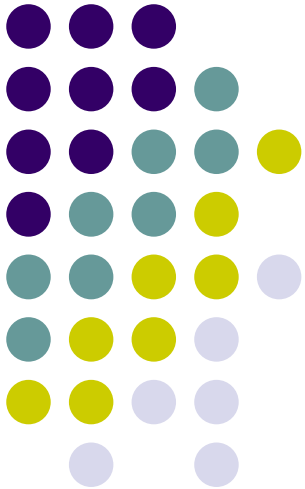
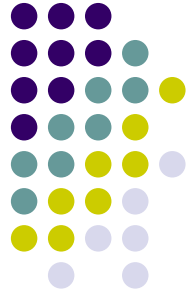
Εφαρμογές στον Παγκόσμιο Ιστό

Προγραμματίζοντας για τον παγκόσμιο ιστό (PHP + MySQL)

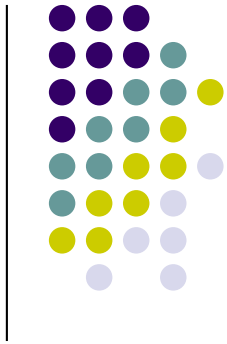


Εισαγωγή



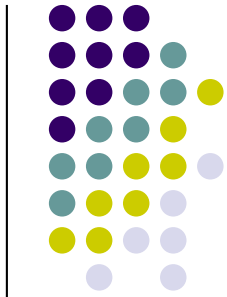
- Απαιτούνται Βασικές Γνώσεις
  - Από το πρώτο μέρος της παρουσίασης
  - Λειτουργία Φορμών
  - Βάσεις Δεδομένων – Γλώσσα SQL

E-Mail



- Λειτουργίες του E-Mail:
  - Η PHP δίνει δυνατότητα αποστολής email μέσω SMTP πρωτοκόλλου
  - Για τη χρήση της default mail() συνάρτησης
    - Τον SMTP server που θα χρησιμοποιήσουμε τον ορίζουμε στο αρχείο php.ini.
  - Σε production περιβάλλοντα συνήθως η mail() ΔΕ χρησιμοποιείται
    - Εναλλακτικά: https://github.com/Synchro/PHPMailer
    - SMTP authentication, attach, HTML mails κ.α.
    - Το email μας δεν κινδυνεύει να χαρακτηριστεί spam

E-Mail



.PHP

```
<?
$message="Text to send through mail";

mail("nobody@example.com", "Subject Text", $message,"From:
    webmaster@example.com");
?>
```

Απλό παράδειγμα με χρήση της mail()

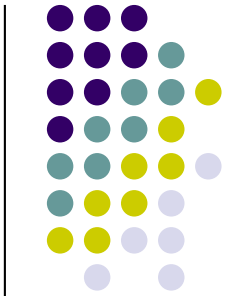
File upload

- Μπορώ να υλοποιήσω για τους χρήστες μου upload αρχείων στον server
- Συνάρτηση move_uploaded_file ()
- Μεταβλητή $_FILES
- php.ini

Παράδειγμα

upload_max_filesize	2M	2M
upload_tmp_dir	c:/wamp/tmp	c:/wamp/tmp
file_uploads	On	On

File Upload – Θέματα ασφάλειας



PHP μέρος 1ο

μπορούν να τις δουν όλοι οι browsers

✓ «Μειονέκτημα»: Server side κώδικας με bugs → Θέματα ασφάλειας (hacked)

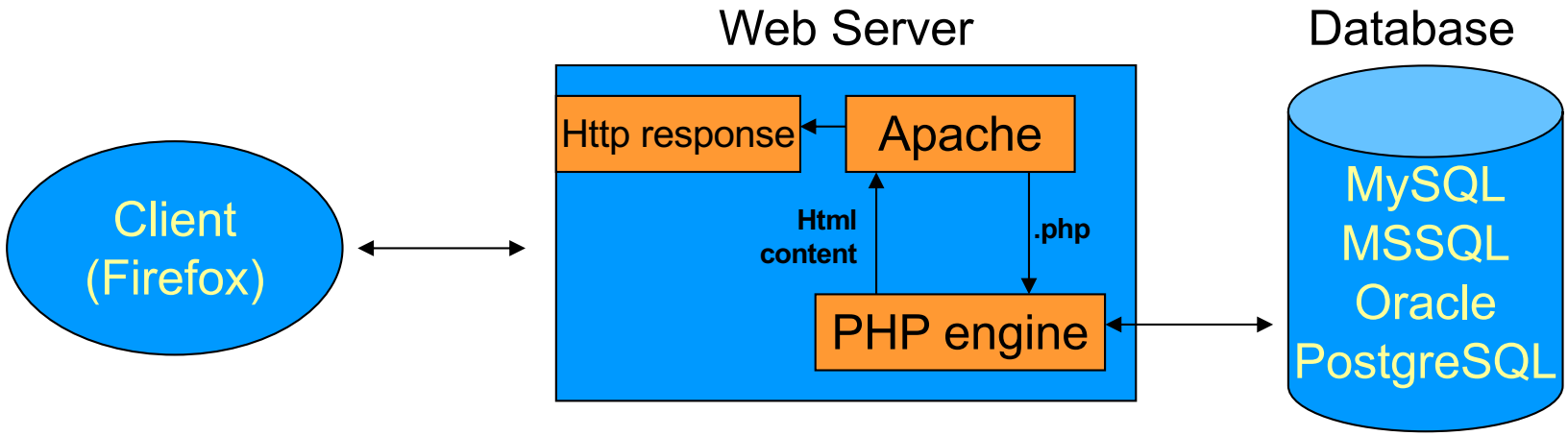
- File upload μόνο αν είναι απαραίτητο (π.χ. εγγεγραμμένοι χρήστες)
- Αποθήκευση των αρχείων εκτός www root
- Αλλαγή ονόματος αρχείου
- Αποφυγή χρήσης συναρτήσεων include για uploaded/ remote files

Warning

Security warning

Remote file may be processed at the remote server (depending on the file extension and the fact if the remote server runs PHP or not) but it still has to produce a valid PHP script because it will be processed at the local server. If the file from the remote server should be processed there and outputted only, readfile() is much better function to use. Otherwise, special care should be taken to secure the remote script to produce a valid and desired code.

Διασύνδεση με ΒΔ



• Για να συνδεθεί η PHP με την MySQL χρησιμοποιούμε τις συναρτήσεις της βιβλιοθήκης **MySQL Improved Extension**

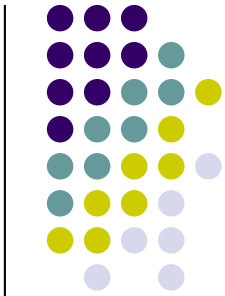
- Object oriented
- Procedural

• Εναλλακτική αποτελεί η βιβλιοθήκη PDO_MYSQL

• Η βιβλιοθήκη MySQL που τυχόν θα δείτε σε online παραδείγματα είναι πλέον "*deprecated*"

- *Τι σημαίνει αυτό;*

Mysql_connect (OO)



.PHP

```
$mysql_link = new mysqli('localhost', ‘user', ‘pass', ‘db_name');

if ($mysql_link->connect_error)
{
    die('Connect Error (' . $mysql_link->connect_errno . ') ‘.
                            $mysql_link->connect_error);
}
echo "Success!";

$mysql_link->close();
```

- Το connection κλείνει αυτόματα με το τέλος του script.
- Είναι όμως καλή πρακτική να το τερματίζουμε με την χρήση της mysqli_close

Mysql_connect (Procedural)

.PHP

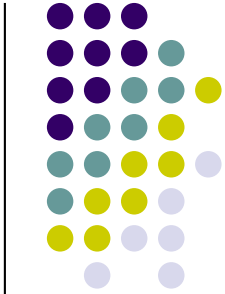
```
$mysql_link = mysqli_connect('localhost', ‘user', ‘pass', ‘db_name');

if (mysqli_connect_errno())
{
    die('Connect Error (' . mysqli_connect_errno() . ') '.
                            mysqli_connect_error());
}
echo "Success!";

mysqli_close($mysql_link);
```

- Το connection κλείνει αυτόματα με το τέλος του script.
- Είναι όμως καλή πρακτική να το τερματίζουμε με την χρήση της mysqli_close

mysql_query



.PHP

```
$my_query = "INSERT into test_table (name) VALUES ('".$_POST['name']."')";

$result = $mysql_con->query($my_query);

if (!$result)
    die('Invalid query: ' . $mysql_con->error);
else
    echo "Updated records: ".$mysql_con->affected_rows;

echo "<p><b>The last id: ".$mysql_con->insert_id."</b></p>";
```

Select

.PHP

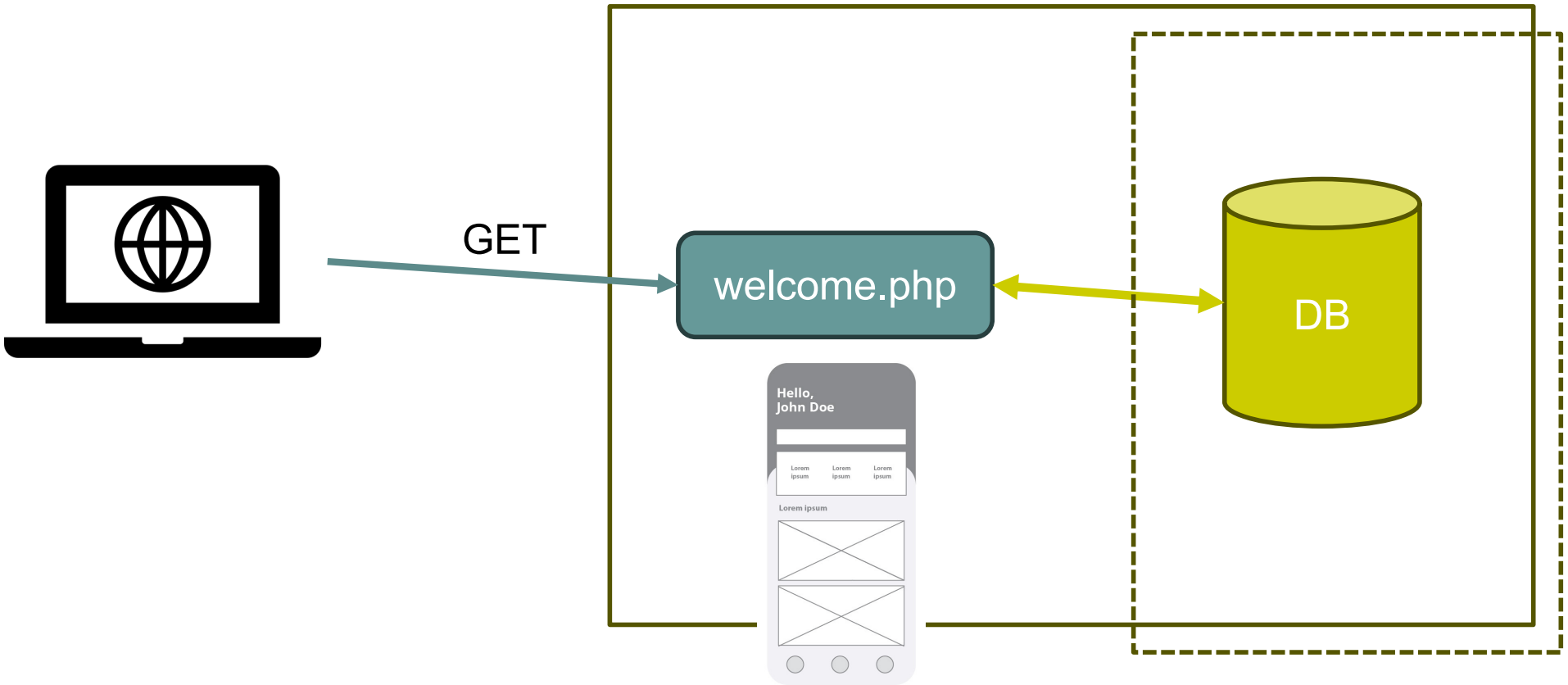
```
while($row = $result->fetch_array())
{
  echo "<tr>";
  echo "<td>" . $row['id'] . "</td>";
  echo "<td>" . $row['name'] . "</td>";
  echo "</tr>";
}
echo "</table>“;
```

- Εκτός της μεθόδου fetch_array μπορεί να χρησιμοποιηθεί και η fetch_row. Χάνουμε όμως αρκετή πληροφορία χωρίς να κερδίζουμε σε ταχύτητα.
- Η fetch_array έχει επιπλέον ένα προαιρετικό όρισμα με τιμές MYSQLI_ASSOC, MYSQLI_NUM, MYSQLI_BOTH

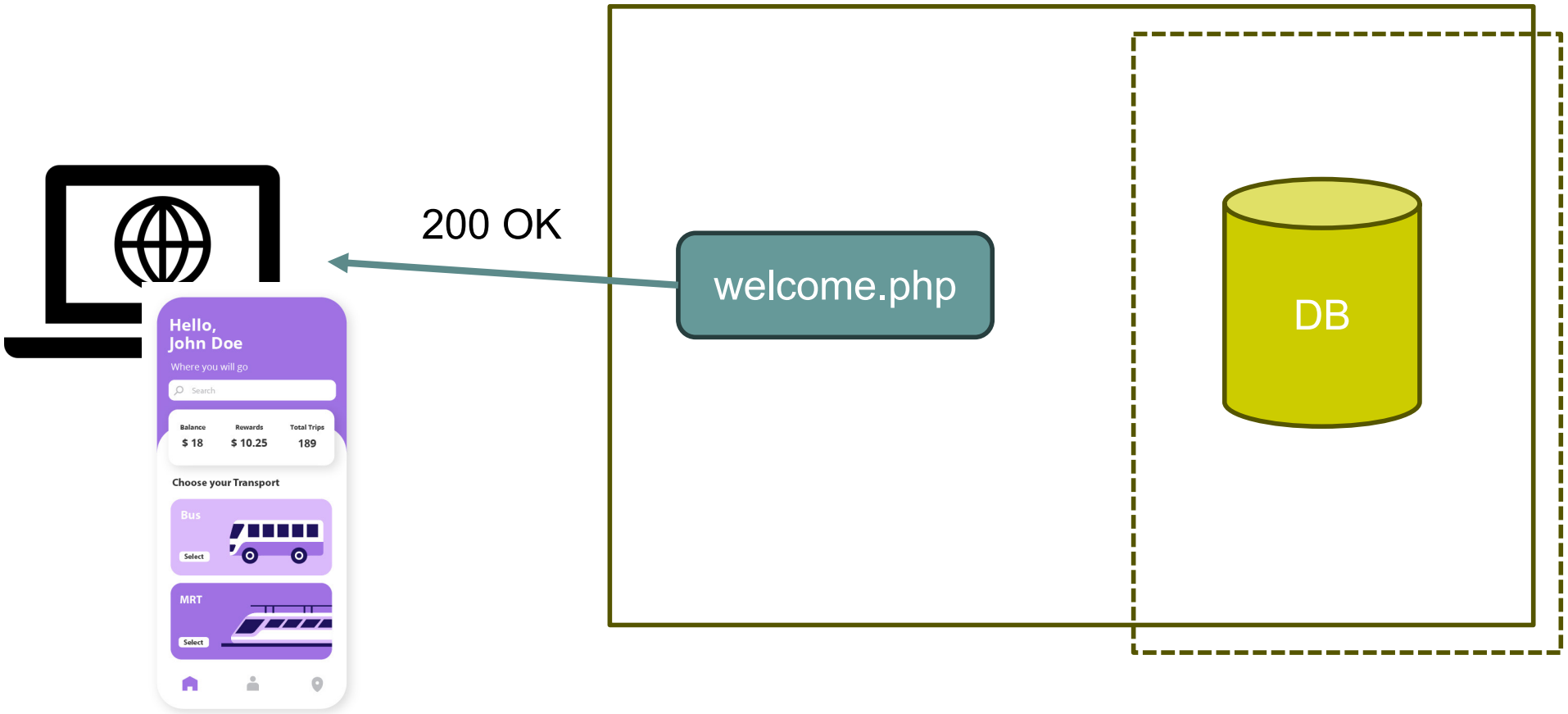
Υποστήριξη Ελληνικών

- UTF-8 παντού!
  - τα .php αρχεία
  - το encoding στη σελίδα που υπάρχει η φόρμα
  - το encoding στη σελίδα που εκτελούνται τα queries
  - το connection μεταξύ php και mysql
  - η βάση, τα tables και τα string πεδία των tables

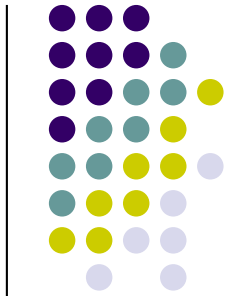
Αρχιτεκτονική



Αρχιτεκτονική

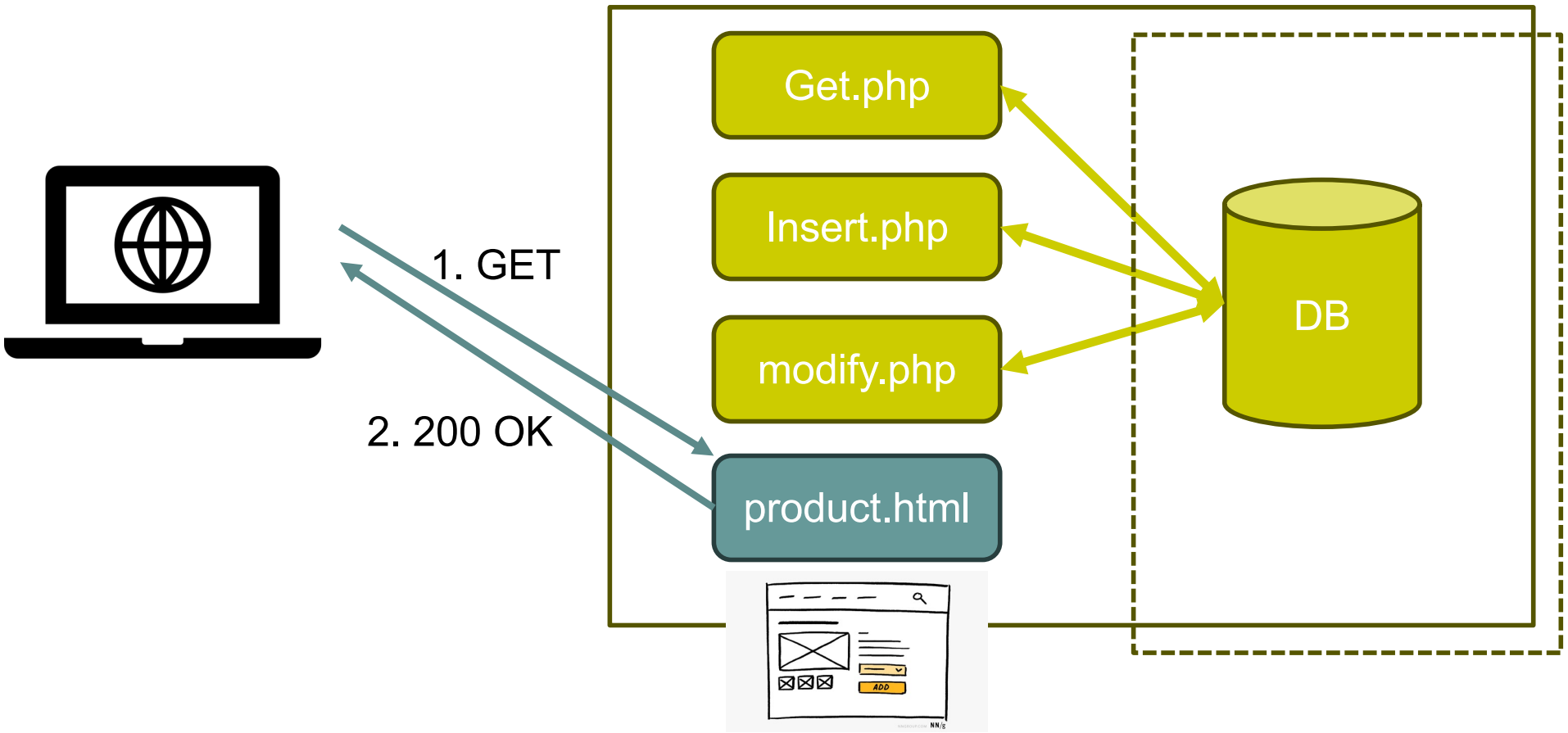


Αρχιτεκτονική

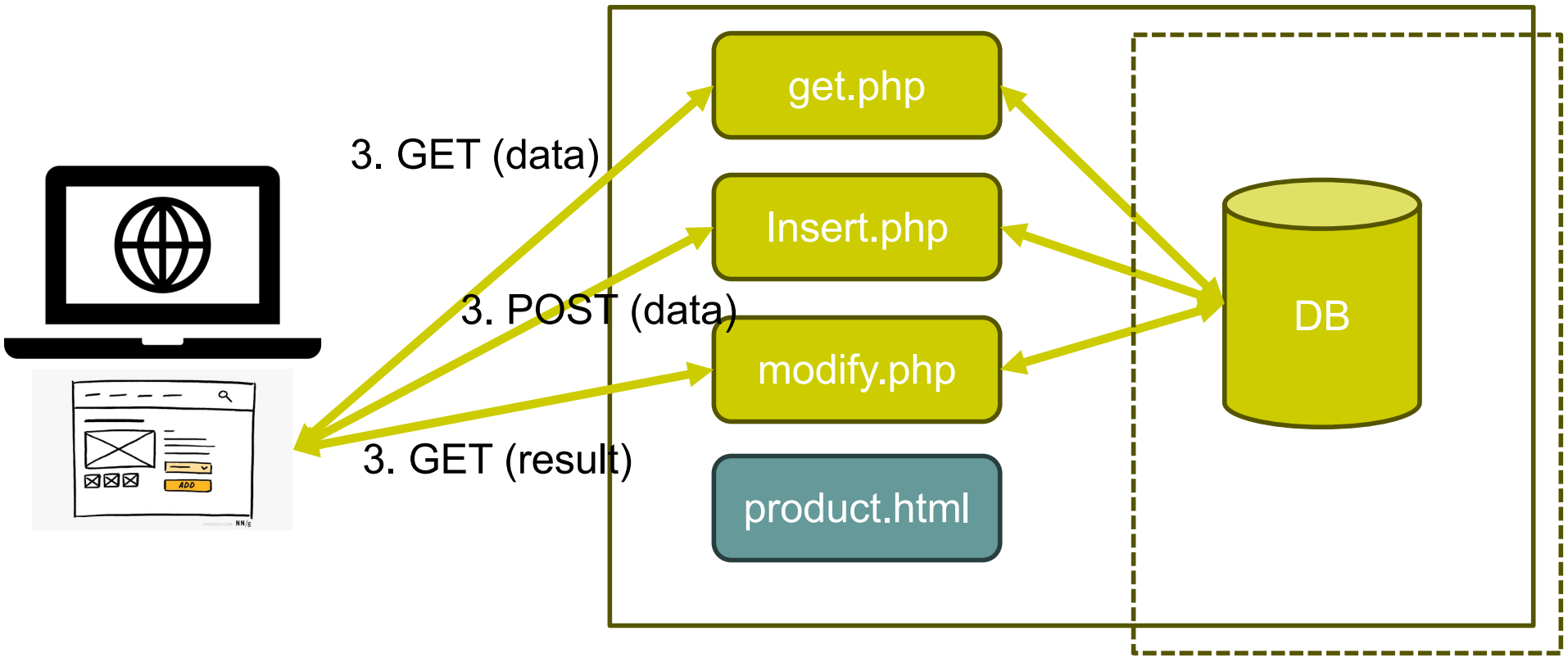


- Ιδανικά θέλουμε:
  - Όλη η HTML/JS να φορτώνεται ΜΙΑ φορά στον πελάτη
  - Τα ΔΕΔΟΜΕΝΑ να διατηρούνται αποκλειστικά στη ΒΔ
  - Κάθε σύνδεση πελάτη-εξυπηρέτη να αφορά μόνο μεταφορά ΔΕΔΟΜΕΝΩΝ και όχι τη μορφοποίησή τους.

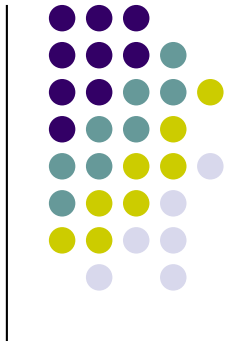
Αρχιτεκτονική



Αρχιτεκτονική

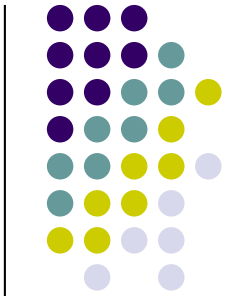


Αρχιτεκτονική



- Πελάτης
  - HTML με «στατικές» και «δυναμικές» περιοχές (placeholders)
    - Δυναμικές: Αυτές για τις οποίες δεν έχουν έρθει ακόμα τα δεδομένα, θα φορτωθούν στη συνέχεια.
  - Όλη η JS που ελέγχει τη μεταφορά δεδομένων, την επεξεργασία τους (pre-/post-), και τον τρόπο επίδειξής τους στον πελάτη.

Αρχιτεκτονική

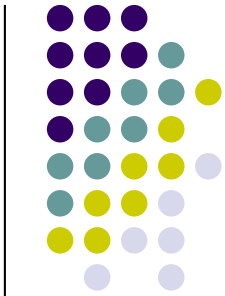


- Εξυπηρέτης – middleware
  - PhP ή άλλη γλώσσα που:
    - Δέχεται αιτήματα για δεδομένα ή αιτήματα για αποθήκευση/τροποποίηση δεδομέμων από πελάτες
    - Λαμβάνει τα δεδομένα από τη ΒΔ ή τοποθετεί/τροποποιεί τα δεδομένα σε αυτή
    - Επιστρέφει στον πελάτη τα αποτελέσματα των αιτημάτων με μορφή που μπορεί να τα αναγνώσει μηχανικά (JSON, XML, κ.α.)
      - Δεδομένα, status codes/messages

Αρχιτεκτονική

- Εξυπηρέτης – ΒΔ
  - Δέχεται queries από το middleware
  - Επιστρέφει αποτελέσματα σε αυτό

Αρχιτεκτονική



- Η αρχιτεκτονική αυτή ονομάζεται REST
  - Representational State Transfer
  - RESTful architecture / RESTful APIs
- Η PhP λειτουργεί ως Application Programming Interface (API) με πολλαπλά endpoints
  - API: μια «πύλη» προς τα **δεδομένα**
  - Ελεγχόμενη και τυποποιημένη πρόσβαση στα **δεδομένα** από τρίτα συστήματα (όχι μόνο τα δικά μας)
  - Μπορεί να υλοποιεί μεθόδους ταυτοποίησης χρήστη για περισσότερη ασφάλεια
  - Μπορεί να υλοποιεί caching για καλύτερη αποδοτικότητα

Υλοποίηση endpoint (get data)

```php
<?php

$servername = "localhost";
$username = "usidas";
$password = "mypassword";
$dbname = "web_project2023";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

$status=[];

// Check connection
if ($conn->connect_error) {
  $status["code"]=0;
  $status["message"]=$conn->connect_error;
}


$sql = "select * from items inner join categories on items.category_id = categories.id order by items.date_added desc limit 0,50";
if ($result->num_rows >= 0) {
  $status["code"]=1;
  $status["message"]="Items retrieved successfully";
  $status["items"]=[];
  $result = $conn->query($sql);
    while($row = $result->fetch_assoc()) {
      array_push($status["items"], $row);
    }

} else {
  $status["code"]=0;
  $status["message"]=$conn->error;
}

header("Content-Type: application/json");

echo json_encode($status);
?>
```

Connect to DB

Initialise response array

Form response status messages

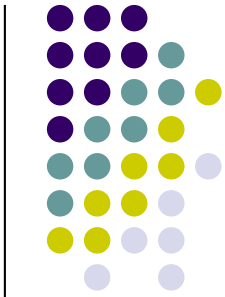
Execute query

Put results into php array

Set response header

Convert and return php array to JSON

Αποτέλεσμα




```
{
    code: 1,
    message: "Items retrieved successfully",
  - items: [
      - {
            id: "6",
            item_name: "Coca Cola",
            category_id: "6",
            date_added: "2023-11-06 21:58:12",
            category_name: "Beverages"
        },
      - {
            id: "6",
            item_name: "water ",
            category_id: "6",
            date_added: "2023-11-06 21:58:12",
            category_name: "Beverages"
        },
      - {
            id: "9",
            item_name: "t22",
            category_id: "9",
            date_added: "2023-11-07 15:52:29",
            category_name: "2d hacker"
        },
      - {
            id: "10",
            item_name: "",
```

Κατανάλωση

Async function to fetch & show data

We will place data here

```
<body onload="get()">
  <h1> Emergency items database </h1>
  <hr>
  <a href="index.php">Home</a> | <a href="export.php">Export items</a> | <a href="ac
  <hr>
  <h2> Add new item </h2>
  <form method="post" action="" id="item_form">
    <h3>New Item name</h3>
    <table>
      <tr>
        <td> Item name </td>
        <td> <input type="text" name="itname" id="itname" required> </td>
      </tr>
      <tr>
        <td> Item category </td>
        <td> <select name="itcats" id="itcats" required/> </td>
      </tr>
    </table>
    <h3>Item Details</h3>
    <table id = 'itemdetails'>
      <!--<tr>
        <td> Item detail name (e.g. "volume") </td>
        <td> <input type="text" name="itdname1" id="itdname1" required> </td>

        <td> Item detail value (e.g. "500ml") </td>
        <td> <input type="text" name="itdvalue1" id="itdvalue1" required> </td>
      </tr>-->
    </table>
    <input type="button" value="Add more details" onclick="add_detail()"/><br><br>
    <input type="button" value="Submit to DB" onclick="send()"/>
  </form>
  <hr>
    <div id="results">[Operation Status]</div>
  <hr>
  <h2> Current items in DB (latest 50) </h2>
  <table id ="item_table">
  </table>


</body>
```

Κατανάλωση

Call API

Create in-memory JSON object from response

Use response data to populate DOM

```
function get(){
  add_detail();
  get_cats();

  var table = document.getElementById("item_table");
  table.innerHTML="";

  var xhr = new XMLHttpRequest();

  xhr.open('GET','get_items.php');
  xhr.send();
  xhr.onreadystatechange = function() {
      if (xhr.readyState == XMLHttpRequest.DONE) {
          data = JSON.parse(this.responseText)['items'];
          //console.log(this.responseText);
          var header = table.createTHead();
          var row = header.insertRow(0);
          var cell1 = row.insertCell(0);
          var cell2 = row.insertCell(1);
          var cell3 = row.insertCell(2);
          cell1.innerHTML = "Item name";
          cell2.innerHTML = "Category";
          cell3.innerHTML = "Date Added";
          var tbody = table.createTBody();

          for (i=0;i<data.length;i++){

            var row = tbody.insertRow(-1);

            // Insert new cells (<td> elements) at the 1st and 2nd position of the "new" <tr> element:
            var cell1 = row.insertCell(0);
            var cell2 = row.insertCell(1);
            var cell3 = row.insertCell(2);

            // Add some text to the new cells:
            cell1.innerHTML = data[i]['item_name'];
            cell2.innerHTML = data[i]['category_name'];
            cell3.innerHTML = data[i]['date_added'];
          }
      }

    }
  }
```

Τελικό αποτέλεσμα

[Operation Status]

Current items in DB (latest 50)

Item name	Category	Date Added
Coca Cola	Beverages	2023-11-06 21:58:12
water	Beverages	2023-11-06 21:58:12
t22	2d hacker	2023-11-07 15:52:29
		2023-11-07 15:52:30
		2023-11-07 15:52:30
		2023-11-07 15:52:30
		2023-11-07 15:52:30
		2023-11-07 15:52:30
		2023-11-07 15:52:30
Glass	Kitchen Supplies	2023-11-18 18:27:29
Pan	Kitchen Supplies	2023-11-18 18:27:29
Paring knives	Kitchen Supplies	2023-11-18 18:27:29
Pots	Kitchen Supplies	2023-11-18 18:27:29
Dishes	Kitchen Supplies	2023-11-18 18:27:29
Boots	Clothing	2023-11-06 21:58:27
Pants	Clothing	2023-11-06 21:58:27