

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>

F05\_b

# 观摩：Android端云整合 与分工策略(b)

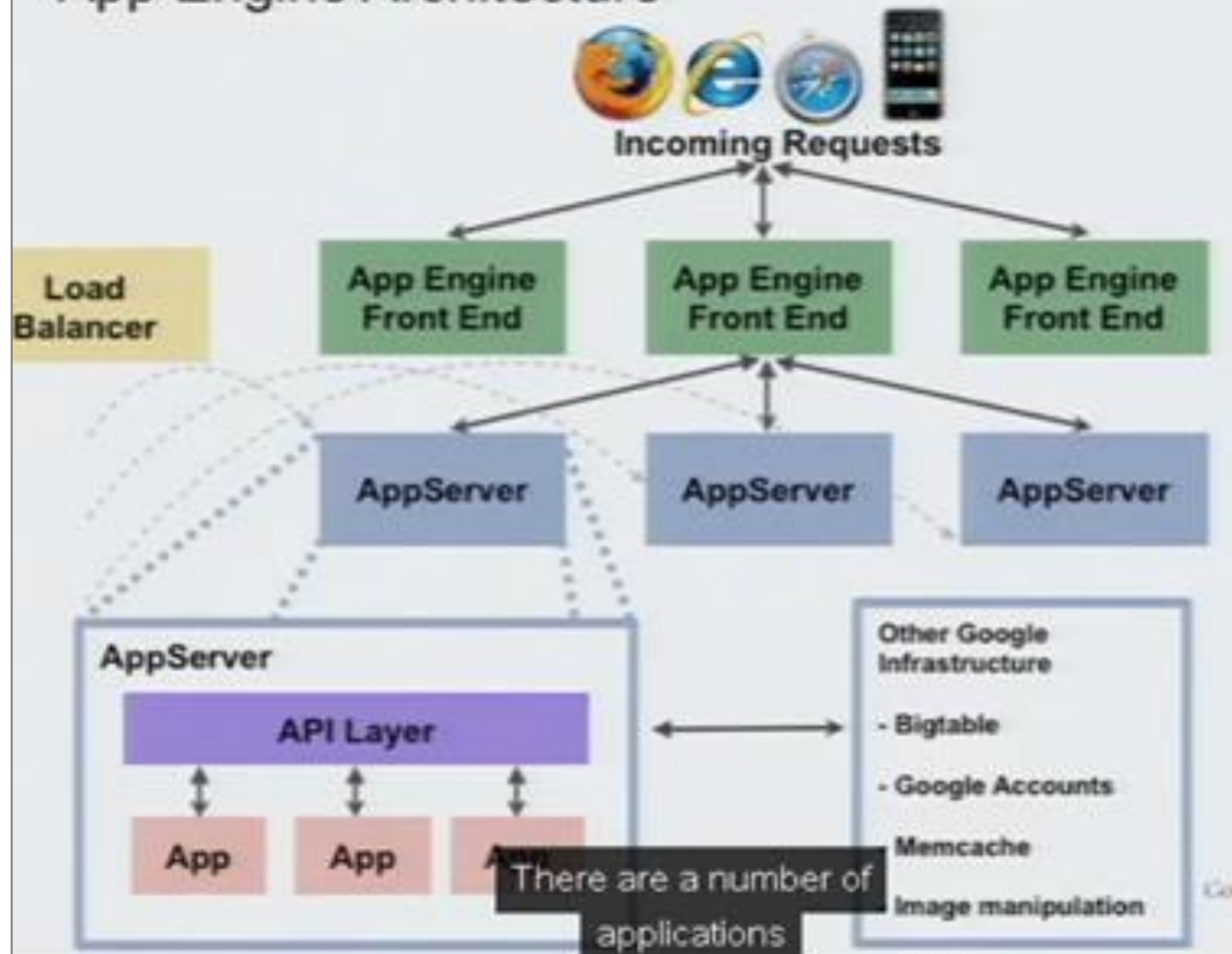
By 高煥堂

## 2、GAE云平台的API

# 以Google的GAE云平台为例

- GAE(Google AppEngine)是Google的云服务引擎，第三方应用开发者能开发App，然后放在Google服务器上执行，不需担心频宽、系统负载、安全维护等问题，一切由Google代管。
- GAE平台的系统架构如下图：

# App Engine Architecture

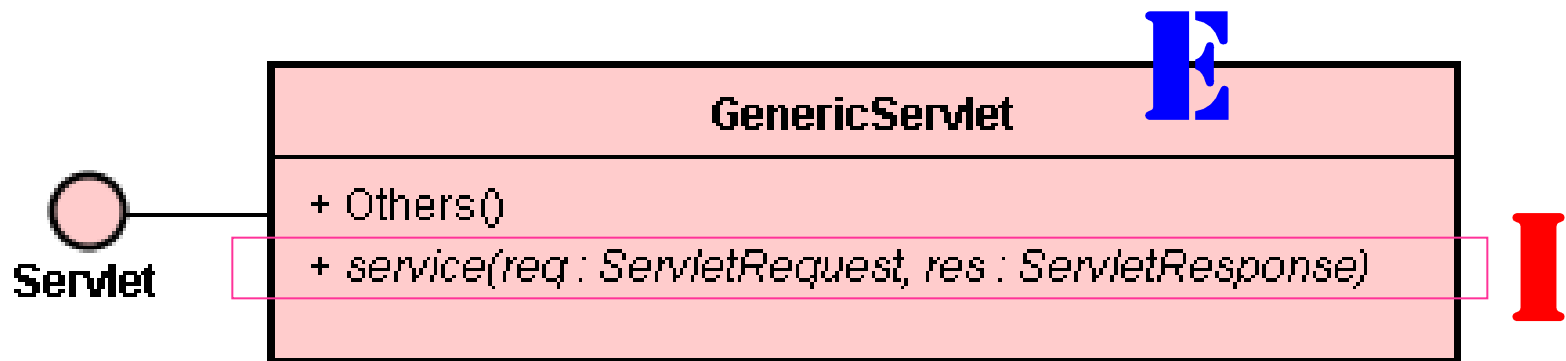


- 从上图可看到，从手机、PC、MID等众多端设备上，都能随时上网发出要求(Incoming Requests)来存取GAE上的服务。
- 在GAE后台的AppServer里，GAE提供了API(即API Layer)来衔接你的云端应用程序。

# GAE与Android之关系

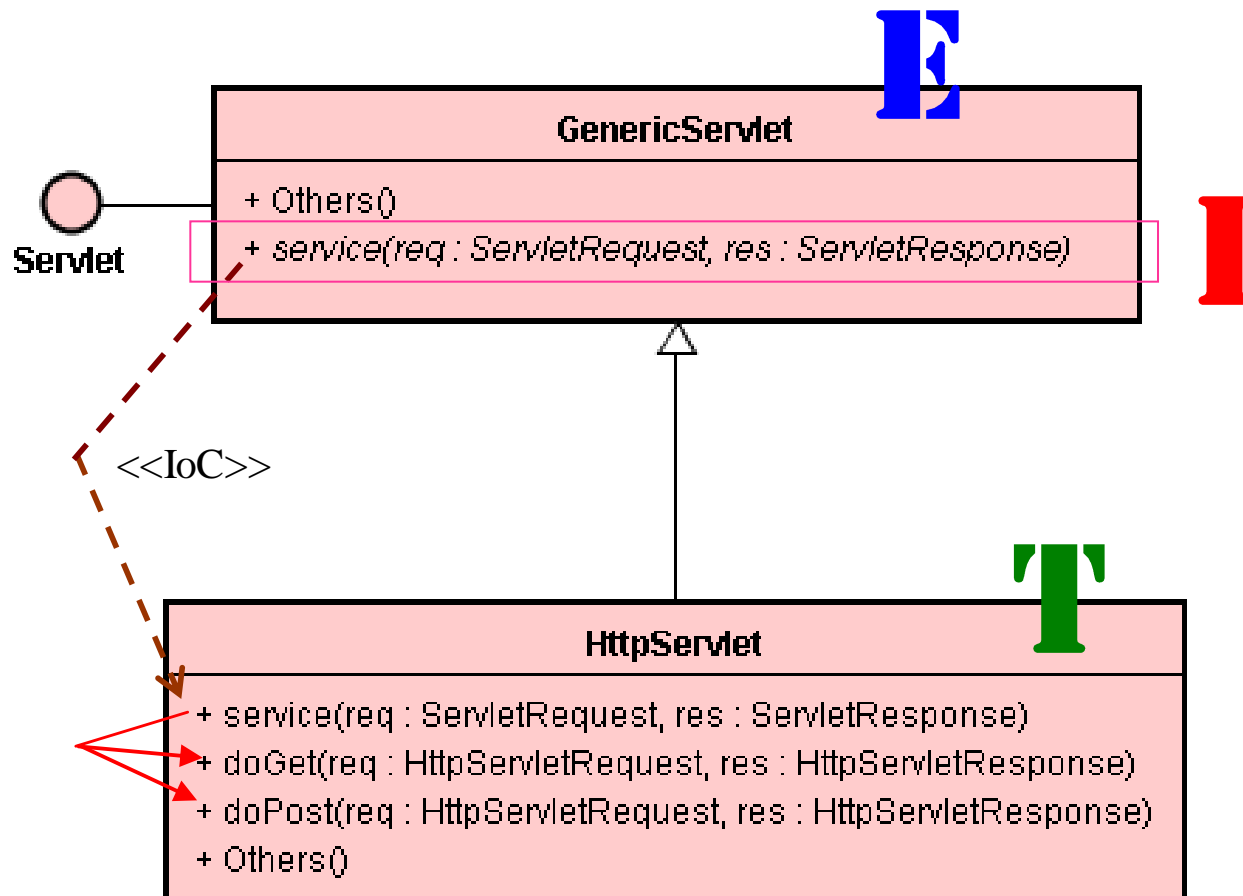
- Android是当今手机的主要软件平台。从AppEngine云平台的角度而观之，手机只是云平台所连接出去的众多端设备之一，所以Android是与GAE相联结的众多端平台之一。
- 相对地，从Android行动端平台的角度而观之，GAE只是Android端所连接出去的众多云平台之一。

- GAE云平台提供了HttpServlet框架，此框架EIT造形来提供主动型的<I>。

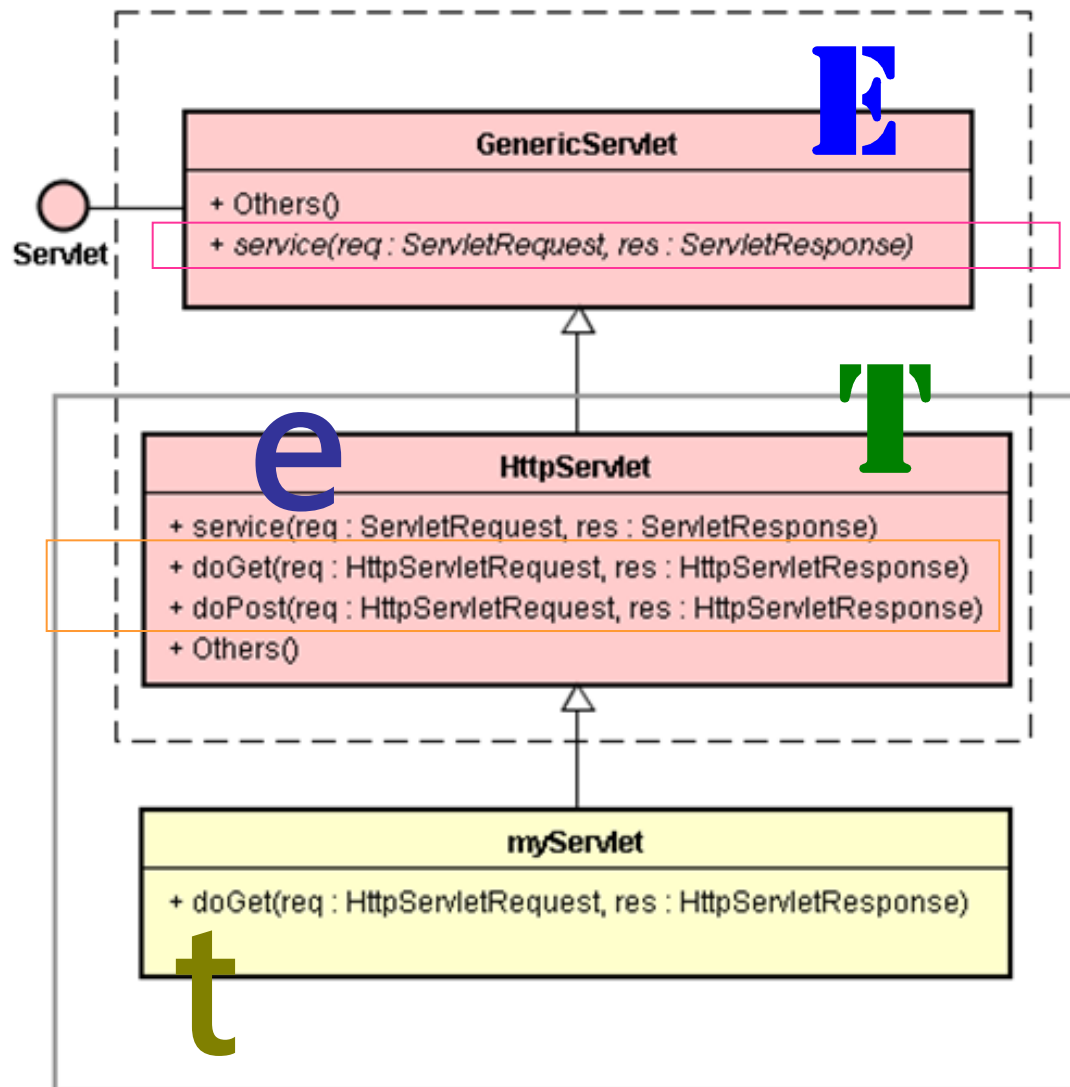




- 这GenericServlet是EIT造形里的<E>。基于主动型API，可加上子类<T>，如下图：



- 其中，GenericServlet基类的service()函数就扮演<I>的角色，执行时，它会反向调用到子类的service()函数，然后继续呼叫到其doGet()或doPost()函数；形成了两层EIT造形了。



# 范例代码

## ★ 地头蛇写GAE云的<T>代码

```
/*---- myServlet.java ----*/  
// .....  
public class myServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    public void doGet(HttpServletRequest req,  
                        HttpServletResponse resp)  
        throws IOException {
```

```
String strCode = req.getParameter("code");
String sv1 = req.getParameter("value1");
String sv2 = req.getParameter("value2");
int code = Integer.valueOf(strCode);
int v1 = Integer.valueOf(sv1);
int v2 = Integer.valueOf(sv2);
int v = 0;
if( code == 0 )  v = v1 + v2;
else            v = v1 * v2;
String result = String.valueOf(v);
resp.setContentType("text/plain");
resp.getWriter().println(result);
}
}
```

## ★ 地头蛇写手机端的<T>代码

```
/*---- ac01.java ----*/  
// .....  
public class ac01 extends Activity implements OnClickListener {  
    // .....  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        //.....  
    }  
    public void onClick(View v) {  
        if(v.getId()== 101)  
            this.invokeServlet();  
        else if(v.getId() == 102) finish();  
    }  
}
```

```
private void invokeServlet(){
    pdg = ProgressDialog.show(this,"please wait...",
                             "GAE accessing...",true);
    h = new EHandler();
    new Thread(){
        public void run(){
            DefaultHttpClient httpclient = new DefaultHttpClient();
            String u = APP_NAME + ".appspot.com";
            ArrayList<NameValuePair> params
                = new ArrayList <NameValuePair>();
            params.add(new BasicNameValuePair("code", "1"));
            params.add(new BasicNameValuePair("value1", "55"));
            params.add(new BasicNameValuePair("value2", "100"));
            HttpResponse response = null;
            try { HttpGet httpget =
                new HttpGet(URIUtils.createURI("https",u, -1, "/my",
                URLEncodedUtils.format(params, "UTF-8"), null));
```

```
// 呼叫GAE的Servlet
    response = httpclient.execute(httpget);
} catch (Exception e) { e.printStackTrace(); }
if( response.getStatusLine().getStatusCode()
    != HttpStatus.SC_OK)
{ status = "doGet Error!"; return; }
status = "doGet OK.";
// 读取回传的内容
InputStream is;
byte[] buffer = new byte[4096];
try { is = response.getEntity().getContent();
    is.read(buffer, 0, 4096);
} catch (Exception e) { e.printStackTrace(); }
res_value = new String(buffer);
pdg.dismiss();
h.sendMessage(0);
}}.start();
}
```



```
//-----  
class EHandler extends Handler {  
    public EHandler()  
        { super(Looper.getMainLooper()); }  
    public void handleMessage(Message msg) {  
        setTitle(status);  
        tv.setText("response: " + res_value);  
    }  
}
```

- 这ac01调用myServlet去做两个整数的相加或相乘。共传送3个参数给myServlet程序。如果第1参数的值为0，就将后续两个参数值相加；反之，如果1参数的值为1，就将两个值相乘。并将其值回传给Android应用程序，显示于画面上

# 结语

- 以上展现了新型API与分工界线：
  - ★两端都写<E&I>的Google成为强龙。
  - ★两端都写<T>的App开发者，成为地头蛇。
  - ★强龙定义API(即<I>)，拥有端云整合架构的话语权。



**~ Continued ~**