使用 HttpURLConnection 来访问网络

知识解析

HTTP 协议中，在客户端用于请求数据的方法常用的有 2 种——get 和 post：

- get：经常用于请求静态页面，也可以将参数附加在 URL 后面请求动态页面，它传递的参数大小有限制
- post：一般用于请求动态页面，它会把参数放在 http 请求的正文内传递。

可以使用 HttpURLConnection 访问网络。

首先要特别提醒，因为网络连接可能会很耗时，所以不要在主线程中进行网络访问！

通过 URL.openConnection()获取 HttpURLConnection 对象，注意需要对结果造型成 HttpURLConnection。

如果有必要，通过 HttpURLConnection 对象的：

setReadTimeout()设置读取数据的超时时间

setConnectionTimeout()设置连接超时时间

setRequestMethod()设置请求方法

setRequestProperty()：设置请求头信息，例如，多线程下载中，可以设置各个下载线程下载数据的范围：setRequestProperty(“Range”, “bytes=开始位置-结束位置");

如果有必要，可以使用 setDoOutput(true)方法设置输出，并且通过往 getOutputStream()流中写入数据的方法来往服务器端传递数据

使用 connect()来连接服务器

使用 getResponseCode()获取响应代码，200 为正常

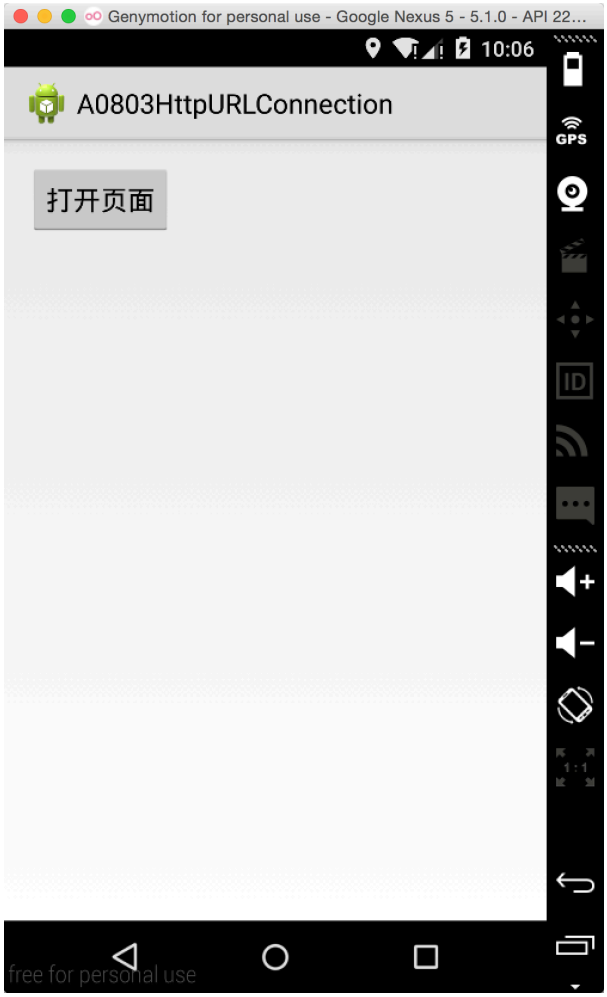
使用 getInputStream()来读取服务器端返回的数据

使用 disconnect()方法来关闭连接

如果需要传递表单数据，可以通过 Uri.Builder 将数据封装到 Uri 中的方法进行传递：

```
conn.setDoOutput(true);
Uri.Builder builder = new Uri.Builder()
        .appendQueryParameter("firstParam",
paramValue1)
        .appendQueryParameter("secondParam",
paramValue2)
        .appendQueryParameter("thirdParam",
paramValue3);
String query = builder.build().getEncodedQuery();
OutputStream os = conn.getOutputStream();
BufferedWriter writer = new BufferedWriter(
           new OutputStreamWriter(os, "UTF-8"));
writer.write(query);
writer.flush();
writer.close();
os.close();
```

功能演示



操作实践

```
    new Thread() {
                    public void run() {
                        try {
                            URL url = new URL("http://www.baidu.com");
                            HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
```

```
                    conn.setReadTimeout(10000);
                    conn.setConnectTimeout(20000);
                    conn.setRequestMethod("GET");
                    conn.setDoOutput(true);
                    conn.setDoInput(true);
                    conn.connect();
                    int responseCode = conn.getResponseCode();
                    if (responseCode == HttpURLConnection.HTTP_OK) {
                        InputStream is = conn.getInputStream();
                        BufferedReader reader = new BufferedReader(new
InputStreamReader(is));
                        String str = "";
                        while ((str = reader.readLine()) != null) {
                            Log.i("Content", str);
                        }
                    } else {
                        Log.e("Err", "Err Code:" + responseCode);
                    }
                } catch (MalformedURLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
```

```
        }.start();
```

职业素质

http 是一个可靠的传输，建立在 TCP/IP 连接之上，缺省端口是 80，其他端口号也可以用。Android 可以用 HttpURLConnection 或 HttpClient 接口来开发 http 程序。

http 通信使用最多的是 Get 和 Post。Post 和 Get 的不同之处在于 Get 的参数放在 URL 字串中，而 Post 的参数放在 http 请求数据中。

HttpURLConnection 继承自 URLConnection，都是抽象类，无法直接实例化对象。其对象主要通过 URL 的 openConnection 方法获得。

openConnection 方法只创建 URLConnection 或 HttpURLConnection 实例，但是不进行真正的连接操作，并且每次 openConnection 都创建一个新的实例。