ibexa

Summit 25

What's new for Ibexa DXP developers in 2025?

Adam Wójs, Director of Engineering @ Ibexa

$ whoami

Adam Wójs

- Director of Engineering @ Ibexa
- > 14 years of experience
- Support > Engineering > Leadership

/adamwojs /adamwojs

ibexa

Agenda

- Key featrures
- Upgrade to Symfony 7
- New Extension Points & APIs
- Documentation

ibexa

Key Features

AI

Collaboration

Discounts

Upgrade to Symfony 7

$ Upgrade to Symfony 7

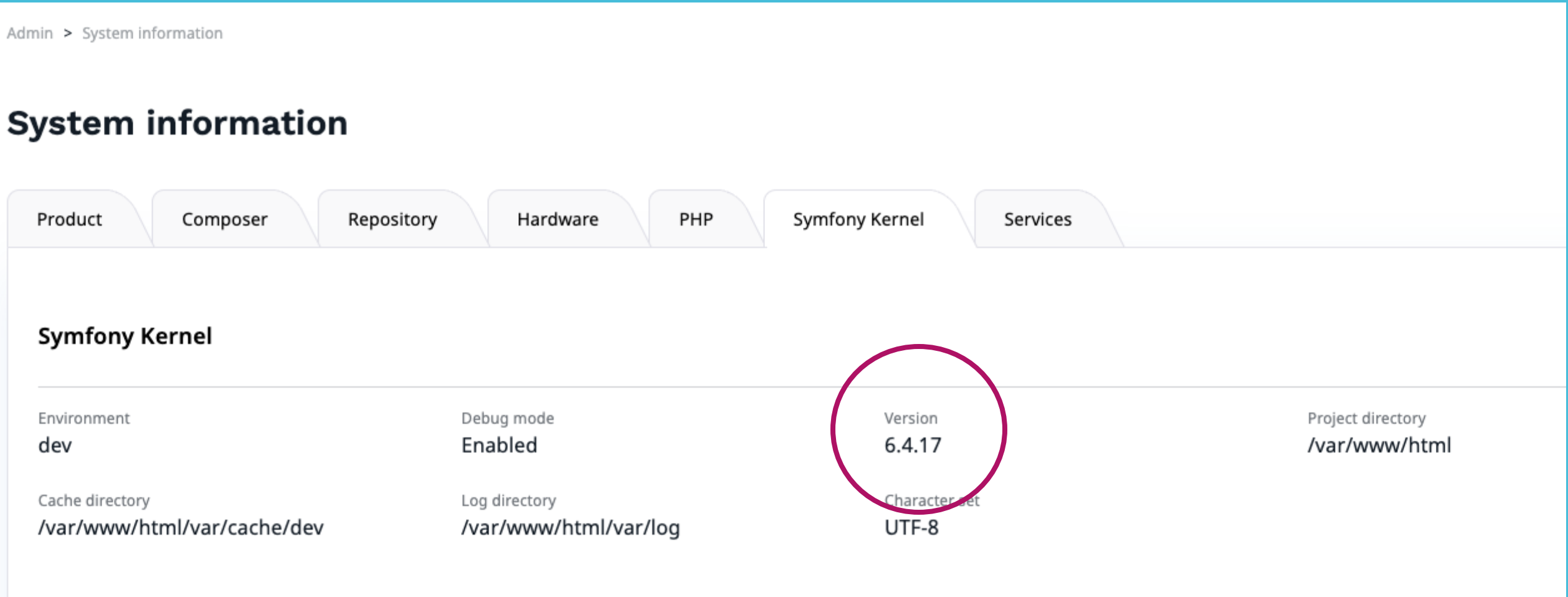
IN PROGRESS

- ✓ Bump minimal PHP version to 8.3
- ✓ Resolve Ibexa DXP deprecations
- ✓ Resolve Symfony 5.4 deprecations (including security layer)
- ⧗ Bump Symfony version to 6.4
- Resolve Symfony 6.x deprecations
- Bump Symfony version to 7.x

ibexa

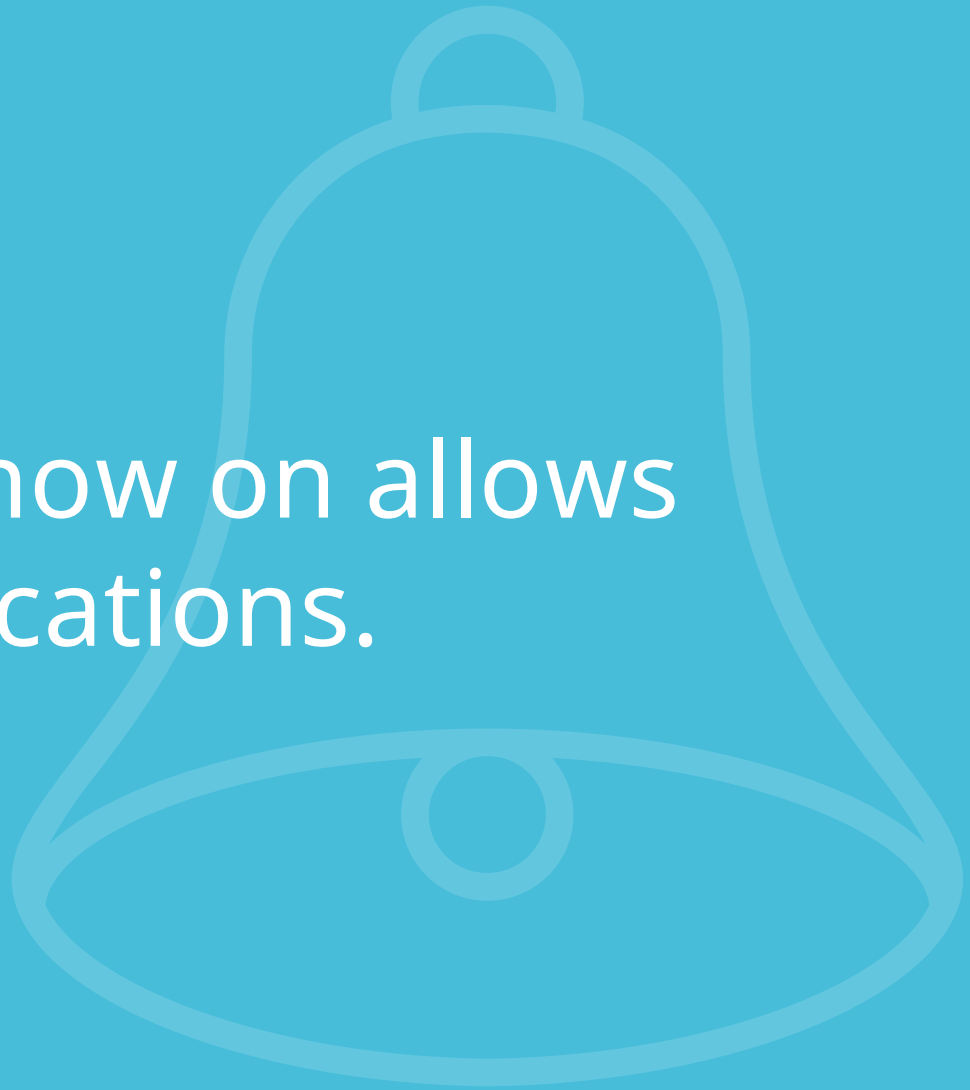
$ Upgrade to Symfony 7

IN PROGRESS



New Extension Points & APIs

$ Sending notifications



ibexa/notification from now on allows you to send internal notifications.

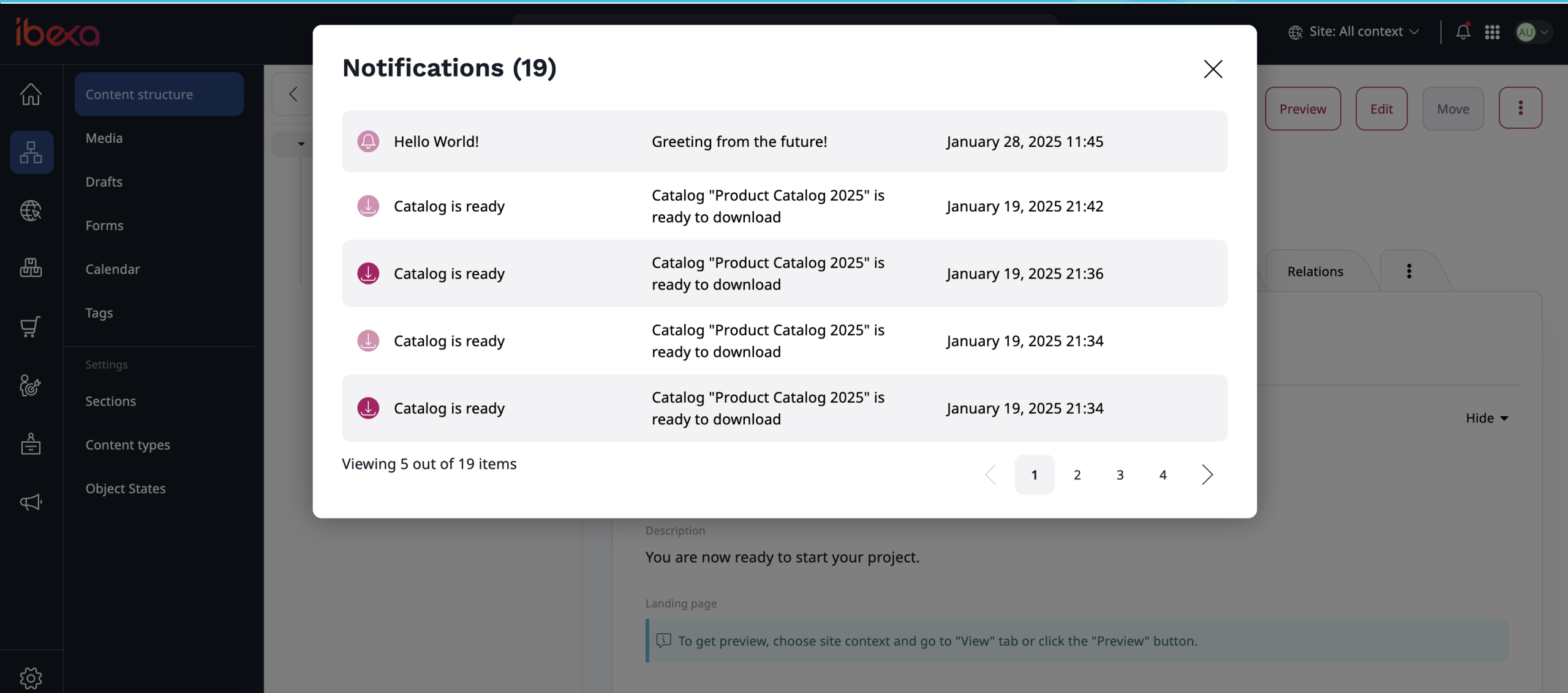
https://github.com/ibexa/notifications/pull/11

ibexa

$ Sending notifications

```php
$user = $userService->loadUserByLogin('john.smith@example.com')

$notification = new SystemNotification('Hello John!', ['ibexa']);
$notification->setIcon('bell');
$notification->setContent("It's works!");
$notification->setRoute(
  new RouteReference(
    'ibexa.content.view',
    [
      'contentId' => 52,
    ]
  )
);

$notifier->send($notification, new UserRecipient($user));
```

$ Sending notifications (demo)



$ Custom notification

```php
final class WelcomeNotification extends Notification implements SystemNotificationInterface
{
    public function asSystemNotification(UserRecipientInterface $recipient, ?string $transport = null): ?SystemMessage
    {
        $message = new SystemMessage($recipient->getUser());
        $message->setType('welcome');
        $message->setContext([
            'subject' => $this->getSubject(),
            'content' => $this->getContent(),
        ]);

        return $message;
    }
}
```

$ Price stamps

> “The idea is to have composition by stamping `PriceInterface` with dedicated implementations of `StampInterface` and performing price changes”

https://github.com/ibexa/product-catalog/pull/1212

$ Price stamp (contract)

```php
namespace Ibexa\Contracts\ProductCatalog\Values\Price;

use Ibexa\Contracts\ProductCatalog\Values\StampInterface;
use Money\Money;

interface PriceStampInterface extends StampInterface
{
    public function getNewPrice(): Money;
}
```

$ Price envelope (contract)

```php
interface PriceEnvelopeInterface
{
    /**
     * Adds one or more stamps.
     *
     * @return static
     */
    public function with(PriceStampInterface ...$stamps): self;

    # ...

    public function last(string $stampFqcn): ?PriceStampInterface;

    public function all(?string $stampFqcn = null): array;
}

```

$ Currency exchange stamp

```php
final class CurrencyExchangeStamp implements PriceStampInterface
{
    public function __construct(
        private Money $originalPrice,
        private Money $convertedPrice,
        private float $rate,
        private DateTimeInterface $timestamp = new DateTimeImmutable()
    ) {}

    public function getNewPrice(): Money
    {
        return $this->convertedPrice;
    }

    # ...
}
```

ibexa

$ Currency exchange stamp

```php
final class CurrencyExchangePriceResolver implements PriceResolverInterface
{
  # ...
  private function convertPrice(PriceInterface&PriceEnvelopeInterface $price, CurrencyInterface $currency): PriceInterface
    {
        $sourcePrice = $price->getMoney();
        $targetPrice = $this->converter->convert($sourcePrice, new Currency($currency->getCode()));
        $rate = $this->getExchangeRate($price->getCurrency(), $currency);

        return $price->with(new CurrencyExchangeStamp($sourcePrice, $targetPrice, $rate));
    }
}
```

$ Currency exchange stamp (demo)

```
demo@ibexa-summit-2025 % ddev php bin/console demo:price-stamp:currency-exchange PIM-25396 --currency=EUR
Target price: € 24.00

demo@ibexa-summit-2025 % ddev php bin/console demo:price-stamp:currency-exchange PIM-25396 --currency=PLN
Target price: PLN 101.04
Source price: € 24.00
Exchange rate: 4.210000
Exchange date: 2025-01-28 12:44:00

demo@ibexa-summit-2025 % ddev php bin/console demo:price-stamp:currency-exchange PIM-25396 --currency=USD
Target price: $ 30.00
Source price: € 24.00
Exchange rate: 1.250000
Exchange date: 2025-01-28 12:45:00
```

$ Symbol attribute

IN PROGRESS

- Symbol attribute type allows to efficiently represent string-based values and enforcing its format in product specifications.
- Example use cases: ISBN, EAN13, Manufacturer Part Number
- Ibexa DXP v4.6 LTS update

https://github.com/ibexa/product-catalog-symbol-attribute

ibexa

$ Custom format

IN PROGRESS

You can define custom symbol attribute formats using YAML configuration:

```yaml
ibexa_product_catalog_symbol_attribute:
    formats:
        manufacturer_part_number:
            name: 'Manufacturer Part Number'
            pattern: '/^[A-Z]{3}-\d{5}$/'
            examples:
                - 'RPI-14645'
                - 'MSS-24827'
                - 'SEE-15444
```

$ Custom format

IN PROGRESS

You can define checksum for format by implementing `ChecksumInterface`:

```php
namespace Ibexa\Contracts\ProductCatalogSymbolAttribute\Value;

use Ibexa\Contracts\ProductCatalog\Values\AttributeDefinitionInterface;

interface ChecksumInterface
{
    public function validate(AttributeDefinitionInterface $attributeDefinition, string $value): bool;
}
```

ibexa

$ Custom format (demo)

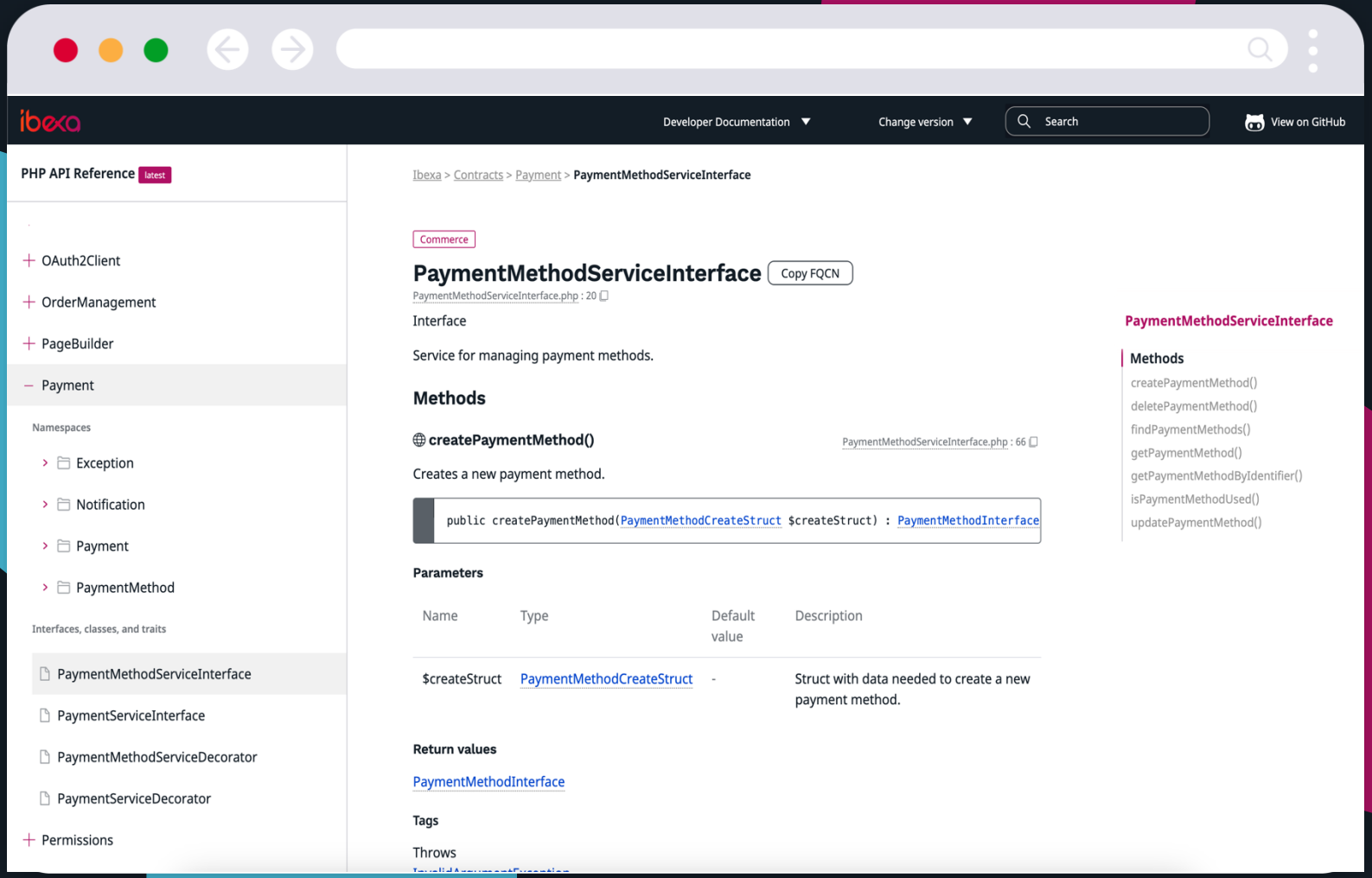
IN PROGRESS

```php
final class ISBN13Checksum implements ChecksumInterface
{
    public function validate(AttributeDefinitionInterface $attributeDefinition, string $value): bool
    {
        $digits = $this->getDigits($value);
        if (count($digits) !== 13) {
            return false;
        }

        $checksum = 0;
        foreach ($digits as $i => $digit) {
            $checksum += $i % 2 === 0 ? $digit : 3 * $digit;
        }

        return $checksum % 10 === 0;
    }

    # ...
}
```

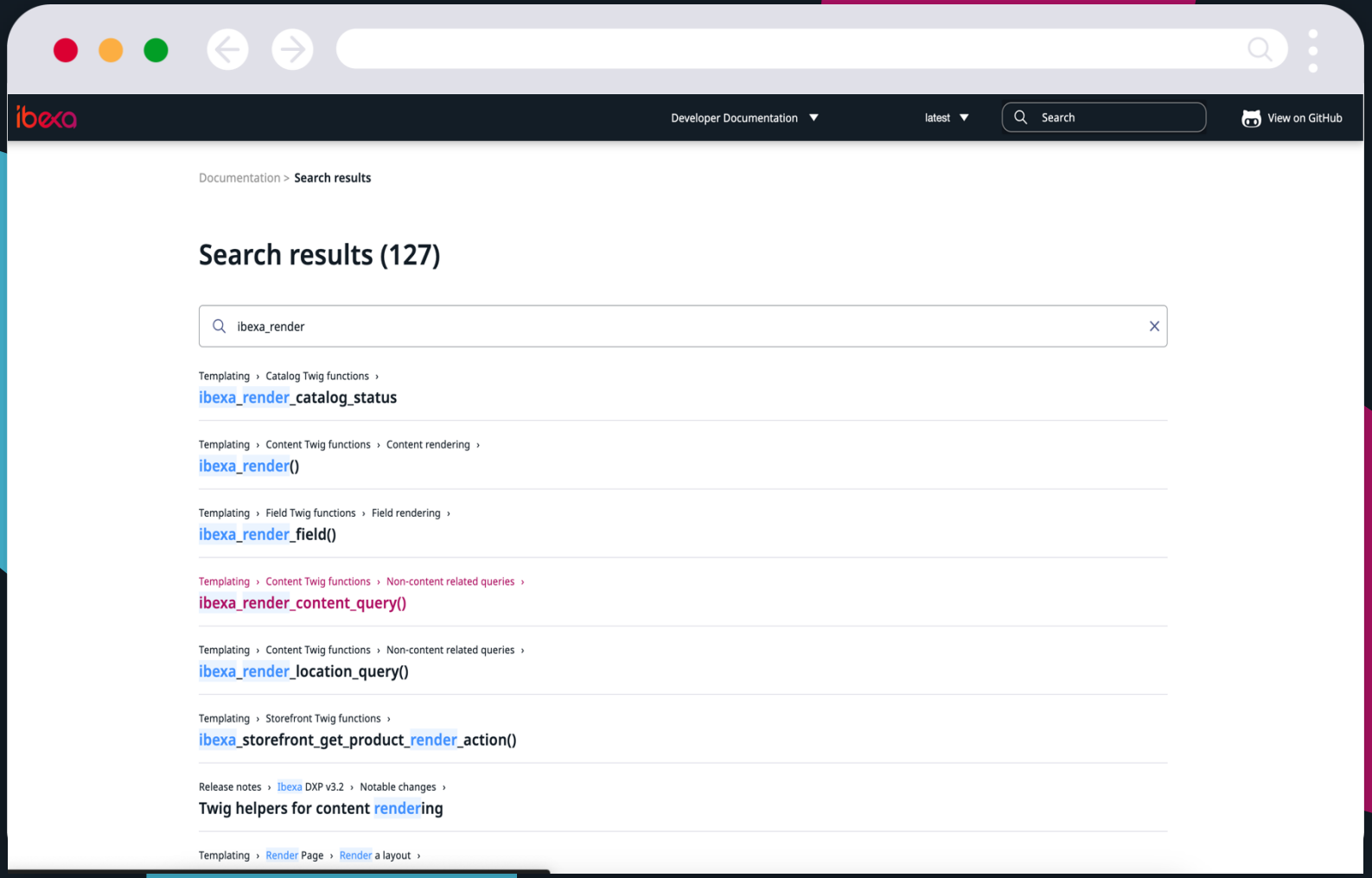
ibexa

Documentation

PHP API Reference



Search results page



Thank you!