

```
package com.example.carpicker;

import androidx.appcompat.app.*;

import android.content.Intent;
import android.net.*;
import android.os.*;
import android.view.*;
import android.widget.*;
import com.squareup.picasso.*;

import java.net.URL;
import java.util.*;

public class MainActivity extends AppCompatActivity {
    // Δημιουργώ αντικείμενο CarBrandList που θα το χρησιμοποιήσω σε κάποια μέθοδο
    private CarBrandList cbl;

    // ορίζουμε το RadioGroup εδώ για να το δούν όλες οι μέθοδοι.
    private RadioGroup rg;

    // ορίζουμε ένα αντικείμενο εικόνας για να το δούν όλες οι μέθοδοι.
    ImageButton myImage;

    // Ορίζω την IP μου για να μην την γράφω σε κάθε μέθοδο
    private final String myIP = "192.168.1.104";

    // ορίζουμε την μεταβλητή imageUrl που είναι η διεύθυνση της εικόνας που θα εμφανιστεί
    στην επόμενη οθόνη
    private String imageUrl;
    private String full_name;

    // Γράφουμε την μέθοδο για το τι θα κάνει το κουμπί όταν πατηθεί.
    public void PickCarOnClick(View v) {
        // Φτιάχνω ένα αντικείμενο spinner. Επιλεγμένο αντικείμενο από την λίστα
        Spinner dropDown = (Spinner) findViewById(R.id.cars);

        // Από το spinner παίρνω το επιλεγμένο αντικείμενο και το αποθηκεύω στο brand
        String brand = String.valueOf(dropDown.getSelectedItem());

        // Σύνδεση του GUI με την κλάση πρέπει να γίνει μέσα στην παρένθεση
        List<String> allModels = cbl.getAllModels(brand);

        // Πρέπει να περάσουμε τα μοντέλα στο GUI στο radiogroup
        rg.removeAllViews();

        // ορίζει την ορατότητα ενός αντικειμένου ImageView σε GONE (αόρατο)
        myImage.setVisibility(View.GONE);

        // Κατασκευή του radiobutton
        for(int i=0;i<allModels.size();i++) {
            RadioButton rb = new RadioButton(this); // Φτιάχνω ένα radio button ανάλογα με τι
            μέγεθος θα έχει η λίστα
            rb.setText(allModels.get(i)); // Προσθέτουν το κείμενο που πήρα από την λίστα ως
            κείμενο σε radiobutton
        }
    }
}
```

```
        rb.setId(100+i); // Βάζω ένα id γιατί πιθανό να το ψάξω με το id
        rg.addView(rb); // Το προσθέτω στο radiogroup
    }

    // Φτιάχνω έναν Listener που θα τον υλοποιήσω εδώ μέσα
    rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {

        /*
        αυτός ο κώδικας χειρίζεται το συμβάν όταν επιλέγεται ένα κουμπί επιλογής σε ένα
        radiogroup.
        Ανακτά το επιλεγμένο κουμπί επιλογής, κατασκευάζει μια συμβολοσειρά URL με την
        επιλεγμένη επωνυμία,
        μοντέλο και χρονική σήμανση, καλεί ένα αίτημα HTTP για να καταγράψει την επιλογή
        χρησιμοποιώντας
        το OkHttp και εμφανίζει ένα μήνυμα toast για να ειδοποιήσει τον χρήστη για την
        επιτυχή καταγραφή.
        */
        @Override
        public void onCheckedChanged(RadioGroup radioGroup, int checkedId) {

            // Ανακτά το επιλεγμένο RadioButton από το RadioGroup με βάση την παράμετρο
            checkedId,
            // η οποία αντιπροσωπεύει το αναγνωριστικό του επιλεγμένου κουμπιού επιλογής.
            RadioButton rb = (RadioButton) findViewById(checkedId);

            // κατασκευάζει μια συμβολοσειρά URL για ένα τελικό σημείο υπηρεσίας ιστού,
            // το οποίο περιλαμβάνει παραμέτρους για την επωνυμία, το μοντέλο και τη
            χρονική σήμανση.
            // Η τιμή της επωνυμίας θεωρείται ότι ορίζεται αλλού και η
            rb.getText().toString() ανακτά το κείμενο
            // του επιλεγμένου κουμπιού επιλογής, που αντιπροσωπεύει το μοντέλο.
            // Η χρονική σήμανση δημιουργείται χρησιμοποιώντας τη νέα
            Date(System.currentTimeMillis())
            // και μετατρέπεται σε συμβολοσειρά.
            String url= "http://" + myIP + "/multimediaDBServices/logHistory.php?brand=" +
            brand + "&model=" + rb.getText().toString() + "&timestamp=" + new
            Date(System.currentTimeMillis()).toString();

            // Ανακτά την εικόνα URI (ομοιόμορφο αναγνωριστικό πόρου) που σχετίζεται με
            την επιλεγμένη επωνυμία και μοντέλο.
            // Η μέθοδος αναζήτησης καλείται σε ένα αντικείμενο CarBrandList cbl,
            μεταβιβάζοντας το όνομα της επωνυμίας
            // και το όνομα του μοντέλου ως ορίσματα. Ανακτά ένα αντικείμενο CarBrand και
            αποκτά πρόσβαση στο URI εικόνας
            // του χρησιμοποιώντας τη μέθοδο getImage().
            imageUrl = cbl.lookup(brand, rb.getText().toString()).getImage();

            // Συνδέει την επωνυμία και το όνομα μοντέλου σε μια ενιαία συμβολοσειρά, που
            αντιπροσωπεύει το πλήρες όνομα της επιλεγμένης επιλογής.
            full_name = brand + " " + rb.getText().toString();

            // Ορίζει την ορατότητα ενός αντικειμένου ImageView που ονομάζεται myImage σε
            VISIBLE. Αυτό κάνει την προβολή εικόνας ορατή στην οθόνη.
            myImage.setVisibility(View.VISIBLE);
```

```
// Εκτυπώνει το URI εικόνας στην κονσόλα για εντοπισμό σφαλμάτων ή
// ενημερωτικούς σκοπούς.
System.out.println(imageUri);

/* Χρησιμοποιεί τη βιβλιοθήκη Picasso για να φορτώσει και να εμφανίσει μια
εικόνα από το παρεχόμενο URI εικόνας
στο myImage ImageView. Η μέθοδος Uri.parse() χρησιμοποιείται για τη μετατροπή
της συμβολοσειράς URI εικόνας σε
αντικείμενο Uri, το οποίο στη συνέχεια μεταβιβάζεται στον Πικάσο για φόρτωση
της εικόνας.
Η μέθοδος resize() χρησιμοποιείται για την αλλαγή του μεγέθους της εικόνας σε
ένα συγκεκριμένο πλάτος 300 pixel,
διατηρώντας παράλληλα την αναλογία διαστάσεων.
*/
Picasso.with(getApplicationContext()).load(Uri.parse(imageUri)).resize(300,
0).into(myImage);

/* ο κωδικός προχωρά στην καταγραφή του ιστορικού επιλογών και στην εμφάνιση
ενός μηνύματος toast που υποδεικνύει
ότι η επιλογή έχει καταγραφεί. Αυτό γίνεται δημιουργώντας ένα στιγμιότυπο του
OkHttpHandler,
καλώντας τη μέθοδο logHistory() με τη δομημένη διεύθυνση URL και
συλλαμβάνοντας και εκτυπώνοντας τυχόν εξαιρέσεις
που προκύπτουν κατά τη διάρκεια της διαδικασίας.
* */
try {
    OkHttpHandler okHttpHandler = new OkHttpHandler();
    okHttpHandler.logHistory(url);
    Toast.makeText(getApplicationContext(), "Selection Logged",
Toast.LENGTH_SHORT).show();
} catch (Exception e) {
    e.printStackTrace();
}
});

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    // Η γραμμή cb1 = new CarBrandList(myIP);
    // δημιουργεί μια νέα "CarBrandList" αντικείμενο και το αρχικοποιεί με την τιμή της
μεταβλητής myIP.
    cb1 = new CarBrandList(myIP);
    super.onCreate(savedInstanceState);

    // ορίζει το περιεχόμενο της δραστηριότητας, δηλαδή ποιο layout θα χρησιμοποιηθεί. Σε
αυτήν την περίπτωση,
    // το layout που ονομάζεται activity_main θα εμφανίζεται στην οθόνη.
    setContentView(R.layout.activity_main);

    // εντοπίζει το στοιχείο με το ID imageButton στο layout και το αποθηκεύει στη
```

μεταβλητή myImage.

```
myImage = (ImageButton) findViewById(R.id.imageButton);  
myImage.setVisibility(View.GONE); // κρύβει το imageButton, καθιστώντας το αόρατο.
```

// εντοπίζει το στοιχείο με το ID cars (πιθανώς ένα Spinner) στο layout και το αποθηκεύει στη μεταβλητή dropDown.

```
Spinner dropDown = (Spinner) findViewById(R.id.cars);
```

// Η γραμμή με το ArrayAdapter δημιουργεί έναν προσαρμοστή (adapter) για το Spinner με τη λίστα των

// brands που παρέχονται από το cbl.getAllBrands().

```
ArrayAdapter<String> arrayAdapter =  
    new ArrayAdapter<String>(this,  
        android.R.layout.simple_spinner_dropdown_item,  
        cbl.getAllBrands());
```

// ορίζει τον προσαρμοστή που δημιουργήθηκε στο πεδίο του Spinner.

```
dropDown.setAdapter(arrayAdapter);
```

// εντοπίζει το RadioGroup με ID rg στο layout και το αποθηκεύει στη μεταβλητή rg.

```
rg = (RadioGroup) findViewById(R.id.rg);
```

// Η επόμενη περιοχή του κώδικα ορίζει έναν click listener για το myImage,

// που όταν πατηθεί, θα εκτελεστεί η μέθοδος sendMessage.

```
myImage.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View view) {
```

```
        sendMessage(view);
```

```
    }
```

```
});
```

```
}
```

// Η sendMessage μέθοδος δημιουργεί ένα Intent για να μεταβεί στη δραστηριότητα CarDetails,

// περνώντας δεδομένα όπως η imageUrl και η full_name, και στη συνέχεια εκκινεί τη νέα δραστηριότητα.

```
public void sendMessage(View view){
```

```
    Intent intent = new Intent(this, CarDetails.class);
```

```
    intent.putExtra("IMAGE", imageUrl);
```

```
    intent.putExtra("FULL_NAME", full_name);
```

```
    startActivity(intent);
```

```
}
```

```
}
```