

Au moins 4 façons

de définir un `DateTime` avec un `Time` à 0



Mathieu Santostefano

 Web developer at [joliCode](#)

 Symfony Core Team Member

 Twitter @welcomattic

 GitHub @welcomattic

 Slack Symfony @welcomattic

Créer un `DateTime` à partir d'une date, ça a un inconvénient

```
1 $dateTime = \DateTime::createFromFormat(format: 'Y-m-d', datetime: '2023-03-23');
2 dump($dateTime);
3
4 ^ DateTime @1679576390 {#3
5     date: 2023-03-23 13:57:29.0 Europe/Paris (+01:00)
6 }
```

Créer un `DateTime` à partir d'une date, ça a un inconvénient

```
1 $dateTime = \DateTime::createFromFormat(format: 'Y-m-d', datetime: '2023-03-23');
2 dump($dateTime);
3
4 ^ DateTime @1679576390 {#3
5     date: 2023-03-23 13:57:29.0 Europe/Paris (+01:00)
6 }
```

Comment reset le temps d'un objet DateTime ?

Comment reset le temps d'un objet DateTime ?

Qui a une idée ?

Avec la méthode `setTime`

```
1 $dateTime = \DateTime::createFromFormat('Y-m-d', '2023-03-23');
2 $dateTime = $dateTime->setTime(0, 0);
3 dump($dateTime);
```

Avec la méthode `setTime`

```
1 $dateTime = \DateTime::createFromFormat('Y-m-d', '2023-03-23');
2 $dateTime = $dateTime->setTime(0, 0);
3 dump($dateTime);
```

```
1 ^ DateTime @1679576390 {#3
2     date: 2023-03-23 00:00:00.0 Europe/Paris (+01:00)
3 }
```

Avec un !

```
1 $dateTime = \DateTime::createFromFormat('!Y-m-d', '2023-03-23');  
2 dump($dateTime);
```

`!` reset en réalité tous les "champs" qui ne sont pas précisés dans le paramètre `format` (à 0 ou 1 ou 1970 selon le champ)

Avec un !

```
1 $dateTime = \DateTime::createFromFormat('!Y-m-d', '2023-03-23');  
2 dump($dateTime);
```

```
1 ^ DateTime @1679576390 {#3  
2     date: 2023-03-23 00:00:00.0 Europe/Paris (+01:00)  
3 }
```

`!` reset en réalité tous les "champs" qui ne sont pas précisés dans le paramètre `format` (à 0 ou 1 ou 1970 selon le champ)

Avec un |

```
1 $dateTime = \DateTime::createFromFormat('Y-m-d| ', '2023-03-23');
2 dump($dateTime);
```

`|` reset tous les "champs" qui n'ont pas encore été parse, donc ceux qui le suivent dans le paramètre `datetime`

Avec un |

```
1 $dateTime = \DateTime::createFromFormat('Y-m-d| ', datetime: '2023-03-23');
2 dump($dateTime);
3
4 ^ DateTime @1679576390 {#3
5     date: 2023-03-23 00:00:00.0 Europe/Paris (+01:00)
6 }
```

`|` reset tous les "champs" qui n'ont pas encore été parse, donc ceux qui le suivent dans le paramètre `datetime`

Avec une date + `midnight` (mais ça marche aussi à midi ou à 16h52)

```
1 $dateTime = new \DateTime('2023-03-23 midnight');
2 var_dump($dateTime);
```

Avec une date + `midnight` (mais ça marche aussi à midi ou à 16h52)

```
1 $dateTime = new \DateTime('2023-03-23 midnight');
2 var_dump($dateTime);


---


1 ^ DateTime @1679576390 {#3
2     date: 2023-03-23 00:00:00.0 Europe/Paris (+01:00)
3 }
```



```
1 $firstDayOfNextMonth = new \DateTime('first day of next month midnight'); // "2023-04-01 00:00:00.000000"
2 $tomorrowAtNoon = new \DateTime('tomorrow noon'); // "2023-03-24 12:00:00.000000"
3 $lastDayOfTheLastWeek = new \DateTime('last day of last week midnight'); // "2023-03-31 00:00:00.000000"
```

All fields are initialised with the current date/time. In most cases you would want to reset these to "zero" (the Unix epoch, 1970-01-01 00:00:00 UTC). You do that by including the ! character as first character in your format, or | as your last. Please see the documentation for each character below for more information.

Merci 

www.php.net/manual/en/datetimeimmutable.createfromformat.php

jolicode.com/blog

blog.welcomatic.com