

# DRUPAL COMMERCE

## Doświadczenia

# Nasze doświadczenia z Commerce

- Ogólnie wrażenia
- Przykłady możliwości + bardziej szczegółowe problemy
  - Eksportowalność ustawień
  - Solr
  - Wykorzystanie Rules

## Plusy - naprawdę, naprawdę elastyczny

- Żadnych założeń co do procesu zakupu (bardziej framework niż sklep)
- Wszystko można nadpisać, dopisać, zmienić przez API
  - np. `hook_entity_info_alter()` pozwala zmienić controller i rozciągnąć każdy entity (order itp).
- Fantastyczne użycie rules:
  - Każde zdążenie jest triggerem
  - Ogromna ilość dostępnych reakcji
  - Wszystko oparte o Entity - badanie elementów w Rules proste

## Problemy – bardzo wczesne etapy rozwoju

- Brak UI do bardzo dużej ilości ustawień
- Brak dokumentacji (czytanie kodu uczy najlepiej)
- Brak podstawowych funkcjonalności:
  - Koszt transportu per item
  - Zarządzanie relacjami displayu z produktem
- Bardzo dużo modułów w dev/alpha
- Podstawowe funkcjonalności wymagają używania wersji dev innych modułów

## Spojrzenie w przyszłość

- \$5mln finansowania dla Commerce Guys
- Bardzo duża elastyczność może pozwolić wyjść poza Drupala
- v2 ma być bardziej przyjazna dla użytkownika

# SZCZEGÓŁOWE ELEMENTY

# Exportability (Features + commerce\_features) 1.

## TAK

- Entities: (zamówienia, nody (węzły)) + pola
- Twoje widoki (katalog itp)

## NIE

- Entities: (line items, payment transactions) – tylko fields & fieldgroups exportable
- Rules – w **kodzie** więc się nie eksportują. Można wyeksportować ich klony które przy instalacji wyłączają podstawowy rule
- Views – w **kodzie** - można stosować tą samą technikę

## Exportability (Features + commerce\_features) 2.

- Ustawienia wyświetlania pseudo pól na display node – trzeba użyć Display Suite (ds)
- Exportowalność coraz lepsza.



# Integracja z Solr

# Apache Solr integracja - moduły

- apachesolr
- search\_api\_solr
- search\_api
- search\_api\_ranges (dev)
- facetapi
- facetapi\_select (<http://drupal.org/sandbox/lynn/1311040>)

# Apache Solr integracja - kroki

1. Instalujemy moduły (wszystkie oprócz deprecated faceted search)
2. Integrujemy Apache solr z search\_api\_solr (INSTALL.txt)
3. Konfiguracja > Wyszukiwanie > Search API
  1. Dodajemy serwer - klasa Solr service
  2. Dodajemy indeks do serwera
4. W indeksie:
  1. Wybieramy pola które chcemy indeksować
  2. Możemy ustawić które z nich będą facetami i jak mają się zachowywać

# Wyświetlanie wyników

- Search API views
- search\_api\_page

# Przedziały cenowe

- Indeksowanie pola Price jest w tej chwili nie możliwe/kłopotliwe:
  - Pole jest matrycą wielu elementów (price components)
  - Cena trzymana jest w wartościach minimalnych (centach/groszach) i użytkownikowi przez field formatter
- Rozwiązanie:
  - Dziesiętne pole średzące pole ceny – można to łatwo zrobić przez Rules (przy zapisaniu/updacie produktu, skopiować cenę podzieloną przez 100)
  - **Minus** – wzięta będzie wartość z bazy i nie uwzględni rekalkulacji cen dokonywanej przez rules oraz żadnych price components (np. Vat)

# Wykorzystanie Rules - złożone mechanizmy cenowe

- Pokazywana jest cena hurtowa obliczona jako:
  - cena hurtowa ta sama dla wszystkich (wpisana z palca) lub,
  - Cena wynikająca z rabatu danego odbiorcy
- 3 pola i 2 reguły
  - User - pole int ze zniżką klienta
  - Product - cena hurtowa dla wszystkich
  - Product - checkbox (czy cena ta sama czy po rabacie)
  - Reguła sprawdzająca rodzaj zniżki i naliczająca unikalną zniżkę
  - Reguła sprawdzająca rodzaj zniżki i pokazująca cenę hurtową

# PYTANIA