nº 6: Consumo de Servicios REST

Crear Servicio Rest.

Al finalizar el laboratorio, el alumno logrará:

- Consumir servicios de tipo rest.

6 - 14

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Lecturas adicionales

Para obtener información adicional, puede consultar:

- Jersey
  - <https://jersey.java.net/>
- Spring Android
  - <https://spring.io/understanding/REST>
  - <https://spring.io/guides/tutorials/bookmarks/>
- Android Volley
  - <https://developer.android.com/intl/es/training/volley/index.html>

6 - 15

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



## Resumen

En este capítulo, usted aprendió:

- La plataforma nos brinda paquetes para consumir servicios de la nube.
- HttpURLConnection, HttpClient son opciones para poder consumir dichos servicios.
- Utilizar una tarea asincrónica cada vez que queramos consumir un servicio.
- Evitar usar el hilo principal, desde la versión Android 3.0. Lanzará la excepción: NetworkOnMainThreadException, si se detecta.

6 - 16

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.





## **Tarea N° 6: Utilizar librería volley para conectarse a servicios externos**

Ampliar conocimientos de Networking para Aplicaciones Android.

Al finalizar el laboratorio, el alumno logrará:

- Notar la diferencia con otros frameworks.
- Enviar data y recibir información desde el app.

