

# Narzędzia programisty Drupala

Grzegorz Dudek  
(grzdud@yahoo.com)

media  
regionalne



# asql

<http://www.steve.org.uk/Software/asql/>

Pozwala analizować logi serwera Apache za pomocą interfejsu SQL.

- asql importuje logi serwera Apache do tymczasowej tabeli SQLite.
- Wykorzystanie SQL pozwala tworzyć skomplikowane zestawienia.
- Szybkość działania (save, restore).
- Możliwość tworzenia aliasów dla często wykonywanych zapytań.

# Drush Make

[http://drupal.org/project/drush\\_make](http://drupal.org/project/drush_make)

Drush make pozwala tworzyć gotowe do użycia strony i rozpowszechniać je pod postacią jednego pliku tekstowego.

- Pobiera core Drupal, moduły, skórki, profile instalacyjne oraz zewnętrzne biblioteki.
- Pobiera i aplikuje patche.
- Daje większą kontrolę nad wersjami modułów i patchami.
- Zmniejsza rozmiar repozytorium aplikacji.

# Selenium + Apache Ant

Selenium <http://seleniumhq.org/>

- Nagrywa i odtwarza aktywność użytkownika na stronie www.
- Główne zastosowanie: testy funkcjonalne.

Apache Ant <http://ant.apache.org/>

- Powstał jako narzędzie do wspomaganie budowania aplikacji napisanych w języku Java.
- Automatyzacja powtarzalnych czynności np. kompilacja, uruchomienie testów, generowanie dokumentacji itp.

# Hacked

<http://drupal.org/project/hacked>

Skanuje aplikację, pobiera raz jeszcze kody wszystkich modułów i skórek a następnie sprawdza, czy zostały zmodyfikowane.

- Potrafi wskazać dokładną linię kodu, w którym nastąpiły zmiany (wymagany moduł Diff).

# Devel

<http://drupal.org/project/devel>

Zestaw pomocnych funkcji ułatwiających pracę programistom.

- Funkcje `dvm()`, `dpm()`, `drupal_debug()`, `ddebug_backtrace()`
- Logowanie i wyświetlanie zapytań do bazy danych.
- Wyświetlanie zużycia pamięci i czasu generowania strony.
- Devel Generate, Devel Node Access.
- Performance: <http://drupal.org/project/performance>
- Theme developer: [http://drupal.org/project/devel\\_themer](http://drupal.org/project/devel_themer)

# Śledzenie zużycia pamięci

<http://drupal.org/node/659980>

## Xdebug Function Traces

- Xdebug pozwala na logowanie wszystkich wywołań funkcji wraz z parametrami i zwracanymi wartościami.
- Logi zawierają czas wykonania funkcji oraz zużycie pamięci.

### **php.ini**

```
xdebug.auto_trace=1
```

```
xdebug.trace_format=1
```

# Śledzenie zużycia pamięci

<http://derickrethans.nl/xdebug-and-tracing-memory-usage.html>

tracefile-analyzer.php

- Skrypt który odczytuje dane zawarte w pliku trace i grupuje je po nazwie funkcji.
- Funkcje mogą być posortowane według Następnie możemy wyświetlać te dane w następujących trybach:
  - liczby wywołań: `calls`
  - zużycia pamięci: `memory-own`
  - czasu wykonywania: `time-own`



# Coder

<http://drupal.org/project/coder>

Moduł Coder wykorzystuje wyrażenia regularne w poszukiwaniu kodu, który:

- Powinien zostać zmieniony z powodu zmian w API Drupala.
- Nie spełnia w sposób satysfakcjonujący standardów kodowania Drupala.

# PHP Code Sniffer

[http://pear.php.net/package/PHP\\_CodeSniffer](http://pear.php.net/package/PHP_CodeSniffer)

Narzędzie do znajdowania kodu, który "brzydko pachnie".

- Narzędzie dla wszystkich aplikacji PHP, nie tylko Drupala.
- Domyślnie dostarcza kilka standardów kodowania, min. PEAR i Zend
- Możliwość tworzenia własnych standardów.

# Drupal Code Sniffer

<http://drupal.org/project/drupalcs>

## Standard kodowania dla Drupala.

- PHP Code Sniffer dostarcza zestaw standardów będących częścią standardów kodowania w Drupalu (PEAR).
- Nie wymaga zainstalowanego Drupala do przeprowadzenia testów.
- Możliwość tworzenia własnych klas testujących (wykorzystanie tokenów zamiast wyrażeń regularnych).
- Łatwiejsza integracja z zewnętrznymi narzędziami typu Eclipse, Vim, Git.

# Vim

<http://www.vim.org/>

Podstawowa konfiguracja:

<http://drupal.org/node/29325>

Rekomendowane pluginy:

<http://drupal.org/node/1389472>

<http://drupal.org/node/1389006>

<https://wincent.com/products/command-t>

# Pytania

media  
regionalne



# Koniec

Dziękuję za uwagę :-)

media  
regionalne

