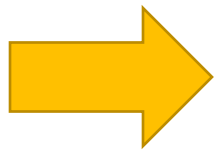


Czy pamiętasz?

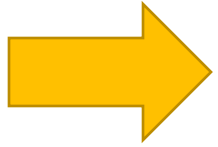
Legenda:



O czym będzie....

O czym było....





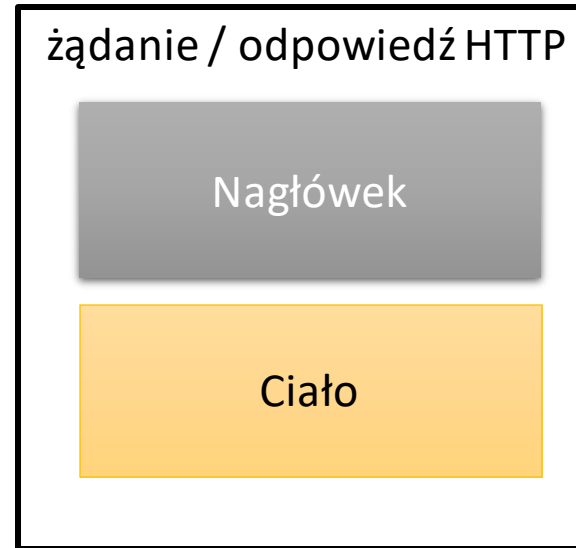
02 Protokoły

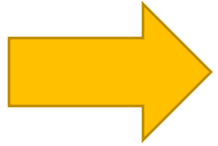
- Jak identyfikowany jest zasób w morzu Internetu?
- Jak trafić na serwer zawierający dokument który chcę odczytać?
 - Co to jest nazwa domenowa?
 - Skąd wiadomo na którym serwerze (o jakim IP) jest dostępna dana domena?
- Jakim protokołem komunikacyjnym zapytać o pożądaną zasób w sieci?
 - Czy jak drugi raz zapytam to mnie pamięta?
 - Czy pod jednym IP może być wiele domen?
- Kto dobiera moje pytania, szuka zasobu i odsyła odpowiedź?
- Jak skrócić drogę do często żądanego zasobu?



02 Protokoły

- URL, URN, URI
- DNS (Domain Name System)
- HTTP
 - Komenda, kod odpowiedzi, status;
 - GET, POST i inne
 - Protokół bezstanowy
 - Wirtualne hosty
- Serwer WWW
- Serwer proxy





StrKlienta_01 Przeglądarka

- Jakie mamy rodzaje zasobów sieciowych?
- HTML (ale jaki?)
 - Czy jest gorszy i lepszy HTML ?
 - Czy HTML jaki umieszczam w sieci musi być poprawny?
- Żądam stronę HTML,
 - a kto prosi o resztę (CSS, fonty, JavaScript, PNG, ...)?
 - Jak przeglądarka rozmawia z serwerem?
 - Czy przeglądarki mogą prosić o treść którą bardziej preferują?
- Jak zbudowana jest przeglądarka?
 - Jak działa komponent wyświetlający dokument HTML?
 - Co jeśli wystąpi błąd krytyczny w aktywnej zakładce? Czy 'wysypie się' cała aplikacja przeglądarki?
 - Co to jest cache /pamięć podręczna w przeglądarce?



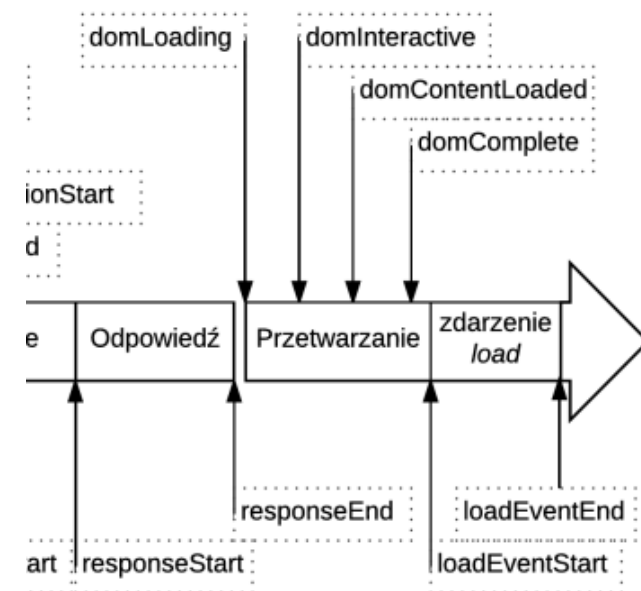
StrKlienta_01 Przeglądarka

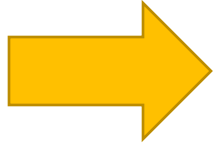
- WebPage, WebSite, WebApplication
- HTML
 - Doctype,
 - walidatory
- Media Type + w jakich nagłówkach żądania/odpowiedzi występuje (obserwujemy ruch sieciowy w narzędziach deweloperskich)
- Negocjowanie treści



StrKlienta_01 Przeglądarka

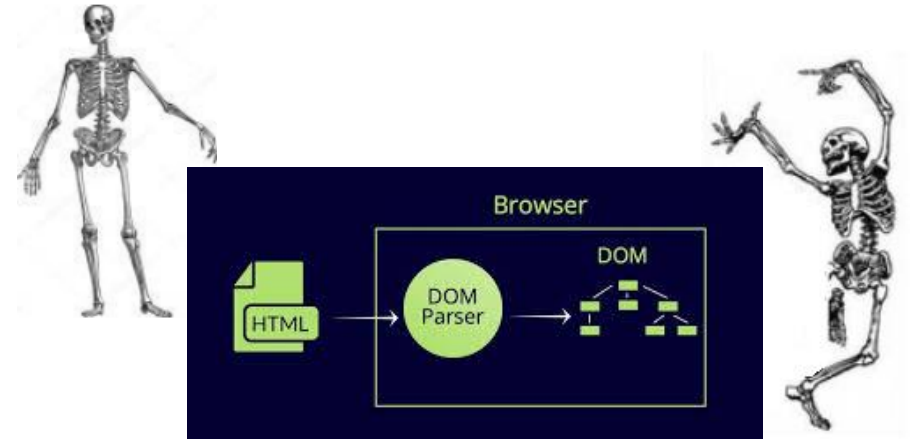
- Działanie przeglądarki:
 - Pobierany Html na progu przeglądarki (optymalizacja, naprawa)
 - Dociągane zasoby w żądaniach przewidujących
 - HTML parsowany,
 - Tworzona reprezentacja obiektowa (drzewo DOM) wczytywanego dokumentu
 - Skrypty osadzone w kodzie HTML wykonywane po napotkaniu `<script>` w trakcie interpretacji modyfikują drzewo DOM
 - Zakończone parsowanie HTML -> gotowe drzewo DOM = zdarzenie DOMContentLoaded
 - CSS pobierane i składane (kaskadowanie, dziedziczenie)
 - Nałożenie na elementy drzewa DOM wyznaczonych stylów
 - Drzewo obiektów wyświetlanych (render tree)
 - Określenie pozycji i wielkości elementów drzewa obiektów wyświetlanych (Layout)
 - Wyrysowanie w oknie przeglądarki (Painting)
 - Wyświetlenie obiektów osadzonych np. plików graficznych
 - Zdarzenie Load zakończone





03 Java Script

- Czy HTML może być dynamiczny?
 - Kto powoduje tą zmianę?
 - Gdzie znajduje się ten nowy zmieniony HTML?
 - Co w nim może się zmieniać?
- Co to jest asynchroniczna komunikacja z serwerem
- Co to jest ECMAScript i po co go wymyślono?
- JavaScript
 - Do czego typowo używamy JS?
 - Gdzie umieścić kod, kiedy się on wykona?
 - Czy działanie JS może być blokowane?





AJAX

03 Java Script

1. XMLHttpRequest + JS -> wysłanie
żądania asynchronicznego

- xhr
- fetch

5. JS + DOM modyfikuje
stronę umieszczając odpowiedź

2. odebranie i przetworzenie żądania;
3. odnalezienie lub wygenerowanie
odpowiedzi;

HTML, text, XML, XHTML, JSON

4. odesłanie odpowiedzi

<https://moja.pg.edu.pl/app/addressBook/>



Aplikacja osadzona we FrontEnd wysła
Ajax'em żądanie o dane, a BackEnd odsyła je
w formacie JSON

Czy można żądaniem asynchronicznym pobrać plik js?

WP aps_csm.js

03 Java Script



- JavaScript Object Notation (JSON)
 - Ależ po co jeszcze jedno cudo?
- Document Object Model
 - co to?
 - gdzie to jest?
 - co można z tym robić?
 - Jak zmienić element już istniejący w pobranym pliku HTML
 - Obrazek po najechaniu na niego myszą zmieniamy na inny obrazek
 - Tekst widoczny w div/p zmieniamy na inny tekst, zmieniamy jego formatowanie
 - Jak sprawdzić poprawność formularza przed jego wysłaniem
 - Jak nie wysyłać formularza gdy pola nie są wypełnione kompletnie
 - Czy zawsze zadziała sprawdzanie formularza przed wysłaniem?



Przykład AJAX_JSON.html

JSON w JS

AJAX żądający podanie aut

```
var message= req.responseText;
```

JSON w PHP

- `json_encode()` - wytworzy string z obiektu

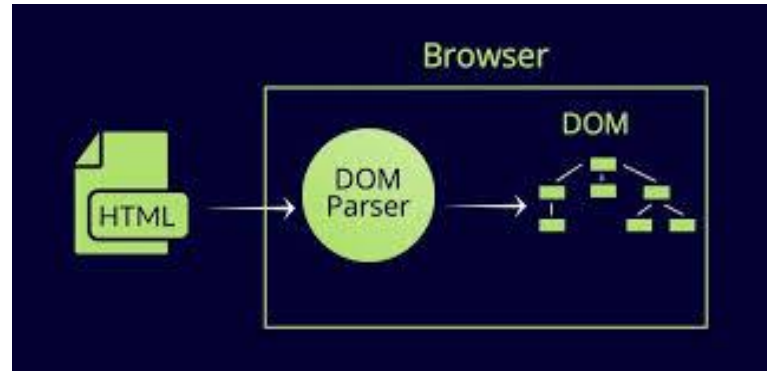
```
<?php
$auto = array("fiat"=>35, "mercedes"=>37, "VW"=>3);
echo json_encode($auto);
?>
```

```
{"fiat":35,"mercedes":37,"VW":3}
```

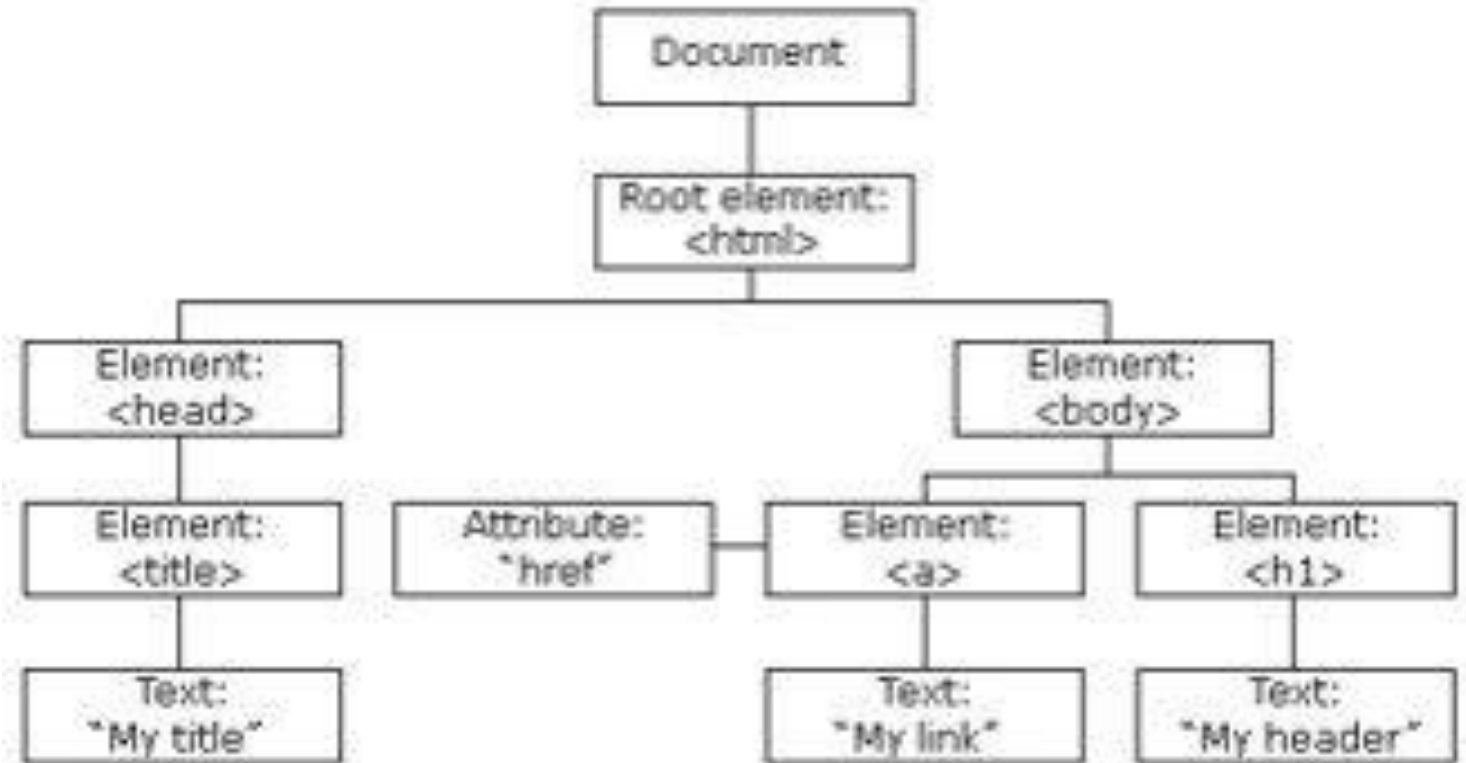
- `json_decode()` - przekonwertuje stringa kodowanego UTF-8 do zmiennej php

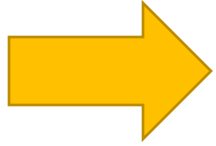
```
var samochody= JSON.parse(message);
var tekst = "Mercedesy w ilości: "+ samochody.mercedes +", a VW w ilości: "+ samochody.VW;
```

```
document.getElementById("userIdMessage").innerText = tekst;
```



```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <a href="" >My link</a>
    <h1>My header</h>
  </body>
</html>
```



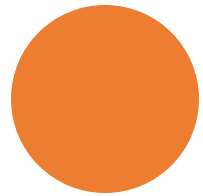


03 Java Script / 04 jQuery

- Storage przeglądarki kontra ciasteczka
 - Jak jedno i drugie obsłużyć w JS?
 - Jak zapisać do storage coś bardziej złożonego niż ciąg znaków?
- \$
 - Jak robić aby się nie narobić i jeszcze zarobić kasę...
- JQuery UI
 - i przy tym jeszcze by wyglądało znośnie...



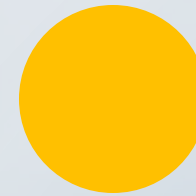
PHP



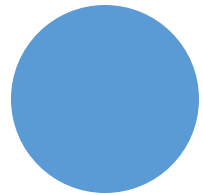
01 Składnia,
przetwarzanie



02 Obsługa żądań
HTTP



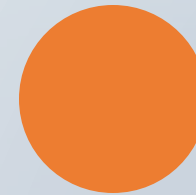
03 współpraca z bazą
MongoDB



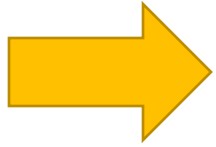
04 Mechanizm sesji



05 Kontrola dostępu



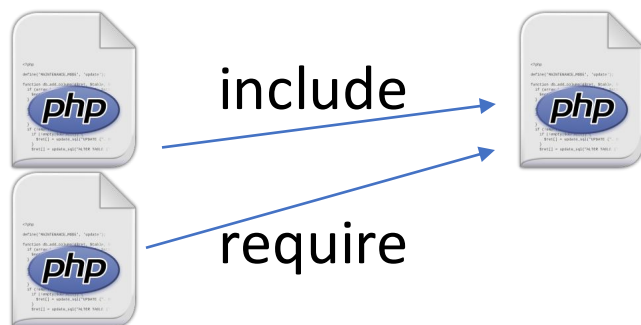
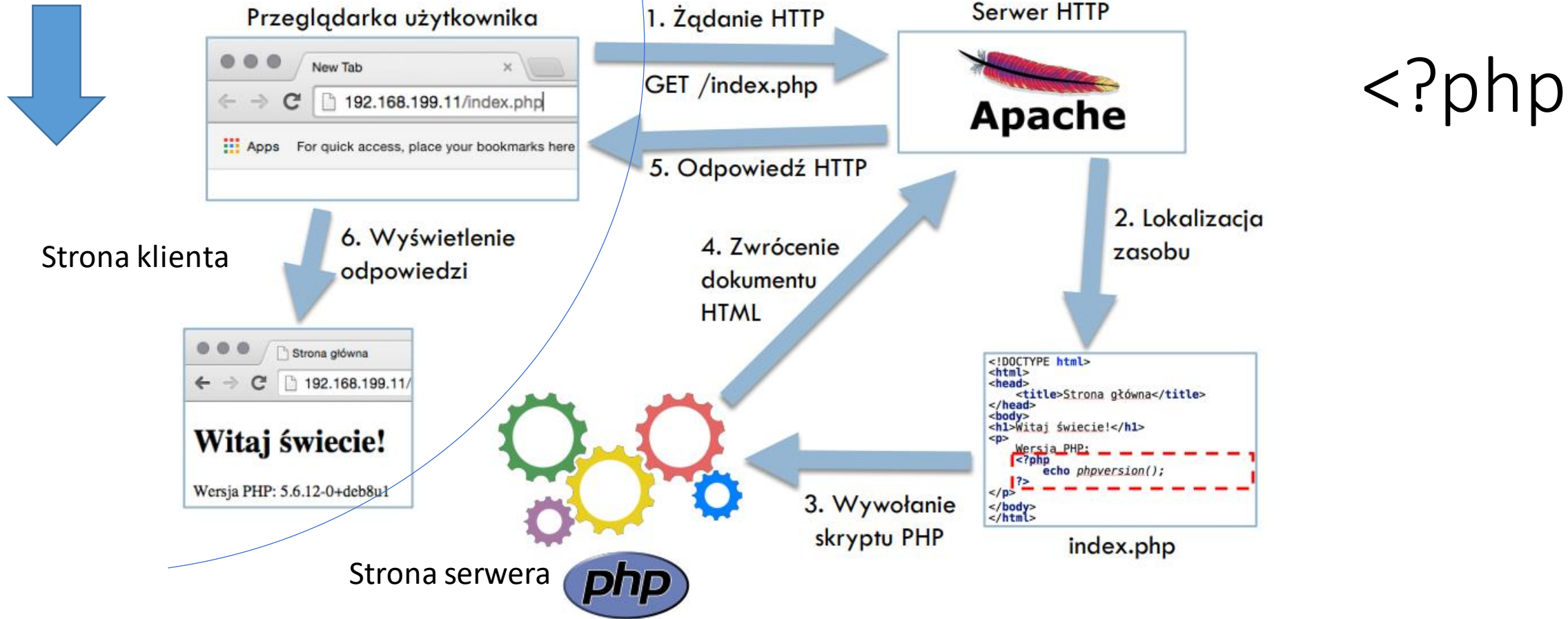
06 wzorce
projektowe (FC,
MVC)



01_PHP

- Żywy czy martwy? <https://kinsta.com/blog/is-php-dead/>
- Jak przetwarzane jest żądanie o zasób PHP
- Jak pisać w PHP (jaka jest jego składnia)
 - `<?php` czy domknięcie jest zawsze potrzebne?
- Ile tych kodów?
 - **x.php = plik źródłowy -> kod pośredni --> kod maszynowy**
- Czy mogę mieć kawałki kodu php lub html w osobnych plikach i je dołączać wedle potrzeb?
- Czy mogę dynamicznie zmieniać nazwy?
 - zmienne zmienne, zmienne funkcje

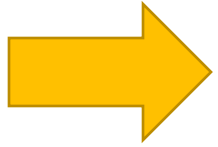




źródłowy plik php

-> **kod pośredni** (niezależny od platformy)

--> **kod maszynowy** (dla dedykowanej platformy)

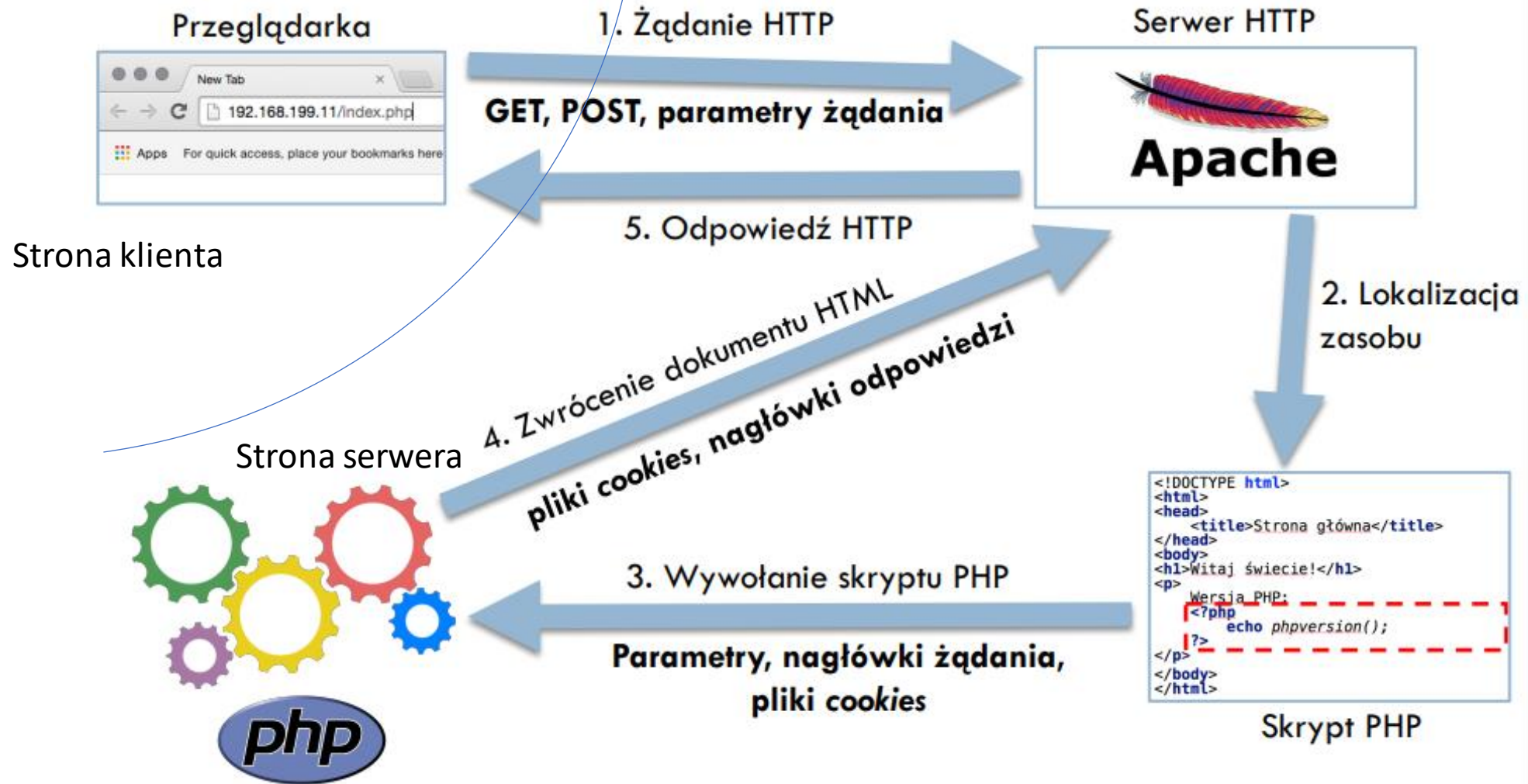


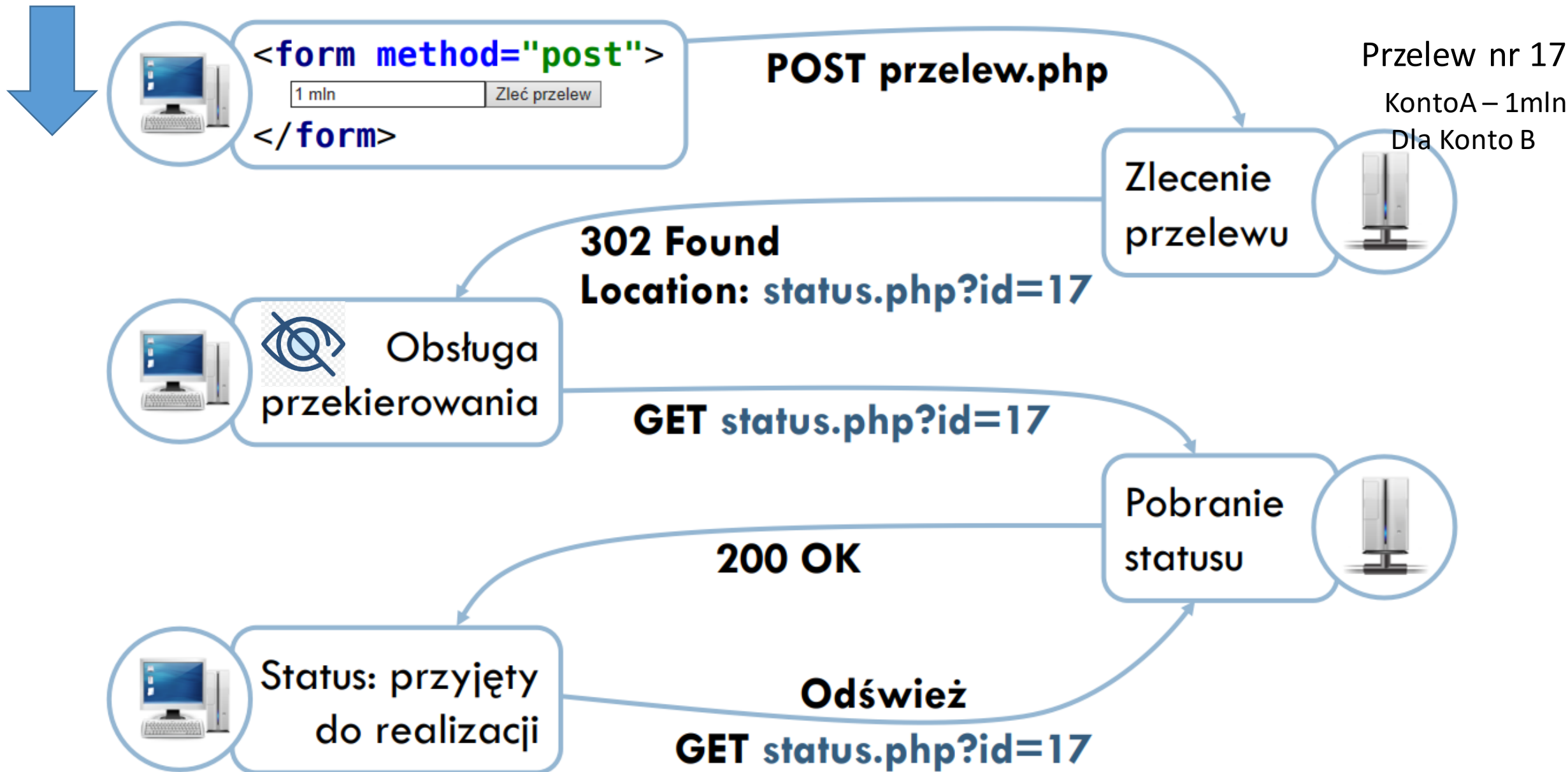
02_obsługa zapytań HTTP

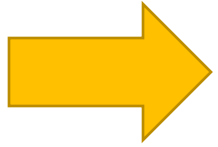
- Jak przekazać parametry dodatkowe?
- Tablice superglobalne
- Jak nie stracić miliona ?
 - POST-REDIRECT-GET
- Jak prawidłowo budować skrypt?(s47)
- Jak operować ciasteczkami?



02_obsługa zapytań HTTP

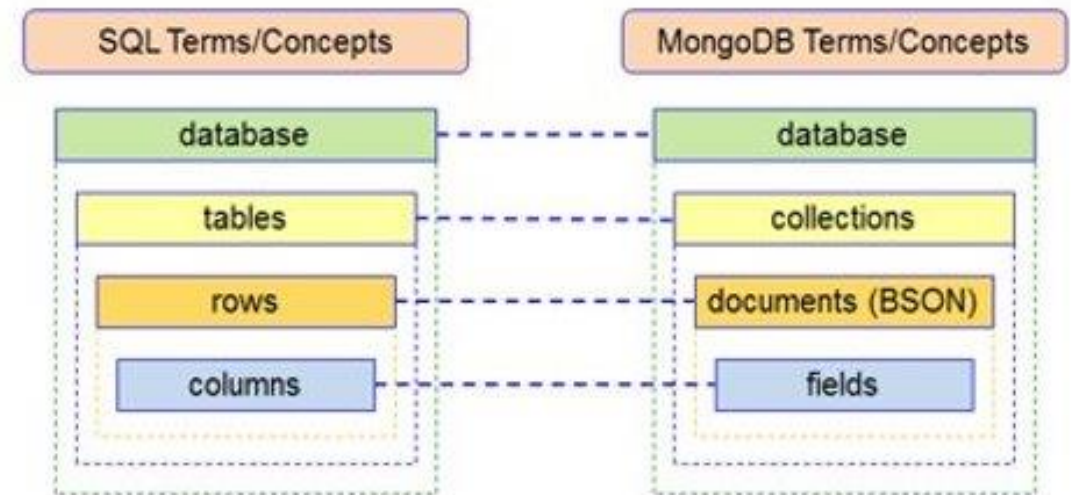






03_MongoDB

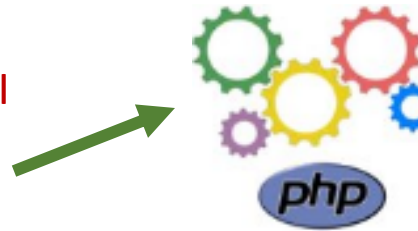
- Co to jest sterownik do bazy?
- Co robi w projekcie PHP composer?
- Co to znaczy nierelacyjna baza danych?
- Instancja, baza, kolekcja, dokument, pole
- Co to jest stronicowanie wyników?



03_MongoDB

```
$products = $db->products->find($query);
```

Biblioteka mongo-php-library oferuje wygodniejsze wysokopoziomowe API



Sterownik mongodb - niskopoziomowy interfejs do komunikacji z bazą danych

Różne biblioteki używane w projekcie PHP - np.:
-mongo-php library;
-PDO, simpleXML,...



Composer

Kod PHP wykorzystujący funkcje z bibliotek



sklep



finanse



CRM

bazy danych



użytkownicy



zamówienia

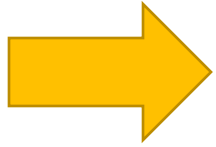


produkty

kolekcje



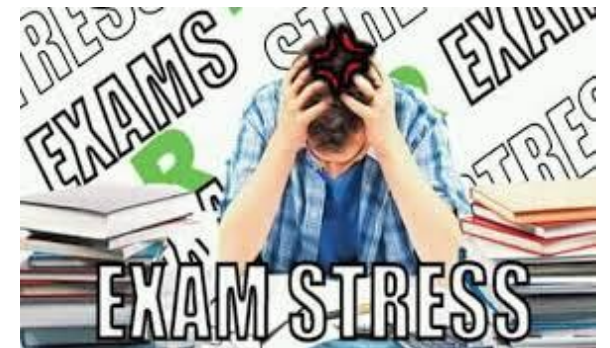
dokumenty



04_Sesja

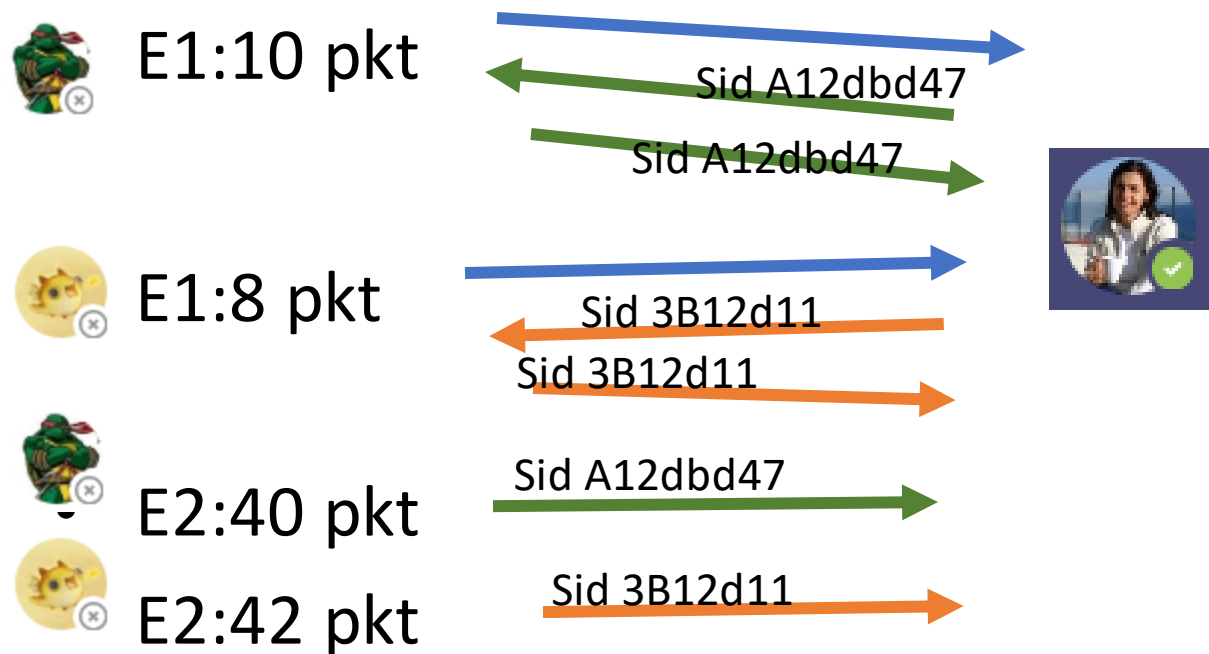
- Dlaczego było konieczne wymyślenie mechanizmu sesji?
- Sesja a stan sesji – czy to jest to samo?
- Jakie są sposoby na przekazanie identyfikatora sesji?
- Jak następuje uzgodnienie właściwego sposobu przekazywania identyfikatora sesji akceptowanego przez obie strony?

To idealny temat w sesji egzaminacyjnej!



04_Sesja

↓
HTTP jest bezstanowy, a jakoś trzeba rozróżnić klientów...



W trakcie powyższej interakcji obserwujemy dwie SESJE



Przekazywanie identyfikatora sesji <->

- ciasteczko
- parametry żądań
 - w URL (GET)
 - ukryte pole formularza (POST)

Sid A12dbd47

E1:10 pkt
E2:40 pkt



Sid 3B12d11

E1:8 pkt
E2:42 pkt



Dwa niezależne STANY SESJI zachowano w:

- pamięci
- plikach
- bazie danych



Bieżąca konfiguracja serwera

Strona klienta – **Cookie obsługiwane**

- 1) wysłanie żądania bez identyfikatora
- 4) odebranie odpowiedzi i zapisanie ciasteczka
- 5) cookie:sid wysłany z każdym następnym żądaniem

Cookie:

PHPSESSID=btneu5v7oljj2u0o8iki8n9g967

- 7) odebranie odp (bez cookie!)
- ... komunikacja jak w krokach 5,6,7

Strona serwera

- 2) session_start();
- wygenerowanie identyfikatora,
 - utworzenie stanu sesji,
 - ustawienie set-cookie: sid w nagłówkach odp.

Set-Cookie:

PHPSESSID=btneu5v7oljj2u0o8iki8n9g967; path=/
Expires:Session – Ciasteczko ważne do zamknięcia

przeglądarki

- 3) wykonanie kodu PHP i odesłanie odpowiedzi z nagłówkiem set-cookie
- 6) session_start() -> odczytanie sid z nagłówka cookie w żądaniu; odnalezienie stanu sesji o tym identyfikatorze (na czas obsługi żądania stan sesji dostępny w tablicy \$_SESSION); wykonanie kodu PHP z ew. modyfikacją stanu sesji; odesłanie odpowiedzi (teraz już cookie nie jest dołączany, bo przyszedł w parametrach żądania, czyli zachowano go w przeglądarce)

```
session.use_cookies=1  
session.use_trans_sid=0  
session.use_only_cookies=1
```



Bieżąca konfiguracja serwera

Strona klienta – **Cookie NIE obsługiwane**

- 1) wysłanie żądania bez identyfikatora
- 4) odebranie odpowiedzi

Set-Cookie: PHPSESSID=ja85ofnb30ld9g3bv4i74c2vi0; path=/ ⚠

i problem z zapisaniem ciasteczka

- 5) cookie:sid NIE jest wysłany z następnym żądaniem (czyli jak w pkt.1)
- 7) odebranie odpowiedzi (z nowym sid!)

Set-Cookie: PHPSESSID=5ni0iv3aq7pp97h0b4ic1ufri6; path=/ ⚠

i problem z zapisaniem ciasteczka

Więc z kolejnym żądaniem znów nie jest odsyłany sid (czyli jak w pkt.1)

Strona serwera

```
session.use_cookies=1
session.use_trans_sid=0
session.use_only_cookies=1
```

- 2) session_start();
- wygenerowanie identyfikatora,
 - utworzenie stanu sesji,
 - ustawienie set-cookie: sid w nagł.odp.

Set-Cookie:

PHPSESSID=btneu5v7oljj2u0o8iki8n9g967;
path=/

Expires:Session – Ciasteczko ważne do zamknięcia przeglądarki

- 3) wykonanie kodu PHP i odesłanie odpowiedzi z nagłówkiem set-cookie
- 6) odebranie żądania bez sid (jak w pkt.2); wygenerowanie nowego sid i stanu sesji; ustawienie nowego set-cookie

Czyli przy tej konfiguracji **nigdy nie ustanowimy sesji!**



Zmieniona konfiguracja serwera



```
session.use_cookies=1  
session.use_trans_sid=1  
session.use_only_cookies=0
```

Strona klienta – **Cookie obsługiwane**

1) wysłanie żądania bez identyfikatora

4) odebranie odpowiedzi (w ciele do elementów html dodany sid)

```
<form method="post" class="inline">  
  <input type="hidden" name="PHPSESSID" value="v7tmholrnn9u4hk2gek7ttg3f0" />  
  <input type="submit" value="Usuń wszystkie" name="clear_cart"/>  
</form>  
<a href="view.php?id=59b134416e9552029727ddd2&PHPSESSID=v7tmholrnn9u4hk2gek7ttg3f0">Laptop X199</a>  
</td>
```

i zapisanie ciasteczka z sid

Set-Cookie: PHPSESSID=v7tmholrnn9u4hk2gek7ttg3f0; path=/

5) kolejne żądanie wysyła sid w **cookie** oraz w parametrach żądania (w url dla GET, ukryte pole formularza dla POST)

7)... dalej następuje komunikacja jak przy używaniu tylko ciasteczka

Strona serwera

2) session_start();

- wygenerowanie identyfikatora,
- utworzenie stanu sesji

3) wykonanie kodu PHP i odesłanie odpowiedzi html z dołączonym sid oraz z nagłówkiem set-cookie

6) session_start() i odnalezienie stanu sesji o danym sid; **sid dostępny w żądaniu (GET/POST) i w nagłówku cookie;**

Generując odpowiedź nie jest ustawiane ciasteczko (skoro **przyszło w nagłówku cookie** tzn. że można było je zapisać w przeglądarce, a to oznacza że **wymiana przez parametry żądania nie jest konieczna!**) odsyłany dokument html nie zawiera już dołączeń sid.

Zaczęto od ciastka i parametrów żądania, a wynegocjowano używanie **tylko cookie** do obsługi sesji!



Zmieniona konfiguracja serwera



```
session.use_cookies=1
session.use_trans_sid=1
session.use_only_cookies=0
```

Strona klienta – Cookie NIE obsługiwane

1) wysłanie żądania bez identyfikatora

4) odebranie odpowiedzi (w ciele do elementów html dodany sid)

```
<form method="post" class="inline">
  <input type="hidden" name="PHPSESSID" value="mf39pff96cft5kc9gdi0cslem0" />
  <input type="submit" value="Usuń wszystkie" name="clear_cart"/>
</form>
```

```
<a href="view.php?id=59b134416e9552029727ddd2&PHPSESSID=mf39pff96cft5kc9gdi0cslem0">Laptop X199</a>
```

i odebrane set-cookie ale problem z zapisaniem ciasteczka z sid

Set-Cookie: PHPSESSID=mf39pff96cft5kc9gdi0cslem0; path=/ ⚠

5) kolejne żądanie wysyła sid tylko w parametrach żądania (w url dla GET, ukryte pole formularza dla POST), bo ciasteczka są wyłączone

Request URL: http://192.168.56.10:8080/03-session/src/web/view.php?id=59b134416e9552029727ddd2&PHPSESSID=mf39pff96cft5kc9gdi0cslem0

7) odebrano dokument z dołączonym sid w kodzie html

```
<div>
  <a href="index.php?PHPSESSID=mf39pff96cft5kc9gdi0cslem0" class="cancel">&laquo; Wróć</a>
  <input type="submit" name="add_to_cart" value="Do koszyka"/>
</div>
```

... dalej następuje komunikacja tylko przez parametry żądania....

Strona serwera

2) session_start();

- wygenerowanie identyfikatora,

- utworzenie stanu sesji

3) wykonanie kodu PHP i odesłanie odpowiedzi html z dołączonym sid oraz z nagłówkiem set-cookie

6) session_start() i odnalezienie stanu seji o danym sid;

sid dostępny w żądaniu (GET/POST)

Generując odpowiedź nie jest ustawiane ciasteczko. Skoro sid nie przyszedł w cookie [bo cookie nie jest wspierane w przeglądarce], a **przyszedł sid w parametrach**, to **konieczne jest odesłanie sid też w parametrach** (odsłany html zawiera dołączenia sid)

Zaczęto od ciastka i parametrów żądania, a wynegocjowano używanie
tylko sid w parametrach żądań do obsługi sesji!

Jak przebiega obsługa sesji przy poniższych konfiguracjach?

Strona klienta – Cookie obsługiwane

?

Strona serwera

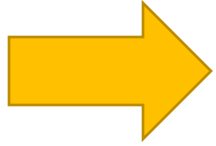
```
session.use_cookies=1  
session.use_trans_sid=1  
session.use_only_cookies=1  
|
```

Strona klienta – Cookie NIE obsługiwane

?

[Pomoc](#) na początek

?



05_Kontrola dostępu

- Na czym polega uwierzytelnianie?
- A na czym autoryzacja?
- Co to jest funkcja mieszająca?
- Kiedy występuje kolizja w wynikach funkcji mieszających?
- Jak zarejestrować nowego użytkownika?
- Jak się poprawnie zalogować?
 - Czemu mam identyfikator sesji zanim wypełnię formularz logowania?
 - Jak nadać nowy identyfikator po zalogowaniu?
- Jak się poprawnie wylogować?
- Komu można ufać?



05_Kontrola dostępu

- Weryfikacja – sprawdzenie tożsamości
- Autoryzacja – nadanie uprawnień
- Funkcja mieszająca /hashująca generuje skrót hasła
- Kolizja - funkcja mieszająca generuje ten sam hash dla różnych haseł;
- Rejestracja slajd 11-12
- Logowanie slajd 13-14
 - w każdym pliku na początku jest `session_start()` nawet jeśli nie było formularza logowania uprzednio wypełnionego;
 - `session_regenerate_id()`;
- `session_destroy()` + usunięcie ciasteczek
- Nie ufaj nikomu:
 - Sanityzacja – usuwanie niebezpiecznych znaków z inputów
 - Walidacja – trzymana dana spełnia kryteria spodziewanego wejścia np. oczekuję stringu lub cyfry
 - Escape output – jak bezpiecznie wyświetlić dane



Sesja z niezalogowanym i zalogowanym użytkownikiem

Strona klienta – Cookie obsługiwane

- 1) wysłanie żądania bez identyfikatora
- 4) odebranie odp i sid oraz zapisanie ciasteczka z sid
- 5) cookie:sid wysłany z każdym następnym żądaniem

Cookie:

PHPSESSID=btneu5v7oljj2u0o8iki8n9g967

- 7) odebranie odp (bez cookie!)
- ... komunikacja w krokach 5,6,7
- 8) wysłanie danych logowania (zawiera sid)
- 10) odebranie nowego sid i zapisanie w ciasteczku; wymiana żądań z nowym sid (jak w 5,6,7)
- 11) wysłanie żądania wylogowania (zawiera sid)
- 13) odczytanie cookie i usunięcie ciasteczka
- ... ewentualnie następne żądanie od kroku 1.

- Jeżeli po pkt 10 klient był bezczynny ponad limit sekund ustalony w `session.gc_maxlifetime`, to stan sesji jest niszczonej po stronie serwera przez garbage collector i choć cookie jest wciąż w przeglądarce i może być ponownie wysłane, to stan sesji dla tego sid nie będzie odnaleziony.
- Jeżeli po pkt 10 klient zamknął przeglądarkę, to po ponownym uruchomieniu przeglądarki cookie nie będzie dostępne; A stan sesji, o sid zgodnym z sid ze zniszczonego ciastka, zostanie po czasie zniszczony przez garbage collector.

Strona serwera

- 2) `session_start();`
- wygenerowanie identyfikatora,
- utworzenie stanu sesji,
- ustawienie set-cookie: sid w nagł.odp.

Set-Cookie:

PHPSESSID=btneu5v7oljj2u0o8iki8n9g967; path=/

Expires:Session – Ciasteczko ważne do zamknięcia przeglądarki

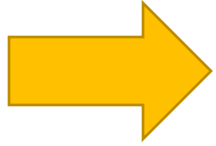
- 3) wykonanie kodu PHP i odesłanie odpowiedzi z nagłówkiem set-cookie

6) `session_start()` -> odczytanie sid z nagłówka cookie w żądaniu; odnalezienie stanu sesji o tym identyfikatorze (na czas obsługi żądania stan sesji dostępny w tablicy `$_SESSION`); wykonanie kodu PHP z ew. modyfikacją stanu sesji; odesłanie odpowiedzi (teraz już cookie nie jest dołączany, bo przyszedł w parametrach żądania, czyli zachowano go w przeglądarce)

9) `session_start()`; pobranie stanu sesji dla otrzymanego sid; weryfikacja danych logowania, regeneracja sid (zamiana bieżącego sid nowym ciągiem -dane stanu sesji pozostają); ustawienie sid w nagł set-cookie

- 12) `session_destroy();` //stan sesji
- `session_unset ();` //tablicę `$_SESSION` w bieżącej obsłudze żądania
- Setcookie (z czasem ujemnym) // set-cookie sid
- Odesłanie odpowiedzi z nagłówkiem set-cookie 'ujemnym czasem'

```
session.use_cookies=1
session.use_trans_sid=0
session.use_only_cookies=1
```



06_Przepływ sterowania

- Jak dojść od chaosu do właściwej struktury projektu?
 - Logika interfejsu, logika obsługi żądań/odpowiedzi i logika biznesowa – i po co dzielić 'włos' na troje?
- Po co tworzyć funkcje biznesowe?
 - Co gdy nastąpi zmiana typu bazy np. z mongoDB na mySQL?
- Co to są wzorce projektowe?
 - Model – Widok - Kontroler;
 - Front controller – czyli po co nam w projekcie portier?
 - Routing – mapowanie adresu na kontroler, czyli jak ukryć wywoływaną logikę?
 - mod_rewrite – co to jest moduł Apache?
 - Czy plik htaccess może być w projekcie tylko jeden?
- Co to są moduły w projekcie?



06_Przepływ sterowania

- Struktura projektu -używać sprawdzonych szablonów projektowych
 - lub rób porządek w chaosie :)
- Funkcje biznesowe wydzielamy by odseparować kod operujący na określonym dostawcy usług bazodanowych od reszty projektu
 - Podmieniamy tylko operacje na bazie w jednym pliku a nie w całym kodzie;
- Wzorce projektowe - gotowe sprawdzone sposoby na rozwiązywanie problemów projektowych;
 - mod_rewrite -[moduły Apache](#) to komponenty programowe, które mogą być dynamicznie dołączane i ładowane w celu rozszerzenia funkcjonalności serwera www.
 - Wiele- np. Bronić dostępu do innych katalogów
- Moduły projektowe wydzielane wg warstw aplikacji lub funkcjonalności - logiczne duże części projektu odpowiedzialne za część funkcjonalności
 - Sklep (produkty, sprzedaż); Statystyki; Monitorowanie stanów magazynowych;

Gratuluję, poznałeś podstawowe zagadnienia

Teraz uprość sobie pracę stosując [frameworki PHP](#)
wykorzystujące właściwe szablony projektów,
uwzględniające wzorce projektowe,
wielojęzyczność witryn,
pracę w grupie,

....



Dziękuję za wspólną przygodę!



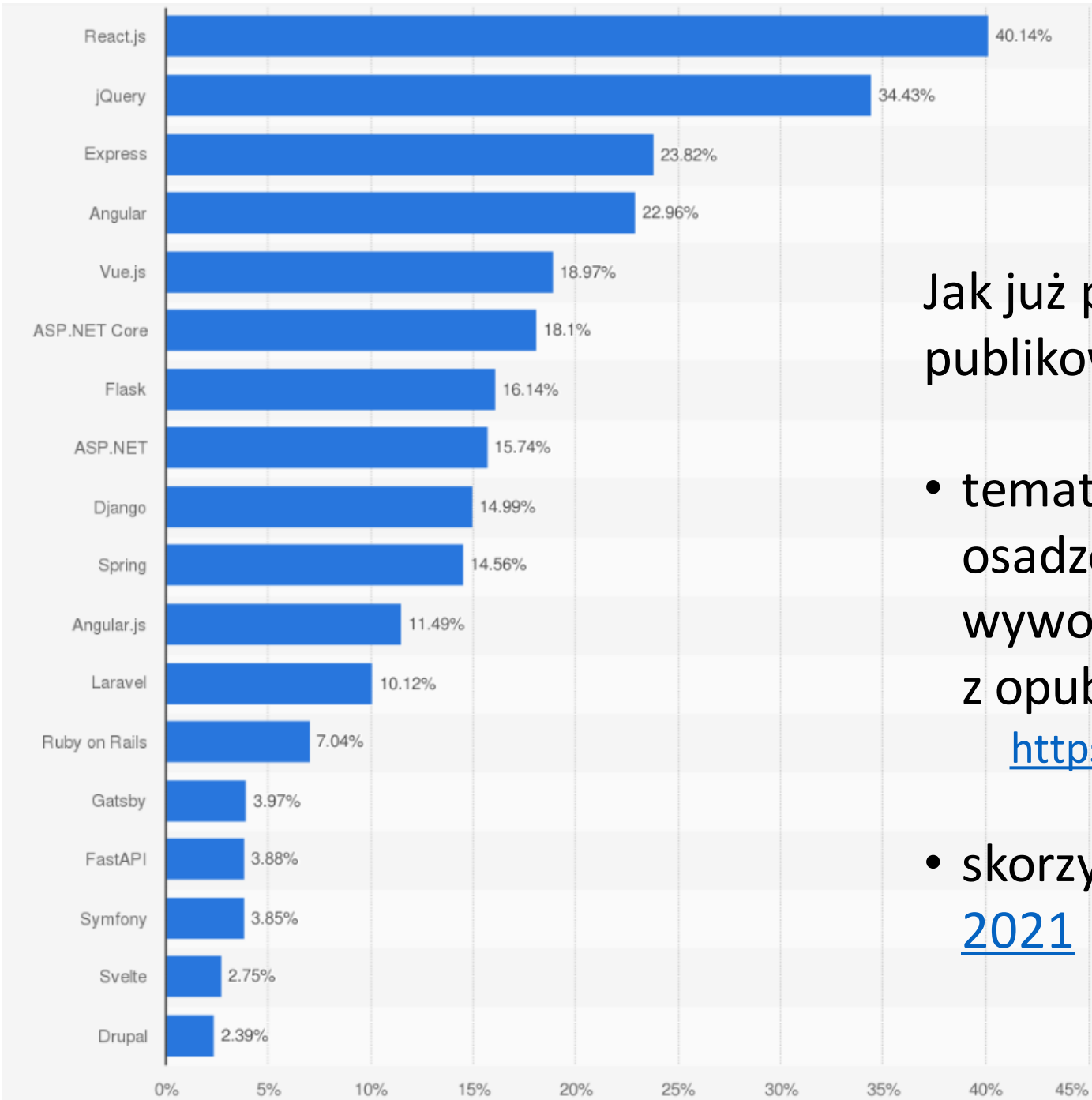
...i co dalej?

Jak już poznasz usługi sieciowe, sposoby ich publikowania oraz żądania typu REST zajmij się:

- tematem MVC po stronie klienta w aplikacji osadzonej w HTML i wywołującej, za pomocą żądań REST, metody z opublikowanych usług backendu

<https://moja.pg.edu.pl/app/addressBook/>

- skorzystaj z popularnych [web frameworks 2021](#)



Życie informatyka to podróż wciąż przez nowe technologie i pytania, więc szerokiej drogi!

Jak wybrać najlepszy stos technologiczny dla aplikacji internetowej?

- Front-End Technology Stack
- Back-End Tech Stack
- Web Tech Stacks of Successful Projects

Czy jest tu miejsce dla mnie?

