

PHPStorm

PHPStorm je pokročilý editor pro vývoj aplikací v jazyce PHP od společnosti [JetBrains](#). V současné době se oprávněně řadí mezi celosvětově nejpoužívanější vývojová prostředí. A není se čemu divit. Vývojářům nabízí opravdu obrovské množství funkcí a nástrojů, které usnadňují a zefektivňují jejich práci. Pojďme se tedy podívat na to, čím si PHPStorm zasloužil tak velkou popularitu.

Vzhled editoru

Témata, barevná schémata

I když se to může zdát jako nepodstatné pro samotný vývoj aplikací, je vzhled editoru a barevné schéma zdrojového kódu často první věc, co si chce programátor přizpůsobit svým představám. Pro programátora je totiž velice důležité, aby byl zdrojový kód co nejčitelnější a k tomu výrazně napomáhá jeho obarvování. V neposlední řadě je důležité i to, aby celkové barevné téma editoru lahodilo jeho oku a nedráždilo.

Témata vývojového prostředí je možné nastavit v sekci *Appearance & Behaviour* > *Appearance*. Obarvování zdrojového kódu, velikost písma a font se pak nastavuje v sekci *Editor* > *Colors & Fonts*. Pokud vám nevyhovují výchozí možnosti, můžete si barvičky nastavit sami nebo se podívat na následující adresy s tématy a schémata, které připravila komunita:

- <https://github.com/daylerees/colour-schemes>
- <http://phpstorm-themes.com/>
- <http://phpstorm-themes.com/theme/friends-symfony>
- <https://github.com/cordova/Symfony2Colors>

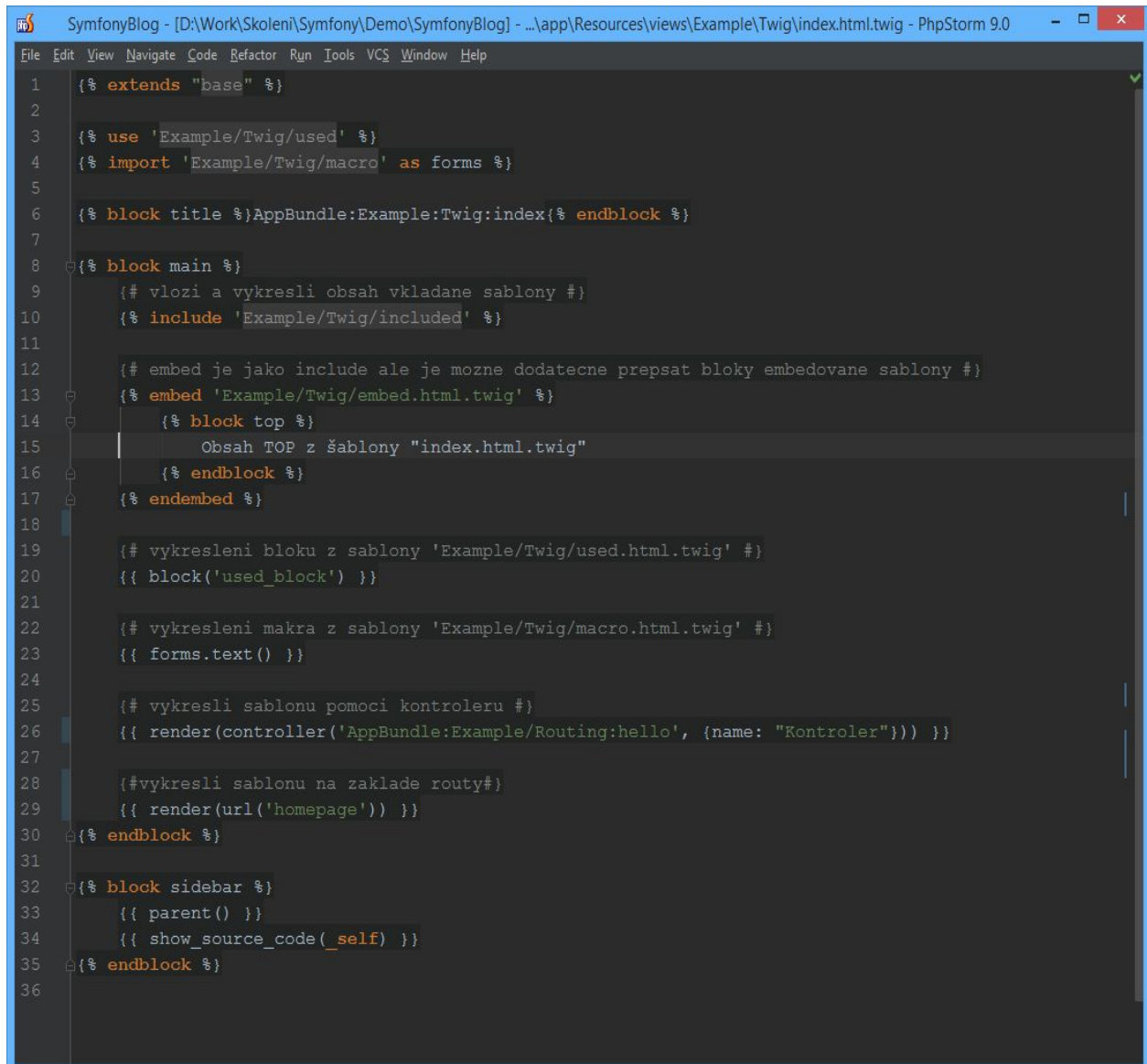
Pokud si vyberete nějaké téma z výše uvedených odkazů, musíte si stažený soubor uložit do konfiguračního adresáře PHPStormu.

- **Windows:** `c:\Users\\.WebIde80\config\colors\`
- **Linux:** `/home/<username>/.WebIde80/config/colors/`

Témata vývojového prostředí ve výchozím chování bohužel nezajistí kompletní přebarvení všech stromových výpisů jako je například výpis adresářové struktury projektu v levém postranním panelu. Pro dokonalý výsledek je tedy nutné ještě doinstalovat plugin s názvem *Color IDE*.

Minimalizmus

Neméně důležitý aspekt efektivního vývoje je koncentrace. Programátor by se během vývoje měl soustředit ideálně pouze na zdrojový kód, který píše a nemělo by ho rozptylovat nic kolem. Proto PHPStorm umožňuje skrýt prvky jako např. toolbar, statusbar, panel s adresářovou strukturou, které přímo nesouvisí se zdrojovým kódem a algoritmem, který vývojář vymýšlí. Okno PHPStormu pak může vypadat například jako na obrázku č. 1.



```
1  {% extends "base" %}
2
3  {% use 'Example/Twig/used' %}
4  {% import 'Example/Twig/macro' as forms %}
5
6  {% block title %}AppBundle:Example:Twig:index{% endblock %}
7
8  {% block main %}
9      {# vlozi a vykresli obsah vkladane sablony #}
10     {% include 'Example/Twig/included' %}
11
12     {# embed je jako include ale je mozne dodatecne prepsat bloky embedovane sablony #}
13     {% embed 'Example/Twig/embed.html.twig' %}
14         {% block top %}
15             Obsah TOP z šablony "index.html.twig"
16         {% endblock %}
17     {% endembed %}
18
19     {# vykresleni bloku z sablony 'Example/Twig/used.html.twig' #}
20     {{ block('used_block') }}
21
22     {# vykresleni makra z sablony 'Example/Twig/macro.html.twig' #}
23     {{ forms.text() }}
24
25     {# vykresli sablonu pomoci kontroleru #}
26     {{ render(controller('AppBundle:Example:Routing:hello', {name: "Kontroler"})) }}
27
28     {#vykresli sablonu na zaklade routy#}
29     {{ render(url('homepage')) }}
30 {% endblock %}
31
32 {% block sidebar %}
33     {{ parent() }}
34     {{ show_source_code(_self) }}
35 {% endblock %}
36
```

Obrázek č. 1

Co když ale při vývoji pracujeme s více soubory a potřebujeme se mezi nimi přepínat? Co když si chceme vytvořit nějaký nový soubor? Co když chceme spustit debugování? Nebude nám to bez pomocných panelů a toolbarů trvat dlouho? Nebude. Pro efektivní způsob programování na PC je optimální pracovat pouze s klávesnicí a vyhnout se myši.

Přiznávám, že přechod z myši čistě na klávesnici může být tvrdý a nějaký čas to potrvá. Jakmile si ale zvyknete, zjistíte, že pracujete rychleji. Pojďme se tedy podívat, co nám PHPStorm nabízí.

Základní principy

Pokud se rozhodnete, že opustíte myš a přejdete na ovládání vývojového prostředí pouze pomocí klávesnice, zapamatujte si dvě základní pravidla. Pokud chcete přepnout fokus z jakéhokoli postranního panelu nebo okna do editoru se zdrojovým kódem, stačí zmáčknout klávesu *ESC*.

Druhé užitečné pravidlo se týká vyhledávání. V PHPStormu platí, že každý výpis položek je snadno filtrovatelný. Stačí když začnete jednoduše psát to, co hledáte a seznam položek se okamžitě smrskne na ty, co obsahují napsaný řetězec.

Navigace

Akce	Výchozí zkratka	Poznámka
<i>Nastavení IDE</i>		
Nastavení	ctrl + alt + S	
Vyhledání a otevření akce	ctrl + shift + A	keymap, plugins
<i>Panely</i>		
Zobrazení adresářové struktury	alt + 1	
Zobrazení drobečkové navigace	alt + home	pak alt + insert
Zobrazení postranních toolbarů	2x alt po sobě	držet alt po 2. stisku
Zobrazení terminálu	alt + F12	
Zobrazení struktury souboru	alt + 7, ctrl + F12	
<i>Otevření, vyhledání souboru, třídy, metody, ...</i>		
Nedávno otevřené soubory	ctrl + E	ctrl + E, ctrl + tab
Vyhledání souboru v projektu	ctrl + shift + F	
Vyhledání a otevření třídy (class)	ctrl + N	
Vyhledání a otevření souboru	ctrl + shift + N	:<line-number> control/*
Vyhledání symbolu (proměnná, metoda, ...)	ctrl + shift + alt + N	

Vyhledání všeho (hledá se v předešlých)	2x shift po sobě	
<i>Informace o souboru, třídě, metodě</i>		
Struktura souboru	ctrl + F12	
Hierarchie třídy (strom dědičnosti)	ctrl + H	
Zobrazení dokumentace k metodě	ctrl + Q	
Zobrazení parametrů metody	ctrl + P	
<i>Navigace v souboru a po souborech, info o definici</i>		
Vyhledání textu	ctrl + F	pak F3 nebo ctrl + L
Přechod na konkrétní řádek	ctrl + G	
Vytvoření/zrušení záložky (bookmark)	ctrl + shift + <num>	
Přechod na záložku (bookmark)	ctrl + <num>	
Náhled definice volané metody	ctrl + shift + I	
Přechod na definici volané metody	ctrl + B	
Zobrazení výskytu použití metody	ctrl + B, alt + F7	ctrl + B jen když je kurzor na definici ctrl + alt + <- navrat zpet
Přechod na rodičovskou třídu	ctrl + U	

Psaní a generování kódu

Při práci v týmu je velice užitečné si dohodnout jednoznačné tzv. coding standards. Jedná se o sadu pravidel určujících, jak má být naformátovaný zdrojový kód. Například jestli se mají pro odsazování používat mezery nebo tabulátory, jestli má být otevírací složená závorka na stejném řádku jako je definice třídy nebo na novém, apod. Pokud tým tyto zásady dodržuje, je kód pro všechny členy týmu dobře čitelný.

PHPStorm proto nabízí možnost si každé jednotlivé pravidlo pro jednotlivý jazyk, ať už PHP, Twig, Javascript, CSS, nebo jiný, přesně definovat pro každý váš projekt. Nastavení najdete v Editor > Code Style. Editor se pak podle těchto pravidel řídí, a když píšete svůj kód, automaticky ho podle toho formátuje. Na základě definovaných pravidel si navíc můžete snadno a automaticky naformátovat všechny již vytvořené soubory ve vašem projektu. Pro rychlé naformátování slouží klávesová zkratka ctrl + alt + L.

PHPStorm se snaží práci programátorů co nejvíce usnadnit a v základu nabízí rychle použitelné šablony pro různé typy souborů nebo pro často se opakující konstrukty. Má také podporu sady nástrojů [Emmet](#).

Šablony souborů (file templates)

Při programování webové aplikace potřebujeme vytvářet řadu různých typů souborů, například HTML, CSS, Javascript, Typescript, soubor s PHP třídou nebo interfacem apod. Přitom vnitřní struktura těchto souborů se stále opakuje. PHPStorm proto nabízí možnost vytvořit si pro jednotlivé typy souborů vlastní šablony. Šablony lze nastavit v sekci *Editor > File and Code Templates*.

Živé šablony (live templates)

Ještě užitečnější funkce než *šablony souborů* jsou tzv. *živé šablony*. Ty přináší možnost vytvořit si šablonky pro často se opakující části zdrojového kódu a uložit si je pod vlastní textovou zkratku. Pokud pak tuto zkratku napíšeme kdekoli v našem zdrojovém kódu a zmáčkneme klávesu *Tab* zkratka se změní na odpovídající šablonu.

V živých šablonách můžeme vytvářet proměnné ve formátu `<nazev-promenne>`. Jakmile se pak zkratka rozvine na odpovídající šablonu, kurzor automaticky (po stisku klávesy *Enter*) skáče na místa s definovanou proměnnou. V šabloně se stejná proměnná může vyskytovat i vícekrát. V takovém případě se text, který napíšeme namísto proměnné píše na všechna místa, kde se tato proměnná vyskytuje.

Jako příklad si můžeme ukázat, jak vytvořit definici živé šablony pro rychlé vytvoření PHP konstruktoru. V nastavení v sekci *Editor > Live Templates* vybereme jazyk PHP. Potom klikneme vpravo na zelenou ikonku se symbolem plus a vybereme *Live Template*. Následně do pole *Abbreviation* vyplníme text `__c` do pole *Description* vyplníme text `konstruktor`. Nyní už do pole *Template* text vyplníme text šablony a uložíme.

```
public function __construct($PARAMETERS$)
{
    $END$
}
```

Emmet

Živé šablony jsou užitečné například při psaní PHP kódu, kde lze nalézt často se opakující konstrukty jako např. definice konstruktoru, funkcí, metod, psaní foreach cyklů atd. Html kódér

by ale mohl namítnout, že struktura HTML kódu, přestože má jasná pravidla, je pro každý web velmi individuální a unikátní. Představme si například HTML kód pro nečíslovaný seznam. Víme, že se vytváří pomocí značek `` a ``, ale při vytváření live šablony nevíme, kolik položek seznamu budeme potřebovat.

V takových případech je možné využít nástroj [Emmet](#), který má v PHPStormu zabudovanou podporu. Princip spočívá v tom, že místo HTML značek píšeme pouze zkrácenou definici v podobě obyčejného textu s názvy značek, názvy HTML atributů, jejich počty a následností. Poté stačí zmáčknout klávesu `TAB` a text se rozvine na plnohodnotný HTML kód. Pro lepší představu si můžete vyzkoušet následující příklad:

```
div#menu-wrapper>ul.menu>li.menu-item.item-$*5>a[href=www.seznam.cz]
```

Akce	Výchozí zkratka	Poznámka
<i>Označení textu, kódu, vkládání</i>		
Označení obsahu celého souboru	ctrl + a	
Označení po slovech	ctrl + shift + [← →]	
Rozšiřující se označení podle kontextu	ctrl + w	alt + UP, alt + DOWN
Vložit ze schránky s historií	ctrl + shift + v	
<i>Formátování kódu</i>		
Naformátování podle zadaných pravidel	ctrl + alt + L	záleží na tom, jaké okno má fokus
<i>Generování kódu</i>		
Zobrazení možností oprav nebo generování	alt + enter	
Implementování metod rodičovské třídy	ctrl + I	
Přetížení metod rodičovské třídy	ctrl + O	
Dokončit rozepsaný příkaz	ctrl + shift + enter	ctrl + enter - rychlejší
Napovídání a dokončování názvů	ctrl + space	
<i>Živé šablony</i>		
Zobrazení dostupných živých šablon	ctrl + J	
Obklopit živou šablonou	ctrl + alt + T	

Refaktoring

Refaktorování, tedy úpravy kódu za účelem zpřehlednění jeho struktury a vylepšení čitelnosti, je pro programátora nevyhnutelná činnost, se kterou se potýká dennodenně. Proto i tady se PHPStorm snaží práci co nejvíce usnadnit.

Akce	Výchozí zkratka	Poznámka
<i>Refaktorování</i>		
Zobrazení okna s možnostmi refaktoringu	ctrl + alt + shift + T	
Vyčlenění metody	ctrl + alt + M	
Vyčlenění proměnné	ctrl + alt + V	
Přesun logických částí kódu	ctrl + shift + arrows	
Přejmenování názvu třídy, metody, proměnné	shift + F6	
<i>Vícekurzorový režim pro rychlé úpravy více míst najednou</i>		
Sloupcový režim	alt + označení myši	
Více kurzorový režim	alt + klik myši	
Více kurzorový režim podle názvu	alt + j	
Více kurzorový režim podle názvu - vše	alt + ctrl + shift + j	

Pluginy

PHPStorm je skvělý editor a už v základu má spoustu funkcí a vychytávek. Přesto ale existuje mnoho specifických jazyků a úkolů, na které nestačí a v takové situaci je nutné se poohlédnout po pluginech. Pro vývoj Symfony2 aplikací jednoznačně doporučujeme následující pluginy:

- PHP Annotation
- Symfony2
- Composer
- Ignore
- Live edit
- <https://github.com/ruslan-polutsygan/phpstorm-adder-remover-generator-plugin>