Jira tinkerers beware: Tomcat does not serve raw files correctly, causing browser hangs
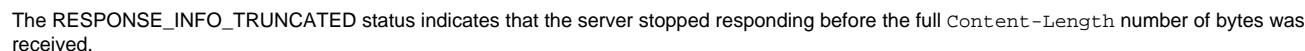
This page is for anyone who has been tempted to add files (e.g. js, css) into JIRA's `atlassian-jira/` directory. It may come as a surprise, but JIRA suffers from the most basic of bugs: it doesn't serve static files correctly. More specifically, if Tomcat serves the file with `gzip` compression, the `Content-Length:` header is incorrectly set.

> ⚠️ This is now filed as a JIRA bug, 🔴 **JRASERVER-66932** - Ineffective static files served through Tomcat GATHERING IMPACT

Let's pick on `jira.jboss.org` for our demo, but you can use any public, non-Cloud JIRA instance fronted by Apache (not nginx, which seems to mask the problem). Try hitting https://jira.jboss.org/robots.txt. This loads, but hangs for 20 seconds.

If you are using Chrome, now ~~view the cached response by viewing~~ ~~chrome://view-http-cache/https://jira.jboss.org/robots.txt~~ **edit:** sadly, Chrome no longer supports chrome://cache and chrome://view-http-cache.:

```
←  →  C    ⊙ Chrome  │ chrome://view-http-cache/https://jira.jboss.org/robots.txt

https://jira.jboss.org/robots.txt

RESPONSE_INFO_TRUNCATED


HTTP/1.1 200
X-AREQUESTID: 352x6852288x2
X-ASEN: SEN-1095081
X-AUSERNAME: anonymous
X-Content-Type-Options: nosniff
Accept-Ranges: bytes
ETag: W/"669-1519645607000"
Last-Modified: Mon, 26 Feb 2018 11:46:47 GMT
Content-Encoding: gzip
Vary: User-Agent
Content-Type: text/plain;charset=UTF-8
Content-Length: 669
Date: Sat, 10 Mar 2018 10:52:03 GMT


00000000: 98 0f 00 00 03 1f 45 00 f3 e3 80 fb a2 c5 2e 00   ......E.........
00000010: 34 cb 83 fb a2 c5 2e 00 5a 01 00 00 48 54 54 50   4.......Z...HTTP
00000020: 2f 31 2e 31 20 32 30 30 00 58 2d 41 52 45 51 55   /1.1 200.X-AREQU
00000030: 45 53 54 49 44 3a 20 33 35 32 78 36 38 35 32 32   ESTID: 352x68522
00000040: 38 38 78 32 00 58 2d 41 53 45 4e 3a 20 53 45 4e   88x2.X-ASEN: SEN
00000050: 2d 31 30 39 35 30 38 31 00 58 2d 41 55 53 45 52   -1095081.X-AUSER
```

The RESPONSE_INFO_TRUNCATED status indicates that the server stopped responding before the full `Content-Length` number of bytes was received.

This can also be replicated on the command-line:

```
jturner@jturner-desktop:~$ curl -D- 'https://jira.jboss.org/robots.txt' --compressed -o /tmp/robots.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0HTTP/1.1 200
X-AREQUESTID: 349x6851675x2
X-ASEN: SEN-1095081
Set-Cookie: atlassian.xsrf.token=AQZJ-FV3A-N91S-UDEU|48bf3940edd26232d07bc5073efd9d190aa239ca|lout;path=/
X-AUSERNAME: anonymous
X-Content-Type-Options: nosniff
Accept-Ranges: bytes
ETag: W/"669-1519645607000"
Last-Modified: Mon, 26 Feb 2018 11:46:47 GMT
Content-Encoding: gzip
Vary: User-Agent
Content-Type: text/plain;charset=UTF-8
Content-Length: 669
Date: Sat, 10 Mar 2018 10:49:29 GMT
 53   669   53   355    0     0     16      0  0:00:41  0:00:21  0:00:20     0
curl: (18) transfer closed with 314 bytes remaining to read
jturner@jturner-desktop:~$
jturner@jturner-desktop:~$ ls -la /tmp/robots.txt
-rw-r--r-- 1 jturner jturner 669 Mar 10 21:49 /tmp/robots.txt
```

The problem is that `Content-Length` is being set to the size of the file (669 bytes), not the gzip-compressed size.

JIRA's static files can be seen in the `atlassian-jira/` directory. For instance, there are a bunch in `/atlassian-jira/static/`. Requesting any of these with compression results in the same hang:

```
jturner@jturner-desktop:~$ time curl -sS 'https://jira.jboss.org/static/util/urls.js' --compressed -o /dev/null
curl: (18) transfer closed with 263 bytes remaining to read
real        0m22.109s
user        0m0.027s
sys         0m0.004s
```

Consequences of the bug

Naturally, any JIRA page that refers to a static file will be affected. The page will load, but then hang for 20 seconds or so as the browser waits for the last resource to load, before giving up.

The thing is, there are actually remarkably few direct references in JIRA. All the files in `/atlassian-jira/static/`, for instance, are actually served through JIRA's file batching mechanism as `batch.js` resources. That is why this bug has gone undetected for so many years.

So far I know of only three situations in which the bug / hang is triggered:

1. `/robots.txt`, as illustrated above.

2. The **JIRA 404 error page**. Try it in your JIRA: add any rubbish to the end of the URL to trigger a 404 error. The 404 error page loads but then spins for 20 seconds. For instance, on https://jira.jboss.org/invalid (please don't click, I've abused them enough), the waterfall shows the 20s hang:

   The 404 page hangs because it directly pulls in the static `/static-assets/metal-all.css` and `/static-assets/metal-all.js` resources.

3. **Any static files you add**. At least, any non-image, not-trivially-small file that Tomcat might want to compress. For instance, say you add a custom Javascript file at `/atlassian-jira/static/custom-file.js`, then include it in a <script> tag on JIRA pages. Your JIRA will now suffer from mysterious 20 second hangs on pages that used to be fast. This is unfortunately how I encountered the bug.

Effects on browsers

The effect on browsers deserves a bit more unpacking. We'll use https://jira.jboss.org/robots.txt again as our example.

- In Firefox, the page loads, but hangs for 20s. Another hit, another 20 second hang.
- In Chrome (64.0 tested), the page loads, but hangs for 20s. On the second hit things get interesting:

Corruption! On a third hit, more corruption, but now there is no hang:

On the second and third hit, if you view to see the state of Chrome's cache, you'll see that the page is flagged as truncated:

```
←  →  C  | ● Chrome | chrome://view-http-cache/https://jira.jboss.org/robots.txt

https://jira.jboss.org/robots.txt

RESPONSE_INFO_TRUNCATED

HTTP/1.1 200
X-AREQUESTID: 382x6862504x1
X-ASESSIONID: 1xujdq3
X-ASEN: SEN-1095081
X-AUSERNAME: anonymous
X-Content-Type-Options: nosniff
Accept-Ranges: bytes
ETag: W/"669-1519645607000"
Last-Modified: Mon, 26 Feb 2018 11:46:47 GMT
Content-Encoding: gzip
Vary: User-Agent
Content-Type: text/plain;charset=UTF-8
Content-Length: 669
Date: Sat, 10 Mar 2018 11:22:24 GMT

00000000: ac 0f 00 00 03 1f 45 00 e4 2a 07 68 a3 c5 2e 00   ......E..*.h....
00000010: c7 11 0a 68 a3 c5 2e 00 70 01 00 00 48 54 54 50   ...h....p...HTTP
00000020: 2f 31 2e 31 20 32 30 30 00 58 2d 41 52 45 51 55   /1.1 200.X-AREQU
00000030: 45 53 54 49 44 3a 20 33 38 32 78 36 38 36 32 35   ESTID: 382x68625
00000040: 30 34 78 31 00 58 2d 41 53 45 53 53 49 4f 4e 49   04x1.X-ASESSIONI
00000050: 44 3a 20 31 78 75 6a 71 33 00 58 2d 41 53 45      D: 1xujdq3.X-ASE
00000060: 4e 3a 20 53 45 4e 2d 31 30 39 35 30 38 31 00 58   N: SEN-1095081.X
00000070: 2d 41 55 53 45 52 4e 41 4d 45 3a 20 61 6e 6f 6e   -AUSERNAME: anon
```

After the third hit, Chrome has marked the corrupt cache entry as valid:

```
HTTP/1.1 200
X-AREQUESTID: 386x6863394x6
X-ASESSIONID: 1xujdq3
X-ASEN: SEN-1095081
X-AUSERNAME: anonymous
Accept-Ranges: bytes
Last-Modified: Mon, 26 Feb 2018 11:46:47 GMT
Date: Sat, 10 Mar 2018 11:26:22 GMT
Vary: User-Agent
X-Content-Type-Options: nosniff
ETag: W/"669-1519645607000"
Content-Encoding: gzip
Content-Type: text/plain;charset=UTF-8
Content-Length: 669
```

```
00000000: ac 0f 00 00 03 0f 45 00 e8 0a 45 76 a3 c5 2e 00   ......E...Ev....
00000010: 2f 0d 48 76 a3 c5 2e 00 70 01 00 00 48 54 54 50   /.Hv....p...HTTP
00000020: 2f 31 2e 31 20 32 30 30 00 58 2d 41 52 45 51 55   /1.1 200.X-AREQU
00000030: 45 53 54 49 44 3a 20 33 38 36 78 36 38 36 33 33   ESTID: 386x68633
00000040: 39 34 78 36 00 58 2d 41 53 45 53 53 49 4f 4e 49   94x6.X-ASESSIONI
00000050: 44 3a 20 31 78 75 6a 64 71 33 00 58 2d 41 53 45   D: 1xujdq3.X-ASE
00000060: 4e 3a 20 53 45 4e 2d 31 30 39 35 30 38 31 00 58   N: SEN-1095081.X
```

and henceforth all requests for the URL will serve binary gibberish at the end.

This unfortunate behaviour appears to be a Chrome bug (edit: now filed as Issue 423318). Searching for RESPONSE_INFO_TRUNCATED will bring up relevant hits, including a stale bug report (https://bugs.chromium.org/p/chromium/issues/detail?id=423318) and people complaining of the same behaviour (https://stackoverflow.com/questions/47311027/response-info-truncated-file-in-chrome-cache). I am not sure what effect this binary corruption has on browsers. Possibly none, as browsers are built to handle any rubbish thrown at them.

Conclusion

There are two takeaways:

- Now you know why JIRA's 404 page always seems slow. ~~I would file a bug, but Atlassian no longer accept bug reports on~~ ~~https://jira.atlassian.com~~ , filed as bug **JRASERVER-66932** - Ineffective static files served through Tomcat `GATHERING IMPACT` indirectly via support request
- If you want to add custom Javascript, CSS or other non-image files to `atlassian-jira`, **you can't rely on Tomcat to serve them**. Better to serve the static files directly via your frontend webserver.