Base64Codec.encodeBytesToBytes generating invalid Base64 string #1831

New issue

Open **rafaelrenanpacheco** opened this issue on 30 Jan · 0 comments

rafaelrenanpacheco commented on 30 Jan • edited

Summary

There's a problem when encoding a payload to JWT when the payload contains `oã` after some other char. We found this problem when generating the access token for an user named `João`, which resulted in an invalid Base64 string.

Actual Behavior

When `Base64Codec.encodeBytesToBytes` encodes the following payload:

```
{"empresa_id":1,"usuario_id":7715,"user_name":"local.joao.1@fesc.io","scope":["write","read"],"nome":"joão anão","exp":158
```

It produces the following Base64 encoded string:

```
eyJlbXByZXNhX2lkIjoxLCJ1c3VhcmlvX2lkIjo3NzE1LCJ1c2VyX25hbWUiOiJsb2NhbC5qb2FvLjFAZmVzYy5pbyIsInNjb3BlIjpbInd yaXRlIiwicmVhZC
```

Which decodes to this:

```
{"empresa_id":1,"usuario_id":7715,"user_name":"local.joao.1@fesc.io","scope":["write","read"],"nome":"j`[i^MNN
K KH XLL
KMMKX XKMY YYL Y[ Y XXKX Y[
```

Expected Behavior

The expected behavior is to decode to the exactly value used to encode.

Version

Spring Boot 2.1.12
spring-security-oauth2 2.3.8.RELEASE
spring-security-jwt 1.1.0.RELEASE

Sample

Put a breakpoint anywhere in the project and run the following code:

```
new String(Base64Codec.encodeBytesToBytes("joão".getBytes(), 0, "joão".getBytes().length, Base64Codec.URL_SAFE))
```

This will encode **joão** to **am_Do28=** which decodes **j`**

If you try to encode `jõao` it will encode and decode just fine. The problem so far is with `oã` with another char before it. Encoding and decoding `oão` for example will also work as expected.

1

spring-issuemaster added the **status: waiting-for-triage** label on 27 Mar

Assignees
No one assigned

Labels
status: waiting-for-triage

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

2 participants