

# Magento Podstawy

## Spotkanie #1

**Trochę teorii.  
Własna skórka.**

Niniejsze materiały stanowią tajemnicę Polcode Sp. z o.o. Zabronione jest rozpowszechnianie ich w jakikolwiek sposób, a także udzielanie informacji o ich treści osobom nieuprawnionym.

---

---

# Agenda

1. Dlaczego Magento?
2. Dokumentacja i dodatkowe informacje.
3. Struktura katalogów.
4. Wiele sklepów na jednej instalacji Magento.
5. EAV.
6. MVC.
7. Skórki i szablony.

---

# 1. Magento – wady i zalety

## **Wady:**

- Mnóstwo plików, bardzo rozbudowana struktura katalogów.
- Bardzo złożone XML powielające informacje z plików.
- Skomplikowana struktura bazy danych wynikająca m.in. z EAV.
- Niska wydajność wynikająca m.in. z EAV.
- Brak dokumentacji.

## **Zalety**

- EAV, rozbudowane zarządzanie atrybutami.
- Nieograniczone możliwości rozbudowy przez moduły.
- Open source (choć bez dokumentacji).
- Skalowalność.

---

## 2. Dokumentacja i dodatkowe informacje

Dostępne w Polcode:

- Magento – kompendium wiedzy  
<https://docs.google.com/a/polcode.pl/document/d/121CDWJIFWDvzOjfCFzLWJTL2ERn27yqKsK06mSXoACI/>
- „No Frills Magento Layout” Alan Storm – biblioteka Polcode (link w podręczniku programisty)
- „Php Architect's Guide to Programming Magento” (biblioteka Polcode)
- Magento Fundamentals screencast -  
<https://screencast.polcode@screencast.polcode.com/Magento/>

---

## 2. Dokumentacja i dodatkowe informacje

### Dodatkowe materiały

- Oficjalne Magento Reference <http://devdocs.magento.com/guides/m1x/>
- Blog Alana Storma - <http://alanstorm.com/category/magento>
- Blog Inchoo <http://inchoo.net/magento/>
- Stack Overflow <http://stackoverflow.com/tags/magento/>
- Magento Stack Exchange <http://magento.stackexchange.com/>

---

## 2. Dokumentacja i dodatkowe informacje

### Moduły

- Moduł Magento Debug - <http://www.magentocommerce.com/magento-connect/magento-debug.html>
- Direct Download Magento Extension <http://freegento.com/ddl-magento-extension.php>

---

## 3. Struktura katalogów

Najważniejsze katalogi w Magento to:

- /app/code - tutaj znajduje się kod źródłowy wszystkich modułów
- /app/design - tutaj znajdziemy szablony i ich pliki konfiguracyjne
- /app/etc - tutaj znajdują się najważniejsze pliki konfiguracyjne
- /app/etc/modules - katalog z plikami inicjalizującymi moduły w Magento
- /app/locale - pliki CSV z tłumaczeniami na inne języki
- /downloader - katalog zawierający Magento Connect Manager
- /lib - katalog z zewnętrznymi bibliotekami PHP
- /js - katalog z zewnętrznymi bibliotekami JavaScript
- /media - obrazki, pliki PDF, SWF itp stanowiące zawartość sklepu
- /skin - pliki CSS, JS a także obrazki itp wykorzystywane przez szablon
- /var - katalog z plikami tymczasowymi (np. cache)

---

## 4. Wiele sklepów na jednej instancji

Magento wprowadza trzy poziomy według których można modyfikować dane.

- **Website** – każda instalacja magento musi posiadać przynajmniej jedną website. Pozwalają na uruchomienie dwóch kompletnie różnych sklepów na jednej instancji
- **Store** – każda strona (website) może posiadać jeden lub więcej sklepów. Główną cechą sklepów jest to, że mogą posiadać całkowicie różne drzewo produktów (oddzielna kategoria główna). Przydatne np. przy tworzeniu różnych katalogów dla różnych grup docelowych, bądź dodatkowy podział produktów
- **Store View** – każdy sklep może posiadać jeden lub więcej sklepów. W 99% przypadków Store View służy do zmiany języka lub innych kluczowych ustawień.

Większość ustawień można zmieniać na jednym z trzech poziomów: *website*, *store view* lub *global/default*.



---

## 5. EAV

Oprócz MVC Magento korzysta z innego, bardzo ważnego wzorca projektowego: **Entity-Attribute-Value**. To właśnie EAV sprawia, że Magento ma tyle problemów z wydajnością a jednocześnie gwarantuje tak dużą elastyczność. W modelu tym wyróżniamy trzy struktury:

- Entity - czyli opisywany rekord (np. produkt)
- Attribute - atrybuty opisujące dany rekord (np. opis, cena)
- Value - wartości tych atrybutów

We wczesnych wersjach Magento każdy model (powoli od tego odchodzą) był opisywany przez zestaw atrybutów wraz z ich wartościami. Listę atrybutów można znaleźć w tabeli *eav\_attribute*:

- entity\_type\_id - wartość typu *integer* unikalna dla każdego modelu
- attribute\_code - unikalna nazwa atrybutu
- backend\_type - typ atrybutu (*text*, *varchar*, *integer*, *date*, *price*)

---

## 5. EAV

Nie jest wskazane przetrzymywanie różnych typów danych w jednym, uniwersalnym polu bazy danych (np. *integer* w *text*). Powodowałoby to problemy z sortowaniem oraz wydajnością samej bazy. Również tworzenie wielu kolumn o różnych typach danych nie jest optymalne: nie da się takich danych efektywnie przeszukiwać ani robić JOIN'ów.

W związku z tym dla każdego typu danych tworzona jest osobna tabela. I tak, dla typu *integer* jest tabela z przyrostkiem *\_int* (np. *catalog\_product\_entity\_int*), dla typu *text* z przyrostkiem *\_text* itd. Każda z tych tabel posiada kolumny *entity\_id*, *attribute\_id* oraz *value*.

Łatwo sobie wyobrazić jak skomplikowane są zapytania pobierające informacje o tylko jednym rekordzie: trzeba pobrać listę atrybutów dla danego modelu, następnie sprawdzić typ każdego z atrybutów i zrobić JOIN z odpowiednią tabelą.

---

## 5. EAV – Attribute Set

Istnieje dodatkowy poziom logiki w przypadku produktów. Oprócz samych atrybutów przypisanych do modelu *Product* jest również dodatkowe pojęcie *Attribute Set*. To tak naprawdę nic innego jak podgrupy atrybutów, a każdy produkt jest opisywany przez dokładnie jedną z nich.

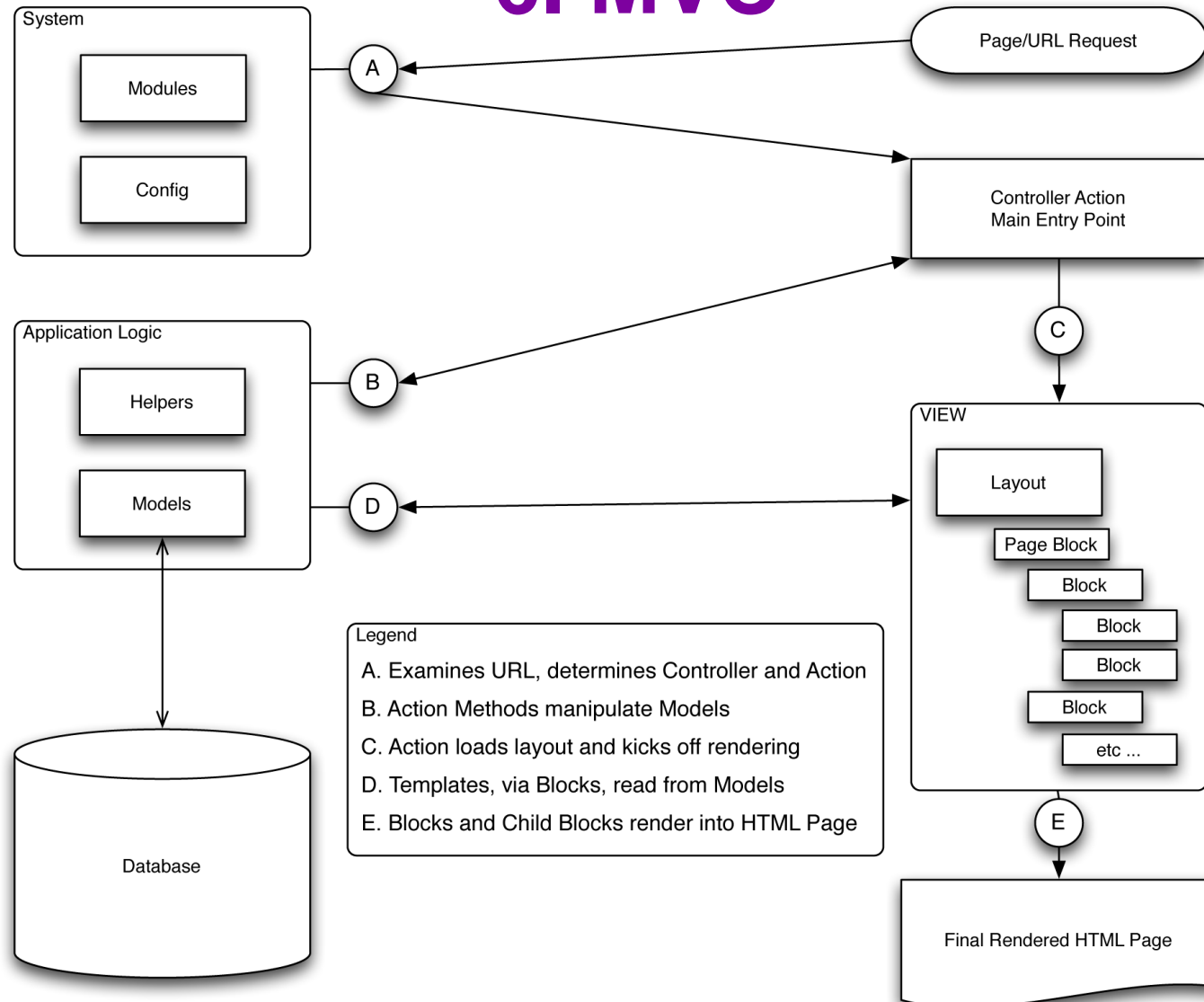
W ten sposób możemy ograniczyć liczbę atrybutów przypisanych do konkretnych produktów. I tak w sklepie komputerowym monitory mogą mieć np. rozmiar a dyski twarde pojemność.

---

## 5. EAV – flat table

Aby rozwiązać problemy wydajnościowe związane z liczbą zapytań wykonywanych przez EAV Magento wprowadziło funkcjonalność *flat catalog*. Jest to typ cache polegający na tworzeniu tabel w bazie danych zawierających osobną kolumnę dla każdego z atrybutów. Pozwala to ograniczyć liczbę zapytań do normalnego poziomu, lecz jest bardzo czasochłonne przy tworzeniu.

# 6. MVC



---

## 6. MVC + Block

- Magento oprócz standardowego wzorca Model-View-Controller wprowadza dodatkowo pojęcie *bloków* (zwanym czasem komponentami).
- Block - klasa która udostępnia zbiór metod dostępnych w samym pliku widoku.
- Pozwala uniknąć bindowania i ustawiania zmiennych na poziomie kontrolera
- Ogranicza powtarzalność kodu - ten sam blok może być wykorzystywany wielokrotnie w różnych częściach strony.
- Przykład: fragment strony pokazujący podsumowanie zawartości koszyka - zamiast dodawać kod w każdym kontrolerze z osobna (bądź wspólnym) całą logikę zawieramy w klasie bloku.
- W Magento układ bloków ma strukturę drzewa. Renderowanie strony zaczyna się zawsze od bloku głównego (*root*), który posiada podbloki (np. *header*, *content*, *footer*), te z kolei kolejne podbloki itd. Nie ma limitu co do ilości zagnieżdżeń.

---

## 7. Szablony Magento

Szablony w magento składają się z dwóch części:

`nazwa_paczki/nazwa_szablону`, na przykład  
`polcode/example`.

Pliki z widokami są wyszukiwane w ustalonej kolejności:

1. Paczka i szablon z ustawień (np. `polcode/example`).
2. Domyślny szablon z paczki (np. `polcode/default`).
3. Domyślny szablon Magento (`base/default`).

---

## 7. Szablony Magento

Tworząc nowy szablon:

- Główną część umieszczamy w szablonie `default`.
- Częściowe zmiany dodajemy w odpowiednich szablonach, np. `xmas`.
- W razie brakującego pliku Magento automatycznie skorzysta z `base/default`.



---

## 7. Szablony Magento

Od wersji **1.9** możemy używać plik w którym podajemy którą skórke nasza ma rozszerzać.

```
<!-- File:
    app/design/frontend/polcode/default/etc/theme.xml
-->

<?xml version="1.0"?>

<theme>

    <parent>rwd/default</parent>

</theme>
```

---

## 7. Szablony Magento

- Własny szablon ustawiamy w *System* → *Configuration* → *General* → *Design* (domyślny szablon dla witryny).
- W *Package* podajemy nazwę paczki (np. *polcode*), a w *Themes* można wpisać nazwę szablonu (puste pola oznaczają szablon *default*).
- Można automatycznie ustawić inny szablon dla określonej daty w *System* → *Design*
- Można również ustawić szablon całoroczny w *System* → *Design*, lecz uniemożliwia to późniejsze zmiany szablonu dla określonych dat (np. święta wielkanocne, bożego narodzenia itd).
- Zmienione pliki XML/PHTML dodajemy do katalogu *app/design/frontend/polcode/default*.

---

## 7. Szablony Magento

- struktura drzewa, oparta na plikach XML
- główny element: *root*
- pliki konfiguracyjne w  
*app/design/frontend/base/default/layout/\*.xml*
- wszystkie pliki są łączone, możliwość wpływania na wygląd każdej strony na poziomie każdego pliku

---

## 7. Szablony Magento

Istnieją dwie główne metody nadpisywania plików XML layoutu:

- przekopiowanie zmienianych plików do katalogu ***layout*** nowej skórki,
- użycie pliku ***local.xml***.

---

## 7. Szablony Magento

- Dla każdego zapytania HTTP brane pod uwagę są następujące ścieżki XPath w plikach layout:
  - `layout/default`
  - `layout/module_default_default`
  - `layout/module_controller_default`
  - `layout/module_controller_action`

---

# 7. Szablony Magento

Przykładowa definicja bloku:

```
<block name="some_block" as="block" type="core/template"
    template="some/block.phtml">
    <block name="some_child_block" as="child_block" type="core/template"
        template="some/child_block.phtml" />
</block>

<reference name="some_block">
    <block name="some_child_block2" as="child_block2" type="core/template"
        template="some/child_block2.phtml" />
</reference>
```

Najważniejsze atrybuty:

- *name* - nazwa bloku wykorzystywana w szablonach XML
- *as* - nazwa bloku wykorzystywana w widokach
- *type* - określa plik z klasą opisującą blok
- *template* - ścieżka do pliku z widokiem

---

## 8. Praca domowa

1. Pobranie i instalacja Magento.
2. Uruchomienie katalogu z przykładowymi produktami.
3. Uruchomienie dwóch sklepów w dwóch wersjach językowych.
4. Stworzenie własnej skórki na podstawie istniejącej  
– zmiana elementów w min. 1 pliku phtml oraz min. 1 css, dodatkowym plusem będzie użycie własnego pliku xml zmieniającego layout.

*Mile widziana własna inwencja, w wypadku braku własnych pomysłów można zmienić kolorystykę domyślnej skórki RWD oraz przenieść okienko wyszukiwania z header do footer.*

*Im więcej wprowadzonych zmian różnego typu, tym wyższa ocena.*