PHP
i komunikacja klient-serwer

PAWEŁ RAJBA

Spis treści

Wprowadzenie

Komunikacja pomiędzy formularzem a skryptem PHP, czyli klientem z serwerem

Metody wysyłania danych z formularza

- Metoda GET
- Metoda POST

Tablice zawierające przesłane dane:

- $_GET
- $_POST

Tworzenie formularza

Z istotniejszych atrybutów formularza są

- name – nazwa
- action – adres wysłania danych
- enctype – sposób kodowanie danych z formularza
  - application/x-www-form-urlencoded
  - multipart/form-data
  - text/plain
- method – sposób wysłania: get lub post

Kontrolki formularza: input, select, textarea

Grupowanie za pomocą fieldset i legend

Obsługa danych z formularza

Pola tekstowe

- Po stronie formularza
  - <input type="text" name="nazwisko" size="11"
    maxlength="11" value="">
    - Wprowadzamy tekst ,,Kowalski''
- Po stronie skryptu PHP
  - echo $_GET["nazwisko"]; // pojawi się napis ,,Kowalski''
- Uwaga:
  - w formularzu można zastosować konstrukcję:
    <input type="text" name="adres[]" (...)>
    wtedy po stronie skryptu $_GET["adres"] będzie tablicą

Obsługa danych z formularza

Pola typu radio

- Po stronie formularza
  - <input name="plec" type="radio" value="m" checked>
    <input name="plec" type="radio" value="k">
    - Zaznaczamy ,,m''
- Po stronie skryptu PHP
  - echo $_GET["plec"]; // pojawi się napis ,,m''
- Uwaga: jeśli nie będzie nic zaznaczone, zmienna po stronie serwera nie będzie ustawiona

Obsługa danych z formularza

Pola typu checkbox

- Po stronie formularza
  - <input type="checkbox" name="jezyk[]" value="en">
    <input type="checkbox" name="jezyk[]" value="fr">
    - Zaznaczamy oba
- Po stronie skryptu PHP
  - echo print_r($_GET["jezyk"]);
    // dostaniemy tablicę o dwóch wartościach: en i fr
- Uwaga: jeśli nie będzie nic zaznaczone, zmienna po stronie serwera nie będzie ustawiona

Obsługa danych z formularza

Pola pojedynczego wyboru

- Po stronie formularza
  - <select name="miasto">
    <option value="warszawa">Warszawa</option>
    <option value="niebieski">Wrocław</option></select>
    - Zaznaczamy Wrocław
- Po stronie skryptu PHP
  - echo $_GET["miasto"] // Pojawi się ,,Wrocław''
- Uwaga: jeśli nie będzie nic zaznaczone, zmienna po stronie serwera nie będzie ustawiona

Obsługa danych z formularza

Pola wielokrotnego wyboru

- Po stronie formularza
  - <select name="kolory[]" size="4" multiple>
    <option value="green">Zielony</option>
    <option value="blue">Niebieski</option>
    <option value="red">Czerwony</option></select>
    - Wybieramy wszystkie trzy
- Po stronie skryptu PHP
  - echo print_r($_GET["kolory"]);
    // dostaniemy tablicę o wartościach: green, blue, red
- Uwaga: jeśli nie będzie nic zaznaczone, zmienna po stronie serwera nie będzie ustawiona

Obsługa danych z formularza

Pole tekstowe

- Po stronie formularza
  - <textarea cols="20" rows="10" name="opis">X</textarea>
- Po stronie skryptu PHP
  - echo $_GET["opis"] // Pojawi się wprowadzony tekst

Pola ukryte

- Po stronie formularza
  - <input type="hidden" name="zmienna" value="wartosc">
- Po stronie skryptu PHP
  - echo $_GET["zmienna"]; // Pojawi się tekst ,,wartosc''

Obsługa danych z formularza

Przykład

- dane.php

Weryfikacja wysłanych danych

Weryfikacja

- Po stronie klienta (opcjonalna, poprawia wygodę)
- Po stronie serwera (obowiązkowa)

Sposoby przeprowadzenia weryfikacji

- Kontrola typów (is_int, is_array, is_string, ...)
- Wyrażenia regularne (ereg, eregi, ereg_replace,...)
- Zbiór dopuszczonych wartości

Które wysłane dane sprawdzać?

Jak wizualizować błędy w formularzu?

Wysyłanie plików

Formularz, z którego wysyłamy pliki

- Powinien mieć enctype="multipart/form-data"
- Może zawierać konstrukcję ustawiającą maksymalny rozmiar pliku (musi być przez input..file)
  - <input type="hidden" name="MAX_FILE_SIZE" value="30000">
  - Interpretowane po stronie przeglądarki i PHP
    - Po stronie przeglądarki nie zawsze jest poprawnie interpretowane
- Powinien zawierać co najmniej jeden element
  - <input type="file" name="plik" (...)>

Wysyłanie plików

Co możemy ustawić w pliku php.ini?

- file_uploads = On | Off
  - włącza/wyłącza możliwość uploadowania plików
- upload_tmp_dir = ścieżka
  - katalog, w którym będą składowane wysłane pliki
  - domyślnie nieustawione – rolę takiego katalogu pełni tymczasowy katalog w systemie (np. C:\Windows\Temp)
- upload_max_filesize = rozmiar (domyślnie 2MB)
  - maksymalny akceptowany rozmiar uploadowanego pliku
- post_max_size = rozmiar (domyślnie 8MB)
  - maksymalny akceptowany rozmiar formularza

Wysyłanie plików

Po wysłaniu pliku w skrypcie PHP dostępna jest zmienna $_FILES["plik"], która ma pola

- name – oryginalna nazwa pliku po stronie klienta
- type – typ zawartości wysłanego pliku, np. text/html
  - Jest to typ MIME
- size – rozmiar wysłanego pliku
- tmp_name – nazwa pliku po stronie serwera
- error – kod błędu

Wysyłanie plików

Kody błędów

- 0 – nie wystąpił błąd - plik wysłany poprawnie
- 1 – wielkość pliku przekracza wartość maksymalną określoną w dyrektywie upload_max_filesize
- 2 – wielkość pliku przekracza wartość maksymalną określoną w polu MAX_FILE_SIZE formularza
- 3 – plik został wysłany częściowo
- 4 – żaden plik nie został wysłany

Odpowiadające im stałe: 0 – UPLOAD_ERR_OK,
1 – UPLOAD_ERR_INI_SIZE, 2 – UPLOAD_ERR_FORM_SIZE,
3 – UPLOAD_ERR_PARTIAL, 4 – UPLOAD_ERR_NO_FILE

Wysyłanie plików

Funkcje do obsługi wysłanych do serwera plików

- bool is_uploaded_file(string nazwa_pliku)
  - upewnia nas, że plik faktycznie został przysłany metodą POST, a skrypt nie będzie pracował na plikach, na których nie powinien
- bool move_uploaded_file(string src, string dest)
  - po sprawdzeniu, że plik został przysłany metodą POST jest on przenoszony do nowej lokalizacji

Wysyłanie plików

Przykład

- upload.php

Cookies

Co to tak naprawdę są te ciacha?

Jak działa mechanizm ,,ciastek''?

- Kwestia buforowania
- Parametry ciastek
  - nazwa, wartość
  - data ważności – czas, po którym ciastko zostanie usunięte
  - ścieżka – ścieżka, z której ciastko będzie wysyłane
  - domena – domena, z której ciastko będzie wysyłane
  - bezpieczne – czy do przesłania ciacha wymagany jest SSL

Cookies

Tworzenie ciastka w PHP

- int setcookie(string nazwa, [, string wartość
  [, int data_ważności [, string ścieżka
  [, string domena [, int bezpieczne]]]]])

Usuwanie ciastka w PHP

- odbywa się przez podanie wstecznej daty ważności – przeglądarka uruchamia wtedy mechanizm usunięcia ciastka
- ciasteczko może być usunięte tylko z takimi parametrami, z jakimi zostało utworzone

Cookies

Kilka uwag

- niektóre argumenty można pominąć podając napisy puste
- ustawione cookie będą widoczne dopiero po przeładowaniu strony
- każda przeglądarka składuje cookies osobno
- niekiedy przeglądarki różnie obsługują cookies

Cookies

Przykład

- cookie1.php
- cookie2.php
- cookie3.php

Nagłówki HTTP

Do wysyłania nagłówka HTTP mamy funkcję

- int header(string treść_nagłówka)

Pobranie list nagłówków do wysłania

- array headers_list (void)

Sprawdzenie, czy nagłówki zostały już wysłane

- bool headers_sent (void)

Nagłówki HTTP

Przykłady:

- header("Location: http://www.onet.pl/");
- header("Cache-Control: max-age=0,
  must-revalidate, proxy-revalidate");
- $download_size = filesize($file_server_path);
  header("Content-type: application/x-download");
  header("Content-Disposition: attachment;
  filename=$file_download_name;");
  header("Accept-Ranges: bytes");
  header("Content-Length: $download_size");
  @readfile($file_server_path);

Buforowanie

Do obsługi buforowania mamy funkcje

- bool ob_start ([callback output_callback])
  - rozpoczęcie buforowania
- bool ob_end_clean (void)
  - kończy buforowanie i czyści bufor wyjściowy
- bool ob_end_flush (void)
  - kończy buforowanie i wypisuje zawartość
- void ob_clean (void)
  - czyści bufor wyjściowy

Buforowanie

Do obsługi buforowania mamy funkcje cd.

- string ob_get_clean (void)
  - zwraca aktualną zawartość bufora i czyści bufor
- string ob_get_flush(void)
  - wypisuje zawartość bufora
- zwraca go jako napis
  - kończy buforowanie
- int ob_get_length(void)
  - zwraca rozmiar bufora wyjściowego

Przykład: bufory.php

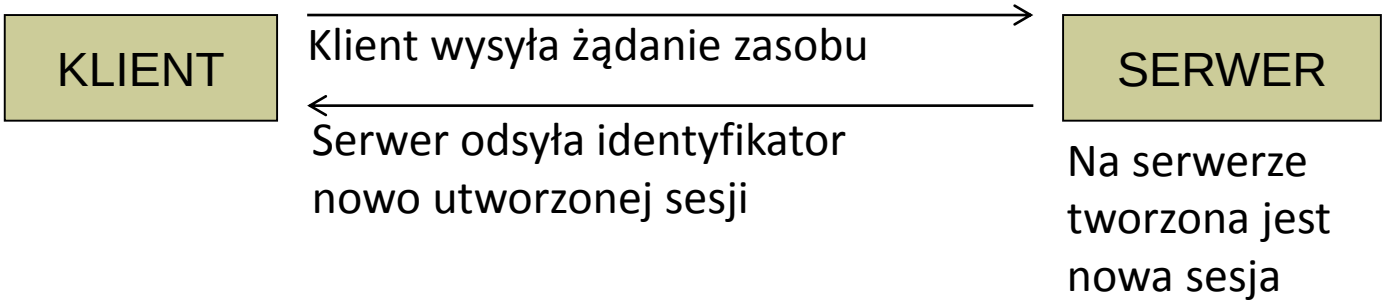
Sesje

Wprowadzenie

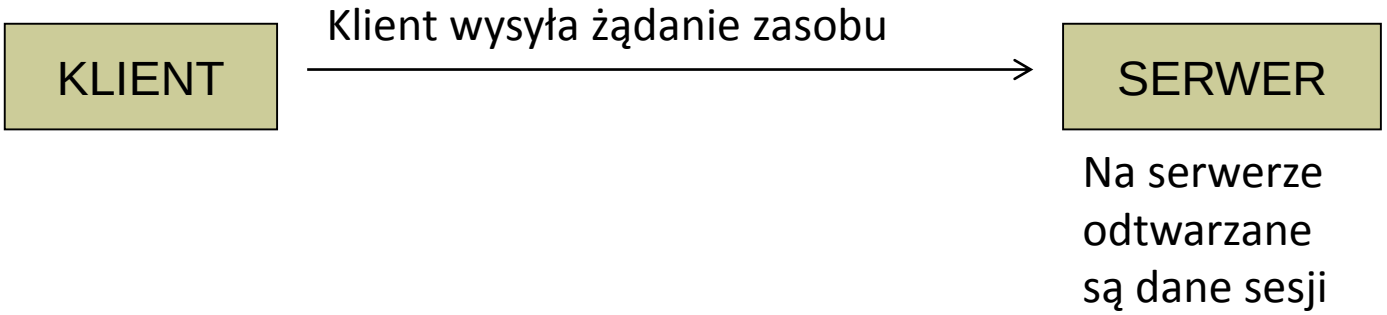
- Po co nam sesje?
- Rola klienta i serwera
- Identyfikator sesji
  - Cookie
  - Zmienna GET lub POST

Sesje

Utworzenie sesji



Korzystanie z sesji



Sesje

Zagrożenia

- W zasadzie jedno: przechwycenie identyfikatora

W jaki sposób ktoś może przechwycić?

- Cookie jest źle skontruowane i wysyłane szerzej niż trzeba, ktoś wysyła nam link (np. pocztą), klikamy i identyfikator ląduje na serwerze napastnika
- Fizyczne włamanie do komputera i wykradzenie id.
- Włamanie i modyfikacja pliku hosts
- Podsłuchiwanie

Sesje

Jak się bronić?

- Ustawienie krókiego czasu nieaktywności, np. 3 min
- Ustawienie krótkiego czasu ważności sesji (ciastka sesyjnego), np. 15 min
- Kontrola ciągłości zmiennej UserAgent
- Monitorowanie ruchu sieciowego, np. żądania następują po sobie w odstępie 10ms (łatwo przegiąć)
- Wyłapywanie różnorodnych adresów IP (ryzykowne)
- Nie przechowywać po stronie klienta za dużo danych (nie nadużywać cookies)

Sesje

Podstawowy sposób obsługi sesji w PHP

- Utworzenie/odtworzenie sesji
  - bool session_start()
- Zniszczenie danych sesji
  - bool session_destroy()
    - Uwaga: ta funkcja nie usuwa ciastka sesyjnego! Do tego może się przydać konstrukcja: setcookie(session_name(),"",0,"/")
- Zmienne sesji
  - Obsługujemy na dwa sposoby
    - poprzez rejestrację zmiennej jako sesyjnej (przestarzałe)
    - poprzez tablicę $_SESSION

Sesje

Zmienne sesji poprzez tablicę $_SESSION

- ustanowienie zmiennej sesyjnej
  - $_SESSION["zmienna"] = "wartość"
- usunięcie zmiennej sesyjnej
  - unset($_SESSION["zmienna"])
- usunięcie wszystkich zmiennych
  - $_SESSION = array()
- sprawdzenie, czy zmienna istnieje
  - isset($_SESSION["zmienna"])

Sesje

Zmienne sesji poprzez rejestację (niezalecane)

- bool session_register(mixed nazwa [, mixed ...])
  - rejestruje zmienną globalną jako zmienną sesji
- bool session_unregister(string nazwa)
  - wyjestruj zmienną z bieżącej sesji
- void session_unset(void)
  - zwalnia wszystkie zmienne sesji
- bool session_is_registered(string nazwa)
  - sprawdza, czy zmienna jest zarejestrowana

Przykład: licznik.php

Sesje

Inne funkcje do zarządzania sesjami

- string session_name ([string nazwa])
  - pobierz i/lub ustaw nazwę dla sesji
  - wywołujemy przed session_start()
  - używana w identyfikatorze sesji, domyślnie PHPSESSID
  - jeśli chcemy zmienić nazwę, to musimy ją wywoływać na każdej stronie – w przeciwnym razie od razu przywracana jest wartość z session.name z php.ini
- string session_id ([string id])
  - pobierz i/lub ustaw identyfikator sesji
  - jeżeli zmienić identyfikator bieżącej sesji, to musimy wywołać funkcję przed session_start()

Sesje

Inne funkcje do zarządzania sesjami cd.

- void session_write_close(void)
  - zapisuje dane i kończy sesję
- void session_commit(void)
  - alias do session_write_close()
- string session_save_path([string ścieżka])
  - pobierz i/lub ustaw ścieżkę zapisu bieżącej sesji
- bool session_regenerate_id(void)
  - tworzy nowy identyfikator dla sesji, dostępne od PHP 4.3.2

Sesje

Inne funkcje do zarządzania sesjami cd.

- string session_encode(void)
  – koduje dane sesji do postaci stringu
- bool session_decode(string dane) – odtwarza dane sesji ze stringu; najpierw trzeba użyć session_start
- void session_set_cookie_params(int czas_życia
  [, string ścieżka [, string domena]])
  – ustawia parametry ciastka sesji
- array session_get_cookie_params(void)
  – zwraca parametry ciasteczka sesji:
  - lifetime, path, domain, secure

Sesje

Niszczenie sesji

- Pierwszy sposób

```
<?php
// Jeśli nazwa sesji nie jest domyślna, należy
// wywołać w tym miejscu sesion_name("nazwa"),
session_start(); session_unset(); session_destroy();
?>
```

- Drugi sposób

```
<?php
// Jeśli nazwa sesji nie jest domyślna, należy
// wywołać w tym miejscu sesion_name("nazwa"),
session_start();
$_SESSION = array();
session_destroy();
?>
```


Sesje

Propagowanie identyfikatora w URLu

- kiedy stosować?
- stała SID
- opcja --enable-trans-sid
- propagowanie, a wydajność – różnica ok. 10%

Zapisywanie danych sesji w bazie danych

- do czego może się to przydać? (personalizacja)
- funkcje session_encode() i session_decode()

Przykład: sesje.php

Sesje

Implementacja sesji opartych na bazach danych

- Na czym to polega?
  - Identyfikatory oraz dane sesji są składowane w bazie danych, a nie za pomocą plików tekstowych
- Jak to zrobić?
  - Zaimplementować odpowiednie funkcje:
    - open(), close(), read(), write(), destroy(), gc()
  - Przekazać te funkcje do wywołania funkcji
    - bool session_set_save_handler(string open, string close, string read, string write, string destroy, string gc)
  - Wywołanie tej funkcji powinno być przed wywołaniem funkcji session_start()

Sesje

Implementacja sesji opartych na bazach (...) c.d.

- Zalety
  - Ułatwia rozwiązanie utworzenia klastru serwerów i kontrolowania sesji w tym modelu
  - Rozwiązuje problem dostępu do katalogu, w którym przechowywane są dane sesji (np. dostępu do nazw plików)
  - Jest bardziej efektywna przy większych obciążeniach

Sesje

Opcje w pliku konfiguracyjnym php.ini

- session.save_path
  - określa, gdzie będą tworzone pliki z danymi sesji
  - domyślnie dla WAMP5: c:/Program Files/wamp/tmp
- session.name
  - określa nazwę sesji, która będzie też nazwą ciastka
  - domyślnie PHPSESSID
- session.auto_start
  - określa, czy sesja ma być uruchamiana automatycznie na początku wywołania
  - domyślnie 0 (wyłączone)

Sesje

Opcje w pliku konfiguracyjnym php.ini

- session.cookie_lifetime
  - określa w sekundach długość życia ciastka sesyjnego
  - 0 oznacza "dopóki przeglądarka nie została zamknięta"
  - domyślnie 0
- session.use_cookies
  - określa, czy identyfikator sesji będzie po stronie klienta pamiętany za pomocą ciastka; domyślnie 1 (włączone)
- session.use_only_cookies
  - określa, czy do pomiętania id sesji mają być używane tylko ciastka; domyślnie 0 (ze względu na kompatybilność)

Access Control

Terminologia

- Identification
- Authentication
- Authorization
- Access Control

Access Control

Składowanie haseł:

- baza danych
- serwer LDAP (np. Active Directory)
- serwer RADIUS

Przechowywanie w postaci:

- jawnej
- funkcji skrótu
- zaszyfrowanej

Access Control

Etapy dostępu do danych

- Logowanie
  - pobranie danych użytkownika (np. login + hasło)
  - weryfikacja
  - odnotowanie faktu weryfikacji w logach
  - rozpoczęcie sesji
- Użycie sesji
  - podtrzymanie sesji
  - autoryzacja dostępu do zasobów
- Wylogowanie
  - zakończenie sesji
  - odnotowanie faktu w logach
  - usunięcie danych związanych z sesją (ważne)

Access Control

Przykłady

- simplelogin.php, gratulacje.html
- simplelogin2.php
- login.php, authenticated.php

