Android Developers Blog

29 SEPTEMBER 2011

Android's HTTP Clients

[This post is by Jesse Wilson from the Dalvik team. —Tim Bray]

Most network-connected Android apps will use HTTP to send and receive data. Android includes two HTTP clients: HttpURLConnection and Apache HTTP Client. Both support HTTPS, streaming uploads and downloads, configurable timeouts, IPv6 and connection pooling.

Apache HTTP Client

DefaultHttpClient and its sibling AndroidHttpClient are extensible HTTP clients suitable for web browsers. They have large and flexible APIs. Their implementation is stable and they have few bugs.

But the large size of this API makes it difficult for us to improve it without breaking compatibility. The Android team is not actively working on Apache HTTP Client.

HttpURLConnection

HttpURLConnection is a general-purpose, lightweight HTTP client suitable for most applications. This class has humble beginnings, but its focused API has made it easy for us to improve steadily.

Prior to Froyo, HttpURLConnection had some frustrating bugs. In particular, calling `close()` on a readable InputStream could poison the connection pool. Work around this by disabling connection pooling:

```
private void disableConnectionReuseIfNecessary() {
    // HTTP connection reuse which was buggy pre-froyo
    if (Integer.parseInt(Build.VERSION.SDK) < Build.VERSION_CODES.FROYO) {
        System.setProperty("http.keepAlive", "false");
    }
}
```

In Gingerbread, we added transparent response compression. HttpURLConnection will automatically add this header to outgoing requests, and handle the corresponding response:

```
Accept-Encoding: gzip
```

Take advantage of this by configuring your Web server to compress responses for clients that can support it. If response compression is problematic, the class documentation shows how to disable it.

Since HTTP's `Content-Length` header returns the compressed size, it is an error to use getContentLength() to size buffers for the uncompressed data. Instead, read bytes from the response until InputStream.read() returns -1.

We also made several improvements to HTTPS in Gingerbread. HttpsURLConnection attempts to connect with Server Name Indication (SNI) which allows multiple HTTPS hosts to share an IP address. It also enables compression and session tickets. Should the connection fail, it is automatically retried without these features. This makes HttpsURLConnection efficient when connecting to up-to-date servers, without breaking compatibility with older ones.

In Ice Cream Sandwich, we are adding a response cache. With the cache installed, HTTP requests will be satisfied in one of three ways:

- Fully cached responses are served directly from local storage. Because no network connection needs to be made such responses are available immediately.
- Conditionally cached responses must have their freshness validated by the webserver. The client sends a request like "Give me /foo.png if it changed since yesterday" and the server replies with either the updated content or a `304 Not Modified` status. If the content is unchanged it will not be downloaded!
- Uncached responses are served from the web. These responses will get stored in the response cache for later.

Use reflection to enable HTTP response caching on devices that support it. This sample code will turn on the response cache on Ice Cream Sandwich without affecting earlier releases:

```
private void enableHttpResponseCache() {
    try {
        long httpCacheSize = 10 * 1024 * 1024; // 10 MiB
        File httpCacheDir = new File(getCacheDir(), "http");
```

Back to developer.android.com

SEARCH

Search

ARCHIVE

▶ 2015 (44)
▶ 2014 (73)
▶ 2013 (48)
▶ 2012 (41)
▼ 2011 (68)
▶ December (7)
▶ November (7)
▶ October (5)
▼ September (5)
Android's HTTP Clients
Euro Android Developer Labs
Preparing for Handsets
Thinking Like a Web Designer
More Carrier Billing Options on Android Market
▶ August (3)
▶ July (7)
▶ June (3)
▶ May (5)
▶ April (6)
▶ March (8)
▶ February (8)
▶ January (4)
▶ 2010 (72)
▶ 2009 (63)
▶ 2008 (40)
▶ 2007 (8)

COMMUNITY

Android Developers
1 circle +1

Subscribe
20127 readers BY FEEDBURNER
Google

```
            Class.forName("android.net.http.HttpResponseCache")
                .getMethod("install", File.class, long.class)
                .invoke(null, httpCacheDir, httpCacheSize);
        } catch (Exception httpResponseCacheNotAvailable) {
        }
    }
```

You should also configure your Web server to set cache headers on its HTTP responses.

Which client is best?

Apache HTTP client has fewer bugs on Eclair and Froyo. It is the best choice for these releases.

For Gingerbread and better, HttpURLConnection is the best choice. Its simple API and small size makes it great fit for Android. Transparent compression and response caching reduce network use, improve speed and save battery. New applications should use HttpURLConnection; it is where we will be spending our energy going forward.

Posted by Unknown at 9:17 AM
Labels: Connectivity, http, network

Links to this post

Create a Link

Newer Post Home Older Post