TW 1 – Přednáška č. 3
Obsluha formulářů v PHP, koláčky, sezení

Martin Klíma

Computer Graphics Group

Proměnné prostředí

- **$_SERVER**: pole hodnot nastavené serverem.
- **$_GET:** pole hodnot parametrů z HTTP GET
- **$_POST:** pole hodnot parametrů z HTTP POST
- **$_SESSION:** pole hodnot session proměnných
- **$_COOKIE:** pole hodnot cookie poslaných klientem
- **$_REQUEST:** sloučené pole $_GET, $_POST a $_COOKIE. Hodnoty se přepisují v pořadí určeném v php.ini
- **$_ENV:** pole hodnot systémových proměnných
- **$_FILES:** pole uploadnutých souborů

$_SERVER

- server_variables.php
- all_server_vars.php

$_POST, $_GET

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<title>Pokusny formular</title>
</head>
<body>

<form method="GET" action="all_request_vars.php">
    <input type="hidden" name="skryte_pole1" value="hodnota_skryteho_pole">

    <input type="text" name="text1" size="20"><br>

    <textarea rows="2" name="textarea1" cols="20"></textarea><br>

    <input type="checkbox" name="checkbox1" value="on"><br>

    <input type="submit" value="odeslat" name="tlacitko_odeslat">
    <input type="reset" value="obnovit" name="tlacitko_obnovit">
</form>

</body>
</html>
```

$_GET, $_POST pokrač.

test_form.html,
all_request_vars.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Všechny request proměnné </title>
</head>
<body>
<h1>POST</h1>
<table border="1" style="border-collapse: collapse">
<?php
        foreach ($_POST AS $key=>$value) {
                echo "<tr><td>$key</td><td>$value</td></tr>";
        }
?>
</table>

<h1>GET</h1>
<table border="1" style="border-collapse: collapse">
<?php
        foreach ($_GET AS $key=>$value) {
                echo "<tr><td>$key</td><td>$value</td></tr>";
        }
?>
</table>

<h1>REQUEST</h1>
<table border="1" style="border-collapse: collapse">

<?php
        foreach ($_REQUEST AS $key=>$value) {
                echo "<tr><td>$key</td><td>$value</td></tr>";
        }
?>
</table>

</body>
</html>
```

! POZOR – chyba ve výpisu nebezpečných znaků

! POZOR – chyba ve výpisu nebezpečných znaků

! POZOR – chyba ve výpisu nebezpečných znaků

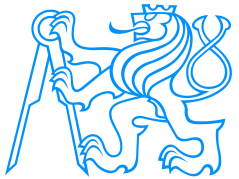
Více hodnot se stejným jménem?

také v test_form_2.html

```
<select name="select_1[]" multiple="multiple">
    <option value="1">Okurky</option>
    <option value="2">Jablka</option>
    <option value="3">Brambory</option>
</select>
<br>
<input type="checkbox" name="checkbox_m[]" value="A">A<br>
<input type="checkbox" name="checkbox_m[]" value="B">B<br>
<input type="checkbox" name="checkbox_m[]" value="C">C<br>
```

Úspěšná a neúspěšná pole

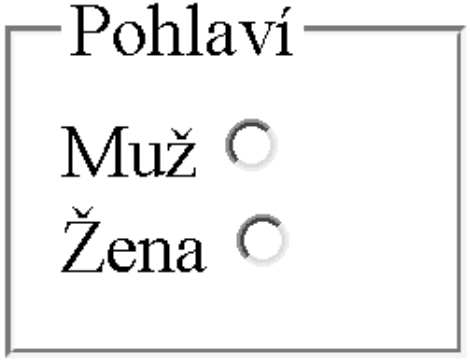
- Úspěšné: jeho hodnota se propaguje na server jako dvojice klíč – hodnota
- Neúspěšné: server se o něm nic nedozví



Úspěšná a neúspěšná pole pokr.

Atribut disabled="disabled" – pole neúspěšné

Atribut readonly="readonly" – pole úspěšné



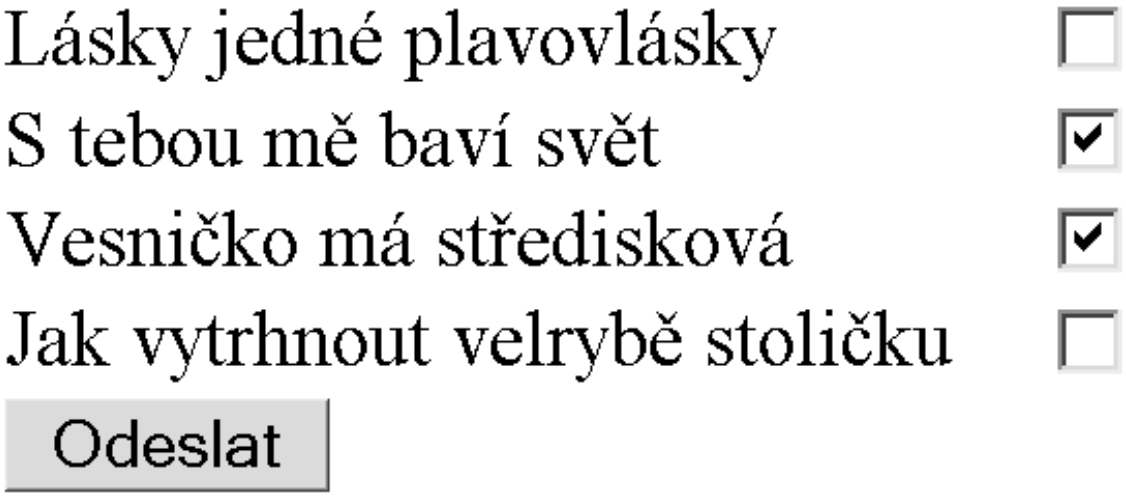
Pozor na input typu radio – co je zaškrtnuto?
jak "odvybrat" položku?

Při zpracování se nedozvím, která pole existovala, nevím tedy, z čeho se uživatel vybíral

HTTP je bezestavový protokol => komplikace

Přiklad problému

- Vyberte svůj oblíbený film:



- Aplikační logika:
  - vybraným zvýšíme ohodnocení o 1
  - nevybraným snížíme ohodnocení o 1

Které to jsou?

Možná řešení

oblibene_filmy.html

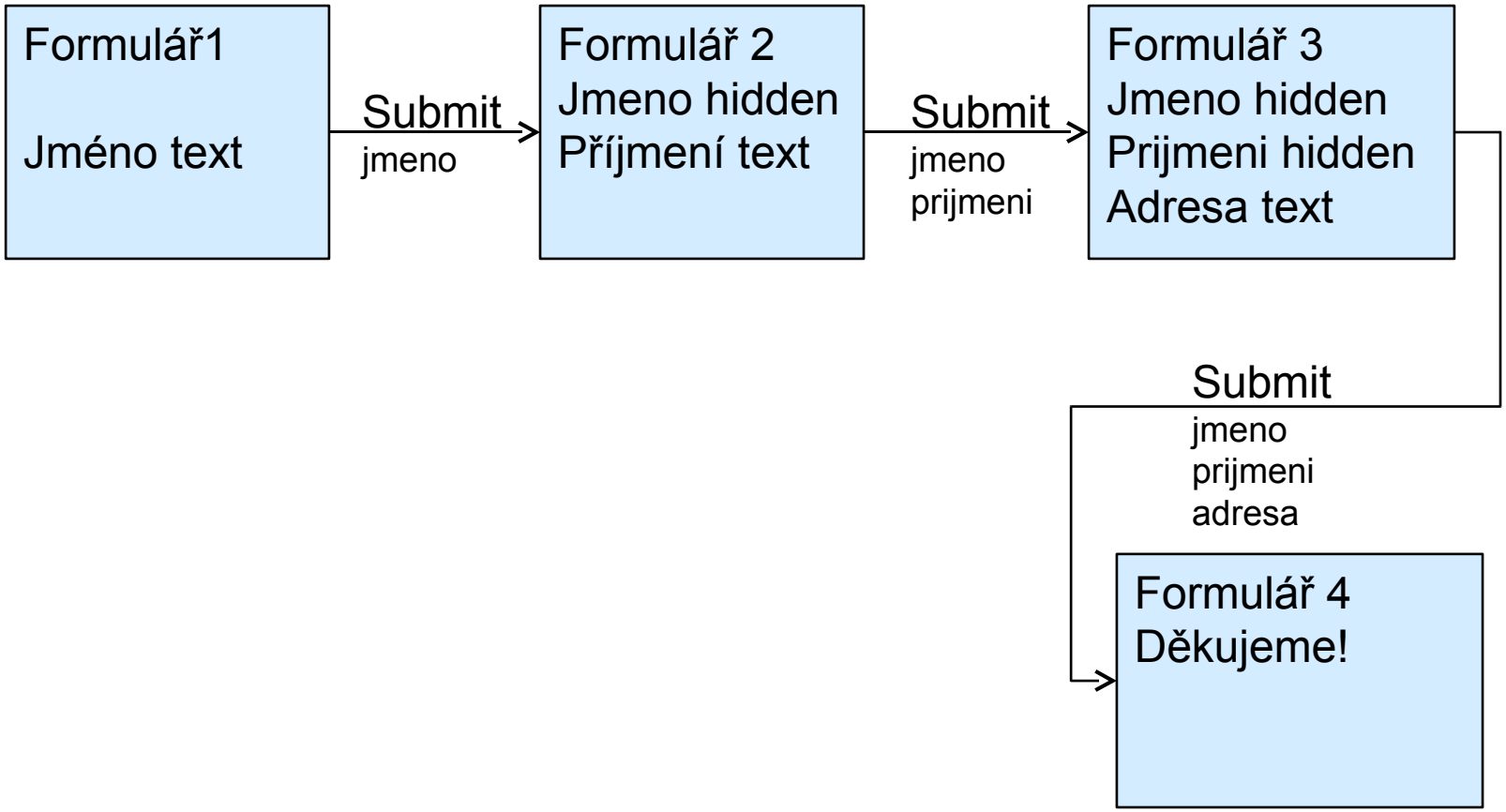
1. Při zpracování se opět dotážeme na původní množinu externího zdroje (db)
   - Co když se to mezitím změnilo?
2. V dotazu pošleme i celou množinu ze které uživatel vybíral
   - Hidden pole

```
<form method="POST" action="oblibene_filmy.php">
    <input type="hidden" name="vsechny_filmy" value="f1 f2 f3 f4">

    <label for="f1">Lásky jedné plavovlásky</label>
    <input type="checkbox" id="f1" name="oblibene_filmy[]" value="f1"><br>
    <label for="f2">S tebou mě baví svět</label>
    <input type="checkbox" id="f2" name="oblibene_filmy[]" value="f2"><br>
    <label for="f3">Vesničko má středisková</label>
    <input type="checkbox" id="f3" name="oblibene_filmy[]" value="f3"><br>
    <label for="f4">Jak vytrhnout velrybě stoličku</label>
    <input type="checkbox" id="f4" name="oblibene_filmy[]"  value="f4"><br>

    <input type="submit"  name="odeslat" value="Odeslat">
</form>
```

Zpracování v php

```
<?php
  $vsechny_filmy= split(" ", $_POST['vsechny_filmy']);
  foreach ($vsechny_filmy as $aktualni_film) {
    echo '<div>'.htmlspecialchars($aktualni_film);
    if (in_array($aktualni_film, $_POST['oblibene_filmy'])) {
      echo " +1 bod";
    } else {
      echo "-1 bod";
    }
    echo '</div>';
  }


 } else { echo "Formulář dosud nebyl odeslán"; }

?>
```

Wizardy


Formulář1
Jméno text
Submit
jmeno
Formulář 2
Jmeno hidden
Příjmení text
Submit
jmeno
prijmeni
Formulář 3
Jmeno hidden
Prijmeni hidden
Adresa text
Submit
jmeno
prijmeni
adresa
Formulář 4
Děkujeme!

Ukázka wizard 2/3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Wizard 2/3</title>
</head>

<body>
<h1>Krok 2/3</h1>
<form method="POST" action="wizard3.php">
        <input type="hidden"
value="<?php echo htmlspecialchars($_POST['jmeno']); ?>" name="jmeno">
        <label for="prijmeni">Příjmení</label>
        <input type="text" id="prijmeni" name="prijmeni"> <br>
        <input type="submit" name="odeslat" value="Odeslat">
</form>

</body>
</html>
```

Ukázka wizard poděkování

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Wizard 3/3</title>
</head>

<body>
<h1>Děkujeme</h1>
 Jméno: <?php echo htmlspecialchars(($_POST['jmeno'])); ?><br>
 Příjmení: <?php echo htmlspecialchars(($_POST['prijmeni'])); ?><br>
 Adresa: <?php echo htmlspecialchars(($_POST['adresa'])); ?><br>
</body>
</html>
```

Používání hodnot z formuláře

- Při používání hodnot z formuláře pozor na několik případů

Případ č.1 - špatně, post může obsahovat nebezpečné znaky

```
 <input type="text"
value="<?php echo $_POST['jmeno']; ?>" name="jmeno">
```

Případ č.2 – špatně, post může obsahovat znak '

```
 <input type="text"
value='<?php echo htmlspecialchars($_POST['jmeno']); ?>'
name="jmeno">
```

Případ č.3 – dobře

```
 <input type="text"
value="<?php echo htmlspecialchars($_POST['jmeno']); ?>"
name="jmeno">
```

Případ č.4 = dobře

```
 <input type="text"
value='<?php echo htmlspecialchars($_POST['jmeno'], ENT_QUOTES); ?>'
name="jmeno">
```

Používání hodnot z formuláře pokr.

Magic quotes

Myšlenka: z dat z formuláře se často sestavují SQL dotazy do MySQL databáze.

magic_quotes ON

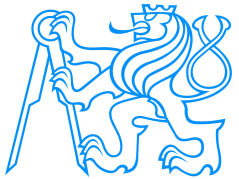
' -> \' " -> \" \ -> \\

Systém automaticky použije funkci addslashes() na GET, POST a COOKIES proměnné

Opak addslashes je stripslashes

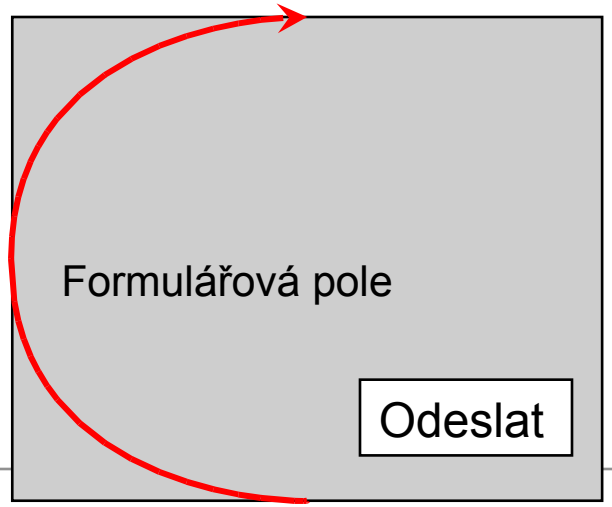
Životní cyklus formuláře

1. Zobrazení formuláře s iniciálními daty (nebo prázdného)
2. Uživatel vyplní (chybně) formulář a odešle
3. Zpracování na straně serveru
    1. Ověření správnosti dat
    2. Pokud OK, přechod dál
    3. Pokud není OK, vygeneruj stejný formulář s vyplněnými daty

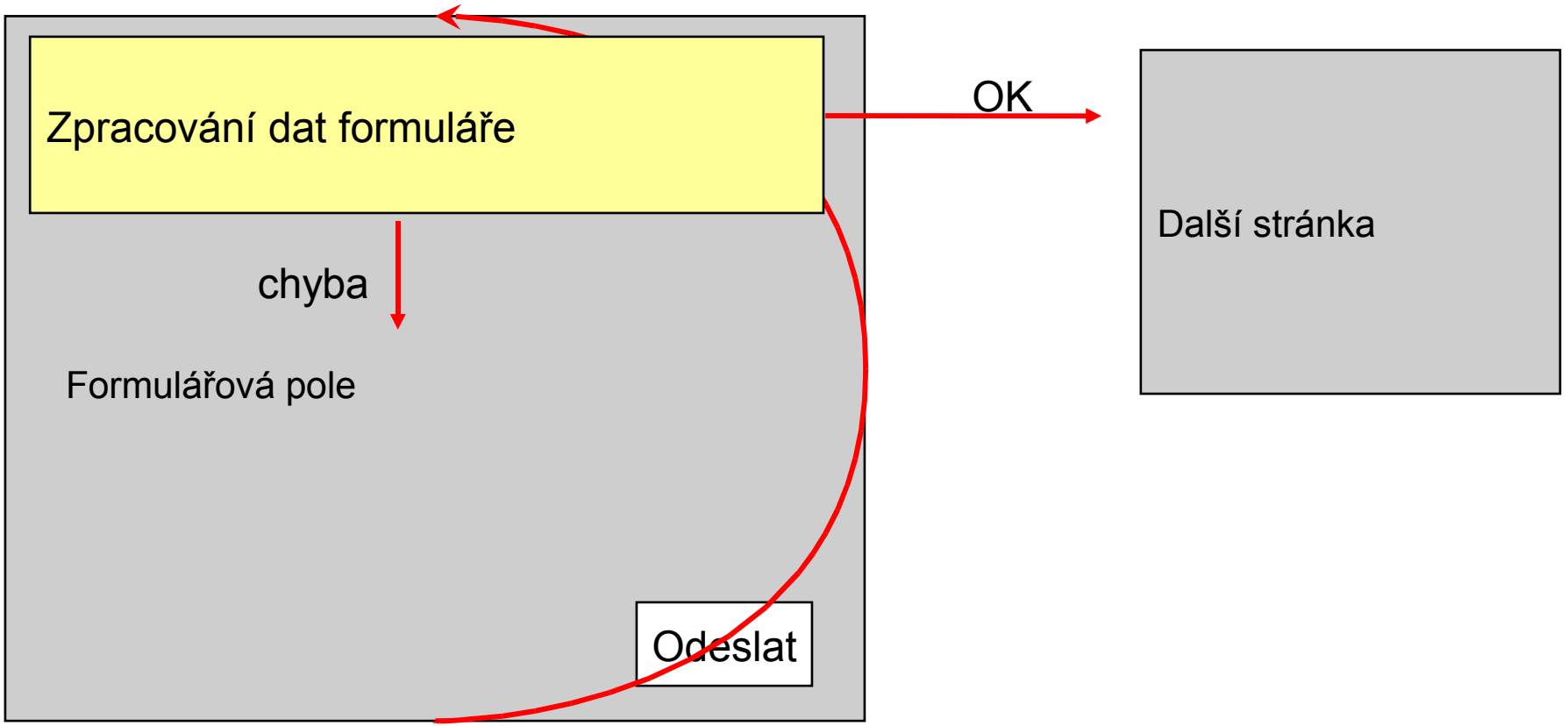


Životní cyklus formuláře

```
<form method="POST" action="<?echo $_SERVER["PHP_SELF"];?>">

        <!-- formularova pole -->
        Jméno: <input type="text" name="jmeno"><br>

        Příjmení: <input type="text" name="prijmeni" ><br>

        <!-- tlacitka -->
        <input type="submit" value="Odeslat" name="odeslat">
        <input type="reset" value="Reset" name="tlacitko_obnovit">
</form>
```



Životní cyklus formuláře



Implementace v PHP

```php
function over($jmeno, $prijmeni) {
  $jm = trim($jmeno);
  $pr = trim($prijmeni);
  return (strlen($jm)>=4 && strlen($pr)>=4);
}

function vratZPost($co) {
  if (isset($_POST[$co]))
     return htmlspecialchars($_POST[$co]);
  return "";
}
```

```php
<?php
$hlaska = "";

if (isset($_POST['odeslat'])) {
 if (over($_POST['jmeno'], $_POST['prijmeni'])) {
    include ("dekujeme.php");
    exit();
} else {
    $hlaska = "Špatně vyplněné položky, prosím opravte!";
   }
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Životní cyklus formuláře</title>
</head>
<body>
<?php if ($hlaska != "") echo htmlspecialchars($hlaska); ?>
<form method="POST" action="<?php echo $_SERVER["PHP_SELF"];?>">

        Jméno: <input type="text" name="jmeno"
        value="<?php echo vratZPost("jmeno");?>"><br>

        Příjmení: <input type="text" name="prijmeni"
        value="<?php echo vratZPost("prijmeni");?>" ><br>

        <input type="submit" value="Odeslat" name="odeslat">
        <input type="reset" value="Reset" name="tlacitko_obnovit">
</form></body></html>
```

Práce se soubory

- Možnost uploadu souborů

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
<title>Upload souborů</title>
</head>

<body>

<form
action="<?php echo $_SERVER['PHP_SELF']; ?>"
method="POST" enctype="multipart/form-data">
Zadejte soubor: <input type="file" name="soubor1"><br>
<input type="submit" name="odeslat" value="Odeslat">
</form>

</body>
</html>
```

Práce se soubory

- Existuje superglobální proměnná $_FILES, která obsahuje 2D pole hodnot

$_FILES['userfile']['name'] – původní jméno souboru

$_FILES['userfile']['type'] – mime type souboru

$_FILES['userfile']['size'] – velikost souboru

$_FILES['userfile']['tmp_name'] – jméno docasné souboru na serveru, kam byl soubor uložen

$_FILES['userfile']['error'] – chyba, ke které případně došlo

Práce se soubory (upload)

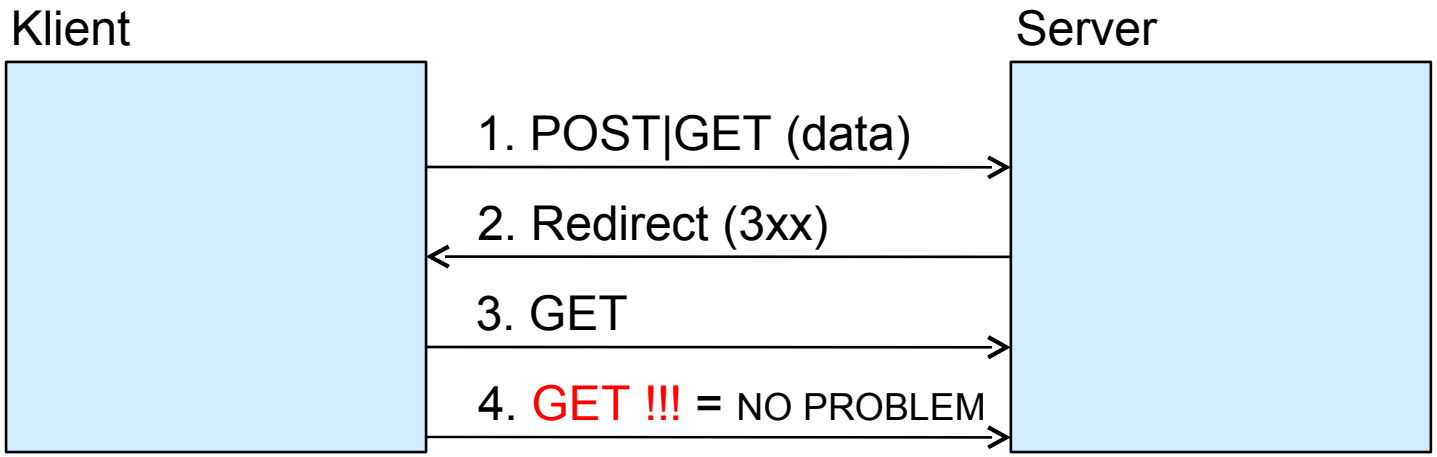
```php
<?php
if (isset($_POST['odeslat'])) {
        // obsluha formulare

        $jmeno_souboru = $_FILES['soubor1']['name'];
        $tmp_jmeno  = $_FILES['soubor1']['tmp_name'];
        $velikost = $_FILES['soubor1']['size'];
        $typ_souboru = $_FILES['soubor1']['type'];

        if (!move_uploaded_file($tmp_jmeno, "c:".DIRECTORY_SEPARATOR.
$jmeno_souboru)) {
                $hlaska = "error ".$_FILES['soubor1']['error'];
        }
}
?>
```

Dvojí odeslání dat

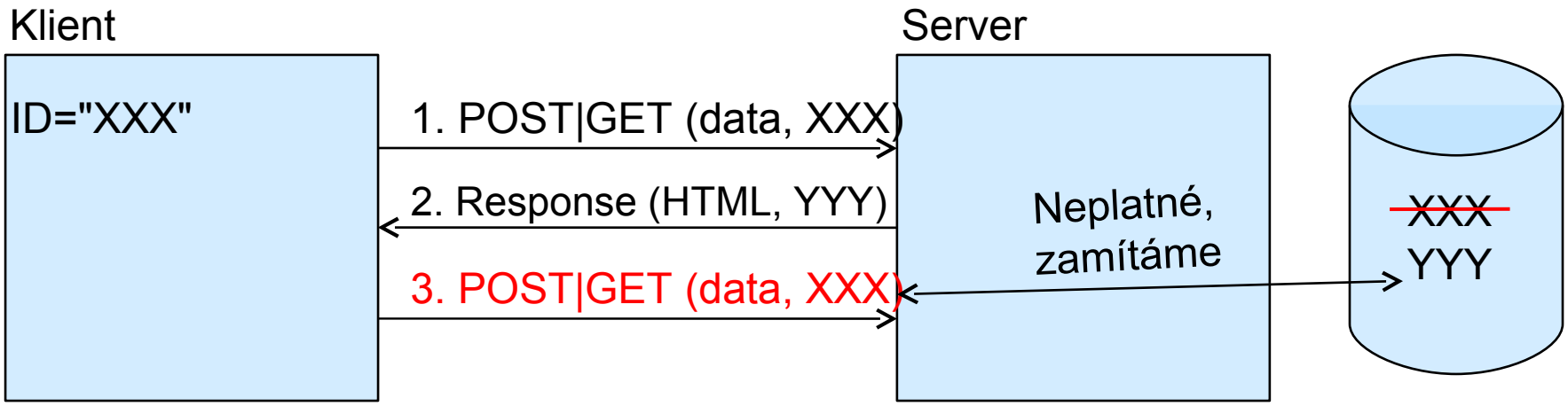
- Uživatel vícekrát stisknul tlačítko submit
- Uživatel se vrátil funkcí zpět
- Někdo nás chce hacknout

Ochrana: docela problém, máme bezestavový protokol, tj. nikdo si nic nepamatuje.

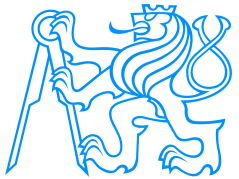
Řešení č.1 – ne zcela správné, ale časté



Řešení


Klient
Server
ID="XXX"
1. POST|GET (data, XXX)
2. Response (HTML, YYY)
3. POST|GET (data, XXX)
Neplatné,
zamítáme
XXX
YYY

UDRŽOVÁNÍ STAVU



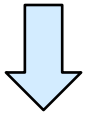
Udržování stavu aplikace

1. Pomocí skrytých polí (viz příklad wizard)
2. Pomocí obohacování odkazů
3. Pomocí cookies
4. Pomocí sessions

Obohacování odkazů

- Veškeré odkazy z dokumentu budou obohaceny o identifikátor s daty sezení
- Není to moc elegantní
- Může to selhat
- Pokud se děje automaticky, náročné na výkon

```
<a href="dalsi_stranka.php">Další stránka</a>
<form action="obsluha.php" method="post">
</form>
```



```
<a href="dalsi_stranka.php?session_id=XXX">Další stránka</a>
<form action="obsluha.php?session_id=XXX" method="post">
</form>
```

Pomocí cookies

- Informace uložená serverem na klientovi
- Klient posílá tuto informaci zpět pokud
  - URI spadá do vymezeného rozsahu
  - Doména odpovídá
  - Nevypršela platnost

- Využití:
  - Počítadlo přístupů
  - Další statistiky
  - **Udržování stavu aplikace**

Cookies

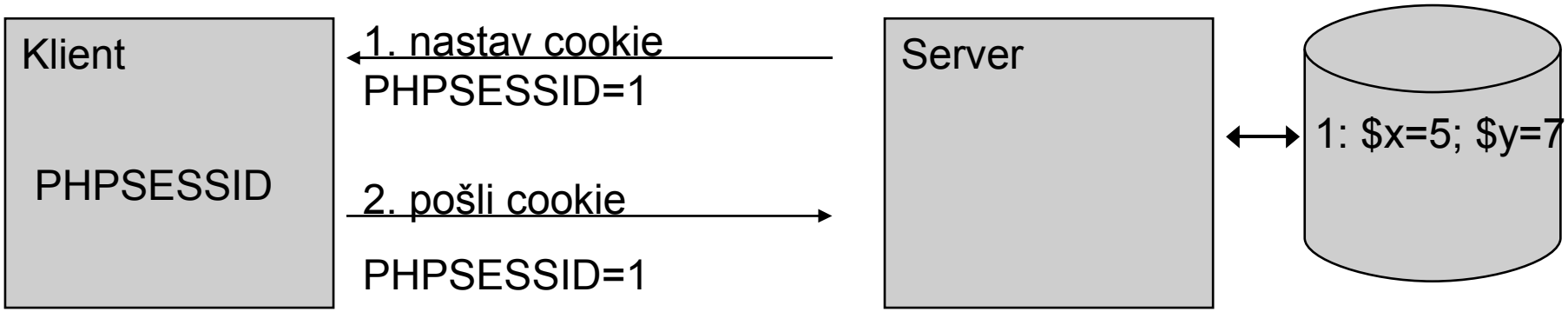
set_cookie.php

pocitadlo.php

```
<?

setcookie("mojecookie1","hodnota1");
setcookie("mojecookie2","hodnota2");

?>
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<title>Nastav Cookie</title>
</head>
<body>
<p>Tento skript nastavi cookie a přečte existující cookie.</p>
<?
foreach ($_COOKIE AS $key=>$value) {
    echo "$key = $value <br>";
}

?>
</body>
</html>
```

Sessions

- Zjednodušení práce
- Podpora stavu
- Kombinace cookies a lokální databáze



Příklad SESSIONS

sessions.php

```
<?
session_start();

$_SESSION['x'] = 5;
$_SESSION['y'] = 7;

?>
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<title>Sessions</title>
</head>
<body>
<p>Tento skript vypise seznam session promennych.</p>
<?
foreach ($_SESSION AS $key=>$value) {
      echo "$key = $value <br>";
}

?>
</body>
</html>
```

Dotazy?

DĚKUJI ZA POZORNOST