

Narzędzia deweloperskie

Projektowanie i programowanie systemów internetowych I

mgr inż. Krzysztof Rewak

4 marca 2018

Wydział Nauk Technicznych i Ekonomicznych

Państwowa Wyższa Szkoła Zawodowa im. Witelona w Legnicy

Plan prezentacji

1. Edytor czy IDE?
2. Przeglądarki internetowe
3. Oprogramowanie dla programisty
4. Podsumowanie

Edytor czy IDE?

Kod musi zostać jakoś zapisany.

Z grubsza można wyodrębnić trzy sposoby na zapisywanie kodu:

- CLI
- edytory tekstu
- IDE

CLI (ang. *Command Line Interface*), czyli interfejs wiersza poleceń, to najbardziej podstawowy sposób interakcji człowieka z maszyną.

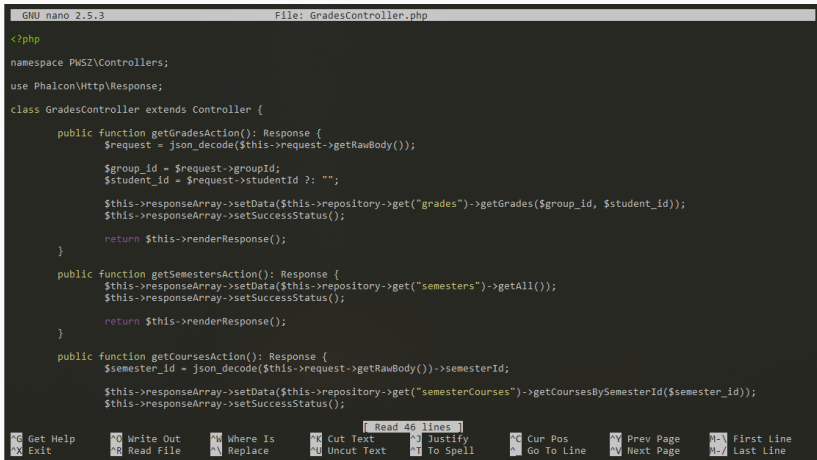
Jest dostępny na systemach UNIX-owych (kolejne dystrybucje linuksa, macOS) oraz Windowsie:

- sh, csh, **zsh**
- Terminal
- cmd.exe, Windows PowerShell, **Cmder**

Z poziomu CLI można nie tylko wydawać polecenia, ale również edytować pliki z kodem. Najpopularniejsze (nie należy mylić z najlepszymi) rozwiązania to:

- vi/vim
- nano
- Emacs
- gedit

Istnieją miłośnicy takiego sposobu programowania, jednakże należy pamiętać, że są to rozwiązania w dużej mierze prymitywne w przypadku pracy nad poważniejszymi projektami.



```
GNU nano 2.5.3 File: GradesController.php

<?php
namespace PWSZ\Controllers;
use Phalcon\Http\Response;

class GradesController extends Controller {

    public function getGradesAction(): Response {
        $request = json_decode($this->request->getRawBody());

        $group_id = $request->groupId;
        $student_id = $request->studentId ?: "";

        $this->responseArray->setData($this->repository->get("grades")->getGrades($group_id, $student_id));
        $this->responseArray->setSuccessStatus();

        return $this->renderResponse();
    }

    public function getSemestersAction(): Response {
        $this->responseArray->setData($this->repository->get("semesters")->getAll());
        $this->responseArray->setSuccessStatus();

        return $this->renderResponse();
    }

    public function getCoursesAction(): Response {
        $semester_id = json_decode($this->request->getRawBody())->semesterId;

        $this->responseArray->setData($this->repository->get("semesterCourses")->getCoursesBySemesterId($semester_id));
        $this->responseArray->setSuccessStatus();
    }
}

Read 46 lines
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      ^V Prev Page    ^_ First Line
^X Exit          ^R Read File    ^A Replace      ^U Uncut Text   ^T To Spell     ^G Go To Line   ^N Next Page    ^-/ Last Line
```

Rysunek 1: Windowsowy shell Cmder z uruchomionym nano na linuxsie

Programiści bardzo często używają edytorów tekstu - często równocześnie z pracą z IDE. Cechuje jest szybkość działania, możliwość pracy na dużych plikach, wiele kombinacji klawiszowych ułatwiających pracę oraz możliwość rozszerzania przez dodatki lub wtyczki.

Kiedy edytor tekstu wystarczy?

- przy projektach w językach ze słabym lub dynamicznym typowaniem, gdzie IDE często nie spełnia stuprocentowo swojej roli
- przy mniejszych projektach
- jeżeli programista nie potrzebuje „magii” IDE

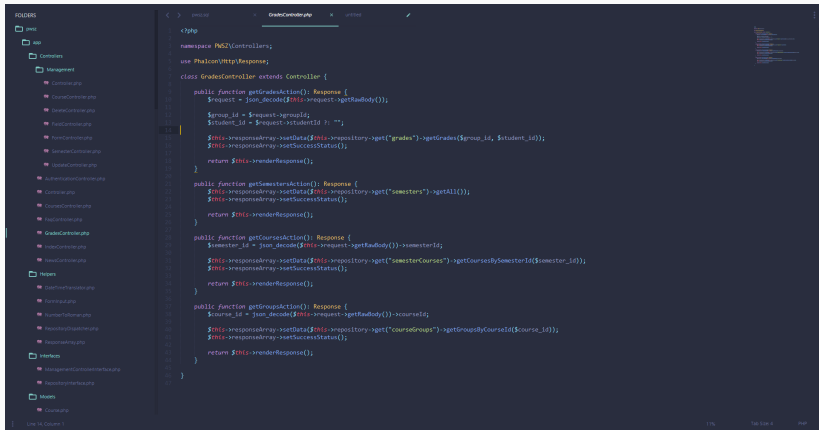
Edytory tekstu

Popularne edytory tekstu:

- Notepad++
- **Sublime Text 3**
- Atom
- Visual Studio Code

Warto zapoznać się z listami najpopularniejszych rozszerzeń do wybranego edytora, a także nauczyć się najpotrzebniejszych skrótów klawiszowych.

Edytory tekstu



Rysunek 2: Sublime Text 3

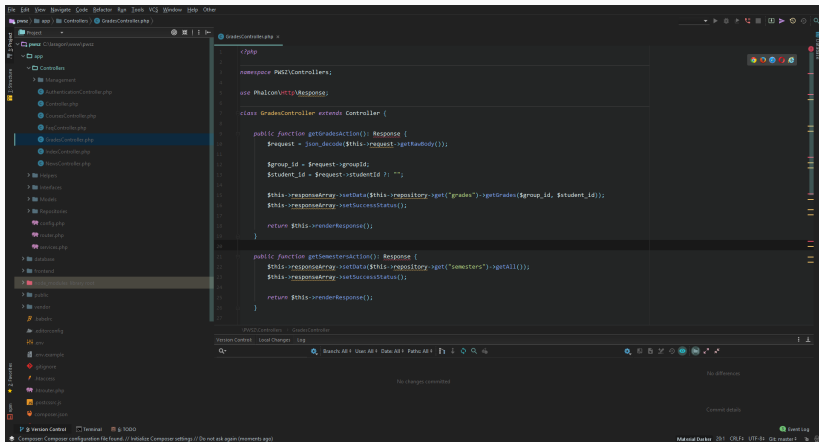
Integrated development environment, czyli **zintegrowane środowisko programistyczne**, to specjalistyczne oprogramowanie służące do tworzenia oprogramowania.

Niektóre zalety i wady korzystania z IDE:

- + wszystkie narzędzia w jednym miejscu
- + interpretacja kodu
- + pełna możliwość konfiguracji
- - zazwyczaj kosztowne
- - obciążające komputer
- - niektórzy powiedzą, że rozleniwiające

Przykładowe i popularne IDE:

- Microsoft Visual Studio (C#, C++, VB i inne)
- Eclipse (przede wszystkim Java)
- NetBeans (Java)
- Code::Blocks (C, C++)
- rozwiązania JetBrains:
 - PhpStorm (PHP)
 - PyCharm (Python)
 - IntelliJ IDEA (Java)
 - WebStorm (JavaScript)
 - CLion (C, C++)
 - RubyMine (Ruby)



Rysunek 3: PhpStorm

Przeglądarki internetowe

Przeglądarki internetowe

Przeglądarka internetowa to program pozwalający przede wszystkim na wyświetlanie stron internetowych. Współczesne przeglądarki oczywiście oferują wiele innych funkcji, jednak nas będą interesowały tylko te, które mogą być przydatne dla programisty.

Przeglądarki internetowe

Najpopularniejsze obecnie przeglądarki internetowe:

- Google Chrome
- Mozilla Firefox
- Internet Explorer
- Edge
- Safari
- Opera

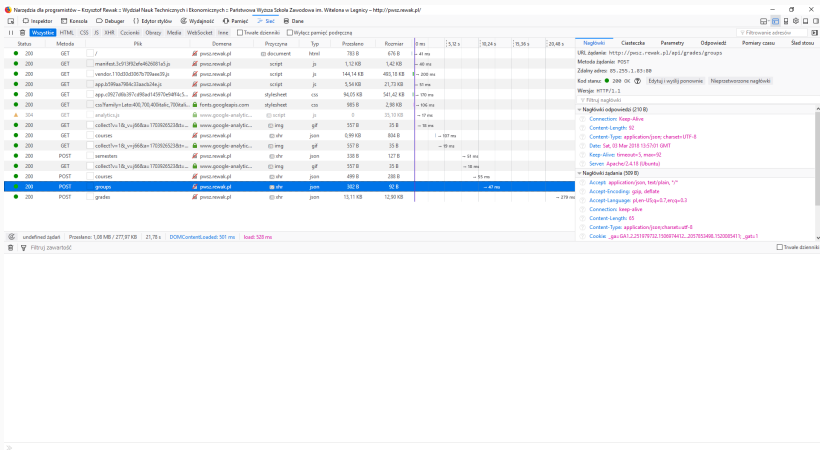
Przeglądarki internetowe

Przeglądarki internetowe oferują najczęściej dwa podstawowe narzędzia dla programistów.

Pierwszym jest prymitywne wyświetlenie źródła strony i zazwyczaj można je wywołać kombinacją klawiszy Ctrl+U. Źródło strony to nic innego jak przedstawienie w formie kodu HTML-a, który buduje daną stronę.

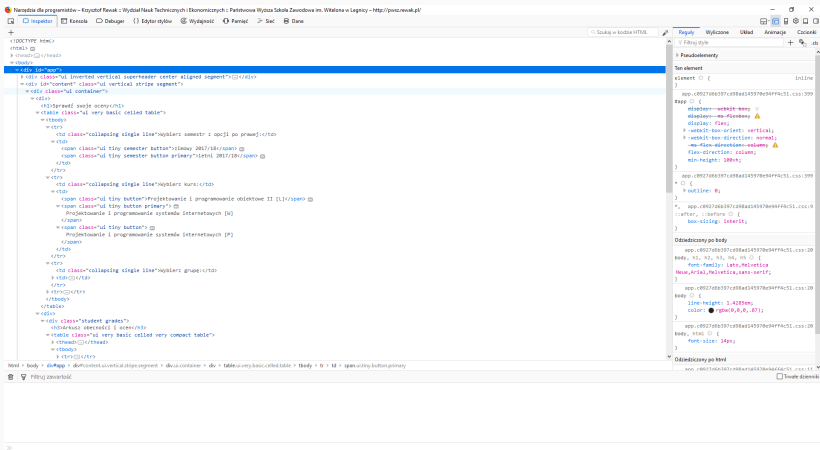
Drugi to narzędzia deweloperskie (najczęściej pod F12) i do nich możemy zaliczyć m. in. inspektora drzewa DOM, konsolę JavaScript, debugger, edytor stylów, podgląd połączeń w sieci czy narzędzia statystyczne.

Przeglądarki internetowe



Rysunek 4: Narzędzia deweloperskie w Firefoksie

Przeglądarki internetowe



Rysunek 5: Inspektor w Firefoksie

Oprogramowanie dla programisty

Środowiska programistyczne

Wygodną sprawą są predefiniowane środowiska programistyczne, które są zebranymi najpotrzebniejszymi serwisami, które są wymagane do tworzenia aplikacji. Wartym spróbowania jest Laragon, który oferuje następujące zestawy:

- WAMP, czyli Apache lub Nginx, PHP, MySQL, npm, git i inne
- ROR, czyli Ruby, Postres, npm, Redis i git
- MEAN, czyli Node, Mongo i git
- Python, Postres, Redis i git
- Spring, czyli Java, Maven, git
- Golang, czyli Go, Postgres, Redis i git

Powinno to zaspokoić potrzeby każdego początkującego programisty, a jednocześnie nie zmusi go do wielogodzinnej konfiguracji środowiska.

Łączenie się z serwerem

Po wdrożeniu systemu internetowego na zewnętrzny serwer z pewnością znajdzie potrzeba ponownego połączenia się z aplikacją. W środowisku Windows pomogą następujące programy:

- ssh: Cmder lub PuTTY
- scp: WinSCP
- ftp: Total Commander

W przypadku pracy z systemem internetowym wykorzystującym bazy danych warto zainteresować się narzędziami przeznaczonymi do zarządzania bazami:

- HeidiSQL
- MySQL Workbench
- phpMyAdmin

Podgląd baz danych

The screenshot shows the HeidiSQL interface connected to a MySQL database. The left sidebar displays the database structure for 'Laragon\pwrz\course_group'. The main window shows the 'course_groups' table with columns: name, semester_course_id, created_at, and updated_at. The table contains 11 rows of data, including course names like 'KOMP102 - p 15:15' and 'KOMP102 - pm 17:00' across different semesters.

#	name	semester_course_id	created_at	updated_at
1	KOMP102 - p 15:15	1	2017-10-01 15:40:15	2017-10-01 15:40:15
2	KOMP102 - p 15:15	1	2017-10-01 15:40:15	2017-10-01 15:40:15
3	KOMP102 - pm 17:00	1	2017-10-01 15:41:09	2017-10-01 15:41:09
4	KOMP102 - w 15:15	1	2017-10-01 15:41:36	2017-10-01 15:41:36
5	KOMP102 - w 15:15	2	2017-10-01 15:42:05	2017-10-01 15:42:05
6	KOMP102 - w 15:15	3	2018-02-12 15:57:57	2018-02-12 15:57:57
7	KOMP102 - w 15:15	3	2018-02-12 15:57:57	2018-02-12 15:57:57
8	KOMP102 - w 15:15	3	2018-02-12 15:57:57	2018-02-12 15:57:57
9	KOMP102 - pm 17:00	3	2018-02-12 15:57:57	2018-02-12 15:57:57
10	KOMP102 - pm 17:00	4	2018-02-12 15:57:57	2018-02-12 15:57:57
11	KOMP102 - pm 17:00	5	2018-02-12 15:57:57	2018-02-12 15:57:57

At the bottom, the SQL console shows the following commands:

```
24 SHOW CREATE TABLE `pwrz`.`course_groups` ;
25 SHOW COLLATION;
26 SHOW ENGINES;
27 SELECT * FROM `pwrz`.`course_groups` LIMIT 1000;
28 SHOW CREATE TABLE `pwrz`.`course_groups` ;
```

Rysunek 6: HeidiSQL podłączona do bazy danych

Systemy zarządzania zależnościami

Każdy język ma własny system zarządzania zależnościami. Pomagają one w procesie deweloperskim przy dodawaniu i utrzymywaniu wersji zewnętrznych modułów, paczek, pakietów lub innych zależności.

Warto wiedzieć jaki język ma jaki system:

- PHP: Composer
- .NET: NuGet
- Java: Maven
- JavaScript: npm
- Ruby: RubyGems
- Python: Pipenv

Jednym z podstawowych narzędzi pracy w każdym sensownym projekcie powinien być system kontroli wersji. Obecnie najpopularniejszym jest Git, ale korzysta się również z Subversion (SVN) lub Mercuriala. Z Gita można korzystać na kilka sposobów:

- z poziomu CLI
- z poziomu IDE
- wykorzystując dedykowane oprogramowanie:
 - Sourcetree
 - Github Desktop
 - GitKraken

Wielu programistów korzysta z tzw. maszyn wirtualnych. Są to odizolowane uruchomienia innych programów, a w szczególności systemów operacyjnych. Popularnym narzędziem tego typu jest **VirtualBox**.

Innym ciekawym rozwiązaniem jest **Vagrant**. Pozwala on na zautomatyzowanie procesu konfiguracji i ustawiania środowiska deweloperskiego. Korzysta pod spodem z rozwiązań VirtualBoksa i z poziomu skryptów konfiguracyjnych jest w stanie stworzyć dowolny zestaw narzędzi programistycznych.

Jeszcze inną rzeczą jest **Docker** i konteneryzacja. Zamiast wirtualizować cały system operacyjny można tworzyć wydzielone kontenery z konkretnymi serwisami takimi jak bazy danych, interpretery języków lub inne serwisy. Takie zestaw pasujących do siebie klocków jest łatwy do przenoszenia i konfiguracji.

Podsumowanie

Bibliografia i ciekawe źródła



<https://www.sublimetext.com/>



<https://laragon.org/download/>



<https://laragon.org/download/>

Pytania?

Kod prezentacji dostępny jest w repozytorium git pod adresem
<https://bitbucket.org/krewak/pwsz-ppsi>



Wszystkie informacje dot. kursu dostępne są pod adresem
<http://pwsz.rewak.pl/kursy/4>

