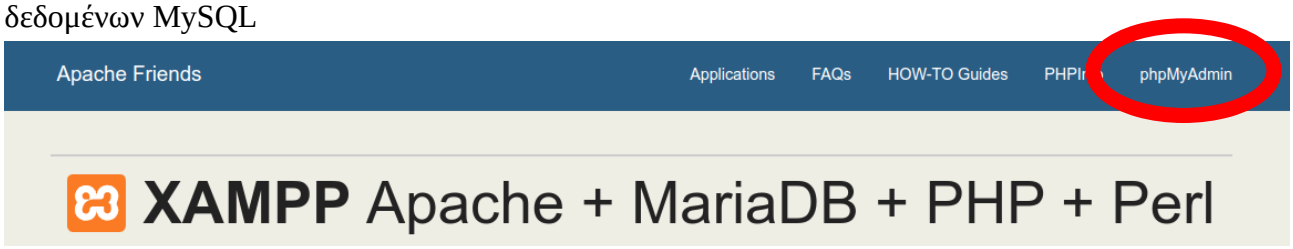
Walkthrough: Διάλεξη 08 - Κλήση απομακρυσμένων λειτουργιών (DB)

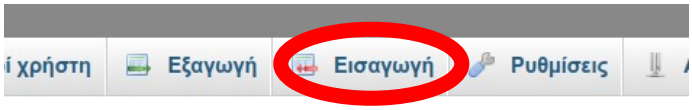
Στόχος μας είναι η δημιουργία μίας εφαρμογής η οποία θα συνδέεται σε μία βάση δεδομένων μέσω υπηρεσιών και http, θα ανακτά δεδομένα και θα τα εμφανίζει.

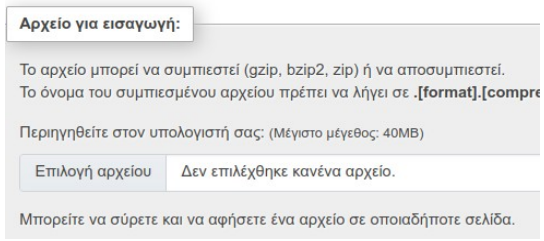
Εισαγωγή δεδομένων και σχήματος βάσης δεδομένων

Αφού εκκινήσουμε το xampp, και συγκεκριμένα τους διακομιστές Apache και MySQL, μεταβαίνουμε στη διεύθυνση http://localhost/ και επιλέγουμε το σύνδεσμο πάνω δεξιά, phpMyAdmin, ώστε να μεταβούμε στο ομώνυμο διαδικτυακό εργαλείο διαχείρισης βάσεων δεδομένων MySQL



Με την έναρξη του phpMyAdmin, επιλέγουμε “Βάσεις Δεδομένων” και στην επόμενη σελίδα που θα ανοίξει επιλέγουμε **“Εισαγωγή”**. Στη σελίδα που θα ανοίξει, στο πεδίο “Αρχείο για εισαγωγή” επιλέγουμε το αρχείο cars.sql.txt, που έχουμε μεταφέρει στον υπολογιστή μας από





το openeclass (Αρχικός κατάλογος » Διάλεξη 08 - Κλήση απομακρυσμένων λειτουργιών (DB) » 02. Δημιουργία Βασης Δεδομένων Cars). Έπειτα, επιλέγουμε το πλήκτρο **“Εισαγωγή”** από το κάτω μέρος της σελίδας. Το αποτέλεσμα είναι η δημιουργία της βάσης δεδομένων cars, που αποτελείται από δύο πίνακες: τον πίνακα history, όπου θα καταγράφεται το ιστορικό ενεργειών στην εφαρμογή, και τον πίνακα models, ο οποίος περιέχει πληροφορίες για τα οχήματα που θα εμφανίζονται στην εφαρμογή.

Δημιουργία υπηρεσιών

Επόμενο βήμα μας είναι η δημιουργία υπηρεσιών σε http, οι οποίες θα επικοινωνούν μέσω http requests με την εφαρμογή, μεταφέροντας δεδομένα σε μορφή json και θα διαβάσουν και θα γράφουν στη βάση δεδομένων. Θα χρησιμοποιήσουμε PHP για τη δημιουργία των υπηρεσιών. Στο φάκελο htdocs δημιουργούμε έναν φάκελο carsDBServices και μέσα σε αυτόν, δύο αρχεία, ένα για κάθε υπηρεσία. Έτσι, έχουμε τα παρακάτω αρχεία:

Αρχείο logHistory.php που χρησιμοποιείται για την καταγραφή ιστορικού:

```php
<?php
	$data = array();
	$brand = $_GET["brand"];
	$model = $_GET["model"];
	$timestamp = $_GET["timestamp"];

	$host="localhost";
	$uname="root";
	$pass="";
	$dbname="cars";
```

```
        $dbh = mysqli_connect($host,$uname,$pass) or die("cannot connect");
        mysqli_select_db($dbh, $dbname);

        $sql = "INSERT into history values('" . $brand . "','" . $model . "','" .
$timestamp . "')";
        echo $sql;
        mysqli_query($dbh, $sql);
        mysqli_close($dbh);
?>
```

και αρχείο populateDropDown.php, που χρησιμοποιείται για να παρέχει δεδομένα στα οπτικά στοιχεία της εφαρμογής:

```
<?php
        $data = array();

        $host="localhost";
        $uname="root";
        $pass="";
        $dbname="cars";

        $dbh = mysqli_connect($host,$uname,$pass) or die("cannot connect");
        mysqli_select_db($dbh, $dbname);

        $sql = "SELECT brand, GROUP_CONCAT(model) AS grouped_models FROM models GROUP BY
brand";
        $result = mysqli_query($dbh, $sql);
        while ($row = mysqli_fetch_array($result)) {
                $data[$row['brand']] = $row['grouped_models'];
        }

        header("Content-Type: application/json");
        echo json_encode($data);
        mysqli_close($dbh);
?>
```

ΠΡΟΣΟΧΗ! Πιθανόν να χρειαστεί να αλλάξετε τα στοιχεία χρήστη (`$uname`,`$pass`) για να λειτουργήσουν οι υπηρεσίες.

Για να ελέγξουμε ότι λειτουργούν οι υπηρεσίες, επισκεπτόμαστε είτε από browser είτε με χρήση Postman το http://localhost/carsDBServices/populateDropDown.php

Αν όλα λειτουργούν σωστά, θα πρέπει να εμφανίζεται το παρακάτω:

```
{"Nissan":"Sunny","Toyota":"Auris,Aygo,Corrolla,Yaris","VW":"Golf,Polo"}
```

Δημιουργία frontend

Για τη δημιουργία του frontend θα βασιστούμε σε προηγούμενη υλοποίηση την οποία βρίσκουμε στην εξής τοποθεσία:

Αρχικός κατάλογος » Διάλεξη 05 - Τοπική Αποθήκευση » Ενδεικτική Λύση

Προετοιμασία

Σε αυτή την υλοποίηση θα προσπελάσουμε το διαδίκτυο και θα χρησιμοποιήσουμε μία εξωτερική βιβλιοθήκη για να πραγματοποιήσουμε τις κλείσεις http. Για το λόγο αυτό θα πρέπει να προχωρήσουμε σε αλλαγές στα παρακάτω αρχεία:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.carpicker">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CarPicker"
        android:networkSecurityConfig="@xml/network_security_config"
        android:usesCleartextTraffic="true">

        <meta-data
            android:name="com.google.android.actions"
            android:resource="@xml/network_security_config" />

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

network_security_config.xml

Δημιουργία αρχείου network_security_config.xml. Δημιουργούμε φάκελο xml εντός του φακέλου res και εντός του φακέλου xml δημιουργούμε αρχείο network_security_config.xml. Δεξί κλικ: New-→ Folder → XML Resources Folder

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">192.168.1.5</domain>
    </domain-config>
</network-security-config>
```

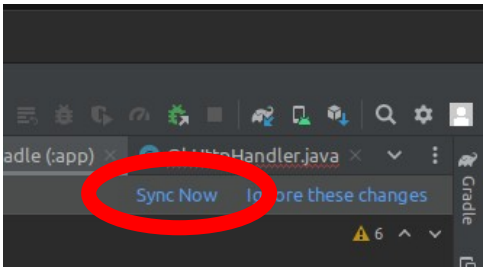
Προσοχή! Η διεύθυνση IP που βλέπετε παραπάνω θα πρέπει να αντικατασταθεί από τη διεύθυνση ip του υπολογιστή στο οποίο τρέχουν οι υπηρεσίες στις οποίες θα συνδεθούμε. Αν τρέχουν στον ίδιο υπολογιστή, βρίσκουμε την IP διεύθυνσή του (π.χ. μέσω ipconfig από γραμμή εντολών).

build.gradle

Στο build.gradle (Module App) κάνουμε την παρακάτω προσθήκη:

```
implementation 'com.squareup.okhttp3:okhttp:4.9.0'
```

Δεν ξεχνάμε να επιλέξουμε Sync Now.



Κώδικας Java

Δημιουργία κλάσης OkHttpHandler.java

```
package com.example.carpicker;

import android.os.*;
import org.json.*;
import java.util.*;
import okhttp3.*;

public class OkHttpHandler {

    public OkHttpHandler() {
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
    }

    ArrayList<CarBrand> populateDropDown(String url) throws Exception {
        ArrayList<CarBrand> cbList = new ArrayList<>();
        OkHttpClient client = new OkHttpClient().newBuilder().build();
        RequestBody body = RequestBody.create("",
MediaType.parse("text/plain"));
        Request request = new Request.Builder().url(url).method("POST",
body).build();
        Response response = client.newCall(request).execute();
        String data = response.body().string();
        //System.out.println("My Response: " + data);
        try {
            JSONObject json = new JSONObject(data);
            Iterator<String> keys = json.keys();
            while(keys.hasNext()) {
                String brand = keys.next();
                String models = json.get(brand).toString();
                cbList.add(new CarBrand(brand, models));
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }

        return cbList;
    }

    public void logHistory(String url) throws Exception {
        OkHttpClient client = new OkHttpClient().newBuilder().build();
        RequestBody body = RequestBody.create("",
MediaType.parse("text/plain"));
        Request request = new Request.Builder().url(url).method("POST",
body).build();
        Response response = client.newCall(request).execute();
        System.out.println("My Response: " + response);
    }
}
```

CarBrandList.java

Δημιουργούμε έναν νέο constructor για να χρησιμοποιήσουμε τη νέα κλάση:

```
public CarBrandList(String ip) {
    String url= "http://"+ip+"/carsDBServices/populateDropDown.php";

    try {
        OkHttpHandler okHttpHandler = new OkHttpHandler();
        cbList = okHttpHandler.populateDropDown(url);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

MainActivity.java

Πραγματοποιούμε τροποποιήσεις στο MainActivity.java ώστε να χρησιμοποιήσουμε τη νέα κλάση:

Προσθέτουμε ως attribute το παρακάτω (Προσοχή! Βάζετε τη διεύθυνση του διακομιστή όπου εκτελούνται οι υποηρεσίες):

```
private final String myIP = "192.168.1.69";
```

Αντικαθιστώ τον constructor της CarBrandlist στην onCreate()

```
cbl = new CarBrandList(myIP);
```

και την OnCheckedChange()

```
public void onCheckedChanged(RadioGroup radioGroup, int checkedId) {
    RadioButton rb = (RadioButton) findViewById(checkedId);
    String url= "http://"+myIP+"/carsDBServices/logHistory.php?brand=" + brand +
"&model=" + rb.getText().toString() + "&timestamp=" + new
Date(System.currentTimeMillis()).toString();
    try {
        OkHttpHandler okHttpHandler = new OkHttpHandler();
        okHttpHandler.logHistory(url);
        Toast.makeText(getApplicationContext(), "Selection Logged",
Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```