FROM DESKTOP TO WEB APP

Business College Helsinki
Kari KA Häkkinen
2019-09-15

Learn:
*** algorithms**
*** thinking**

ALGORITHM

- algorithm is a description on how one specific problem will be solved
- in this material we'll use two different algorithms to solve one specific problem …
- … which is to sum figures from 1 to given number N

ALGORITHM #1: (SLOW)

- first algorithm is …
  - easy to find out
  - slow when figures are big
- big O(n)
  - linear time consumption
- set sum = 0
- go through figures from 1 to N
  - calculate sum=sum+current figure

ALGORITHM #2: (FASTER)

- second algorithm is …
  - not so obvious
  - much quicker when figures are big
- big O(1)
  - constant time consumption

- set sum = 0
- sum biggest and smallest (call it BisMa)
- multiply with integer division N/2
- add half of BisMa, if N is odd

SUMMING 1…N

Algorithm #1 *[big O(N)]*:
Loop through 1 ... N
Sum them
...

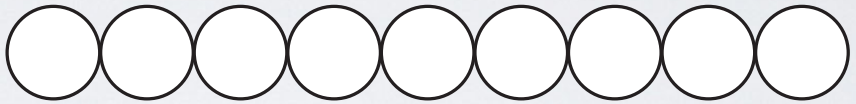
1 2 3 4

1 2 3

1 2

1

1 2 3 4 5 6 7 8 9



1+4 = 5
4:2=2
2x5=10

1 x (6/2)
1+5 = 6
5:2=2 kokonaista + 1
2x6 + (1 x (6/2))=12

(1+4 = 5
(4 div 2) x hesu+ (4 mod 2) * (hesu) = 2
2x5 =10)

Algorithm #2 *[big O(1)]*:
Smallest to be summed is **one=**1.
Biggest number to sum is **biggest.**
helpersum=one+biggest
Divide helpersum with two (**intdivpertwo**=**helpersum/2**)
Need remainder (modulus) too (**moduluspertwo**=**helpersum/2**)

IMPLEMENTATION WITH C

```
[PoliTrukki:dataStructure&algorithm karikahakkinen$ ./conut 44444
 Summing numbers from 1 to 44444 ...

 Algorithm #1: Sum is 987656790
 Time used: 0.000131

 44444 - 987656790
 Algorithm #2: Sum is 987656790
 Time used: 0.000000

[PoliTrukki:dataStructure&algorithm karikahakkinen$ ./conut 44443
 Summing numbers from 1 to 44443 ...

 Algorithm #1: Sum is 987612346
 Time used: 0.000145

 44443 - 987612346
 Algorithm #2: Sum is 987612346
 Time used: 0.000001
```

Learn:
*** programming language C**

```
#include <stdio.h>
#include <inttypes.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

#define BASE    10

int main(int argc, char **argv){

char*   endptr;
long    tocount;
long    i;
long    sum=0;
long    hesu;
clock_t start, end;
double cpu_time_used;


if (argc<2){
    printf("%s: use: %s count\n", argv[0], argv[0]);
    exit(-1);
}

tocount = strtoimax(argv[1],&endptr,BASE);

printf ("Summing numbers from 1 to %ld ...\n\n",tocount );

start = clock();
for (i=0;i<=tocount;i++){
    sum+=i;
}
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

printf ("Algorithm #1: Sum is %ld\n",sum );
printf ("Time used: %lf\n\n",cpu_time_used);

start = clock();
hesu = (tocount+1) ;
sum=((tocount/2)*hesu) + ((tocount%2)*(hesu/2));
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
printf("%ld - %ld\n",tocount, sum);
printf ("Algorithm #2: Sum is %ld\n",sum );
printf ("Time used: %lf\n\n",cpu_time_used);
exit(0);
}
```

WEB APPLICATION ENVIRONMENT

WEB Application Environment

Php Backend

(here: calculator)

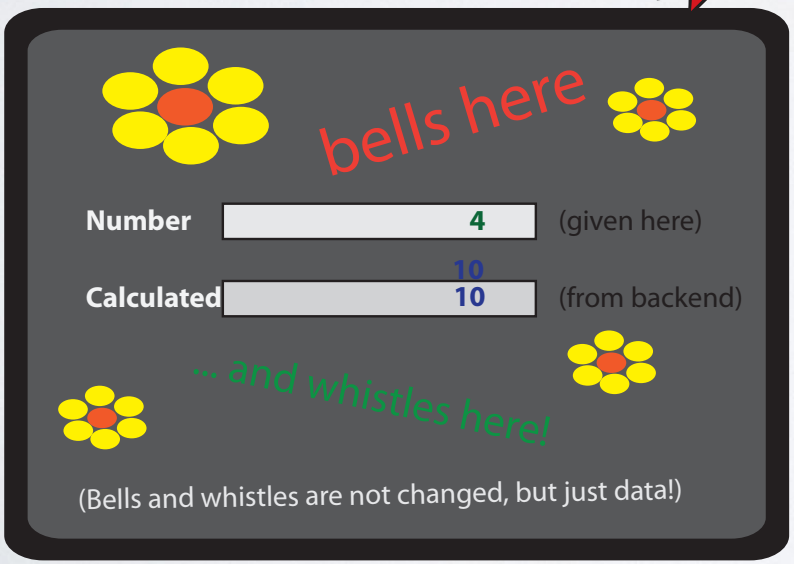
Network Data Transfers
restless API CRUD
JSON data (or XML)

GET, POST, ...
Here: input data e.g.GET -> 4

data as JSON
AJAX ... XMLHttpRequest
Calculated data as json
(alg#1 sum & time + alg#2 sum & time)

Frontend,
(html, JavaScript)

Changing data updated only



CRUD (Restless API):
* Create (POST)
* Read (GET)
* Update (PUT)
* Delete (DELETE)

Learn:
*** virtualizing + LAMP**
*** XAMPP**
*** FileSystem**
*** JavaScript, PHP, JSON**
*** DataStructures**

JSON THAT WE'LL USE

```
{"testing":
{"input":"4","output":
[{"sum":"800020000","time":
"0.000121"},
{"sum":"800020000","time":"
0.000001"}]}}
```

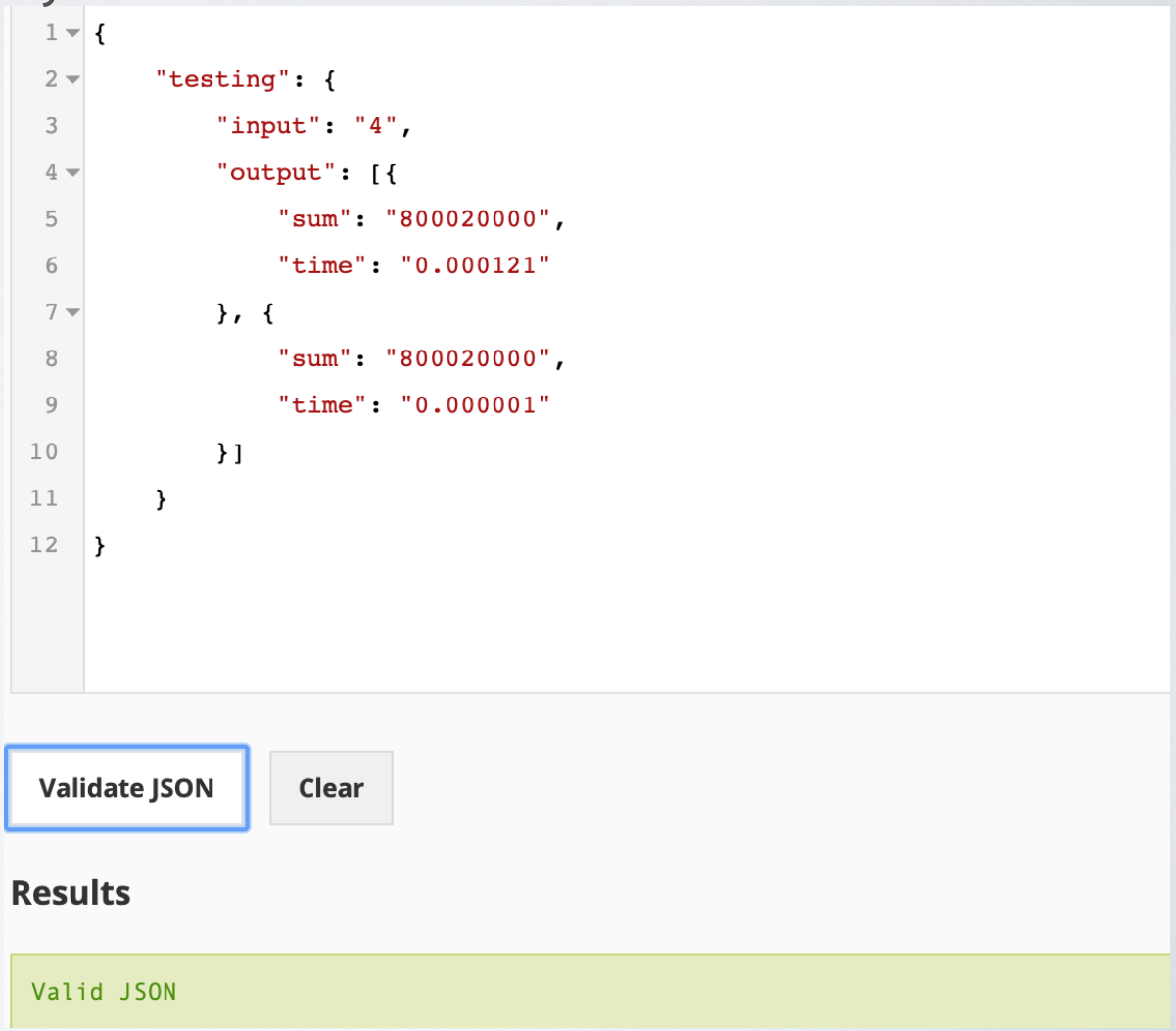
But is it *valid JSON?*

Yes, according

jsonlint.com

jsonlint.com

Learn:
*** validating JSON**



CREATING FIRST BACKEND

- It is easy to create first JSON backend by copying valid JSON to myjson.com

- then you can continue developing frontend without working, real backend

myjson.com

Learn:
*** using internet services in program development**

BUILDING OWN BACKEND #1

Learn:
*** basic html**

- it is easy to echo json to humans …

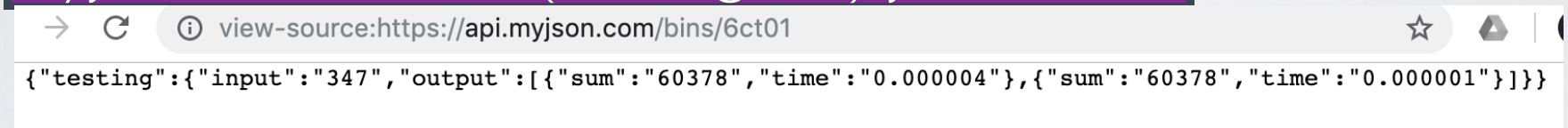
```
<!doctype html>
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <p>{"testing":{"input":"4","output":
[{"sum":"800020000","time":"0.000121"},
{"sum":"800020000","time":"0.000001"}]}}</p>
  </body>
</html>
```

BUILDING OWN BACKEND #2

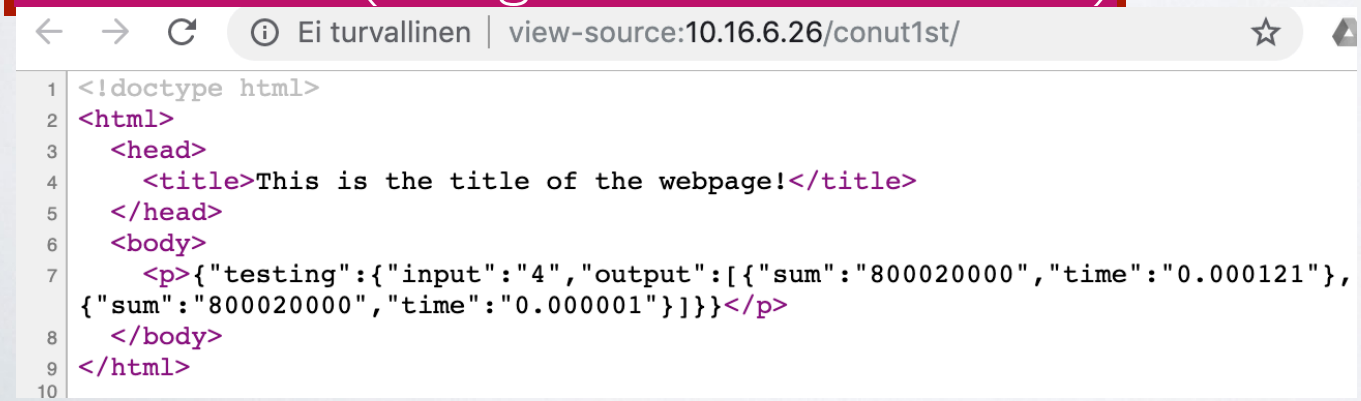
Learn:
*** using web browser developer tools**

- … but it is not ok to our frontend!

myjson.com source (this is good) json source



our source (not good, but ok starter!)



ALGORITHMS, IMPLEMENTED USING PHP

This echoes still html page - not json as meta!

Learn:
*** php functions**

```
<!doctype html>
<html>
  <head>
    <title>This is the title of the webpage!</title>
  </head>
  <body>
    <?php
       function soslow(int $nmbrin) {
     // https://www.php.net/manual/en/function.microtime.php
     $start = microtime(true);
     $sum=0;
     for ($i=0;$i<=$nmbrin;$i++){
        $sum += $i;
     };
     $time_elapsed_secs = microtime(true) - $start;
     $time=sprintf("%f",$time_elapsed_secs);
     return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
}
       function faster(int $tocount) {
               $start = microtime(true);
               $sum=0;
     $hesu = ($tocount+1) ;
     $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*($hesu/2));
               $time_elapsed_secs = microtime(true) - $start;
               $time=sprintf("%f",$time_elapsed_secs);
               return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
       }
      $in=40000000;
       $jsonStr  = '{"testing":{"input":"' . $in . '","output":[';
       $jsonStr .= soslow($in);        <- calling soslow() - returns json
       $jsonStr .= ',';
       $jsonStr .= faster($in);        <- calling faster() - returns json
       $jsonStr .= ']}}' ;
       echo '<p>' . $jsonStr . '</p>';        <- calling faster()
    ?>
  </body>
</html>
```

ALGORITHMS ... PHP IMPLEMENTED - JSON BACKEND

echoes JSON

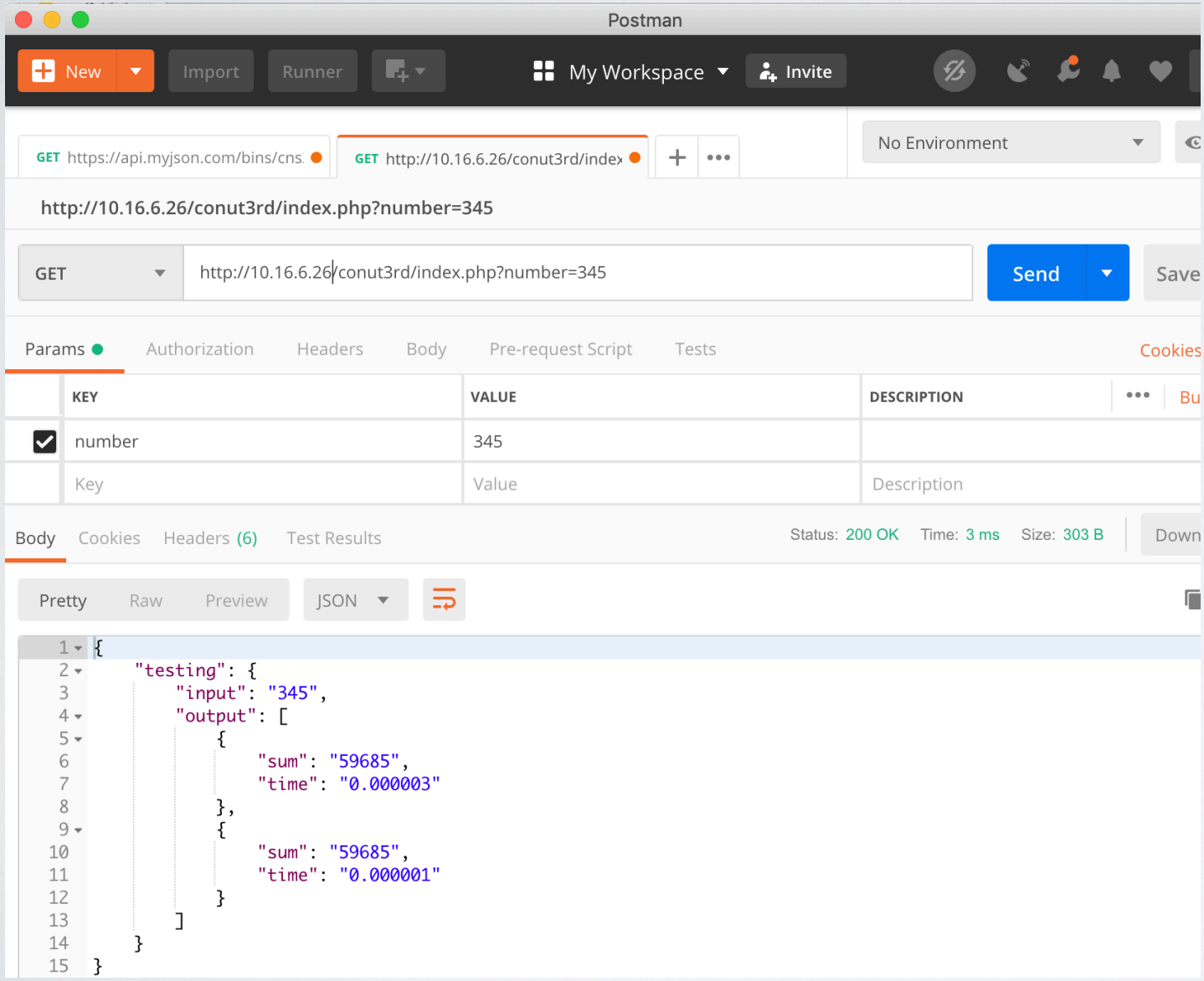
json as content-type

Learn:
*** www content-types**

```
<?php

// https://alexwebdevelop.com/php-json-backend/

 function soslow(int $nmbrin) {
  // https://www.php.net/manual/en/function.microtime.php
  $start = microtime(true);
  $sum=0;
  for ($i=0;$i<=$nmbrin;$i++){
    $sum += $i;
  };
  $time_elapsed_secs = microtime(true) - $start;
  $time=sprintf("%f",$time_elapsed_secs);
  return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
  }

  function faster(int $tocount) {
    $start = microtime(true);
    $sum=0;
    $hesu = ($tocount+1) ;
// PHP integer division intdiv()-> https://secure.php.net/manual/en/function.intdiv.php
    $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*(intdiv($hesu,2)));
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
  }

  https://www.w3schools.com/php/php_superglobals.asp
  echo 'given number was -> ' . $_GET['number'] ;
$in=$_GET['number'];

  $jsonStr  = '{"testing":{"input":"' . $in . '","output":[';
  $jsonStr .= soslow($in);
  $jsonStr .= ',';
  $jsonStr .= faster($in);
  $jsonStr .= ']}}' ;

    header('Content-Type: application/json');  //json backend header
  echo $jsonStr;
?>
```

TESTING BACKEND

Learn:
*** Postman**
*** curl**



FIRST FRONTEND

Learn:
* **basic JavaScript**
* **AJAX**
* **URLs and URIs**
* **GETs, PUTs, ...**
* **CRUD principles (RESTless API)**

```
<!DOCTYPE html>
<html>
<body>

<h2>The XMLHttpRequest Object</h2>

<button type="button" onclick="loadDoc()">Request data (4 in this example)</button>

<p id="demo"></p>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "http://192.168.100.15/conut3rd/index.php?number=4", true);
  xhttp.send();
}
</script>

</body>
</html>
```

SECOND FRONTEND

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id=”demo">Enter positive integer into field above!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://192.168.100.15/conut3rd/index.php?number="+numberValue
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("demo").innerHTML = this.responseText;
        }
      };
      xhttp.open("GET", URL, true);
      xhttp.send();
    }
  </script>

</body>
</html>
```

Learn:
*** basic JavaScript**
*** AJAX**

THIRD FRONTEND

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object
</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Enter positive integer into field above!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://10.16.6.26/conut3rd/index.php?number="+numberValue;
      var XHR = new XMLHttpRequest();
      XHR.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          let jsonobj = JSON.parse(this.responseText);
          let outPutText="With input data "+jsonobj.testing.input+":<br><br>"+
           "Algorithm #1 gives result "+jsonobj.testing.output[0].sum +
           " in time "+jsonobj.testing.output[0].time+".<br><br>"+
           "Algorithm #2 gives result "+jsonobj.testing.output[1].sum +
           " in time "+jsonobj.testing.output[1].time + "<br><br>";
     document.getElementById("demo").innerHTML = outPutText;
        }
      };
      XHR.open("GET", URL, true);
      XHR.send();
    }
  </script>

</body>
</html>
```

parse JSON string ->JS object…

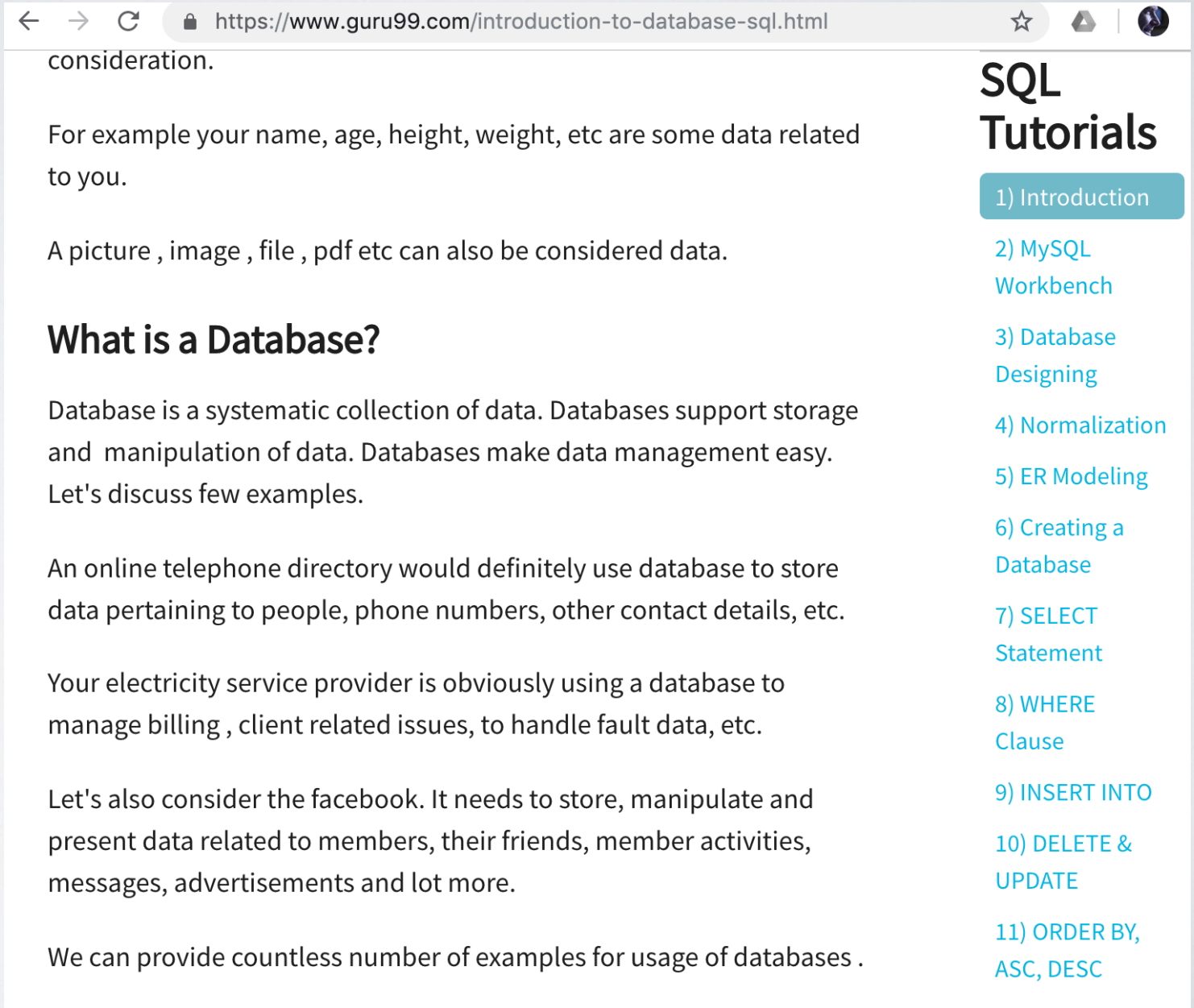
… and use it

Learn:
*** objects principles**
*** OOP principles**
*** JSON parsing in JavaScript**
*** JavaScript objects**

DATABASE BASICS

What is data, database, DBMS

https://www.guru99.com/introduction-to-database-sql.html



consideration.

For example your name, age, height, weight, etc are some data related to you.

A picture , image , file , pdf etc can also be considered data.

What is a Database?

Database is a systematic collection of data. Databases support storage and manipulation of data. Databases make data management easy. Let's discuss few examples.

An online telephone directory would definitely use database to store data pertaining to people, phone numbers, other contact details, etc.

Your electricity service provider is obviously using a database to manage billing , client related issues, to handle fault data, etc.

Let's also consider the facebook. It needs to store, manipulate and present data related to members, their friends, member activities, messages, advertisements and lot more.

We can provide countless number of examples for usage of databases .

SQL Tutorials

LEARNING SQL

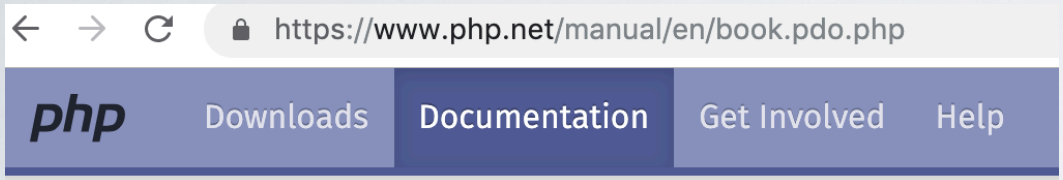
www.sqlcourse.com

www.sqlcourse2.com

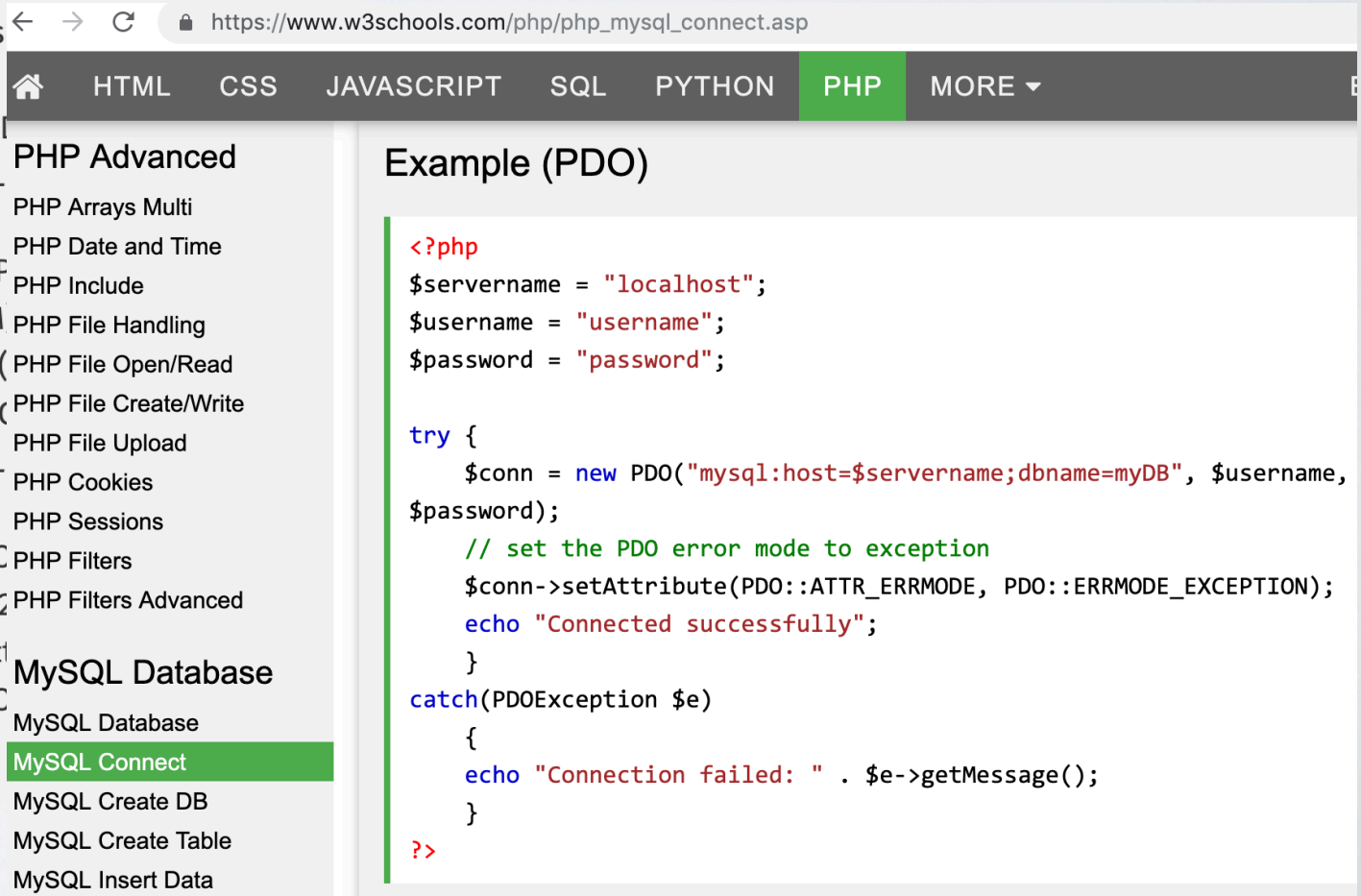
PHP CONNECTION TO SQL DATABASE

https://www.w3schools.com/php/php_mysql_connect.asp

https://www.php.net/manual/en/book.pdo.php

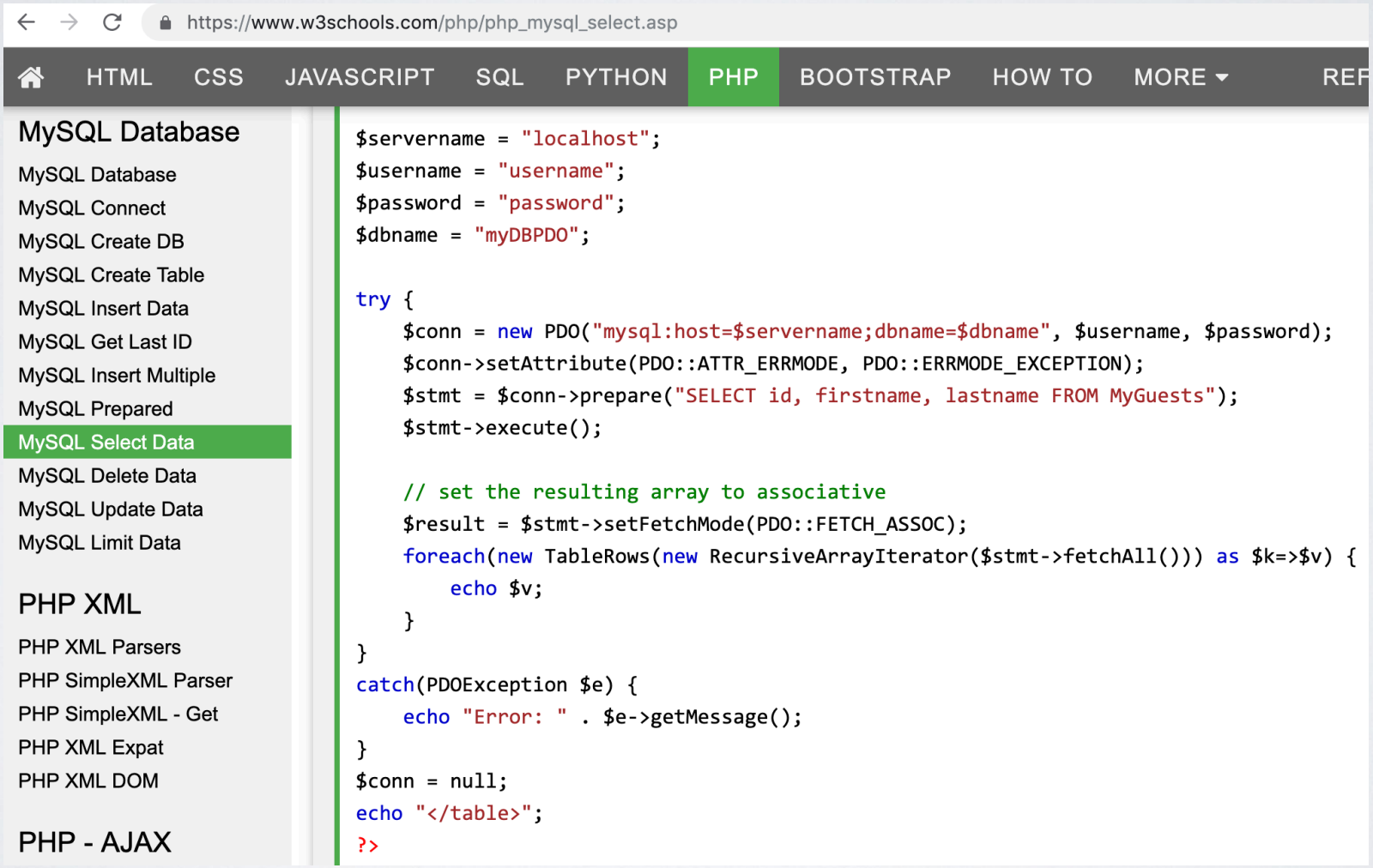


- PDOException — The PDOException class
- PDO Drivers
  - CUBRID (PDO) — CUBRID Functions (PI
  - MS SQL Server (PDO) — Microsoft SQL Functions (PDO_DBLIB)
  - Firebird (PDO) — Firebird Functions (F
  - IBM (PDO) — IBM Functions (PDO_IBM
  - Informix (PDO) — Informix Functions (
  - MySQL (PDO) — MySQL Functions (PDO
  - MS SQL Server (PDO) — Microsoft SQL (PDO_SQLSRV)
  - Oracle (PDO) — Oracle Functions (PDO
  - ODBC and DB2 (PDO) — ODBC and DB2
  - PostgreSQL (PDO) — PostgreSQL Funct
  - SQLite (PDO) — SQLite Functions (PDO
  - 4D (PDO) — 4D Functions (PDO_4D)



PHP PDO SELECT

https://www.w3schools.com/php/php_mysql_select.asp



be01.php

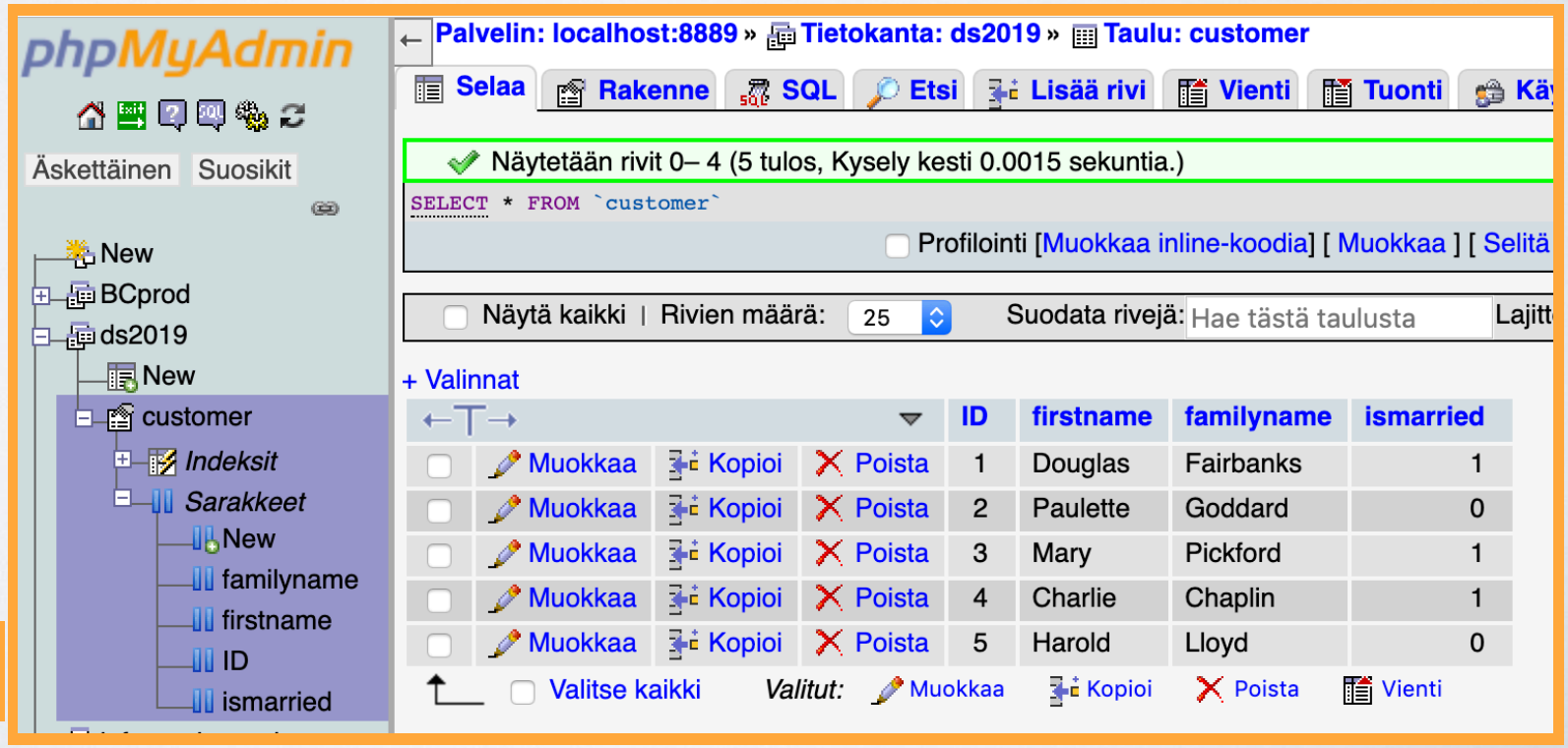
```
<?php
// https://alexwebdevelop.com/php-json-backend/
 function soslow(int $nmbrin) {
  // https://www.php.net/manual/en/function.microtime.php
  $start = microtime(true);
  $sum=0;
  for ($i=0;$i<=$nmbrin;$i++){
    $sum += $i;
  };
  $time_elapsed_secs = microtime(true) - $start;
  $time=sprintf("%f",$time_elapsed_secs);
  return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
  }

  function faster(int $tocount) {
    $start = microtime(true);
    $sum=0;
    $hesu = ($tocount+1) ;
// PHP integer division intdiv()-> https://secure.php.net/manual/en/function.intdiv.php
    $sum=((intdiv($tocount,2))*$hesu) + (($tocount%2)*(intdiv($hesu,2)));
    $time_elapsed_secs = microtime(true) - $start;
    $time=sprintf("%f",$time_elapsed_secs);
    return ('{"sum":"'.$sum.'","time":"' . $time . '"}');
  }

// https://www.w3schools.com/php/php_superglobals.asp
//  echo 'given number was -> ' . $_GET['number'] ;
  $in=$_GET['number'];


  $jsonStr  = '{"testing":{"input":"' . $in . '","output":[';
  $jsonStr .= soslow($in);
  $jsonStr .= ',';
  $jsonStr .= faster($in);
  $jsonStr .= ']}}' ;

  header('Content-Type: application/json');  //json backend header
  echo $jsonStr;
?>
```

fe01.html

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Syötä positiivinen luku yläpuolella olevaan kenttään!</p>

  <script>
    function loadDoc() {
      numberValue = document.getElementById("number").value;
      URL="http://localhost:8888/datastructures/backend/be01.php?number="+numberValue;
      var xhr = new XMLHttpRequest();
      xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
          let jsonobj = JSON.parse(xhr.responseText);
          let outPutText="Syötteellä " + jsonobj.testing.input + "<br><br>" +
              "algoritmi #1 antoi tulokseksi " + jsonobj.testing.output[0].sum +
              " ajassa " + jsonobj.testing.output[0].time + ".<br><br>" +
              "Algoritmi #2 antoi tulokseksi " + jsonobj.testing.output[1].sum +
                        " ajassa " + jsonobj.testing.output[1].time + "<br><br>";
          document.getElementById("demo").innerHTML = outPutText;
        }
      };
      xhr.open("GET", URL, true);
      xhr.send();
    }
  </script>

</body>
</html>
```

Structure

DATA BASE



Data

fe02.html

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<body>
  <h2>The XMLHttpRequest Object</h2>
  <input id="number">
  <button type="button" onclick="loadDoc()">Request data</button>
  <p id="demo">Write actors id into field above!</p>

<script>
  function loadDoc() {
    numberValue = document.getElementById("number").value;
    URL="http://localhost:8888/datastructures/backend/be02.php?actorid="+ numberValue;
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
      if (xhr.readyState == 4 && xhr.status == 200) {
        let outPutText = xhr.responseText;
        document.getElementById("demo").innerHTML = outPutText;
      }
    };
    xhr.open("GET", URL, true);
    xhr.send();
  }
</script>

</body>
</html>
```

be02.php on next slide >>

be02.php

```php
<?php
// https://www.w3schools.com/php/php_superglobals.asp
//  echo 'given actorid was -> ' . $_GET['number'] ;
  $in=$_GET['actorid'];
  $servertype = "mysql";
  $servername = "localhost";
  $username   = "ds2019";
  $password   = "TopSecret";
  $dbname     = "ds2019";

  try {
      $conn = new PDO("$servertype:host= $servername;dbname= $dbname", $username, $password);
      // set the PDO error mode to exception
      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
      $conn = $conn->prepare("SELECT ID, firstname, familyname, ismarried FROM `customer` WHERE ID=".$in);
      $conn->execute();

      // set the resulting array to associative
      $result = $conn->setFetchMode(PDO::FETCH_ASSOC);

      foreach ($conn->fetchAll() as $k=>$v){ // Associative Array https://www.w3schools.com/php/php_arrays.asp
        $booleanIsMarried = (1  == $v['ismarried']) ? 'true' : 'false' ;  // ternary operator
        $output= '{"firstname":"'.$v['firstname'].'","familyname":"'.$v['familyname'].'","ismarried":"'. $booleanIsMarried .'"}';
      }
  }
  catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
  }
header('Content-Type: application/json');  //json backend header
echo $output;
?>
```