Caching with PSR-6

Laravel Barcelona 🇪🇸

@laravelbcn

@hannesvdvreken

Hi, my name is Hannes.

madewithlove.be

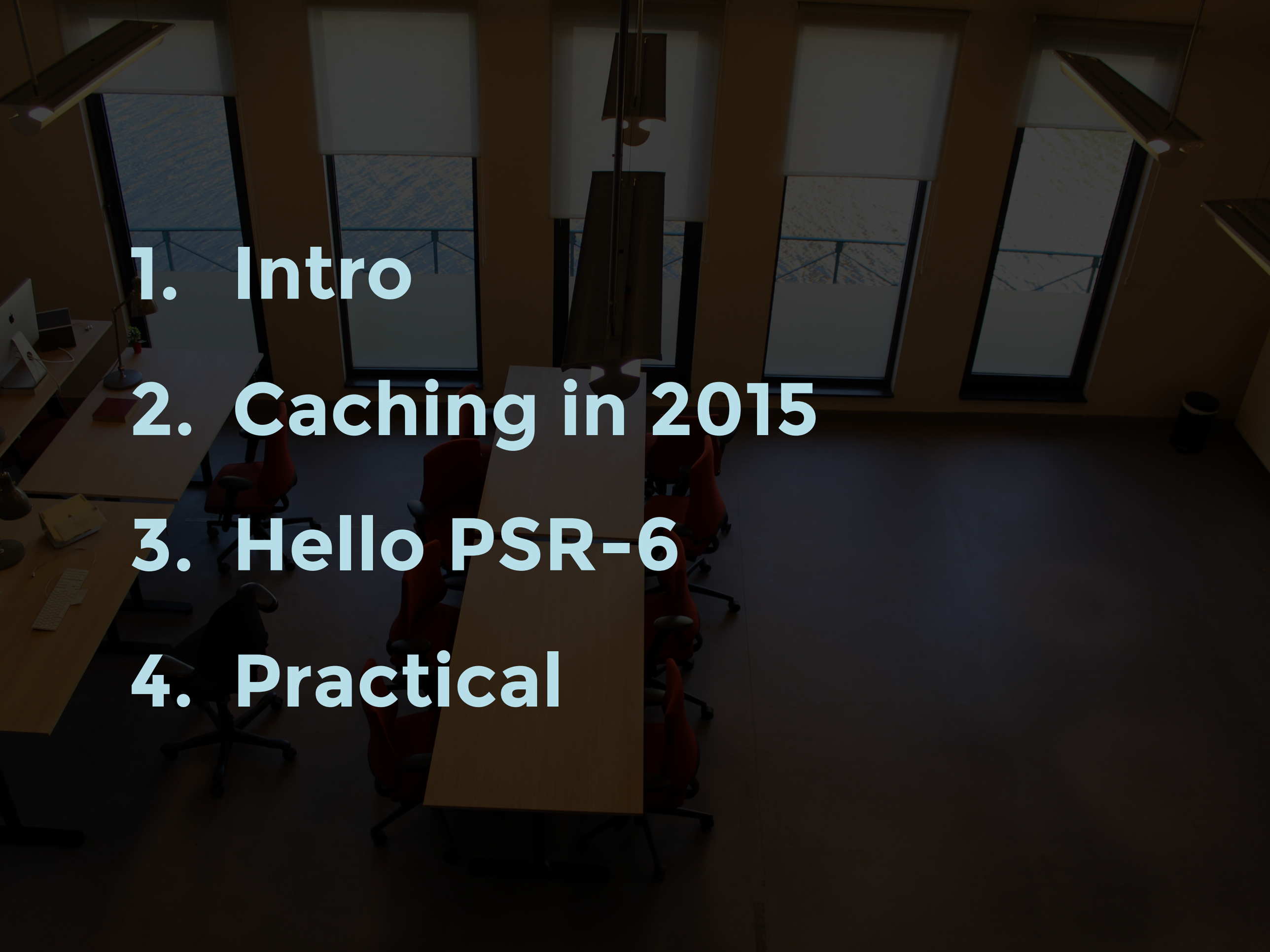
20 × 😍

🇧🇪 🇵🇹 🇫🇷 🇰🇪 🇨🇦 🇧🇷

300m
350m
300m
200m
250m
Parc de l'Oreneta
150m
SARRIÀ
100m
100m
B-20
Club de tennis
edralbes
Puig d'Ossa
300m
100m
PEDRALBES
150m
200m
250m
UPC Campus
Nord
Avingu
50m
LES CORTS
Parc de Cervantes
150m
Sant Just
Desvern
Ronda de Dalt
100m
50m
Ronda del Mig
Esplugues de
Llobregat
B-23
50m
N-340
Parc dels Torrents

PSR-6

Caching in PHP anno 2016



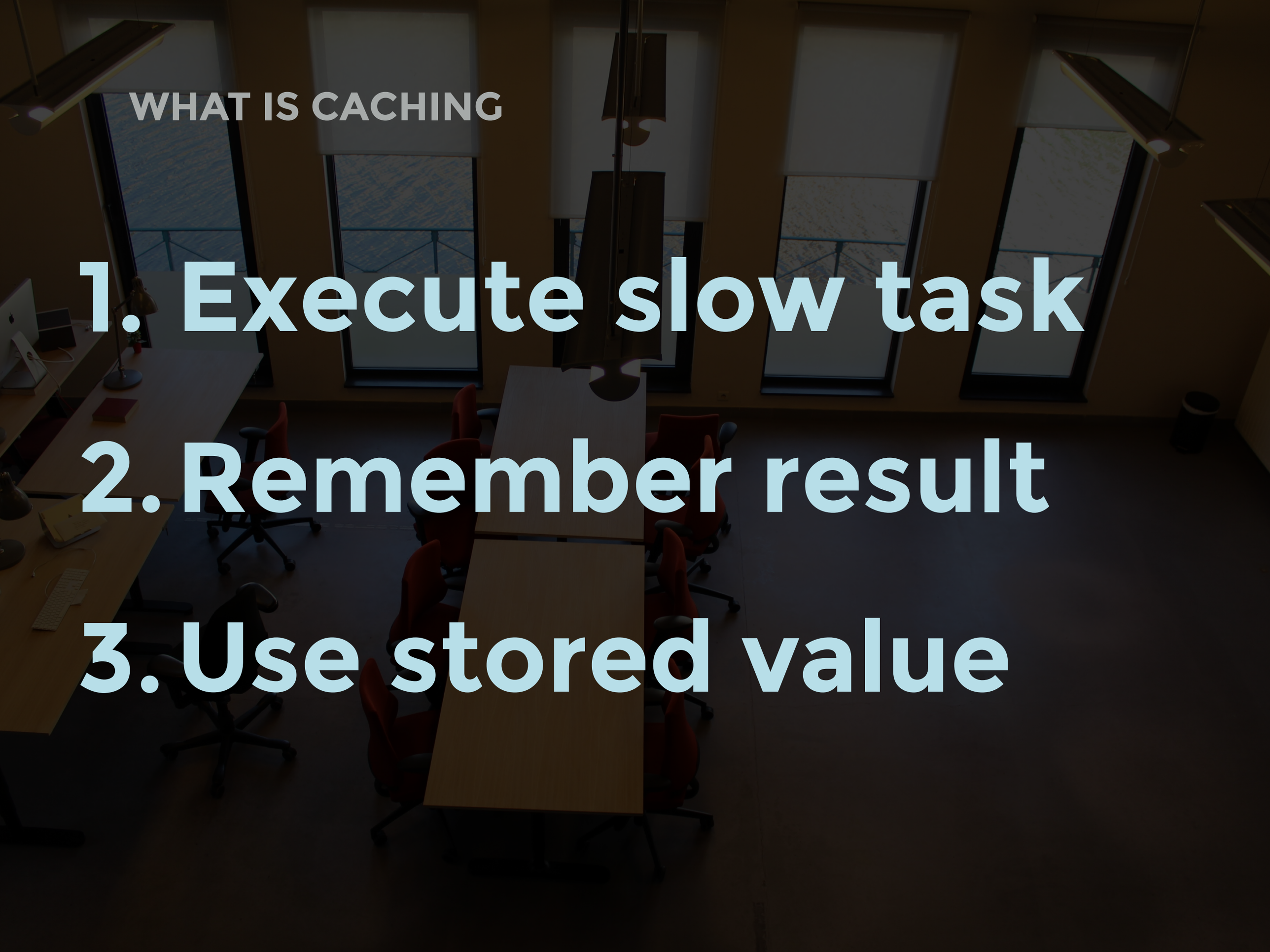
1. Intro
2. Caching in 2015
3. Hello PSR-6
4. Practical

1. Intro

what is caching?

WHAT IS CACHING

Speeding up your app



WHAT IS CACHING

1. Execute slow task

2. Remember result

3. Use stored value

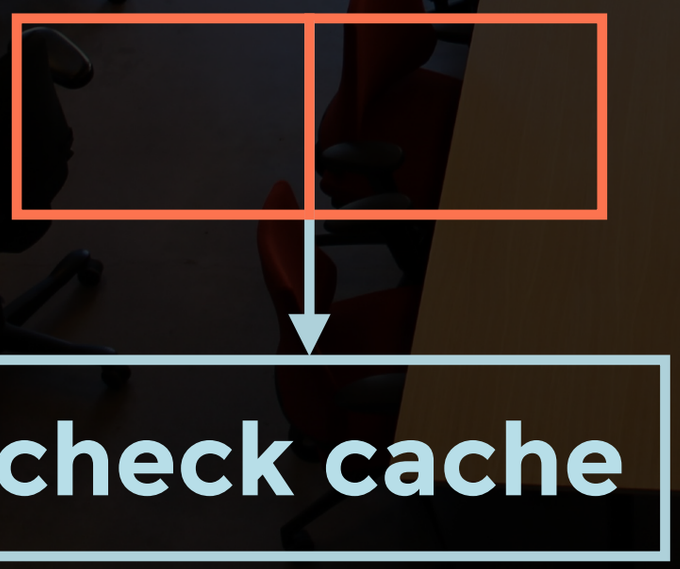
Should not cause app failure

WHAT IS CACHING - EXAMPLES

- HTTP Request
- Slow DB call
- Process image/zip/...

Should not cause app failure

WHAT IS CACHING



Should not cause app failure

At different layers:

User-Agent

Webserver (Varnish/Nginx)

Application

Opcache

Should not cause app failure

2. Caching in 2015

the state of caching (pre PSR-6)

APPLICATION CACHING IN 2015

1. Libraries do it

2. Frameworks do it

3. Cache libs do it

APPLICATION CACHING IN 2015

Frameworks & cache libs have their own:

- interfaces
- support for caching systems

APPLICATION CACHING IN 2015

Adapters everywhere!

If no adapter available

- Write your own
- Store in different cache stores

3. Hello PSR-6

what does that mean?

INTRODUCING PSR-6

Finalised & accepted in December 2015

INTRODUCING PSR-6

Repository - Entity

INTRODUCING PSR-6

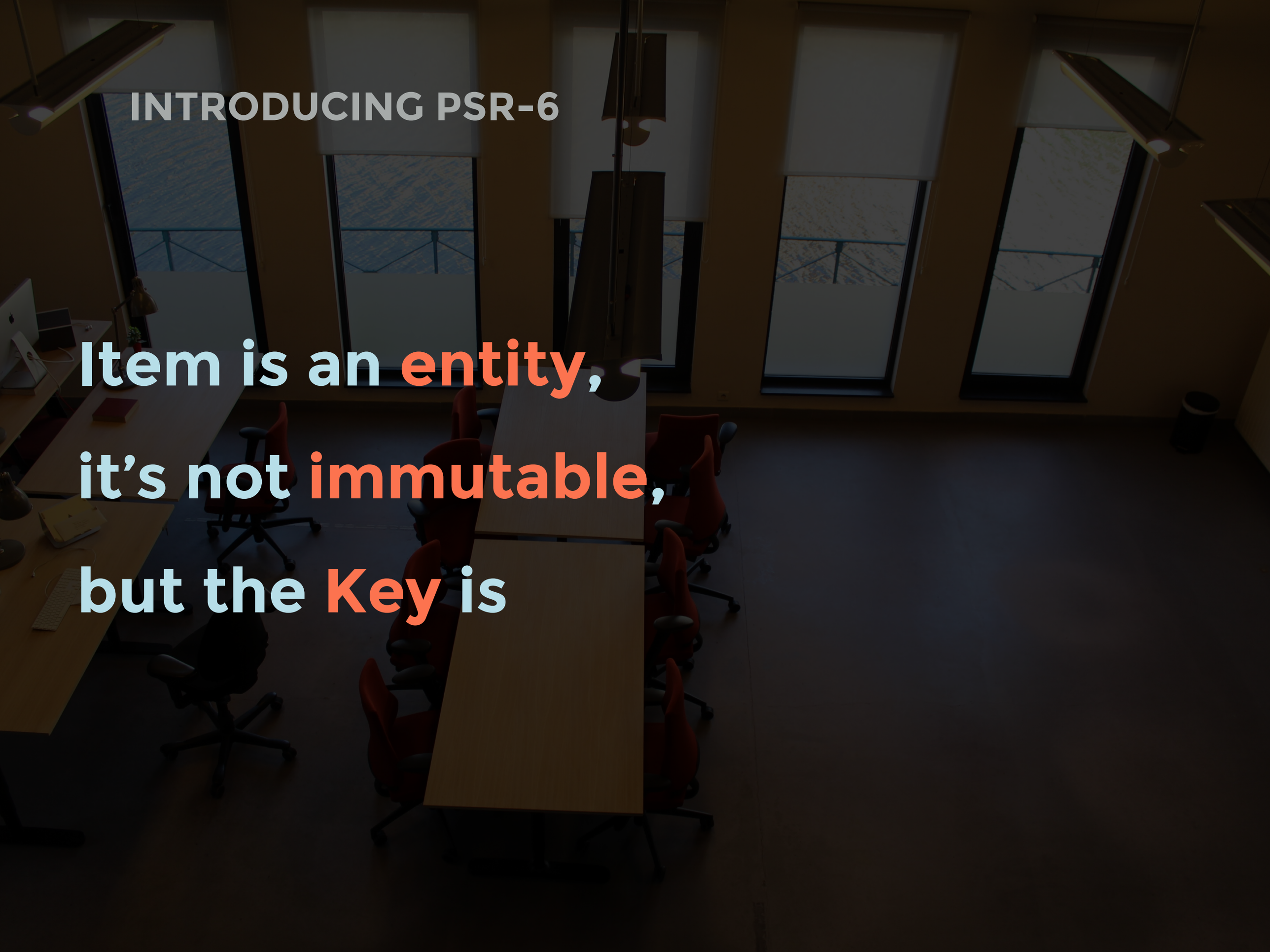
```
use Psr\Cache\CacheItemPoolInterface;
use Psr\Cache\CacheItemInterface;

$item = $pool->getItem($key);

$item->getKey();
```

INTRODUCING PSR-6

Lemme see that code

Don’t instantiate your own Items.

```
$item = $pool->getItem(‘key’)
    ->set($value)
    ->expiresAfter(3600);
$pool->save($item);
```



INTRODUCING PSR-6

Item is an entity,

it's not immutable,

but the Key is

INTRODUCING PSR-6

CacheItemInterface

has no

getExpiresAt

Pool has support for multi-actions

```
$pool->getItems($keys);
$pool->saveDeferred($item);
$pool->commit();
```

INTRODUCING PSR-6

Repository - Entity model

allows extensions

INTRODUCING PSR-6

Cache features:

- Stampede protection:

Parallel incoming requests executing long process to update cache value

INTRODUCING PSR-6

Cache features:

- Stampede protection
- Taggable cache

```
$item->addTags(['cat-1']);
$pool->clearTags(['cat-1']);
```

INTRODUCING PSR-6

Cache features:

- Stampede protection
- Taggable cache
- Hierarchical cache

```
$pool->delete('tree/*');
```

4. Practical

who uses it, and how can I use it?

Packagist | Dependent Packages - psr/ | mwl

https://packagist.org/providers/psr/cache-implementation

Packagist *The PHP Package Repository* Browse Submit hannesvdvreken

Search packages...

Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer.

The following packages provide psr/cache-implementation

tedivm/stash

The place to keep your cache.

PHP ↓ 414 515 ★ 657

symfony/cache

Symfony implementation of PSR-6

PHP ↓ 13 429 ★ 13

cache/hierarchical-cache

A helper trait and interface to your PSR-6 cache to support hierarchical keys.

PHP ↓ 13 284 ★ 0

cache/array-adapter

A PSR-6 cache implementation using a php array. This implementation supports tags

PHP ↓ 4 659 ★ 2

cache/memcached-adapter

A PSR-6 cache implementation using Memcached. This implementation supports tags

PHP ↓ 6 302 ★ 3

Packagist | cache/filesystem-adapter - | mwl

https://packagist.org/packages/cache/filesystem-adapter

Packagist *The PHP Package Repository* Browse Submit hannesvdvreken

Search packages...

★ cache/filesystem-adapter

`composer require cache/filesystem-adapter`

A PSR-6 cache implementation using filesystem. This implementation supports tags

0.3.0 2016-02-25 08:03 UTC

requires

- php: ^5.5|^7.0
- psr/cache: ^1.0
- cache/adapter-common: ^0.3
- cache/taggable-cache: ^0.4
- league/flysystem: ^1.0

requires (dev)

- phpunit/phpunit: ^4.0|^5.1
- cache/integration-tests: ^0.9

suggests

None

provides

- psr/cache-implementation: ^1.0

conflicts

None

replaces

None

START USING IT

Integration tests for implementations:

cache/integration-tests

Packagist | cache/integration-tests - P | mwl

https://packagist.org/packages/cache/integration-tests

Packagist *The PHP Package Repository* Browse Submit hannesvdvreken

Search packages...

★ cache/integration-tests

`composer require cache/integration-tests`

Integration tests for PSR-6 cache implementations

0.10.0 2016-04-23 22:57 UTC

requires

- php: ^5.4|^7
- psr/cache: ~1.0

requires (dev)

- phpunit/phpunit: ^4.0|^5.0

suggests

None

provides

None

conflicts

None

replaces

None

© MIT 43677ae075f81e38dff3d4645ecc9b0ace74a6fc

Aaron Scherer <aequasi@gmail.com>, Tobias Nyholm <tobias.nyholm@gmail.com>

#cache #test #psr6

START USING IT

```
class PoolIntegrationTest extends CachePoolTest
{
    public function createCachePool()
    {
        return new CachePool();
    }
}
```

Upgrade paths

START USING IT - UPGRADE

Libraries

START USING IT - UPGRADE

Libraries (next major versions) have a PSR-6 caching decorator

Packagist | Dependent Packages - psr/ | mwl

https://packagist.org/packages/psr/cache-implementation/dependents

Packagist *The PHP Package Repository* Browse Submit hannesvdvreken

Search packages...

psr/cache-implementation dependents (6)

aequasi/cache-bundle

PHP ⬇ 72 355 ★ 36

Creates services in Symfony 2, for cache, that can also be used with doctrines three cache types. It also provides functionality for session handler support, and Router support.

cache/psr-6-doctrine-bridge

PHP ⬇ 3 265 ★ 2

PSR-6 Doctrine bridge

markup/contentful

PHP ⬇ 6 597 ★ 8

A client library to integrate with the Contentful APIs (Content Delivery and Content Management) for PHP 5.4+.

markup/contentful-bundle

PHP ⬇ 5 314 ★ 3

A Symfony2 bundle to integrate with the Contentful APIs (Content Delivery and Content Management) for PHP 5.4+.

markup/wistia-bundle

PHP ⬇ 856 ★ 1

The Markup Wista Bundle contains functionality for communicating with the Wistia Data API

START USING IT - UPGRADE

Frameworks

START USING IT - UPGRADE

Start using PSR-6 enabled libraries with adapters for current FW's implementation.

Example:

Laravel PSR-6 bridge

Packagist | madewithlove/illuminate-ps | mwl

https://packagist.org/packages/madewithlove/illuminate-psr-cache-bridge

Packagist *The PHP Package Repository* Browse Submit hannesvdvreken

Search packages...

★ madewithlove/illuminate-psr-cache-bridge

`composer require madewithlove/illuminate-psr-cache-bridge`

Abandon Delete Update Edit

1.0.0 2016-06-04 19:28 UTC

requires

- php: >=5.5.0
- illuminate/contracts: ^5.0
- psr/cache: ^1.0

requires (dev)

- illuminate/cache: ^5.0
- phpunit/phpunit: ^4.8
- nesbot/carbon: ^1.3.0

suggests

None

provides

- psr/cache-implementation: 1.0.0

conflicts

None

replaces

None

MIT 82a8cd0cab73e94c4684cf1a6d9e3ee1666096ec

RECAP

1. Intro
2. Caching in 2015
3. Hello PSR-6
4. Practical

[VOTE][Entrance] Simple Cache PSR

Jordi Boggiano 1:18 PM (28 minutes ago)

The vote ended Saturday, and passed! Thanks everyone.

Here are the stats:

Number of Non-Voters	13
Positive Votes	25
Negative Votes	1
Abstain Votes	0
Voters	26
Total Number of Members	39
Quorum Number	13
Quorum Met	Yes
Percentage Voted	66%

Hello PSR-16

Thank you!

https://joind.in/talk/xxxxx

@laravelbcn

@hannesvdvreken

Questions & discussions

@laravelbcn

@hannesvdvreken