TIME AND DATE FOR PROGRAMMERS

TIME AND DATE FOR PROGRAMMERS

MICHAŁ PIPA

PHPERS SUMMIT, POZNAŃ, 2017-09-02

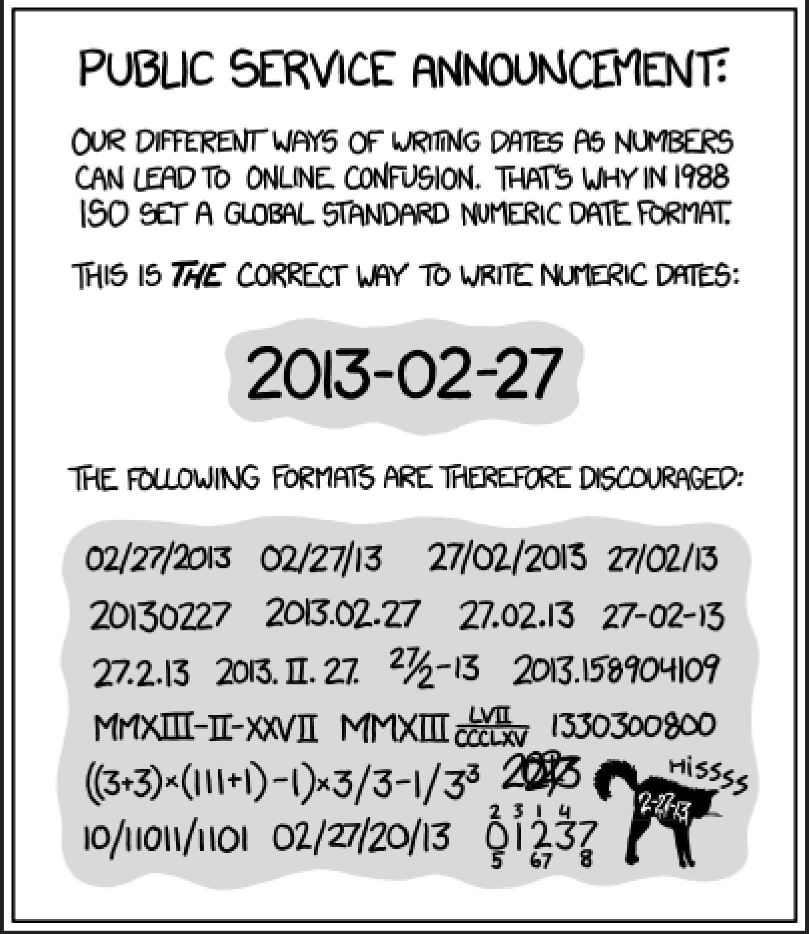
ABOUT ME:

- PHP & Symfony
- PHPers
- szkolenia, konsulatcje

PHPCE

SUMMIT2017

DATE FORMAT



PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. $^{27}\!/_{2}$-13 2013.158904109
MMXIII-II-XXVII MMXIII $\frac{\text{LVII}}{\text{CCCLXV}}$ 1330300800
$((3+3)\times(111+1)-1)\times3/3-1/3^3$ 2013
10/11011/1101 02/27/20/13 $\underset{5}{\overset{2}{0}}\overset{3}{1}\underset{67}{\overset{1}{2}}\overset{4}{3}\underset{8}{7}$

ISO 8601 was published on 06/05/88 and most recently amended on 12/01/04.

ISO 8601

2017-02-24T23:04:17Z

2017-02-24T23:04:17+00:00

RFC 3339

1985-04-12T23:20:50.52Z

1996-12-19T16:39:57-08:00

RFC 2616

Sun, 06 Nov 1994 08:49:37 GMT

UNIX TIMESTAMP

number of seconds since midnight, 1 January 1970 (UTC)

UNIX TIMESTAMP

1488024000

1 500 000 000

Fri Jul 14 04:40:00 CEST 2017

Avoid date formats not specifying time zone

~~2016-01-10 13:14:15~~

Programmers are not always aware of ISO 8601. Use it everywhere, UI can be an exception of this rule.

ISO WEEK NUMBER

The ISO 8601 definition for week 01 is the week with the Gregorian year's first Thursday in it.

LEAP YEARS

Criteria to identify leap year:

- year can be evenly divided by 4
- if year can be evenly divided by 100, it is not a leap year
- unless year can be evenly divided by 400

LEAP YEARS

- 1896
- 1904
- ...
- 1996
- 2000
- 2004

~~next year = year + 365 days~~

next year = year + 1 year

EASTER

COMPUTUS

The calculation used to determine the calendar date of Easter

Check if your programming language has function for Easter date

LEAP SECONDS

UNIVERSAL TIME (UT1)

Based on Earth's rotation

COORDINATED UNIVERSAL TIME (UTC)

Based on Atomic time

LEAP SECOND

- Earth slows down it's rotation
- extra second can be added or removed from UTC
- occurs on midnight UTC
- usually last day of June or December

LEAP SECOND INSERTION

- 23:59:58
- 23:59:59
- **23:59:60**
- 00:00:00

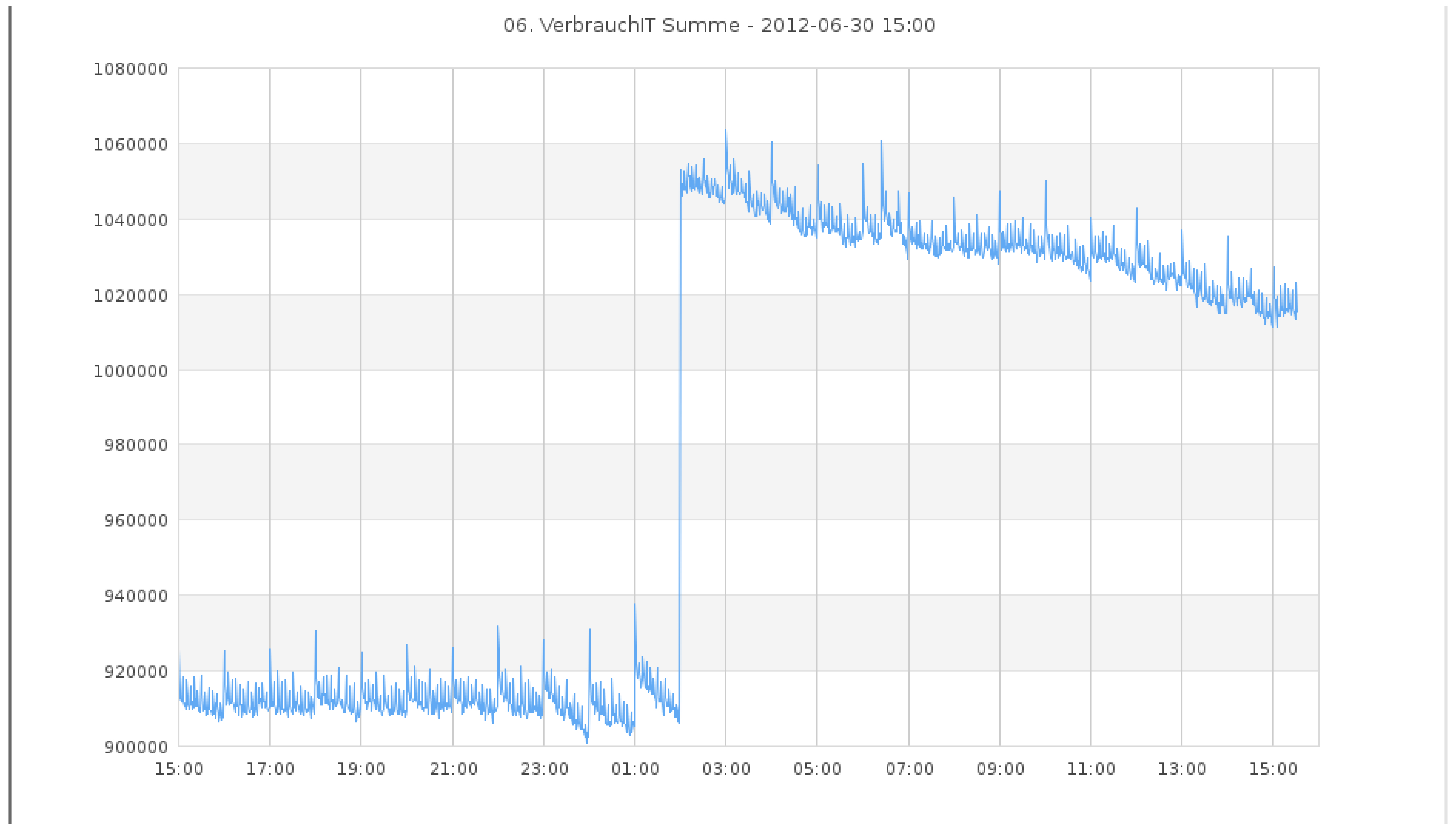
LEAP SECOND REMOVAL

- 23:59:57
- 23:59:58
- 00:00:00

Unix timestamp ignores leap seconds

LEAP SECOND INSERTION

- 1483228798
- 1483228799
- **1483228799**
- 1483228800



06. VerbrauchIT Summe - 2012-06-30 15:00

HOW TO HANDLE LEAP SECOND? (LINUX)

- kernel backward step
- daemon backward step
- ignore leap second and correct by slewing
- client slew
- server slew

TIME ZONES

TIME ZONE NAMING

- CET
- Europe/Warsaw

TIME ZONE OFFSET IS NOT ALWAYS FULL HOUR

Asia/Kolkata +05:30

Asia/Kathmandu +05:45

DAYLIGHT SAVING TIME (DST)

Time difference between Warsaw and Sydney

Date: 2017-02-25

Warsaw: UTC + 1

Sydney: UTC + 11

Difference: 10 hours

Date: 2017-03-27

Warsaw: UTC + 2

Sydney: UTC + 11

Difference: 9 hours

Date: 2017-04-03

Warsaw: UTC + 2

Sydney: UTC + 10

Difference: 8 hours

Date: 2017-10-02

Warsaw: UTC + 2

Sydney: UTC + 11

Difference: 9 hours

Date: 2017-10-30

Warsaw: UTC + 1

Sydney: UTC + 11

Difference: 10 hours

DST OFFSET IS NOT ALWAYS FULL HOUR

Australia/Lord_Howe +10:30

Australia/Lord_Howe +11:00 (DST)

Africa/Casablanca suspends DST for about a month (Ramadan)

DST in 2016 for Africa/Casablanca

from 2016-03-27 to 2016-06-05

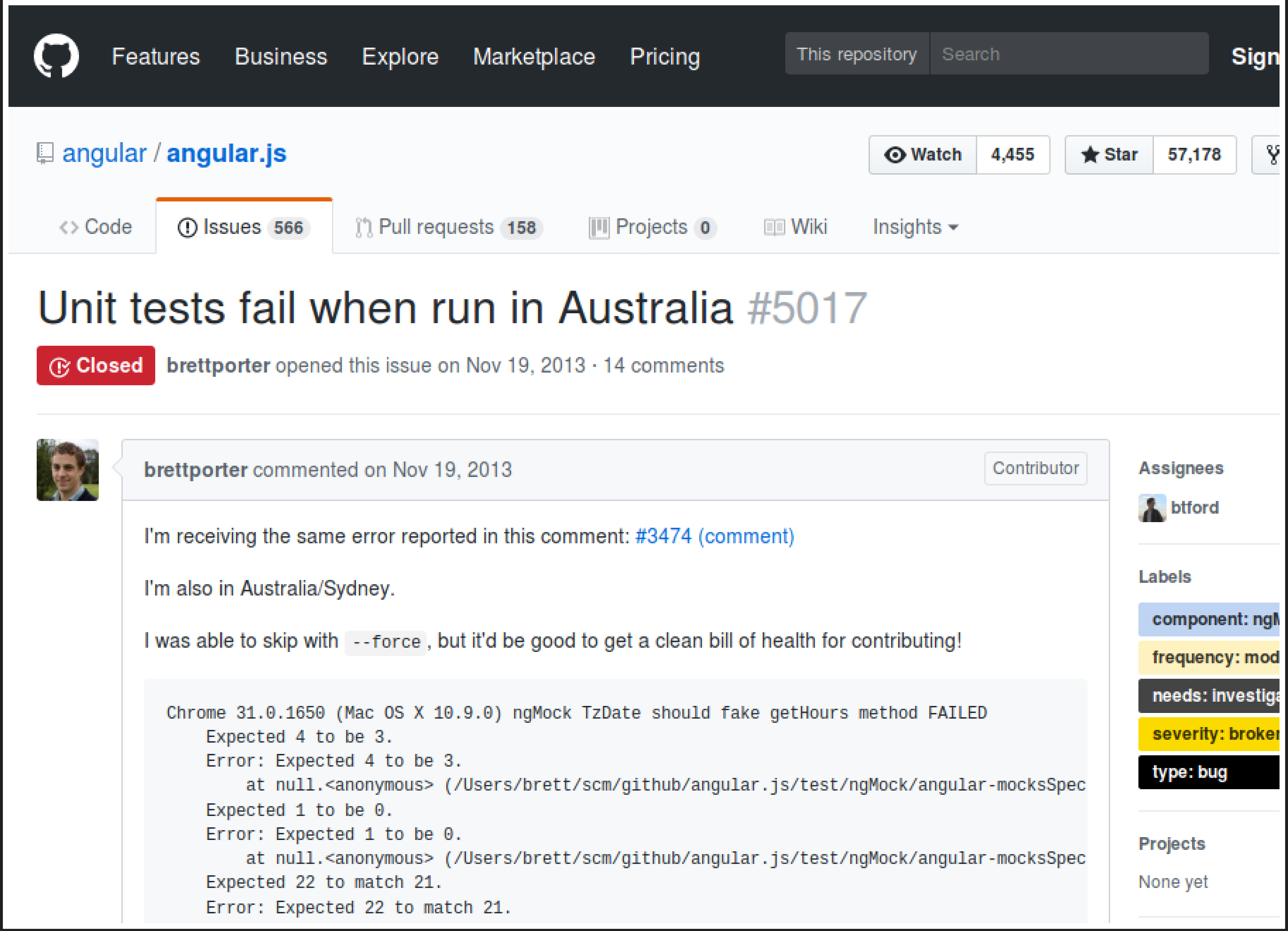
from 2016-07-10 to 2016-10-30

[Date Prev][Date Next] [Thread Prev][Thread Next] [Date Index] [Thread Index]

Bug#457938: tzdata: Argetina adds DST from 30 dec till 16 mar

- *To*: Debian Bug Tracking System <submit@bugs.debian.org>
- *Subject*: Bug#457938: tzdata: Argetina adds DST from 30 dec till 16 mar
- *From*: Margarita Manterola <marga@debian.org>
- *Date*: Thu, 27 Dec 2007 09:10:18 -0300
- *Message-id*: <20071227121018.532.30401.reportbug@ulises.amadeus>
- *Reply-to*: Margarita Manterola <marga@debian.org>, 457938@bugs.debian.org

```
Package: tzdata
Version: 2007j-2
Severity: normal

Hi!

As a show of how third world we are, our Congress approved yesterday a
law that says that we are to start using DST in 3 days.

The change means the time will go forward on December 30th 0 hour, and
it will go backwards on March 16th (the exact moment for the end is not
mentioned in any piece of information I could find, but I suspect we
will have two 23 hours).

Just to be clear, from Dec 30 till Mar 16 we will be GMT-2 and then
back to GMT-3

But, the president will go on deciding the dates for the DST change as
time passes, so it won't be the same for next year.

Link to news articles about this:
http://www.lacapital.com.ar/contenidos/2007/12/27/noticia_0024.html
http://www.lanacion.com.ar/EdicionImpresa/economia/nota.asp?nota_id=974279&pid=3720941&toi=5257
```



Unit tests fail when run in Australia #5017

Closed **brettporter** opened this issue on Nov 19, 2013 · 14 comments

brettporter commented on Nov 19, 2013

I'm receiving the same error reported in this comment: #3474 (comment)

I'm also in Australia/Sydney.

I was able to skip with `--force`, but it'd be good to get a clean bill of health for contributing!

```
Chrome 31.0.1650 (Mac OS X 10.9.0) ngMock TzDate should fake getHours method FAILED
    Expected 4 to be 3.
    Error: Expected 4 to be 3.
        at null.<anonymous> (/Users/brett/scm/github/angular.js/test/ngMock/angular-mocksSpec
    Expected 1 to be 0.
    Error: Expected 1 to be 0.
        at null.<anonymous> (/Users/brett/scm/github/angular.js/test/ngMock/angular-mocksSpec
    Expected 22 to match 21.
    Error: Expected 22 to match 21.
```

TIME AND DATE

HOW TO MEASURE ELAPSED TIME?

```
start = now()
f()
end = start - now()
```

If you want to measure elapsed time use monotonic clock

If your code depends on time make it an input parameter

Instead of this:

```
function f()
{
    $now = time();
    // ...
}
```

Do this:

```php
function f($time)
{
    $now = $time;
    // ...
}
```

In your application do not call system clock

```
interface Clock
{
    public function now(): DateTimeImmutable;
}
```

```php
class SystemClock
{
    public function now(): DateTimeImmutable
    {
        return new DateTimeImmutable();
    }
}
```

THANK YOU!