PHP - Relace, autentizace, chyby

Relace

- Aplikace jako e-shop potřebují udržovat stav
- Klientské relace
  - Data ukládána na straně klienta
  - Při každém požadavku přenášena mezi klientem a serverem
- Serverové relace
  - Data ukládána na serveru
  - Objem přenesených dat mezi klientem a serverem je malý
  - Relaci je přiřazen identifikátor a ten je pak přenášen

Klientské relace

- Analogie s návštěvou lékaře – pacient přichází s kartou
- Problémy
  - Riziko ztráty
  - Obtížná přeprava
  - Neoprávněné nahlédnutí/sfalšování
- Implementace přes cookie
  - $_COOKIE['name']
  - setcookie('name', 'value', platnost);
    time()+3600
  - Vkládání přes serialize(…), výběr přes unserialize(stripslashes(…))
- Uživatel může cookies měnit – nutné šifrování

Serverové relace

- Analogie s návštěvou lékaře – přinesu pouze číslo karty
- Metody
  - Cookie – session.use_cookies = 1
    - Název cookie: PHPSESSIONID (session.name v php.ini)
  - Munging řetězce dotazu: session.use_trans_sid = 1
- Cookie bezpečnější, estetičtější
- Spuštění relace: session_start()
- Vložení dat: $_SESSION['name'] = value
- Vypršení relace: session.cookie_lifetime v PHP.ini (implicitně 0 = vyprší po zavření prohlížeče)
- Vyčištění relace:

  $_SESSION = array(); session_destroy();

Autentizace

- Analogie lyžařský vlek – zakoupení jízdenky
- Ověření identity uživatele – shoda jména a hesla s údaji v db
- Metody
  - Základní – integrována v HTTP
  - Munging řetězce dotazu
  - $_SERVER['REMOTE_IP'] – takhle ne!
  - Klientská relace – cookie
  - Serverová relace - session

Základní autentizace

- Autentizační schéma integrováno v protokolu HTTP
- Jakmile server obdrží neautorizovaný požadavek na stránku, odpoví hlavičkou: www-Authenticate: Basic realm="MyRealm"
- realm je libovolný název vašeho jmeného prostoru
- Vyskakovací okno pro zadání jména a hesla
- Jméno a heslo se předá skriptu v $_SERVER['PHP_AUTH_USER'] a $_SERVER['PHP_AUTH_PW']
- Lze použít k ochraně souborů

Munging řetězce dotazu

- Identita se předává v řetězci dotazu
- Nehezké adresy
- Bezpečnostní problém

Klientské relace – cookie

- Vytvoření cookie u uživatele
- Musí být šifrováno
- Expirace
  - po každém požadavku
  - po každém intervalu
- Identifikátor uživatele v cookie

Serverová relace – session

- Uložení informace o přihlášeném uživateli do session

Zpracování chyb (1)

- Vnější chyby – program se nechová tak, jak by se o něj očekávalo, např. spojení s db se nezdaří
- Chyby v kódu – bugy, chybná logika/překlep
- PHP má tři úrovně závažnosti chyb
  - `E_NOTICE` – upozorňuje na něco, co funguje, ale možná ne tak, jak jste zamýšleli (např. použití proměnné, které ještě nebyla přiřazena hodnota)
  - `E_WARNING` – nezpůsobuje zastavení nebo změnu toku provádění skriptu (např. vnější chyby `fopen, mysql_connect`
  - `E_ERROR` – chyby, ze kterých se nelze zotavit, zastaví provádění skriptu (např. vytvoření instance neexistující třídy)

Zpracování chyb (2)

- Vyvolání uživatelské chyby:

  `trigger_error($message, E_USER_NOTICE)`
- V `php.ini` lze řídit úroveň chyb, které proniknou až do vašeho skriptu
  - `error_reporting = E_ALL ~ E_NOTICE`
  - `display_errors = on/off`
  - `log_errors = on/off`
  - `error_log = cesta_k_souboru`

    (logování: `error_log("Retezec chyby");`)
  - `track_errors = on`, poslední chybové hlášení bude uloženo v `$php_errormsg`

Výjimky

- Kde obsloužit chyby – lokálně nebo u uživatele knihovny?
- Výjimka je struktura řídící tok programu
- Výjimky jsou objekty
- PHP má zabudovanou třídu `Exception`
- Nezachycená výjimka je fatální chybou

```
try {
  blok, kde může být výjimka vyhozena
}
catch (Exception $e) {
  obsluha výjimky
}
```

Implementace vlastní výjimky

- Oddědit od `Exception`
- Přepsat konstruktor:

  `__construct($message=false, $code=false)`
- Z `__FILE__` a `__LINE__` lze zjistit poslední volání
- `debug_backtrace()` – zpětné stopování
- Při obsluze např. `print_r($e)`