

# Podstawy rozwoju szablonu

## Podstawy rozwoju szablonu

- Podstawy rozwoju szablonu
  - Koncepcja
    - Model
    - Widok
    - Kontroler
    - Jak działa szablon
    - Specyficzne foldery szablonu
  - Generalne informacje
    - Jak działa szablon
    - Specyficzne foldery szablonów
    - Podgląd na folder tematów
  - Technologie
    - Podstawy
    - Smarty 3
    - Bootstrap v3
    - Sass i Compass
  - Porady
    - Rozszerzenia przeglądarki
    - Najlepsze porady

## Koncepcja

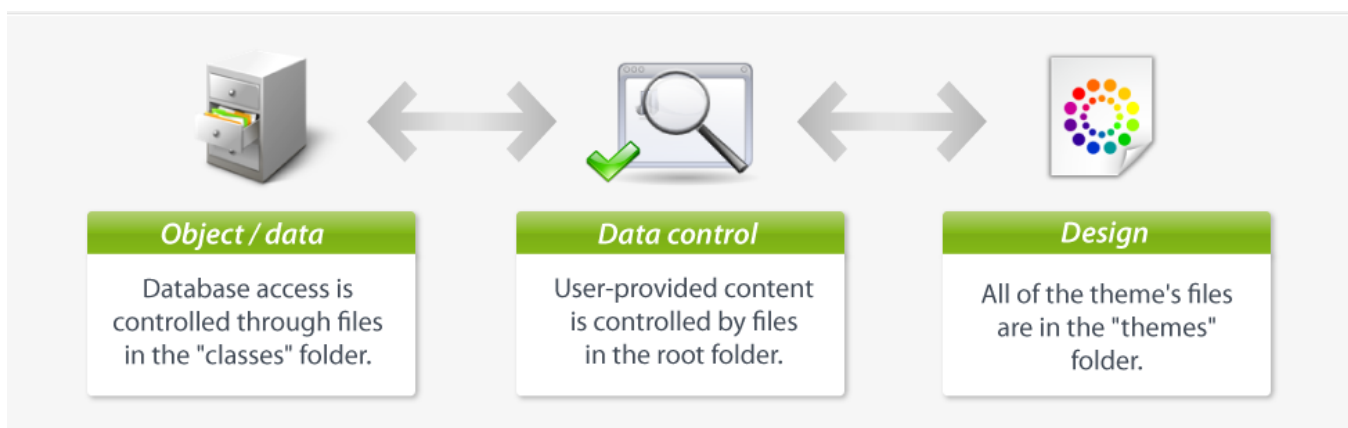
System szablonów PrestaShop pozwala developerom i programistom łatwo zbudować swój własny szablon, pod warunkiem, że mają solidne podstawy techniczne.

System szablonów jest zbudowany na 3 - warstwowej architekturze :

- **Poziom danych (obiekty danych)**. Dostęp do bazy danych jest kontrolowany przez pliki w folderze `/classes`.
- **Warstwa aplikacji (kontrola danych)**. Treść dostarczana przez użytkowników jest kontrolowana przez pliki w folderze głównym.
- **Poziom prezentacji (projekt)**. Wszystkie pliki tematu są w folderze `/themes`.

Szablon należy do warstwy prezentacji: Szablon plików znajduje się w folderze `/themes`, będzie obsługiwać dane dostawiane przez warstwę aplikacji i ich powrót do warstwy aplikacji.


# PrestaShop's 3-tier architecture



Zobacz w Wikipedii więcej informacji na temat architektury 3 warstwowej: [http://en.wikipedia.org/wiki/Multitier\\_architecture#Three-tier\\_architecture](http://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture)

## Architektura 3 - warstwowa ma wiele zalet:

- Ułatwia odczytanie kodu.
- Programiści mogą dodawać i edytować kod szybciej.
- Graficy i integratorzy HTML mogą pracować z zakresem powierzonych folderów /themes bez odczytania czy zrozumienia choćby jednej linijki kodu PHP.
- Programiści mogą pracować na dodatkowych danych i modułach, które mogą wykorzystać integratorzy HTML.

 Jest to ta sama zasada jak w Model-View-Controller (MVC) architekturze, tylko jest to prostszy i bardziej przystępny sposób.  
Dowiedz się więcej na temat MVC w Wikipedii: <http://en.wikipedia.org/wiki/Model-view-controller>

### Model

Model pokazuje zachowanie aplikacji: przetwarzania danych, interakcji bazy danych etc.

Opisuje lub zawiera dane, które zostały przetworzone przez aplikację. Zarządza tymi danymi i zapewnia ich integralność.

### Widok

Widok jest interfejsem użytkownika z którym użytkownik wchodzi w interakcję.

Jego pierwszym zadaniem jest wyświetlanie danych, które są udzielane przez model. Jego drugą rolą jest obsługa wszystkich działań ze strony użytkownika (kliknięcia myszy, wybranie elementów, przyciski, etc.), a także wysyłanie tych zdarzeń do kontrolera.

Widok nie przetwarza, tylko wyświetla widok przetwarzania wykonywanego przez model i komunikuje się z użytkownikiem.

### Kontroler

Kontroler ma za zadanie synchronizować zdarzenia pomiędzy Modelem a Widokiem i aktualizować obydwa jeżeli jest taka potrzeba. Otrzymuje on wszystkie zdarzenia użytkownika i inicjuje potrzebne czynności do wykonania.

Jeżeli jest potrzeba zmian danych, Kontroler zapyta Model o zmianę danych, a Model z kolei powiadomi, że dane zostały zmienione, tak aby widok mógł się aktualizować samoczynnie.

### Jak działają szablony

Szablon PrestaShop to zestaw plików, które można edytować w celu zmiany wyglądu sklepu internetowego.

Oto kilka ważnych informacji:

- Wszystkie tematy i pliki mają swoje lokalizacje w folderze głównym /themes.
- Każdy motyw ma swój własny sub-folder w folderze głównym tematów.
- Każdy temat jest wykonany z pliku szablonów (.tpl), pliku obrazów (.gif, .jpg, .png), z jednego lub więcej plików CSS (.css), i czasami także z plików JavaScript (.js).
- Każdy temat ma podgląd 180\*200 preview.jpg pliku w folderze umożliwiający właścicielowi sklepu zobaczenie wyglądu sklepu bezpośrednio z panelu administracyjnego, a następnie wybór odpowiedniego tematu.

### Specyficzne foldery szablonu

Trzeba by wiadomym, że istnieje 5 głównych folderów PrestaShop:

- `/modules`: To gdzie moduły są zlokalizowane. Moduł zawiera pliki szablonów, które mogą być w części tematycznej przeddefiniowane.
- `/themes`: Jest miejscem gdzie wszystkie tematy są zlokalizowane. Domylny temat 1.5 jest w folderze `/default` (w 1.4, był w `/prestashop`).
- `/mails`: Tutaj są zlokalizowane wszystkie e-maile szablonu. E-mail szablonu powinien dokładnie odzwierciedlać styl i projekt głównego szablonu. Każdy sub-folder zawiera szablony dla danego języka.
- `/img`: W tym miejscu są zlokalizowane wszystkie obrazy sklepu. **Specyficzne obrazy dotyczące szablonu są w waszym folderze `/img`**
- `/pdf`: Tutaj są zlokalizowane wszystkie modele dokumentów.

## Generalne informacje

### Jak działają szablony

Szablon PrestaShop jest zestawem plików, które można edytować w celu zmiany wyglądu waszego sklepu internetowego.

Oto kilka ciekawostek:

- Wszystkie motywy mają swoją lokalizację w folderze głównym `/themes`.
- Każdy z motywów ma swój własny sub-folder w głównym folderze tematów.
- Każdy z motywów jest wykonany z pliku szablonów (`.tpl`), plików obrazów (`.gif`, `.jpg`, `.png`), z jednego lub więcej plików CSS (`.css`), i czasami nawet z plików JavaScript (`.js`).
- Każdy z nich ma `180*200 preview.jpg` plik obrazka w folderze, umożliwiający właścicielowi sklepu zobaczenie wyglądu szablonu bezpośrednio z pozycji panelu administracyjnego i następnie wybór odpowiedniego motywu.

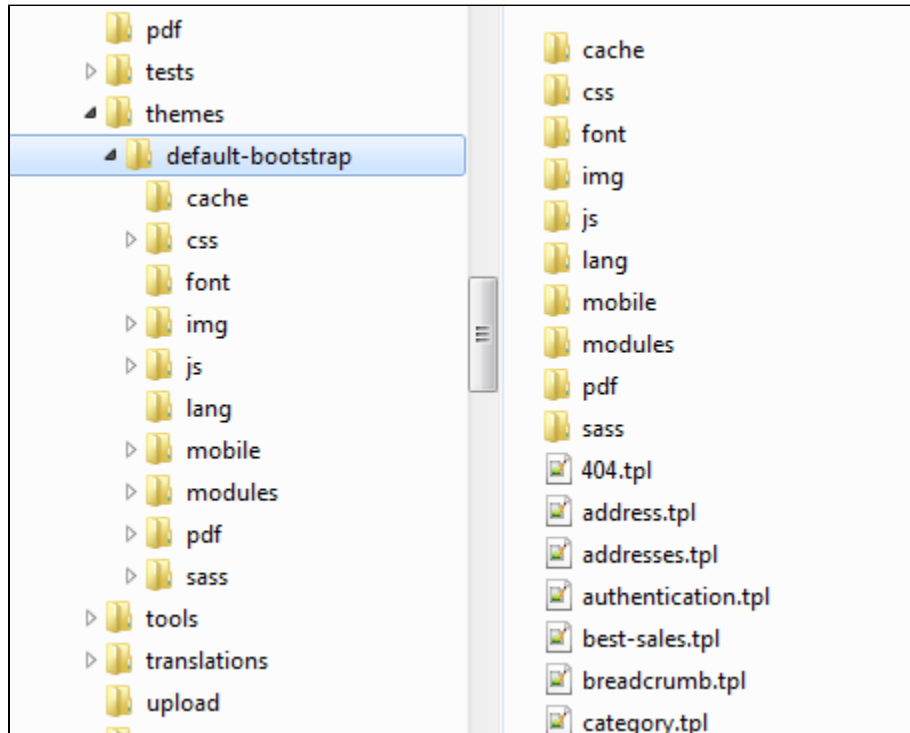
### Specyficzne foldery szablonów

Jako developer musisz zwrócić uwagę na 5 głównych folderów:

- `/modules`: to gdzie wszystkie moduły są zlokalizowane. Moduły szablonów mają swoje pliki, możesz również przeddefiniować części tematyczne.
- `/themes`: tutaj są zlokalizowane wszystkie motywy.
  - Domylny motyw 1.6 jest w folderze `/default-bootstrap`.
  - Domylny motyw 1.5 był w folderze `/default`.
  - Domylny motyw 1.4 był w folderze `/prestashop`.
- `/mails`: Tutaj są zlokalizowane wszystkie szablony. Szablony e-mail powinny idealnie odzwierciedlać styl i design głównego motywu. Każdy sub-folder zawiera szablony dla danego języka.
- `/img`: Tutaj są zlokalizowane obrazy sklepu. **Specyficzne motywy obrazków są zapisywane w waszym folderze `/img`.**
- `/pdf`: Tutaj są zlokalizowane modele dokumentów.

### Podgląd na folder tematów

Oto podgląd na strukturę tematów w PrestaShop (tutaj domylny motyw):



- Folder `/css`, zawiera wszystkie pliki CSS
  - Folder `sass` zawiera wszystkie foldery `Sass.scss` Pliki ródowe, za nim zostaną skompilowane w plikach CSS.
- Folder `/font` zawiera pliki trzcionek.
- Folder `/img` zawiera wszystkie obrazki.
- Folder `/js` zawiera wszystkie pliki JavaScript.
- Folder `/lang` zawiera wszystkie tłumaczenia tematu. Jego prawa dostępu powinny być ustawione na CHMOD 666 (na przykład), tak, aby narzędzie panelu administracyjnego do tłumaczenia mogło odczytywać i zapisywać do niego.
- Folder `/mobile` zawiera mobilną wersję strony.
- Folder `/modules` zawiera potrzebny szablon plików do wielu modułów.
- Folder główny zawiera tylko pliki TPL (pliki Smarty), jak również plik miniatur `preview.jpg`.

## Technologie

### Podstawy

Tak jak każda strona, szablony PrestaShop są wyświetlane jako mix HTML, CSS i JavaScript. Te trzy języki są podstawą każdej strony internetowej, dlatego powinieneś je znać, zanim podejmiesz się edycji istniejącego tematu lub utworzysz nowy.

Domyślny szablon (`/themes/default-bootstrap`) używa HTML5, CSS3 i JQuery-steruje JavaScript (<http://jquery.com/>). Używamy jQuery v1.11+.

### Smarty 3

W swojej istocie motyw PrestaShop jest oparty na silniku szablonu Smarty 3 (<http://www.smarty.net/>), co pozwala programistom i developerom łatwo zbudować swój motyw, z niewielką wiedzą techniczną.

Smarty 2 jest od dawna przestarzała i nie powinna być używana do motywów w PrestaShop 1.6.

### Bootstrap v3

Od kiedy wersja 1.6 PrestaShop zawiera strukturę czoo Bootstrap (<http://getbootstrap.com/>). Jest to zbiór narzędzi HTML i CSS które pozwalają zbudować programistom strony (co oznacza, że można je łatwo dostosować do dowolnej wielkości ekranu). Początkowo służyło to do wewnętrznych wzorców i projektów w firmie Twitter, ale szybko urosło do najczęściej używanej ramy struktury czoo.

PrestaShop używa Bootstrapa dla obydwu narzędzi szablonu zarówno czooowego jak i panelu administracyjnego.

 Projektanci tworzący nowy panel administracyjny muszą używać Bootstrapa.

Projektanci tworzący nową stronę czoo powinni użyć Bootstrapa. Nie jest to obowiązkowe, ale Bootstrap jest mocno wkomponowany w PrestaShop.

Bootstrap początkowo używa mniej dynamicznego arkusza stylów LESS (<http://lesscss.org/>), ale PrestaShop używa Bootstrapa dla portu Sass (<https://github.com/twbs/bootstrap-sass>), Sass jest językiem który inspirował LESS. To umożliwiło użycie ramy Compass.

Przewodnik dla projektanta poruszy ten temat gdzieś o Bootstrapie w późniejszym rozdziale.

## Sass i Compass

Wersja 1.6 także daje możliwość użycia języka skryptu Sass (<http://sass-lang.com/>), i komponuje się z ramą Compassa.

Sass (skrót skadniowy arkusza stylów) jest preprocesorem CSS3 czyli warstwą pomiędzy arkuszami stylów których jeste autorem i plikami CSS które są przeglądane (jak powiedział Dan Cederholm: <http://alistapart.com/article/why-sass>). Ma on na celu ułatwienie użycia podczas tworzenia plików CSS poprzez rozszerzenie języka CSS:

- Dodaje zmienne wstawki i funkcje (dziedziczy).
- To pomaga w zapobieganiu powtórzeniu kodu.
- To sprawia, że łatwiej jest zachować kod.
- To sprawia, że pisać się go szybciej.
- etc.

Zamiast bezpośredniego tworzenia plików `.css`, projektant tworzy plik `.sass` albo `.scss` który jest interpretowany do standardu pliku CSS przy użyciu wiersza polecenia lub narzędzia online.

Należy pamiętać, że można to zrobić zmieniając sposób w jaki używane jest zwykły proces CSS: Jak znowu pisze Dan Cederholm, "Sass jest przedzeniem CSS3, a jego SCSS ("Sassy CSS") skadnia jest rozszerzeniem CSS3. Oznacza to, że każdy dokument CSS3 jest równie ważnym dokumentem SCSS."

Ze swojej strony, Compass (<http://compass-style.org/>) jest ramą Sassa – czyli kombinacją narzędzi jest stworzony, aby pomóc budować i zarządzać plikami CSS używając języka Sass. Jest napisany w Ruby.

Ten przewodnik projektanta zagłęb się mocniej w Sass i Compass w następnym rozdziale.

## Porady

### Rozszerzenia przeglądarki

Wszyscy projektanci stron i programiści powinni stosować następujące rozszerzenia przeglądarki:

- Firefox: używa narzędzi programistów (zobacz <https://developer.mozilla.org/en-US/docs/Tools>). Mogą tobie nawet pomóc przy pracy z Sassem: <https://hacks.mozilla.org/2014/02/live-editing-sass-and-less-in-the-firefox-developer-tools/>

- Firefox: alternatywnie instaluje wtyczkę Firebug (<http://getfirebug.com/>), darmowe rozszerzenie dla atwiejzego porównania i debugowania pomiędzy twoim CSS a wyjciem.
- Firefox/Chrome: zainstalowa programist stron (<http://chrispederick.com/work/web-developer/>), bezpatne rozszerzenie dodajce wiele rcznie stworzonych narzdzi przez programistów do twojej przegldarki.
- Safari 5+: uywa inspektora stron (zobacz [https://developer.apple.com/library/safari/documentation/AppleApplications/Conceptual/Safari\\_Developer\\_Guide/Introduction/Introduction.html](https://developer.apple.com/library/safari/documentation/AppleApplications/Conceptual/Safari_Developer_Guide/Introduction/Introduction.html)).
- Chrome: uywa narzdzi programisty (zobacz <https://developer.chrome.com/devtools/index>).
- Opera 9.5+: uywa Dragonfly, w peni funkcjonalne rodowisko debugowania (zobacz <http://www.opera.com/dragonfly/>).
- Internet Explorer 8+: uywa narzdzi programisty (zobacz <http://msdn.microsoft.com/en-us/library/dd565628%28v=vs.85%29.aspx>).
- Internet Explorer 6 and 7: install the IE Developer Toolbar (see <http://www.microsoft.com/en-us/download/details.aspx?id=18359>), albo uyj Firebug Lite (zobacz <https://getfirebug.com/firebuglite>).

Zapewnij one wiele przydatnych narzdzi, wśród których DOM explorer, edytor CSS, network inspector, itp., a to ogromna pomoc podczas debugowania HTML, CSS, JavaScript, i nawet da Ajaxa.

## Najlepsze porady

Tutaj znajduje si niewyczerpujca lista najlepszych porad którymi powiniene si kierowa kiedy kreujesz widok:

1. Nie mieszaj kodu HTML i PHP; uywaj tagów Smarty w celu uzyskania dynicznej strony.
2. Nie mieszaj kodu HTML i CSS; kod CSS wstawiaj w separator pliku .css.
3. Zawsze uprawamaciaj swój kod HTML i CSS uywaj walidatora W3C: <http://validator.w3.org/> dla HTML i XHTML, <http://jigsaw.w3.org/css-validator/> dla CSS.
4. Nie rób zapyta SQL z kontrolera PHP (.php w pliku root PrestaShop); uywaj istniejce metody z klasy PrestaShop, albo uyj nowej metody dla tej klasy.
5. Zawsze sprawdzaj czy metoda której potrzebujesz czasem ju nie istnieje w dostpnych klasach.
6. Nigdy nie edytuj plików wasnych PrestaShop, zawsze twórz swój kod w moduach w celu uatwienia aktualizacji.
7. Zawsze upewnij si e tworzysz przejrzysty i czytelny kod, na przyszło twój kod bdzie atwy do odczytania przez inne osoby.
8. Komentarze do kodu i nazwy metod napisz w prostym jzyku angielskim.
9. Kiedy edytujesz motyw na stronie produkcyjnej sklepu, zawsze pamitaj eby najpierw umieci sklep w trybie konserwacji poprzez panel administracyjny "Konserwacja " w zakadce preferencje.
10. Uy waj nowoczesnych przeglądarek takich jak, Firefox (najnowsza wersja non-beta), Google Chrome (najnowsza wersja non-beta), IE10+ albo Opera (najnowsza wersja non-beta) – i upewnij si e twoi znajomi rodzina rónie!
11. Jeli to tylko moliwe uy waj CSS sprites (czytaj <http://www.alistapart.com/articles/sprites> i <http://www.alistapart.com/articles/sprites2/>).
12. Wycz pamie podrczn swojej przegldarki kiedy pracujesz.