

MICROOH 麦可网

# Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

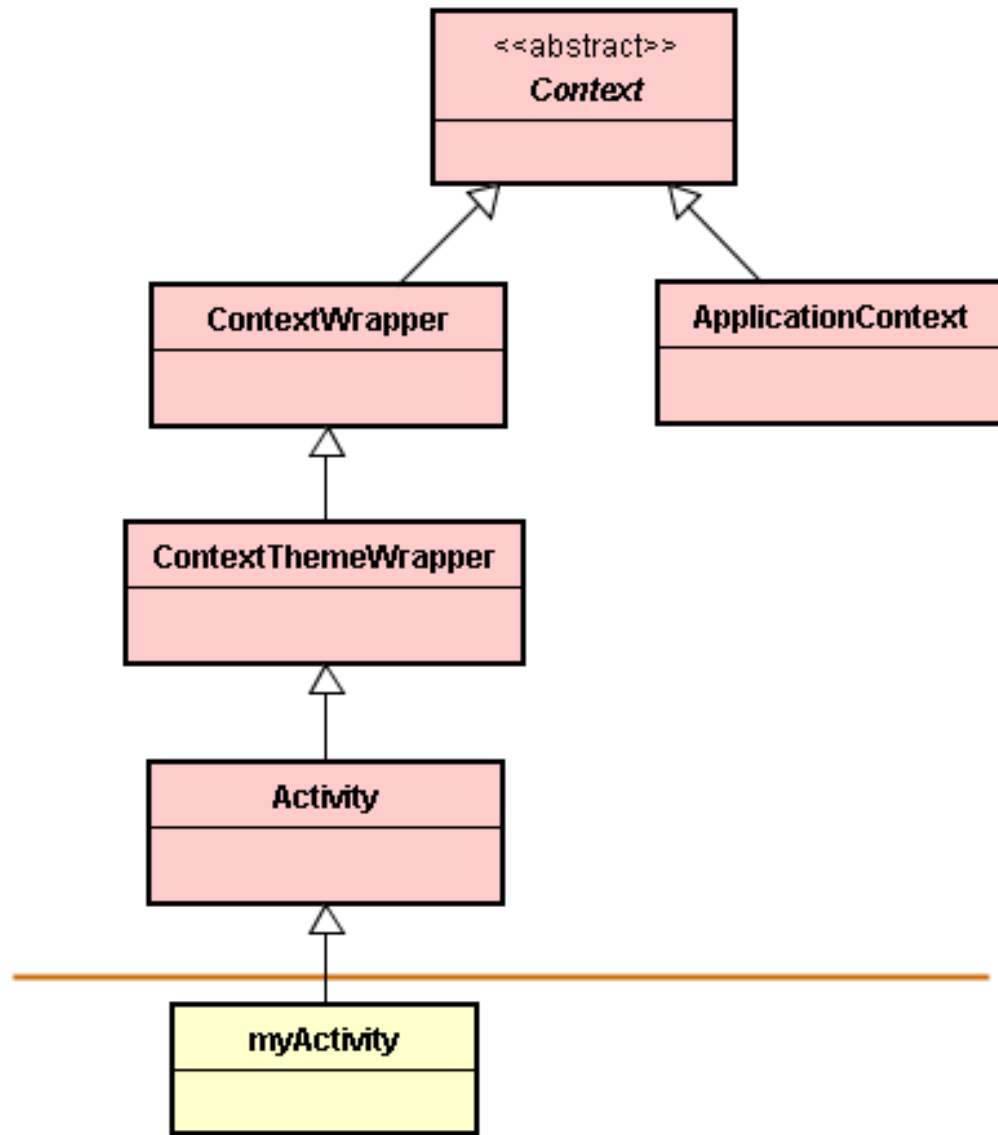
F07\_b

# 观摩：Android+i-Jetty网页 的设计(b)

By 高煥堂

## 2、I-Jetty 与Android沟通 接口：Context

- I-Jetty里的Servlet程序如何与同一只手机里的其它App沟通呢？答案是：透过Android的Context接口。在Android应用框架(Framework)里定义了一个纯粹抽象父类别，其主要的类继承体系如下：



- I-Jetty是一支Android应用程序(\*.apk)，其扮演Container角色，让许多支Servlet程序可以在Android手机里执行。
- 然而，这些Servlet程序经常需要与同一只手机里的其它App沟通，例如启动晨钟或闹铃等等。
- 此时，只要能取得Android框架的Context接口，就能调用Android的服务了。

- 这个Context界面位居幕后，仅从App代码的表面，常常看不到，然而它却是Android平台框架的最主要服务窗口。
- Context接口提供数十个大家很常用的服务，例如startService()、sendBroadcast()函数等。

- 由于Activity也是继承Context抽象父类别，意味着Activity也是实作(Implements)这个Context界面。
- 所以我们在Android应用程序里(如myActivity子类)随时可调用Context接口里的函数，如下述的程序范例：



// Android程序范例

//.....

```
public class ac01 extends Activity implements OnClickListener {  
    private Button btn, btn3;  
    @Override protected void onCreate(Bundle icle) {  
        super.onCreate(icle);  
        //.....  
        setContentView(layout);  
    }  
    public void onClick(View v) {  
        if(v == btn)  
            startService(new Intent(this, myService.class));  
        else if(v == btn3)  
            finish();  
    }  
}
```

- 其中的指令：

*startService(new Intent(this, myService.class));*

- 相当于指令：

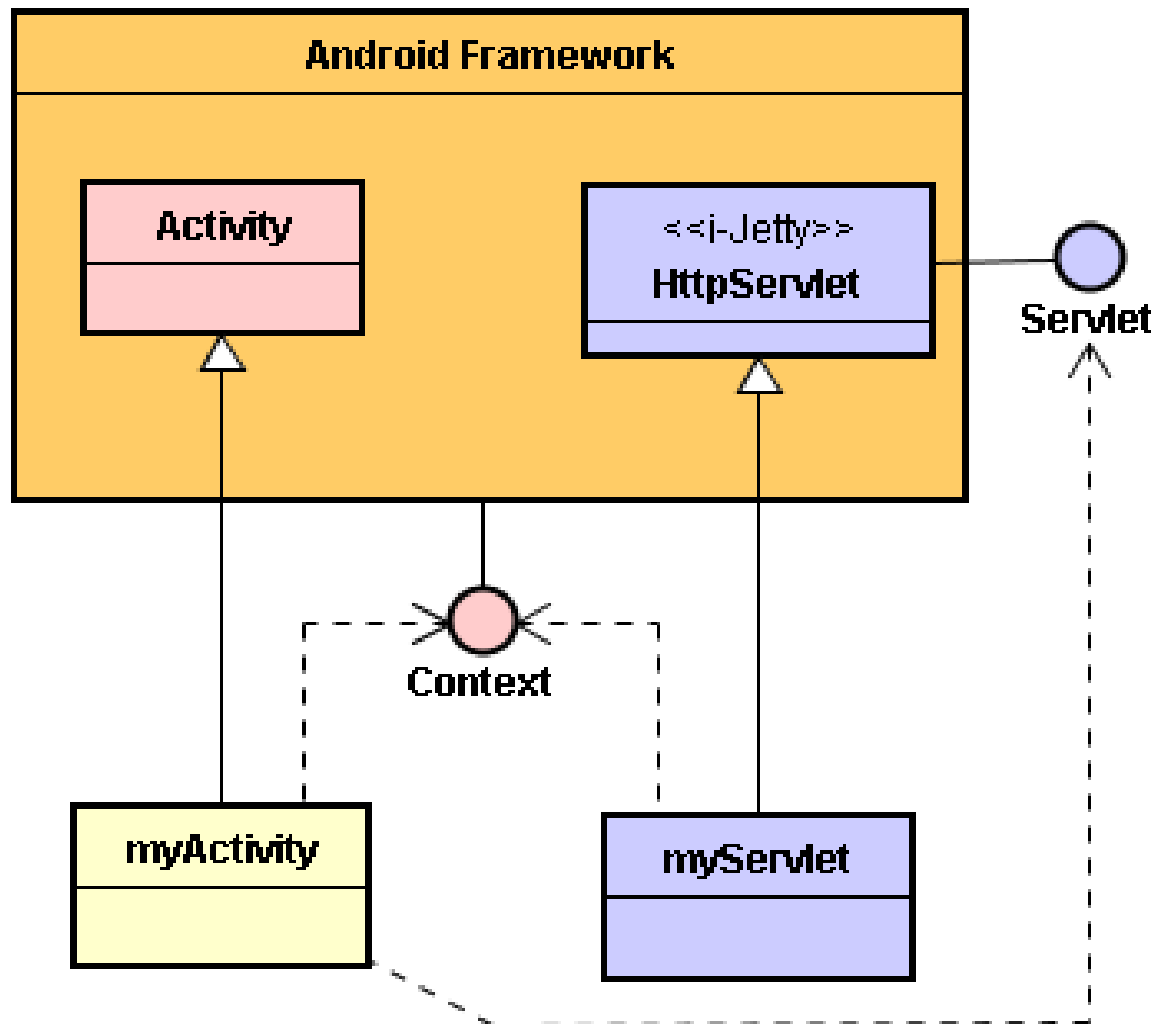
*this.startService(new Intent(this, myService.class));*

- 意味着，其呼叫到Context接口的startService()函数。

- 其幕后机制是，myActivity类调用到ContextWrapper类的Context接口(的startService()函数)，然后转而调用其mBase的(例如ApplicationContext的对象)Context接口的startService()函数。
- 除了startService()函数之外，Context接口还含有许多其它函数，例如getApplicationContext()函数等。

### 3、Android内的I-Jetty 与App通信

- 我们撰写i-Jetty里的myServlet程序时，也能透过Context接口来呼叫startService()函数来启动Android应用程序里的myService子类别，也能呼叫Context接口startActivity()函数来启动Android应用程序里的myActivity子类别，如下图：



- 在myServlet子类别里可利用指令：

```
config.getServletContext().getAttribute(  
    "org.mortbay.ijetty.context");
```

- 指令来取得其当下的Context接口，如下述i-Jetty的范例代码：

```
/* --- i-Jetty的Servlet程序范例 ----*/
```

```
//.....
```

```
public class HelloWorld extends HttpServlet {  
    String proofOfLife = null;  
    public void init(ServletConfig config) throws ServletException {  
        super.init(config);  
        Object o = config.getServletContext().getAttribute(  
            "org.mortbay.ijetty.contentResolver");  
        android.content.ContentResolver resolver  
            = (android.content.ContentResolver)o;  
        android.content.Context androidContext  
            = (android.content.Context)  
                config.getServletContext().getAttribute(  
                    "org.mortbay.ijetty.context");  
        proofOfLife =  
            androidContext.getApplicationInfo().packageName;  
    }  
}
```

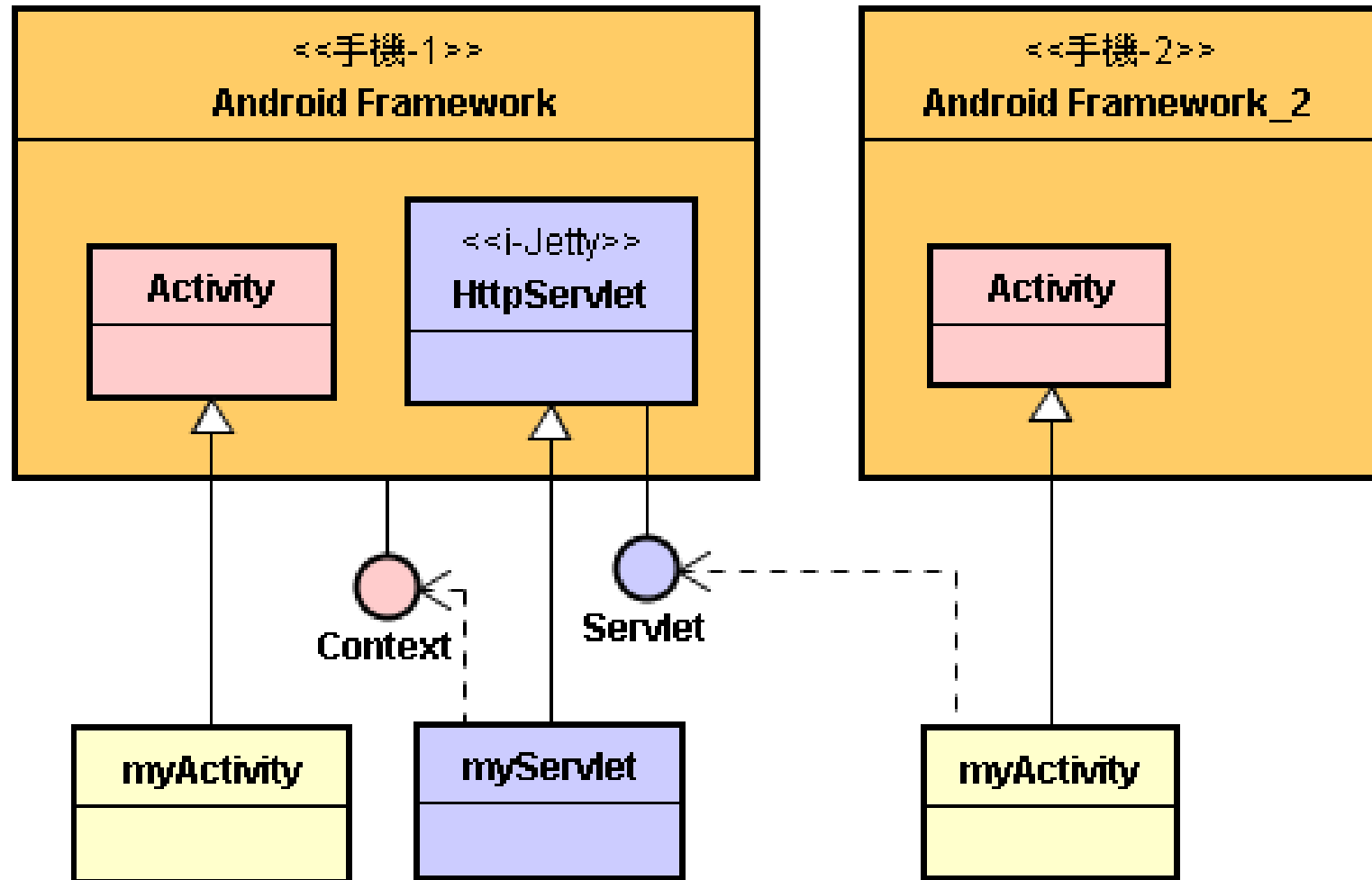


```
public void doPost(HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, IOException  
{ doGet(request, response); }
```

```
public void doGet(HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    ServletOutputStream out = response.getOutputStream();  
    out.println("<html>");  
    out.println("<h1>Hello From Servlet Land!</h1>");  
    out.println("Brought to you by: "+proofOfLife);  
    out.println("</html>");  
    out.flush();  
    }  
}
```

## 4、I-Jetty与Android App的跨机通信

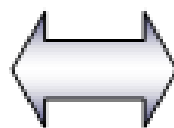
- 上图里是从同一支手机(本机)里的myActivity来调用本机的Servlet接口，这意义比较小。
- 更大的意义是：由别的手机里的myActivity来传送HTTP访问到本机的Servlet接口，调用到本机的myServlet子类；然后由myServlet调用Android的Context接口，而完成两支手机里两个myActivity子类之间的通信。如下图：



*Browser/Mobile*

瀏覽器  
(Browser)

Mobile平台



*Android/TV*

本地  
App

Servlet/HTML5

I-Jetty框架

Android框架

- 于是，Context和Servlet两大接口，成为Android\*\*云的喜鹊桥的两个主要桥墩。



**~ Continued ~**