

Can your software travel in time?





Arne Blankerts

Co-Founder, The PHP Consulting Company

So, what is Time?

2024-02-22 10:00:00

2024-02-22 10:00:00 EST

Unix

Unix Time Stamp

Unix Time Stamp

Seconds since Jan 01 1970 UTC

Unix Time vs. January 19, 2038

How long is a Day?

How long is a Day?

24 Hours?

How long is a minute?

How long is a minute?

60 seconds?

Realtime

```
echo date('Y-m-d H:i:s', \time());
```

```
echo (new DateTimeImmutable('now'))->format('Y-m-d H:i:s');
```



```
CREATE TABLE `records` (  
  `id` BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  // ....  
  PRIMARY KEY (`id`),  
  ) ENGINE=INNODB
```

Houston, we have a Problem!

Let's Fix Time

Let's Fixate Time

```
$now = '@1705500150';  
echo (new \DateTimeImmutable($now)) ->format('Y-m-d H:i:s');
```

Request Time

```
$_SERVER["REQUEST_TIME_FLOAT"]  
$_SERVER["REQUEST_TIME"]
```

```
$now = '@' . $_SERVER["REQUEST_TIME"] ?? \time();  
echo (new \DateTimeImmutable($now)) ->format('Y-m-d H:i:s');
```



```
$now = '@' . $_SERVER["REQUEST_TIME_FLOAT"] ?? \time();  
echo (new \DateTimeImmutable($now))->format('Y-m-d H:i:s');
```

Time Travel?

Time Travel

Approach #1

RFC 5905

RFC 5905

Network Time Protocol

NO.

Time Travel

Approach #2

libfaketime

<https://github.com/wolfcw/libfaketime>

Live Demo

libfaketime with Containers

```
FROM fedora:39
```

```
RUN dnf install -y libfaketime && dnf clean all
```

```
ENV LD_PRELOAD=/usr/lib64/faketime/libfaketime.so.1
```

libfaketime with Containers

```
FROM fedora:39  
RUN dnf install -y libfaketime && dnf clean all  
  
ENV LD_PRELOAD=/usr/lib64/faketime/libfaketime.so.1
```

```
$ podman build --tag faketime -f Containerfile
```

libfaketime with Containers

```
FROM fedora:39
```

```
RUN dnf install -y libfaketime && dnf clean all
```

```
ENV LD_PRELOAD=/usr/lib64/faketime/libfaketime.so.1
```

```
$ podman build --tag faketime -f Containerfile
```

```
$ podman run --rm -it --env FAKETIME=-2d faketime date
```

Time Travel

Approach #3

Application Internal

PSR-20

```
namespace Psr\Clock;  
  
interface ClockInterface {  
    public function now() : \DateTimeImmutable;  
}
```

<https://www.php-fig.org/psr/psr-20/>

**Do not depend on interfaces
controlled by others**

PHP

```
namespace our\app;  
  
interface Clock {  
    public function now(): \DateTimeImmutable;  
}
```

PHP & PSR-20

```
class PsrAdapter implements Clock {  
  
    public function __construct(  
        private Psr\Clock\ClockInterface $other  
    ) {}  
  
    public function now(): \DateTimeImmutable {  
        return $this->other->now();  
    }  
}
```

Do we need an actually
functioning clock?

```
namespace our\app;

class DefinedSequenceClock implements Clock {

    /** @var \DateTimeImmutable[]
    private array $times;
    private int $pos = 0;

    public function __construct(\DateTimeImmutable ...$times) {
        $this->times = $times;
    }

    public function now(): \DateTimeImmutable {
        if ($this->pos === count($this->times)) {
            throw new OutOfRangeException('No entries left.');
```

```
namespace our\app;

$clock = new SequenceClock(
    new DateTimeImmutable('2 days ago 00:00'),
    new DateTimeImmutable('yesterday 00:00'),
    // ...
);
```

```
namespace our\app;  
  
$clock = new SequenceClock(  
    ...$this->sequenceLoader()  
);
```

CRUD

~~CRUD~~

Event Sourcing!

ID	CREATED_AT	AGGREGATE	TYPE	PAYLOAD
1	2024-02-01 09:14:00	b6bc6997-2e71-410e-803d-496c7e08eded	AccountCreated	{....}
2	2024-02-01 09:16:41	b6bc6997-2e71-410e-803d-496c7e08eded	NotificationSent	{....}
3	2024-02-02 11:38:12	b6bc6997-2e71-410e-803d-496c7e08eded	PasswordSet	{....}
4	2024-02-03 08:18:12	b6bc6997-2e71-410e-803d-496c7e08eded	{....}
5	2024-02-03 09:23:53	b6bc6997-2e71-410e-803d-496c7e08eded	{....}

```
// ....

$eventClock = new SequenceClock(
    $this->loadSequenceFor('b6bc6997-2e71-410e-803d-496c7e08eded')
);

// ....

$this->emit(
    new AccountCreated(
        $eventClock->now(),
        // ...
    )
);
```

Time Travel!

Time Travel!

Finally!

```
// ....

$customer = Customer::fromEvents(
    $eventStore->loadUntil(
        'b6bc6997-2e71-410e-803d-496c7e08eded',
        new DateTimeImmutable('1 week ago 12:34:56')
    )
);
```

Thank you.

 @arne@thephpcc.social

 @theseer@phpc.social

 <https://thephp.cc>

 arne@thephp.cc