



Tworzenie witryn internetowych PHP/Java

(mgr inż. Marek Downar)

Rodzaje zawartości



- Zawartość statyczna
 - Treść statyczna (np. nagłówek, stopka)
 - Layout, pliki multimedialne, obrazki, elementy typograficzne,
 - Javascript wzbogacający interakcję z użytkownikiem
- Zawartość dynamiczna – kod PHP
 - Wykorzystanie języka skryptowego PHP w celu dynamicznego ładowania zawartości do warstwy prezentacji,
 - Wykorzystanie PHP w celu dostępu do informacji przechowywanych w bazie (np. MySQL, Postgres, Oracle) bądź też w plikach (np. XML),
 - Wykorzystanie PHP w celu implementacji logiki biznesowej aplikacji
 - Wykorzystanie PHP w celu generowania zawartości częściowej strony (np. dla żądań Ajax)

PHP – dobre zwyczaje



- `<?php ?>` - dlaczego nie `<? ?>`
- Stosowanie wcięć
- W przypadku wstawek (instrukcji jednolinijkowych generujących układ strony) – wykorzystanie rozbudowanej formy instrukcji

```
<?php foreach ($i in $table): ?>
```

```
<?php endforeach;?>
```

Zamiast:

```
<? foreach($i in $table){?>
```

```
<? } ?>
```

PHP – typowy projekt

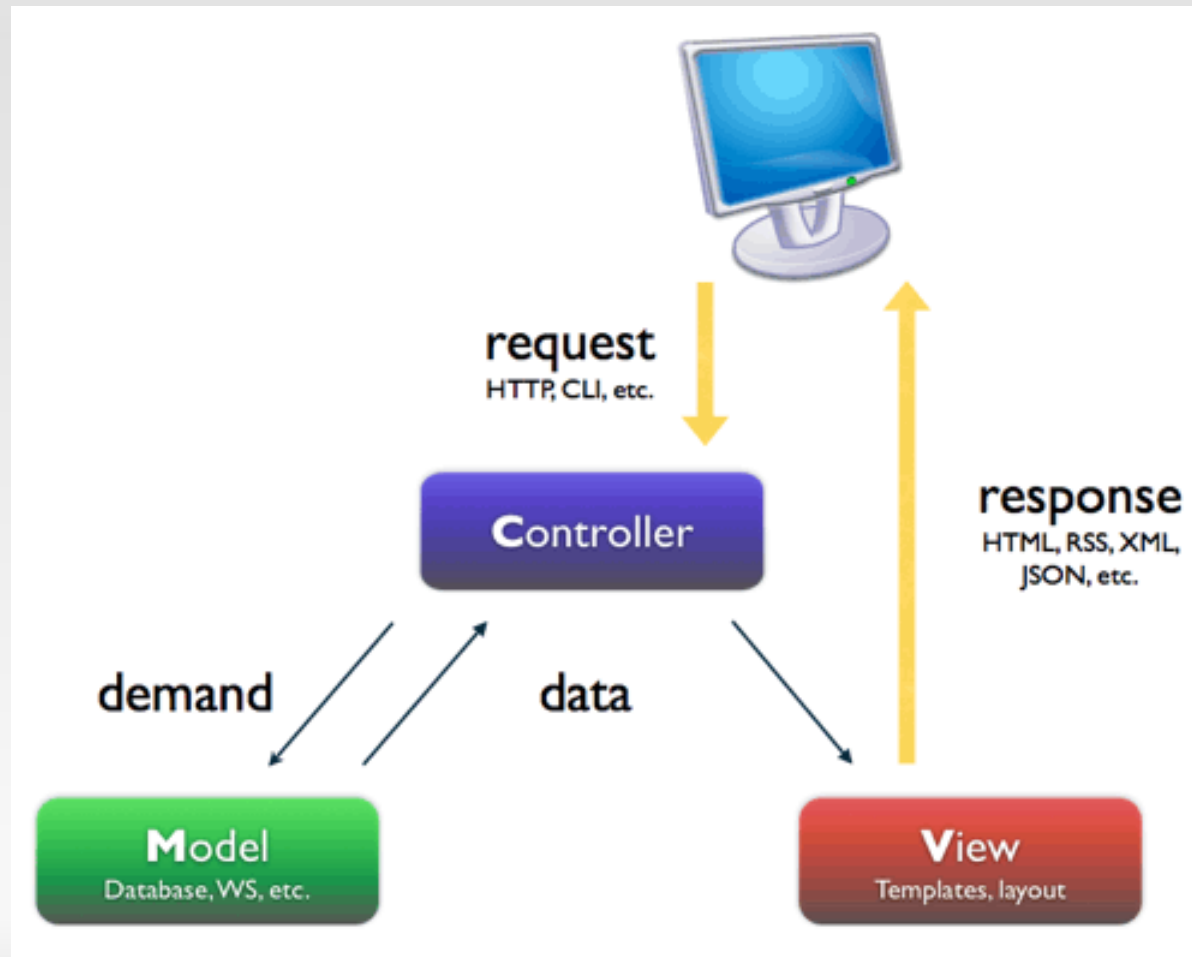


- Logiczny podział aplikacji na standardowe pliki biblioteczne, pliki bibliotek użytkownika, pliki użytkowe, oraz wstawki.
- Importowanie plików przy pomocy instrukcji `require/require_once` oraz `include/include_once`
- Definiowanie nawigacja pomiędzy stronami.
 - Parsowanie i interpretacja otrzymanych parametrów (GET, POST) po stronie serwera

Warstwy aplikacji internetowej



- MVC – Model View Controller



PHP HTTP features



- Redirect

- HTML

- ```
<meta HTTP-EQUIV="REFRESH" content="0; url=
http://www.eti.pg.gda.pl" />
```

- PHP

- ```
header("Location: http://www.eti.pg.gda.pl")
```

- \$_POST, \$_GET, \$_SESSION

- Ograniczenia dla długości przesyłanych danych

Przekazywanie parametrów



- Przekazywanie parametrów pomiędzy żadaniami odbywa się przy pomocy poleceń GET oraz POST,
- Parametry oraz ich wartości znajdują się w tablicach asocjacyjnych `$_GET` oraz `$_POST`.
- W przypadku (błędnej konfiguracji!!!) możliwe jest także odwołanie się do tych parametrów jako do zmiennych kodu PHP
(`register_globals = off !!`)

\$_GET



- W przypadku, gdy parametry żądania przekazywane są od klienta przy pomocy metody GET, trafiają one do tablicy asocjacyjnej \$_GET

<http://www.eti.pg.gda.pl/index.html?q=szukaj>

```
$_GET["q"]="szukaj";
```



- Z reguły służą do przesyłania parametrów przy pomocy metody POST
- Zapewniają zaawansowaną interakcję z użytkownikiem
- Typy w formularzu `input type text password, hidden, select, textarea, itp.`

```
<form action="nazwa_skryptu.php" method="post/get">
```

```
</form>
```

- Enctype – typ zawartości (np. `multipart/form-data` do przesyłania plików)



- W przypadku wysłania parametrów przy pomocy metody POST, znajdą się one w tablicy asocjacyjnej `$_POST` w postaci

```
<form method="post" action="http://www.eti.pg.gda.pl/index.php">
```

```
<input type="text" name="q" value="szukaj" />
```

```
<input type="submit" value="Wyślij" />
```

```
</form>
```

```
$_POST["q"] = "szukaj";
```



- Znajdują się tutaj wartości przypisane w ramach sesji użytkownika
- Sesja jest tymczasowym zbiorem danych istniejącym dopóki nie zostanie wyłączona przeglądarka, bądź nie minie czas jej wygaśnięcia (konfigurowalne z poziomu php.ini)
- Mechanizm nieodłączny w tworzeniu aplikacji internetowych (wymagających np logowania bądź przechowywania koszyka)

■

\$_SESSION – imitacja stanowości



- Zachowanie wartości parametrów w czasie nawigowania po stronie – "imitacja stanowości"

Set-Cookie: name=value; domain=domain.name;
path=path-within-server; [secure]

Cookie: klucz1=wartość1; klucz2=wartość2;...

`$_SESSION` - php.ini



- `session.name`
- `session.use_only_cookies`
- `session.cookie_lifetime`
- `session.auto_start`
- `session.gc_maxlifetime`
- `session.cookie_lifetime`

\$_SESSION



- Możliwość wykorzystania mechanizmu sesji wymaga na samym początku skryptu PHP wywołania funkcji `session_start()`.

```
<?php
    session_start();
    $_SESSION['zmienna1']="wartość";
    $_SESSION['zmienna2']=2;
?>
```

```
<?php
    session_start();
    echo $_SESSION['zmienna1'];
    echo $_SESSION['zmienna2'];
?>
```

\$_REQUEST



- Tablica zbiorcza - zawiera zarówno parametry przekazane przy pomocy `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`
- Stosowana, gdy dla logiki biznesowej nie ma znaczenia sposób dostarczenia wartości dla danej zmiennej



- Zmienne, którym przypisane zostały informacje dotyczące obsługi żądania po stronie serwera

`$_SERVER['PHP_SELF']` – aktualnie wykonywany skrypt, ścieżka względna

`$_SERVER['REQUEST_METHOD']` – metoda, przy pomocy której wygenerowano żądanie

`$_SERVER['QUERY_STRING']` – zawartość URL po znaku ?

`$_SERVER['REMOTE_ADDR']` – adres IP klienta

`$_SERVER['HTTP_USER_AGENT']` – informacje dotyczące przeglądarki klienta

`$_SERVER['HTTP_REFERER']` – strona, z której został wywołany zasób

\$_FILES & \$_ENV



- `$_FILES` - W przypadku ustawienia w formularzu `enctype="multipart/form-data"` znajdują się tutaj pliki typu:

```
<input type="file" name="polepliku" />
```

```
$_FILES['polepliku']
```
- `$_ENV` - zmienne, którym przypisywane są wartości w obrębie środowiska

\$ _COOKIE



- Set-Cookie: <name>=<value>[; expires=<date>][; path=<path>][; domain=<domain name>][; secure]
- Pliki cookie są małymi porcjami informacji zachowywanymi na komputerze użytkownika
- Potencjalne niebezpieczeństwo ze strony aplikacji próbujących wyłudzić informacje od użytkownika (takie, jak IP, bądź operacyjny system).
- W plikach cookie nie powinno się przechowywać żadnych istotnych informacji w postaci haseł bądź innych danych użytkownika
- Pliki cookie są przechowywane w powszechnie znanym katalogu. Każdy kto ma dostęp do komputera użytkownika może potencjalnie otworzyć plik cookie i podejrzeć informacje w nim zawarte.



- Z uwagi na fakt, że niektórzy wyłączają sobie domyślnie obsługę plików cookie – z tytułu potencjalnego niebezpieczeństwa – nie jest dobrym poleganiem na nich w przypadku implementacji logiki aplikacji.
- Należy w nich zawrzeć informacje jedynie takie, które miałyby użytkownikowi ułatwić powrót do aplikacji po ponownym uruchomieniu przeglądarki.



- W celu zachowania zmiennej w pliku cookie należy wykorzystać funkcję `setcookie()`.
 - Nazwa cookie
 - Wartość cookie
 - Czas wygaśnięcia (expiration time)
 - Ścieżka – opcjonalnie
 - Domena – nazwa domeny, dla której ustawiany jest plik cookie – opcjonalnie
 - Czy plik wymaga bezpiecznego połączenia
- `setcookie('cookienamie', 'value', 'expiration time', 'path', 'domain', 'secure connection')`

Cookies – PHP - przykład



```
<?php ;  
    setcookie('username', 'Marek', time()+60);  
    session_start();  
    $_SESSION['param']='wrt';  
?>
```

```
<?php  
    session_start();  
    echo $_COOKIE['username'];  
?>
```