

```
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);
    }

    /**
     * Changed this method to also parse image urls
     * Warning! URLs should not contain commas!
     * μια μέθοδο με το όνομα populateDropDown που επιστρέφει μια ArrayList αντικειμένων
CarBrand.
     */
    ArrayList<CarBrand> populateDropDown(String url) throws Exception {

        // Αρχικοποιεί μια κενή ArrayList για την αποθήκευση των αντικειμένων CarBrand.
        ArrayList<CarBrand> cbList = new ArrayList<>();

        // Δημιουργεί αντικείμενο OkHttpClient, το οποίο είναι μια δημοφιλής βιβλιοθήκη
πελατών HTTP
        // για την υποβολή αιτημάτων HTTP.
        OkHttpClient client = new OkHttpClient().newBuilder().build();

        // Δημιουργεί ένα σώμα αιτήματος με ένα κενό φορτίο συμβολοσειράς και έναν τύπο
πολυμέσων "κείμενο/απλό".
        RequestBody body = RequestBody.create("", MediaType.parse("text/plain"));

        // Δημιουργεί ένα αίτημα HTTP χρησιμοποιώντας το παρεχόμενο URL και ορίζει τη μέθοδο
αιτήματος ως "POST"
        // με το σώμα αιτήματος που δημιουργήθηκε προηγουμένως.
        Request request = new Request.Builder().url(url).method("POST", body).build();

        // Στέλνει το αίτημα HTTP και ανακτά την απάντηση.
        Response response = client.newCall(request).execute();

        // Εξάγει το σώμα απόκρισης ως συμβολοσειρά.
        String data = response.body().string();

        // Εκτυπώνει τα δεδομένα απόκρισης στην κονσόλα για σκοπούς εντοπισμού σφαλμάτων.
        System.out.println("My Response: " + data);

        // Στη συνέχεια, ο κώδικας εισάγει ένα μπλοκ δοκιμής για να αναλύσει τα δεδομένα
απόκρισης ως αντικείμενο JSON.
        try {
            // Δημιουργεί μια παρουσία JSONObject περνώντας τη συμβολοσειρά δεδομένων
απόκρισης.
            JSONObject json = new JSONObject(data);
```

```
// Ανακτά έναν επαναλήπτη πάνω από τα κλειδιά του αντικειμένου JSON.
Iterator<String> keys = json.keys();

while(keys.hasNext()) {
    String brand = keys.next();

    // Στη συνέχεια εξάγει τις τιμές για τα κλειδιά "grouped_models" και "images"
    // από το αντικείμενο JSON για την τρέχουσα επωνυμία
    String models =
json.getJSONObject(brand).getString("grouped_models").toString();
    String images = json.getJSONObject(brand).getString("images").toString();

    // δημιουργεί ένα νέο αντικείμενο CarBrand χρησιμοποιώντας την ανακτημένη
    // επωνυμία, μοντέλα και εικόνες και το προσθέτει στη cbList
    cbList.add(new CarBrand(brand, models, images));
}
} catch (JSONException e) {
    e.printStackTrace();
}

// Επιστρέφει τη συμπληρωμένη ArrayList<CarBrand> που περιέχει τα δεδομένα επωνυμίας
// και μοντέλων που ανακτήθηκαν.
return cbList;
}

/*
    Αυτή η μέθοδος λαμβάνει ένα url παραμέτρου συμβολοσειράς που αντιπροσωπεύει ένα
    τελικό σημείο διεύθυνσης URL.
    Εκτελεί ένα αίτημα HTTP POST στην καθορισμένη διεύθυνση URL για την καταγραφή
    ορισμένων δεδομένων.
    Το αίτημα υποβάλλεται χωρίς κανένα φορέα αιτήματος. Η απάντηση από τον διακομιστή
    καταγράφεται στην κονσόλα.
    */
public void logHistory(String url) throws Exception {
    // Δημιουργεί αντικείμενο της κλάσης OkHttpClient, η οποία είναι υπεύθυνη για την
    // υποβολή αιτημάτων HTTP.
    OkHttpClient client = new OkHttpClient().newBuilder().build();

    // Δημιουργεί ένα κενό σώμα αιτήματος χρησιμοποιώντας τη μέθοδο RequestBody.create().
    // Η μέθοδος MediaType.parse() χρησιμοποιείται για τον καθορισμό του τύπου μέσου του
    // σώματος αιτήματος,
    // που σε αυτήν την περίπτωση είναι "κείμενο/απλό".
    RequestBody body = RequestBody.create("", MediaType.parse("text/plain"));

    // Δημιουργεί ένα αίτημα HTTP POST χρησιμοποιώντας το Request.Builder. Η διεύθυνση
    // URL ορίζεται στο παρεχόμενο url
    // και η μέθοδος αιτήματος ορίζεται σε "POST". Το κενό σώμα αιτήματος που
    // δημιουργήθηκε στο προηγούμενο βήμα
    // περιλαμβάνεται στο αίτημα.
    Request request = new Request.Builder().url(url).method("POST", body).build();

    // Στέλνει το αίτημα HTTP και λαμβάνει την απάντηση χρησιμοποιώντας το OkHttpClient.
    // Η μέθοδος execute() καλείται για να εκτελέσει τη σύγχρονη εκτέλεση της αίτησης.
    Response response = client.newCall(request).execute();
}
```

```
        // Καταγράφει το αντικείμενο απόκρισης στην κονσόλα. Το αντικείμενο απόκρισης
        περιέχει πληροφορίες σχετικά με την
        // απόκριση HTTP, όπως τον κωδικό κατάστασης, τις κεφαλίδες και το σώμα.
        System.out.println("My Response: " + response);
    }
}
```