

*nette*  
FRAMEWORK

Radek Ježdík

# Zadání

- [#1258](#) - Nette\DateTime: inherited createFromFormat returns DateTime
- [#1430](#) - RANGE validations with float value fails on czech locale

# Opraveno

- [#1258](#) - Nette\DateTime: inherited createFromFormat returns DateTime
- ~~[#1430](#) - RANGE validations with float value fails on czech locale~~

# #1258

- třída `Nette\DateTime` dědí od `DateTime` (PHP)
- statická factory metoda `createFromFormat` ale stále vrací `DateTime` (špatný design a nepředvídatelnost)

```
createFromFormat($format, $time[, $timezone])
```

```
createFromFormat('Y-m-d', '2014-05-14')
```

```

89 +
90 +     /**
91 +      * Returns new DateTime object formatted according to the specified format.
92 +      * @param string The format the $time parameter should be in
93 +      * @param string String representing the time
94 +      * @param string|\DateTimeZone desired timezone (default timezone is used if NULL is passed)
95 +      * @return DateTime
96 +      */
97 +     public static function createFromFormat($format, $time, $timezone = NULL)
98 +     {
99 +         if ($timezone === NULL) {
100 +             $timezone = new \DateTimeZone(date_default_timezone_get());
101 +
102 +         } elseif (is_string($timezone)) {
103 +             $timezone = new \DateTimeZone($timezone);
104 +
105 +         } elseif (!$timezone instanceof \DateTimeZone) {
106 +             throw new Nette\InvalidArgumentException('Invalid timezone given');
107 +         }
108 +
109 +         return static::from(parent::createFromFormat($format, $time, $timezone));
110 +     }
111 +
89 112 }

```

## Přetížená metoda

```
1 +<?php
2 +
3 +/**
4 + * Test: Nette\DateTime::createFromFormat().
5 + *
6 + * @author      Radek Ježdík
7 + */
8 +
9 +use Tester\Assert,
10 +     Nette\Utils\DateTime;
11 +
12 +require __DIR__ . '/../bootstrap.php';
13 +
14 +
15 +date_default_timezone_set('Europe/Prague');
16 +
17 +Assert::type( 'Nette\Utils\DateTime', DateTime::createFromFormat('Y-m-d H:i:s', '2050-08-13 11:40:00') );
18 +Assert::type( 'Nette\Utils\DateTime', DateTime::createFromFormat('Y-m-d H:i:s', '2050-08-13 11:40:00', new DateTimeZone('Europe/Prague')) );
19 +
20 +Assert::same( '2050-08-13 11:40:00', (string) DateTime::createFromFormat('Y-m-d H:i:s', '2050-08-13 11:40:00') );
21 +
22 +Assert::same( 'Europe/Prague', DateTime::createFromFormat('Y', '2050')->getTimezone()->getName() );
23 +Assert::same( 'Europe/Bratislava', DateTime::createFromFormat('Y', '2050', 'Europe/Bratislava')->getTimezone()->getName() );
24 +
25 +Assert::error(function(){
26 +     DateTime::createFromFormat('Y-m-d H:i:s', '2050-08-13 11:40:00', 5);
27 +}, 'Nette\InvalidArgumentException', 'Invalid timezone given' );
```

## Test metody

# Zhodnocení

- povedlo se začlenění jedné opravy bugu
- druhý bug se ukázal být složitější a po komunikaci s vývojáři by potřeboval větší refactoring

# Spolupráce s komunitou

- velmi přátelská komunikace, na úrovni
- velký důraz na kvalitu
- na to, že to byla jedna jednoduchá metoda, bylo hodně úprav a komunikace



**Díky za pozornost**