



BARCELONA

DRUPALCON 2015





BARCELONA
DRUPALCON 2015

Decoupling Drupal modules into PHP libraries

Bojan Zivanovic

Coding and Development



BARCELONA
DRUPALCON 2015



Bojan Zivanovic (bojanz)

Commerce Development Lead
@ Commerce Guys

Views Bulk Operations, Inline Entity Form



BARCELONA
DRUPALCON 2015

Commerce Guys

Paris | London | Ann Arbor, MI

Creators of Drupal Commerce and Platform.sh





Drupal Commerce

[View](#)

[Edit](#)

[Version control](#)

[Revisions](#)

[Maintainers](#)

[Automated Testing](#)

Posted by [rszrama](#) on *October 15, 2009 at 11:41pm*

Drupal Commerce is used to build eCommerce websites and applications of all sizes. At its core it is lean and mean, enforcing strict development standards and leveraging the greatest features of Drupal 7 and major modules like Views and Rules for maximum flexibility.

Whereas eCommerce solutions are often developed with an *application* mindset, highlighting what you can *do* with it out of the box, Drupal Commerce was developed with a *framework* mindset, focusing on what you can *build* with it. The core Commerce systems make no hard-coded assumptions about your business model, privileging developers and site builders at the core level to build custom eCommerce solutions to suit.





4 YEARS SINCE THE FIRST RELEASE...

Weekly project usage



Week	7.x	8.x	Total
September 6, 2015	60,462	40	60,502
August 30, 2015	60,527	21	60,548
August 23, 2015	60,072	24	60,096
August 16, 2015	60,533	22	60,555
August 9, 2015	58,714	28	58,742



BARCELONA
DRUPALCON 2015

And now we start from scratch

```
1 <?php  
2  
3 namespace Drupal\commerce\Entity;  
4  
5 |
```



BARCELONA
DRUPALCON 2015

Re-evaluating our problem space





Libraries first

tax

PHP ★ 41 ↗ 4

A PHP 5.4+ tax library.

Updated 23 minutes ago

addressing

PHP ★ 241 ↗ 14

A PHP 5.4+ addressing library, powered by Google's dataset.

Updated 2 days ago

intl

PHP ★ 97 ↗ 6

A PHP 5.4+ internationalization library, powered by CLDR data.

Updated 13 days ago

zone

PHP ★ 32 ↗ 2

A PHP 5.4+ zone management library.

Updated 13 days ago



commerceguys / intl

Unwatch 14

A PHP 5.4+ internationalization library, powered by CLDR data. — Edit

51 commits

1 branch

0 releases

3 contributors



branch: master

intl / +



README.md

intl

A PHP 5.4+ internationalization library, powered by CLDR data.

Features:

- NumberFormatter, inspired by intl.
- Currencies
- Countries
- Languages





BARCELONA
DRUPALCON 2015



commerceguys / **addressing**

Unwatch 18



A PHP 5.4+ addressing library, powered by Google's dataset. — Edit

65 commits

1 branch

0 releases

6 contributors



branch: master

addressing / +



README.md

addressing

A PHP 5.4+ addressing library, powered by Google's dataset.

Stores and manipulates postal addresses, meant to identify a precise recipient location for shipping or billing purposes.

Features:

- Address formats for 200 countries
- Subdivisions (administrative areas, localities, dependent localities) for 40 countries
- Subdivision translations for all of the parent country's (i.e Canada, Switzerland) official languages.
- Validation (via Symfony Validator)
- Form generation (via Symfony Form)
- Postal formatting



commerceguys / zone

Unwatch

5



A PHP 5.4+ zone management library. — Edit

23 commits

1 branch

0 releases

2 contributors



branch: master

zone / +



README.md

zone

build passing

A PHP 5.4+ zone management library. Requires [commerceguys/addressing](#).

Zones are territorial groupings mostly used for shipping or tax purposes. For example, a set of shipping rates associated with a zone where the rates become available only if the customer's address matches the zone.



commerceguy / tax Unwatch 6

A PHP 5.4+ tax library. — Edit

45 commits 1 branch 0 releases 2 contributors

branch: master tax / +

README.md

tax

build passing

A PHP 5.4+ tax management library.

Features:

- Smart data model designed for fluctuating tax rate amounts ("19% -> 21% on January 1st")
- Predefined tax rates for EU countries and Switzerland. More to come.
- Tax resolvers with logic for all major use cases.



Trending developers

These are the organizations and developers building the hot tools today.

Repositories

Developers

Trending: **today** ▾

All languages

Unknown languages

CSS

JavaScript

PHP

Python

☰ Other: **Languages** ▾

ProTip! Looking for recently updated PHP repositories? [Try this search](#)

1



commerceguys (Commerce Guys)

📄 addressing A PHP 5.4+ adres...

2



google (Google)

📄 google-api-php-client A PHP client library...

3

facebook (Facebook)

📄 facebook-php-sdk Use the Facebook ...

4



laravel (The Laravel PHP Framework)

📄 laravel A PHP Framework ...



BARCELONA
DRUPALCON 2015

Influenced/Infiltrated





BARCELONA
DRUPALCON 2015

More to come

The screenshot shows the GitHub interface for the repository 'commerceguys/pricing'. At the top, it displays the repository name, a description 'A PHP 5.3+ library for working with prices.', and statistics: 29 commits, 1 branch, 0 releases, and 3 contributors. A commit by user 'bojanz' is highlighted, with the message 'Remove the currency handling, depend on commerceguys/intl.' and the latest commit hash 'bdb9da3b19'. Below the commit list, the 'README.md' file is open, showing the repository name 'pricing' and its description. On the right side, there are navigation links for Code, Issues (0), Pull Requests (2), Wiki, Pulse, Graphs, and Settings. At the bottom right, there are options to clone the repository using HTTPS, SSH, or Subversion, and buttons for 'Clone in Desktop' and 'Download ZIP'.

commerceguys / pricing

Unwatch 14 Unstar 36 Fork 6

A PHP 5.3+ library for working with prices. — Edit

29 commits 1 branch 0 releases 3 contributors

branch: master pricing / +

Remove the currency handling, depend on commerceguys/intl.

bojanz authored on Aug 22 latest commit bdb9da3b19

File	Commit Message	Time
src	Remove the currency handling, depend on commerceguys/intl.	a month ago
.gitignore	Add gitignore.	3 months ago
LICENSE	Initial commit	3 months ago
README.md	Remove the currency handling, depend on commerceguys/intl.	a month ago
composer.json	Remove the currency handling, depend on commerceguys/intl.	a month ago

README.md

pricing

A PHP 5.3+ library for working with prices.

Depends on [commerceguys/intl](#) for currency information and formatting.

Code

- Issues 0
- Pull Requests 2
- Wiki
- Pulse
- Graphs
- Settings

HTTPS clone URL

<https://github.com/>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP



Why?

- **Re-examine your problem space.**
- **Get additional exposure.**
- **Validate your implementation**
- **Get development/maintenance help.**
- **Allow for backports.**



BARCELONA
DRUPALCON 2015

The (not so) hidden cost





BARCELONA
DRUPALCON 2015

The anatomy of a library



Data model - The interfaces

```
1  <?php
2
3  namespace CommerceGuys\Intl\Currency;
4
5  interface CurrencyInterface
6  {
7      public function getCurrencyCode();
8      public function getName();
9      public function getNumericCode();
10     public function getSymbol();
11     public function getFractionDigits();
12 }
13
14 interface CurrencyEntityInterface extends CurrencyInterface
15 {
16     public function setCurrencyCode($currencyCode);
17     public function setName($name);
18     public function setNumericCode($numericCode);
19     public function setSymbol($symbol);
20     public function setFractionDigits($fractionDigits);
21 }
```



Data model - The default class

```
1 <?php
2
3 namespace CommerceGuys\Intl\Currency;
4
5 /**
6  * Default currency implementation.
7  *
8  * Can be used directly by Doctrine, for implementing applications
9  * that want currencies to be user-editable.
10 */
11 class Currency implements CurrencyEntityInterface
12 {
```



Data model - Traits

```
1  <?php
2
3  namespace CommerceGuys\Addressing\Model;
4
5  /**
6   * Provides the format string with getters, setters and introspection methods.
7   *
8   * Allows external systems to reimplement AddressFormat without needing
9   * to duplicate the format introspection logic.
10  *
11  * @see AddressFormatInterface
12  */
13  trait FormatStringTrait
14  {
```



Data model - Collections

```
218 /**
219  * {@inheritdoc}
220  */
221 public function getChildren()
222 {
223     return $this->children;
224 }
225
226 /**
227  * {@inheritdoc}
228  */
229 public function setChildren(Collection $children)
230 {
231     $this->children = $children;
232
233     return $this;
234 }
235
236 /**
237  * {@inheritdoc}
238  */
239 public function hasChildren()
240 {
241     return !$this->children->isEmpty();
242 }
243
```



doctrine / collections



Data model - Repository

```
10 interface AddressFormatRepositoryInterface
11 {
12     /**
13      * Returns an address format instance matching the provided country code.
14      *
15      * If no matching address format was found, the fallback address format (ZZ)
16      * is returned instead.
17      *
18      * @param string $countryCode The country code.
19      * @param string $locale       The locale (e.g. fr-FR).
20      *
21      * @return AddressFormat The address format instance.
22      */
23     public function get($countryCode, $locale = null);
24
25     /**
26      * Returns all address format instances.
27      *
28      * @param string $locale The locale (e.g. fr-FR).
29      *
30      * @return AddressFormat[] An array of address format instances.
31      */
32     public function getAll($locale = null);
33 }
```



Services

```
5 class ZoneMatcher implements ZoneMatcherInterface
6 {
7     protected $repository;
8
9     public function __construct(ZoneRepositoryInterface $repository)
10    {
11        $this->repository = $repository;
12    }
13
14    public function match(AddressInterface $address, $scope = null)
15    {
16        $zones = $this->matchAll($address, $scope);
17
18        return count($zones) ? $zones[0] : null;
19    }
20
21    public function matchAll(AddressInterface $address, $scope = null)
22    {
23        $zones = [];
24        foreach ($this->repository->getAll($scope) as $zone) {
25            if ($zone->match($address)) {
26                $zones[] = $zone;
27            }
28        }
29
30        return $zones;
31    }
32 }
33
```



Tests (PHPUnit)

```
/**
 * @coversDefaultClass \CommerceGuys\Intl\Currency\Currency
 */
class CurrencyTest extends \PHPUnit_Framework_TestCase
{
    /**
     * @var Currency
     */
    protected $currency;

    public function setUp()
    {
        $this->currency = new Currency();
    }

    /**
     * @covers ::getCurrencyCode
     * @covers ::setCurrencyCode
     * @covers ::__toString
     */
    public function testCurrencyCode()
    {
        $this->currency->setCurrencyCode('USD');
        $this->assertEquals('USD', $this->currency->getCurrencyCode());
        $this->assertEquals('USD', (string) $this->currency);
    }
}
```



BARCELONA
DRUPALCON 2015

Tips & Tricks



Translatable strings

```
5 use CommerceGuys\Enum\AbstractEnum;
6
7 /**
8  * Enumerates available locality types.
9  *
10 * @codeCoverageIgnore
11 */
12 final class LocalityType extends AbstractEnum
13 {
14     const CITY = 'city';
15     const DISTRICT = 'district';
16     const POST_TOWN = 'post_town';
17
18     /**
19      * Gets the default value.
20      *
21      * @return string The default value.
22      */
23     public static function getDefault()
24     {
25         return static::CITY;
26     }
27 }
28
```



Pluggable implementations

```
1  <?php
2
3  $taxTypeResolverEngine = new TaxTypeResolverEngine();
4  $taxTypeResolverEngine->add(new EuTaxTypeResolver);
5  $taxTypeResolverEngine->add(new DefaultTaxTypeResolver);
6
7  $taxTypeResolverEngine->resolve($taxable, $context);
8
```

```
12  /**
13   * Validation constraint for the entity changed timestamp.
14   *
15   * @Plugin(
16   *   id = "EntityChanged",
17   *   label = @Translation("Entity changed", context = "Validation"),
18   *   type = {"entity"}
19   * )
20   */
21  class EntityChangedConstraint extends Constraint {
22
23   public $message = 'The content has either been modified by another user.';
24  }
```



BARCELONA
DRUPALCON 2015

Know when to stop



BARCELONA
DRUPALCON 2015

Drupal integration



BARCELONA
DRUPALCON 2015

Composer, part II

<https://bojanz.wordpress.com/2015/09/18/d8-composer-definitive-intro/>



Entity types

```
8 namespace Drupal\commerce_price;
9
10 use CommerceGuys\Intl\Currency\CurrencyEntityInterface as ExternalCurrencyInterface;
11 use Drupal\Core\Config\Entity\ConfigEntityInterface;
12
13 /**
14  * Defines the interface for currencies.
15  */
16 interface CurrencyInterface extends ExternalCurrencyInterface, ConfigEntityInterface {
17 }
18
```



BARCELONA
DRUPALCON 2015

Library ideas

Which problems to tackle



BARCELONA
DRUPALCON 2015

Hard problems





BARCELONA
DRUPALCON 2015

Boring problems





BARCELONA
DRUPALCON 2015

Are you saving me time?

“Nobody cares about your CartInterface”

– Me



BARCELONA
DRUPALCON 2015

Spreading the drop





BARCELONA
DRUPALCON 2015

Questions?



BARCELONA
DRUPALCON 2015

Sprint: Friday

Sprint with the Community on Friday.

We have tasks for every skillset.

Mentors are available for new contributors.

An optional Friday morning workshop for first-time sprinters will help you get set up.

Follow @drupalmentoring.



[https://www.flickr.com/photos/amazeelabs/
9965814443/in/faves-38914559@N03/](https://www.flickr.com/photos/amazeelabs/9965814443/in/faves-38914559@N03/)



BARCELONA
DRUPALCON 2015

What Did You Think?

Evaluate This Session

barcelona2015.drupal.org/schedule

Thank you!