

# Instalacja WAMP i obsługa



## 1. Co to jest Wamp?

WAMP to skrót od **W**indows **A**pache **M**ySQL **P**HP – odpowiednik Linuxowego LAMP-a. Z poziomu jednego instalatora możemy zainstalować kilka usług na naszym komputerze. Najważniejsze usługi które uruchomimy po instalacji to:

- Apache – serwer WWW. O ile pliki \*.html oraz \*.js o które te pierwsze są rozszerzane możemy otworzyć po prostu za pomocą przeglądarki o tyle skrypty PHP są wykonywane po stronie serwera i dopóki takowego nie ma, nie będą mogły po prostu zadziałać.
- MySQL – serwer bazodanowy. Tworzenie stron w oparciu o PHP bez żadnego serwera bazodanowego praktycznie nie ma sensu. Za pomocą PHP chcemy, aby strona była bardziej żywa – nie była statyczną stroną na której zawsze wyświetla się to samo. Chcielibyśmy dodać post na naszego bloga czy dodać użytkowników do strony – wszystko to zapiszemy w naszej bazie danych.
- PHP – to po prostu moduł do naszego serwera Apache w którym zaszyty jest interpreter PHP, dzięki niemu skrypty napisane przez nas dają widoczny efekt.

Celowo poradnik jest pisany pod Windowsa, ze względu na to, że moje próby obsługi LAMP-a zawsze kończyły się zmarnowanym czasem na naprawianie błędów z którymi nigdy nie spotykałem się na Windowsie.

## 2. Instalacja Wamp-a

Najprostszym sposobem jest oczywiście wejście na stronę <http://www.wampserver.com/en/> i pobranie odpowiadającej nam wersji (x64 lub x86). Sama instalacja przebiega standardowo, po wyklikaniu wszystkich „nextów” powinniśmy móc uruchomić nasz serwer. W zasobniku systemowym powinna pojawić nam się ikonka . Jeżeli będzie ona pomarańczowa () to prawdopodobnie mamy włączonego Skypa który korzysta z tych samych portów co WAMP. Wystarczy wyłączyć Skypa, zrestartować usługi WAMP'a i gotowe.

## 3. Dodanie pierwszej strony

Pracować będziemy na Aliasach – za ich pomocą mapujemy ścieżkę dostępu z naszego systemu plików jako nazwę naszej strony. Po wpisaniu w przeglądarkę `http://127.0.0.1/mojastrona`, zaczniemy wyświetlać pliki z katalogu `D:\www\mojastrona`. Aby to zrobić wystarczy wykonać kilka prostych kroków:

- Tworzymy folder `D:\www\test\`
- Klikamy LPM na ikonkę WAMP'a
- Wyszukujemy opcję `Apache>Alias directories>Add dan alias`
- Wpisujemy nazwę naszego aliasu np. „test” po czym wciskamy ENTER
- Następnie podajemy ścieżkę dostępu do naszego katalogu w którym będziemy trzymać nasze pliki PHP. Należy pamiętać o użyciu notacji linuxowej tzn. ukośniki skierowane w prawo / np. „`D:/www/test/`” - bardzo ważnym elementem jest również ukośnik na końcu ścieżki dostępu, dzięki czemu serwer będzie wiedział, że chodzi nam o katalog.

W ten sposób stworzyliśmy nasz alias. Teraz możemy wejść do naszego katalogu D:/www/test/, stworzyć tam plik index.php a jako zawartość możemy wpisać:

```
<?php  
echo 'hello World';
```

Otwieramy naszą przeglądarkę i wpisujemy <http://localhost/test/>. Naszym oczom powinien ukazać się piękny napis „hello World”.

## 4. Jak działa serwer WWW

Na ten temat można napisać sporych rozmiarów wypracowanie, natomiast ja skupię się na kilku najważniejszych elementach. Po wpisaniu interesującego nas adresu, serwer wyszukuje pliku „index” – czyli głównego pliku strony. Następnie na podstawie jego zawartości tworzy kod HTML który zostanie wysłany do użytkownika. W momencie w którym trafimy na znacznik początku kodu PHP (<?php), nasz interpreter analizuje skrypt i wykonuje wszystkie zadeklarowane w nim funkcje. Ot cała filozofia. Wszystkie skrypty PHP wykonywane są po stronie serwera. Użytkownik nie może podejrzec jego zawartości, a jedynie wyświetlić kod jaki zwróci mu serwer. Tak wygląda fragment pliku wp-config.php – pliku konfiguracyjnego WordPressa:

```
define('DB_NAME', 'wp');  
define('DB_USER', 'root');  
define('DB_PASSWORD', '');  
define('DB_HOST', 'localhost');
```

Widzimy tutaj dane dostępu do bazy danych. Gdyby użytkownik mógł wyświetlić zawartość pliku \*.php, mógłby nam zrobić w niej niemałe zamieszanie.

## 5. Podsumowanie

Po przeczytaniu tego poradnika powinieneś potrafić stworzyć bazę do swojego projektu PHP, czyli działający serwer za pomocą którego będziesz wykonywał swoje skrypty. W kolejnej części omówię składnię PHP, jednak wymagana będzie minimalna wiedza na temat programowania – co to są zmienne, funkcje itp.