

# Βασικές γνώσεις Android σε μορφή ερωτοαπαντήσεων

---

## Τι είναι ένα activity και τι είναι ένα fragment

---

Στο πλαίσιο της ανάπτυξης εφαρμογών για το Android με τη χρήση του Android Studio, τα "activity" και "fragment" είναι δύο βασικά στοιχεία που χρησιμοποιούνται για την δομή και την παρουσίαση της διεπαφής των εφαρμογών.

### Activity:

Ένα "activity" αντιπροσωπεύει μια μονάδα εργασίας μιας εφαρμογής. Σε απλά λόγια, είναι μια οθόνη με την οποία ο χρήστης αλληλεπιδρά. Κάθε φορά που ο χρήστης εκτελεί μια ενέργεια που απαιτεί να αλλάξει την οθόνη ή την κατάσταση της εφαρμογής, δημιουργείται ένα νέο activity. Τα activities μπορούν να περιέχουν τον κώδικα για τη λογική της εφαρμογής, την παρουσίαση των δεδομένων και την αλληλεπίδραση με τον χρήστη.

### Fragment:

Ένα "fragment" είναι ένα μικρό κομμάτι της διεπαφής χρήστη που μπορεί να είναι μέρος ενός activity. Η χρήση των fragments επιτρέπει τη δημιουργία ευέλικτων διεπαφών που μπορούν να προσαρμοστούν σε διάφορα μεγέθη οθονών και συσκευών. Τα fragments μπορούν να αντικατασταθούν ή να συνδυαστούν για να δημιουργήσουν πολύπλοκες διεπαφές. Ένα activity μπορεί να περιέχει ένα ή περισσότερα fragments και να τα εναλλάσσει ανάλογα με την αλληλεπίδραση του χρήστη.

Συνολικά, τα activities και τα fragments αποτελούν τα βασικά στοιχεία της διαμόρφωσης της διεπαφής του χρήστη σε μια εφαρμογή Android, επιτρέποντας την οργάνωση του κώδικα και την αποτελεσματική αλληλεπίδραση με τον χρήστη.

## Πως εισάγω νέα activity και πως νέα fragment;

---

- Εισαγωγή νέου Activity:

1. Ανοίξτε το Android Studio και το έργο (project) σας.
2. Κάντε δεξί κλικ πάνω στον φάκελο "app" στο "Project" παράθυρο.
3. Επιλέξτε "New" και στη συνέχεια "Activity".
4. Επιλέξτε τον τύπο της activity που θέλετε να δημιουργήσετε (π.χ. Empty Activity, Basic Activity, etc.).
5. Δώστε ένα όνομα στο νέο activity και πατήστε "Finish".
6. Θα δημιουργηθεί το νέο activity και τα απαραίτητα αρχεία κώδικα και διεπαφής θα προστεθούν στο έργο σας.

- Εισαγωγή νέου Fragment στο Android Studio:
1. Ακολουθήστε τα βήματα 1 και 2 όπως παραπάνω.
  2. Επιλέξτε "New" και στη συνέχεια "Fragment".
  3. Επιλέξτε τον τύπο του fragment που θέλετε να δημιουργήσετε (π.χ. Fragment (Blank), Fragment (List), etc.).
  4. Δώστε ένα όνομα στο νέο fragment και πατήστε "Finish".
  5. Θα δημιουργηθούν τα απαραίτητα αρχεία κώδικα και διεπαφής για το νέο fragment στο έργο σας.
  6. Για να το εισάγετε πχ στο activity\_main.xml, πρέπει να εισάγετε το **FragmentContainerView** όπου θα σας εμφανιστεί ένα παράθυρο. Εκεί συνδέετε το fragment σας.

## Πως ενώνω το fragment στο activity;

Για να ενώσετε ένα Fragment σε ένα Activity στο Android, πρέπει να ακολουθήσετε τα εξής βήματα:

1. Βεβαιωθείτε ότι στο build.gradle (Module:App) να έχετε εισάγει την εξάρτηση (dependencies) και να πατήσετε το Sync Now. Θα μοιάζει κάπως έτσι το αρχείο σας.

```
implementation 'androidx.appcompat:appcompat:1.6.1'
implementation 'com.google.android.material:material:1.9.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
implementation 'androidx.appcompat:appcompat:1.3.1'
implementation 'androidx.fragment:fragment:1.3.6'
```

### 1. Δημιουργία Fragment:

Πρώτα, πρέπει να έχετε δημιουργήσει το Fragment στο Android Studio όπως περιγράφηκε στις προηγούμενες απαντήσεις.

### 2. Σύνδεση Fragment στο Layout του Activity:

Πηγαίνετε στο XML layout αρχείο του Activity (συνήθως βρίσκεται στον φάκελο res/layout). Εκεί μπορείτε να προσθέσετε το στοιχείο και να χρησιμοποιήσετε το attribute android:name για να ορίσετε το πλήρες όνομα της κλάσης του Fragment που θέλετε να ενσωματώσετε. Το layout αρχείο μπορεί να μοιάζει κάπως έτσι:

```
<fragment
    android:id="@+id/fragment_container"
    android:name="com.example.myapplication.MyFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

### 3. Εμφάνιση Fragment μέσω κώδικα:

Στην Java κλάση του Activity, μπορείτε να χρησιμοποιήσετε το FragmentManager για να

εμφανίσετε το Fragment στο container που ορίσατε στο layout. Ένα παράδειγμα κώδικα σε Java μπορεί να είναι:

```
FragmentManager fragmentManager = getSupportFragmentManager();
FragmentManager.beginTransaction().replace(R.id.fragment_container, myFragment);

// Αντικαταστήστε το MyFragment με το όνομα της κλάσης του
// Fragment που δημιουργήσατε
MyFragment myFragment = new MyFragment();
fragmentManager.beginTransaction().replace(R.id.fragment_container, myFragment);
fragmentManager.beginTransaction().commit();
```

Αυτό θα ενσωματώσει το Fragment στο Activity και θα το εμφανίσει στον καθορισμένο χώρο στο layout του Activity. Είναι σημαντικό να προσέξετε ότι πρέπει να χρησιμοποιήσετε το σωστό όνομα πακέτου και τη σωστή κλάση για το Fragment στον κώδικά σας.

## Παράδειγμα πως μπορώ να αλλάζω fragment μέσα σε activity με το πάτημα κουμπιών

Αφού φτιάξω στο layout το activity\_main.xml, fragment\_fragment\_one.xml, fragment\_fragment\_two.xml, στον κώδικα java προσθέτω το εξής:

```
public void ChangeFragment(View view){
    Fragment fragment;
    if(view == findViewById(R.id.button)){
        fragment = new FragmentOne();
        FragmentManager fm = getFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();

        // Το id του fragment είναι fragment_place.
        ft.replace(R.id.fragment_place, fragment);
        ft.commit();
    }
    if(view == findViewById(R.id.button2)){
        fragment = new FragmentTwo();
        FragmentManager fm = getFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();

        // Το id του fragment είναι fragment_place.
        ft.replace(R.id.fragment_place, fragment);
        ft.commit();
    }
}
```

Στο activity\_main.xml στις ιδιότητες των κουμπιών βάζω στο onClick την μέθοδο ChangeFragment.

## Τι είναι το binding, που χρησιμοποιείται και ένα παράδειγμα

# χρήσης του

Το "binding" σε σχέση με το Android αναφέρεται στο "View Binding", μια τεχνική που επιτρέπει τη σύνδεση (binding) των στοιχείων του διεπαφής χρήστη (UI elements) με τον κώδικα Java της εφαρμογής σας. Αυτό γίνεται μέσω του αυτόματου δημιουργίας ενός binding class που αντιπροσωπεύει το XML layout της διεπαφής σας.

Ο View Binding αντικαθιστά τον παλαιότερο τρόπο χρήσης του findViewById() για την εύρεση και τη σύνδεση των στοιχείων του UI στον κώδικα. Χρησιμοποιώντας το View Binding, μπορείτε να αποφύγετε τα null pointers που μπορούν να προκύψουν από λάθη στην αναζήτηση στοιχείων UI και να έχετε πιο ασφαλή και αποτελεσματική πρόσβαση σε αυτά.

Εδώ είναι ένα απλό παράδειγμα χρήσης του View Binding:

## 1. Layout (activity\_main.xml):

Ας υποθέσουμε ότι έχουμε ένα απλό TextView στο XML layout:

```
<TextView
    android:id="@+id/helloTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, View Binding!" />
```

## 2. Java Code (MainActivity.java):

Στη συνέχεια, μπορούμε να χρησιμοποιήσουμε το View Binding για να συνδέσουμε το TextView στον κώδικα:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import com.example.myapp.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        // Τώρα μπορούμε να χρησιμοποιήσουμε το binding για πρόσβαση στα στοιχεία UI
        binding.helloTextView.setText("Hello from View Binding!");
    }
}
```

Στο παραπάνω παράδειγμα, το `ActivityMainBinding` είναι το binding class που δημιουργείται αυτόματα από το View Binding για το XML layout `activity_main.xml`. Το binding αντιπροσωπεύει τον ολόκληρο πίνακα UI που έχετε ορίσει στο XML, και μπορείτε να προσπελάσετε τα στοιχεία του UI με τη χρήση του binding.

## Τι είναι ένα Bundle και τι Intent;

Τόσο το Bundle όσο και το Intent είναι δύο βασικά στοιχεία στην πλατφόρμα Android που χρησιμοποιούνται για την μεταφορά δεδομένων μεταξύ συνιστωσών μιας εφαρμογής, όπως μεταξύ διαφορετικών activities ή services.

### Bundle:

Το Bundle είναι ένας τύπος δεδομένων που χρησιμοποιείται για τη συσκευασία και μεταφορά δεδομένων μεταξύ συνιστωσών της ίδιας εφαρμογής. Συνήθως χρησιμοποιείται για την παράδοση παραμέτρων και δεδομένων από ένα activity σε ένα άλλο. Τα δεδομένα μέσα σε ένα Bundle οργανώνονται με τη μορφή "key-value" ζευγαριών, όπου κάθε κλειδί (key) αντιστοιχεί σε ένα τύπο δεδομένων (π.χ. String, Integer, Parcelable κλπ.). Τα Bundle συνήθως χρησιμοποιούνται για τη μεταφορά περιορισμένων ποσοτήτων δεδομένων μεταξύ συνιστωσών.

### Intent:

Το Intent αναπαριστά μια προτροπή για μια ενέργεια που πρόκειται να εκτελεστεί. Χρησιμοποιείται για την εκκίνηση δραστηριοτήτων (activities), υπηρεσιών (services), εκδηλώσεων (broadcasts), και άλλων συνιστωσών της εφαρμογής. Μέσω των Intent, μπορείτε να περάσετε δεδομένα και να εκτελέσετε διάφορες ενέργειες. Τα Intent μπορούν να περιέχουν επιπλέον πληροφορίες, όπως κατηγορία, ενέργεια, δεδομένα και παραμέτρους. Στην ουσία, το Intent περιγράφει το τι θέλετε να κάνει η εφαρμογή σας.

Συνολικά, τόσο το Bundle όσο και το Intent είναι βασικά εργαλεία για την μεταφορά δεδομένων και τη διαχείριση της αλληλεπίδρασης μεταξύ διάφορων συνιστωσών μιας εφαρμογής Android.

## Ένα παράδειγμα κώδικα Bundle

Εδώ έχετε ένα απλό παράδειγμα κώδικα που δείχνει πώς να χρησιμοποιήσετε ένα Bundle για να μεταφέρετε ένα ακέραιο αριθμό από ένα Activity σε ένα άλλο:

```
import android.content.Intent;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class FirstActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
    }
}
```

```

        int numberToPass = 42;

        // Δημιουργία ενός Bundle και προσθήκη ενός ακέραιου αριθμού
        Bundle bundle = new Bundle();
        bundle.putInt("number_key", numberToPass);

        // Δημιουργία ενός Intent για το δεύτερο Activity
        Intent intent = new Intent(this, SecondActivity.class);
        intent.putExtras(bundle);

        // Εκκίνηση του δεύτερου Activity με το Intent
        startActivity(intent);
    }
}

```

Στο παράδειγμα αυτό, ο ακέραιος αριθμός 42 προστίθεται στο Bundle με το κλειδί "number\_key". Στη συνέχεια, το Bundle περνάται στο Intent και το δεύτερο Activity εκκινείται με το Intent. Στο δεύτερο Activity, μπορείτε να ανακτήσετε τον ακέραιο αριθμό από το Bundle.

Αυτό είναι ένα απλό παράδειγμα για να δείτε πώς λειτουργεί η χρήση ενός Bundle για τη μεταφορά δεδομένων μεταξύ διαφορετικών συνιστωσών ενός Android εφαρμογής.

## Ένα παράδειγμα με Intenet (μεταφορά τιμής από πρώτο activity στο δεύτερο activity)

Εδώ έχετε ένα απλό παράδειγμα κώδικα που δείχνει πώς να χρησιμοποιήσετε ένα Intent για να μεταφέρετε δεδομένα από ένα Activity σε ένα άλλο:

### 1. Πρώτο Activity (FirstActivity.java):

```

import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class FirstActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);

        Button sendDataButton = findViewById(R.id.sendDataButton);
        sendDataButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        String dataToSend = "Hello from FirstActivity!";

        Intent intent = new Intent(FirstActivity.this, SecondActivity.class);
        intent.putExtra("data_key", dataToSend);
        startActivity(intent);
    }
});
}
}

```

## 2. Δεύτερο Activity (SecondActivity.java):

```

import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        Intent intent = getIntent();
        if (intent != null) {
            String receivedData = intent.getStringExtra("data_key");

            TextView receivedDataTextView = findViewById(R.id.receivedDataTextView);
            receivedDataTextView.setText(receivedData);
        }
    }
}

```

Σε αυτό το παράδειγμα, το πρώτο Activity περιέχει ένα κουμπί. Όταν πατηθεί το κουμπί, δημιουργείται ένα Intent που έχει ως στόχο το δεύτερο Activity. Επίσης, μεταφέρει ένα String δεδομένο με το κλειδί "data\_key". Στο δεύτερο Activity, το String δεδομένο ανακτάται από το Intent και εμφανίζεται σε ένα TextView.

Αυτό το παράδειγμα δείχνει πώς μπορείτε να χρησιμοποιήσετε το Intent για τη μεταφορά δεδομένων και την εναλλαγή μεταξύ διαφορετικών συνιστωσών ενός Android εφαρμογής.

## Πότε χρησιμοποιώ Binding, πότε Bundle και πότε Intent;

Τα Binding, Bundle και Intent είναι τρία διαφορετικά μέσα που χρησιμοποιούνται σε διάφορες περιστάσεις στο Android για διαφορετικούς σκοπούς:



### 1. View Binding:

Χρησιμοποιείται για να συνδέσετε τα στοιχεία του διεπαφής χρήστη (UI) με τον κώδικα σε ένα Activity ή Fragment. Είναι μια αναβαθμισμένη τεχνική από τη χρήση του findViewById() που είναι πιο ασφαλής και αποτελεσματικός. Χρησιμοποιείται για να αποφύγετε τα null pointers και να έχετε πιο συνεπή και αξιόπιστη πρόσβαση στα στοιχεία UI.

### 2. Bundle:

Χρησιμοποιείται για τη μεταφορά δεδομένων μεταξύ διαφορετικών components, όπως Activities και Fragments, εντός μιας εφαρμογής Android. Χρησιμοποιείται συνήθως για να περάσετε παραμέτρους ή πληροφορίες από ένα Activity σε ένα άλλο, όταν κάποιος χρήστης παρακολουθεί μια συγκεκριμένη δράση.

### 3. Intent:

Το Intent χρησιμοποιείται για να εκκινήσετε διάφορες δραστηριότητες (Activities) ή να αλληλεπιδράσετε με άλλες εφαρμογές. Χρησιμοποιείται για την αποστολή εντολών ή τη μεταφορά δεδομένων προς άλλες εφαρμογές, ανοίγοντας π.χ. μια συγκεκριμένη σελίδα ιστού σε έναν προεπιλεγμένο περιηγητή.

Συνοψίζοντας, χρησιμοποιήστε το Binding για να συνδέσετε τα στοιχεία του UI με τον κώδικα, το Bundle για να περνάτε δεδομένα μεταξύ διαφορετικών components της εφαρμογής και το Intent για να επικοινωνείτε με διάφορες δραστηριότητες και εφαρμογές.

## Τι είναι τα layouts και πως χρησιμοποιούνται; Ένα μικρό παράδειγμα.

---

Τα layouts στο Android αναφέρονται στην οργάνωση και διάταξη των διαφόρων οπτικών στοιχείων (Views) σε μια οθόνη. Τα layouts καθορίζουν πώς τα διάφορα στοιχεία θα τοποθετηθούν και θα αλληλεπιδρούν μεταξύ τους στο γραφικό περιβάλλον της εφαρμογής σας.

Τα διάσημα layouts που μπορείτε να χρησιμοποιήσετε είναι:

- **LinearLayout:** Τα στοιχεία τοποθετούνται μία πάνω στο άλλο ή δίπλα στο άλλο, στον ίδιο άξονα (οριζόντιο ή κατακόρυφο).
- **RelativeLayout:** Τα στοιχεία ορίζονται σε σχέση με τα άλλα, χρησιμοποιώντας κανόνες θέσης, όπως "από πάνω του", "δίπλα από", κ.λπ.
- **ConstraintLayout:** Τα στοιχεία τοποθετούνται με βάση περιορισμούς (constraints) που καθορίζουν τη θέση και τις σχέσεις τους με άλλα στοιχεία.
- **FrameLayout:** Τα στοιχεία επικαλύπτονται ένα πάνω στο άλλο, και το πάνω στοιχείο είναι το πιο ορατό.

Εδώ έχουμε ένα παράδειγμα χρήσης του LinearLayout:

**Layout (activity\_main.xml):**



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" <!-- Οριζόντιο ή κατακόρυφο -->
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, this is a TextView!"
        android:layout_gravity="center" />
    <!-- Κείμενο στο κέντρο του γονικού στοιχείου -->

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me!"
        android:layout_gravity="center" />
    <!-- Κουμπί στο κέντρο του γονικού στοιχείου -->

</LinearLayout>

```

Στο παραπάνω παράδειγμα, χρησιμοποιούμε το LinearLayout με κατακόρυφη διάταξη (vertical orientation) για να τοποθετήσουμε ένα `TextView`

## Απλό παράδειγμα κώδικα πως μεταφέρομαι από το MainActivity στο SecondActivity

Παραπάνω, στο παράδειγμα με το Intent αναφέρεται αυτό. Αλλά ας το δούμε ξεχωριστά. Αρχικά βεβαιωθείτε ότι στο AndroidManifest.xml υπάρχει η εγγραφή (μετά το .FirstActivity):

```

<activity
    android:name=".SecondActivity"
    android:label="@string/title_activity_second" >
    <intent-filter>
        <action android:name="gr.uom.myApp.SecondActivity" />

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

```

Στην συνέχεια στο MainActivity.java να έχετε την εγγραφή:

```

public class MainActivity extends ActionBarActivity {
    private static Button button_sbm;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        OnClickListener();
    }

    public void OnClickListener() {
        // Φτιάχνω το αντικείμενο κουμπιού
        button_sbm = (Button)findViewById(R.id.button);

        // Του ορίζω έναν Listener
        button_sbm.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    // Εδώ του ορίζω ποιο Activity να ξεκινήσει.
                    // Παραπάνω του περάσαμε και τιμές.
                    Intent intent = new Intent("gr.uom.myApp.SecondActivity");
                    startActivity(intent);
                }
            }
        );
    }
}

```

## Πως παίρνω τιμή από ένα πεδίο;

Αν υποθέσουμε ότι έχετε ένα EditText πεδίο στο layout του Activity και θέλετε να πάρετε την τιμή που έχει εισαχθεί σε αυτό, μπορείτε να το πετύχετε ως εξής:

### 1. Layout (activity\_main.xml):

Στο XML layout αρχείο του Activity, θα πρέπει να έχετε ένα EditText πεδίο στο οποίο ο χρήστης μπορεί να εισάγει δεδομένα.

```

<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter something" />

```

### 2. Java Code (MainActivity.java):

Στην κλάση του Activity, μπορείτε να ανακτήσετε την τιμή που έχει εισαχθεί στο πεδίο με τη χρήση της μεθόδου `getText()` του `EditText`.

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = findViewById(R.id.editText);

        // Παράδειγμα: Πάρτε την τιμή από το πεδίο όταν πατηθεί ένα κουμπί
        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String enteredText = editText.getText().toString();
                // Κάντε κάτι με την enteredText, όπως εμφανίστε τη σε ένα TextView
            }
        });
    }
}

```

Στο παράδειγμα αυτό, το **editText.getText().toString()** χρησιμοποιείται για να ανακτήσει το κείμενο που έχει εισαχθεί στο EditText πεδίο. Αυτό το κείμενο στη συνέχεια μετατρέπεται σε String με τη χρήση της μεθόδου toString(), και μπορείτε να το χρησιμοποιήσετε όπως επιθυμείτε (π.χ. εμφανίστε το σε ένα TextView ή εκτελέστε άλλες ενέργειες με αυτό).

## Πως εμφανίζω το μήνυμα που πήρα παραπάνω σε ένα Toast μήνυμα (με πάτημα κουμπιού);

Εδώ είναι ο κώδικας για να εμφανίσετε το κείμενο που πήρατε από το πεδίο εισαγωγής σε ένα Toast μήνυμα:

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

```

```

private EditText editText;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    editText = findViewById(R.id.editText);

    Button button = findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String enteredText = editText.getText().toString();

            // Εμφάνιση του μηνύματος με Toast
            Toast.makeText(MainActivity.this, "Entered Text: " + enteredText,
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

Στον κώδικα αυτό, η μέθοδος **makeText()** του Toast χρησιμοποιείται για να δημιουργήσει και να εμφανίσει ένα Toast μήνυμα. Ο πρώτος παράμετρος είναι η κλάση που καλεί το Toast (συνήθως το τρέχον Activity), η δεύτερη παράμετρος είναι το κείμενο που θέλετε να εμφανίσετε, και η τρίτη παράμετρος είναι η διάρκεια εμφάνισης του Toast (μπορεί να είναι **Toast.LENGTH\_SHORT** ή **Toast.LENGTH\_LONG**).

Όταν πατηθεί το κουμπί, το κείμενο που έχει εισαχθεί στο πεδίο εισαγωγής θα εμφανίζεται σε ένα Toast μήνυμα.

## Πως παίρνω την εισαγωγή του χρήστη από ένα πεδίο και μετά το υψώνω στην 3 (κάνω κάποια ενέργεια έστω) και το εμφανίζω στην οθόνη (με πάτημα κουμπιού).

Εδώ έχετε ένα παράδειγμα κώδικα που δείχνει πώς να πάρετε την είσοδο του χρήστη από ένα πεδίο, να την υψώσετε στην τρίτη δύναμη και να την εμφανίσετε στην οθόνη:

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

```

```

private EditText inputEditText;
private TextView resultTextView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    inputEditText = findViewById(R.id.inputEditText);
    resultTextView = findViewById(R.id.resultTextView);

    Button calculateButton = findViewById(R.id.calculateButton);
    calculateButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Πάρτε την είσοδο του χρήστη από το πεδίο
            String userInput = inputEditText.getText().toString();

            if (!userInput.isEmpty()) {
                // Μετατροπή της εισόδου σε αριθμό
                double number = Double.parseDouble(userInput);

                // Υψώστε τον αριθμό στην τρίτη δύναμη
                double result = Math.pow(number, 3);

                // Εμφανίστε το αποτέλεσμα στην οθόνη
                resultTextView.setText("Result: " + result);
            }
        }
    });
}

```

Στο παραπάνω παράδειγμα, το πεδίο εισαγωγής (EditText) χρησιμοποιείται για την είσοδο του χρήστη, και το αποτέλεσμα εμφανίζεται σε ένα TextView μετά τον υπολογισμό. Ο κώδικας επιβεβαιώνει εάν η είσοδος δεν είναι κενή πριν προσπαθήσει να τη μετατρέψει σε αριθμό. Εάν η μετατροπή είναι επιτυχής, υπολογίζεται η τρίτη δύναμη του αριθμού και το αποτέλεσμα εμφανίζεται στην οθόνη.

## Παράδειγμα επιλογής από ένα Radio Button

Έστω το activity\_main.xml

```

<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:layout_below="@+id/button"
    android:layout_alignParentLeft="true"

```

```

        android:layout_alignParentStart="true"
        android:id="@+id/rg_animals">

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Dog"
            android:id="@+id/radioButton_dog"
            android:checked="false" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cat"
            android:id="@+id/radioButton_cat"
            android:checked="false" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cow"
            android:id="@+id/radioButton_cow"
            android:checked="false" />

    </RadioGroup>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:id="@+id/button"
        android:layout_marginTop="25dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>

```

Τώρα πρέπει να φτιάξουμε το MainActivity.java

```

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity {
    private static RadioGroup radio_g;

```

```

    private static RadioButton radio_b;
    private static Button button_sbm;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        onClickListenerButton();
    }

    public void onClickListenerButton() {
        radio_g = (RadioGroup)findViewById(R.id.rg_animals); // Φτιάχνω το RadioGroup
        button_sbm = (Button)findViewById(R.id.button); // Φτιάχνω και το πλήκτρο

        button_sbm.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    // Παίρνει το id του επιλεγμένου radiobutton
                    int selected_id = radio_g.getCheckedRadioButtonId();
                    // Βρίσκει το αντικείμενο του επιλεγμένου radiobutton
                    radio_b = (RadioButton)findViewById(selected_id);

                    Toast.makeText(MainActivity.this,radio_b.getText().
                        toString(),Toast.LENGTH_SHORT ).show();
                    // Την μετατρέπει σε string και την εμφανίζει στην οθόνη
                }
            }
        );
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

```



```
}  
}
```

## Παράδειγμα κώδικα πως γεμίζει ένα spinner, επιλέγεται κάτι και το δείχνει στην οθόνη σε Toast μήνυμα

Εδώ είναι ένα παράδειγμα κώδικα που δείχνει πώς μπορείτε να γεμίσετε ένα Spinner, να επιλέξετε κάτι και να το εμφανίσετε σε ένα Toast μήνυμα:

### 1. Layout (activity\_main.xml):

Ένα παράδειγμα απλού Spinner στο XML layout:

```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:padding="16dp">  
  
    <Spinner  
        android:id="@+id/spinner"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />  
  
    <Button  
        android:id="@+id/showResultButton"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Show Result" />  
  
</LinearLayout>
```

### 2. Java Code (MainActivity.java):

```
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {
```

```

private Spinner spinner;
private Button showResultButton;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    spinner = findViewById(R.id.spinner);
    showResultButton = findViewById(R.id.showResultButton);

    // Δημιουργία λίστας με δεδομένα για το Spinner
    List<String> items = new ArrayList<>();
    items.add("Option 1");
    items.add("Option 2");
    items.add("Option 3");

    // Δημιουργία του ArrayAdapter και σύνδεση του με τον Spinner
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_dropdown_item, items);
    spinner.setAdapter(adapter);

    // Ενέργεια επιλογής αντικειμένου από τον Spinner
    spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view,
            int position, long id) {
            // Τίποτα εδώ, η λειτουργία αυτή έχει μετακινηθεί στο κουμπί
        }

        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
            // Τίποτα εδώ
        }
    });

    showResultButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String selectedOption = spinner.getSelectedItem().toString();
            Toast.makeText(MainActivity.this, "Selected: " + selectedOption,
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

Σε αυτό το παράδειγμα, ο Spinner γεμίζει με επιλογές "Option 1", "Option 2" και "Option 3". Όταν επιλέγετε μια επιλογή από τον Spinner, εμφανίζεται ένα Toast μήνυμα που δείχνει την επιλεγμένη επιλογή.

# Παράδειγμα κώδικα για χρήση βιβλιοθήκης Picasso σε μια εικόνα η οποία να αλλάζει με το πάτημα ενός κουμπιού.

Η βιβλιοθήκη Picasso χρησιμοποιείται για τη φόρτωση και την προβολή εικόνων από διάφορες πηγές (όπως τοπικά αρχεία, διαδικτυακές διευθύνσεις URL κ.λπ.). Εδώ είναι ένα παράδειγμα κώδικα με σχόλια που δείχνει πώς μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη Picasso για να εμφανίσετε μια εικόνα που αλλάζει με το πάτημα ενός κουμπιού:

## 1. Layout (activity\_main.xml):

Προσθέστε ένα ImageView και ένα Button στο XML layout:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:src="@drawable/placeholder_image" />

    <Button
        android:id="@+id/changeImageButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Change Image" />

</LinearLayout>
```

## 2. Java Code (MainActivity.java):

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import com.squareup.picasso.Picasso;

public class MainActivity extends AppCompatActivity {

    private ImageView imageView;
    private Button changeImageButton;

    // Οι διαθέσιμες εικόνες
```

```

private String[] imageUrls = {
    "https://example.com/image1.jpg",
    "https://example.com/image2.jpg",
    "https://example.com/image3.jpg"
};
private int currentIndex = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    imageView = findViewById(R.id.imageView);
    changeImageButton = findViewById(R.id.changeImageButton);

    // Φόρτωση της πρώτης εικόνας
    loadImage(imageUrls[currentIndex]);

    changeImageButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Εναλλαγή της εικόνας μεταξύ των διαθέσιμων
            currentIndex = (currentIndex + 1) % imageUrls.length;
            loadImage(imageUrls[currentIndex]);
        }
    });
}

// Μέθοδος για τη φόρτωση εικόνας με τη βοήθεια της Picasso
private void loadImage(String imageUrl) {
    Picasso.get()
        .load(imageUrl)
        .placeholder(R.drawable.placeholder_image) // Ενδιάμεση εικόνα
        .error(R.drawable.error_image) // Εικόνα αν συμβεί λάθος
        .into(imageView);
}
}

```

Σε αυτό το παράδειγμα, το Picasso φορτώνει την επιλεγμένη εικόνα από την λίστα `imageUrls` και την εμφανίζει στο `ImageView`. Κάθε φορά που πατιέται το κουμπί "Change Image", η επιλεγμένη εικόνα αλλάζει. Επίσης, υπάρχουν προσδιορισμένες εικόνες "placeholder" και "error" που εμφανίζονται κατά τη φόρτωση και σε περίπτωση λάθους, αντίστοιχα. Αντικαταστήστε τα URLs με τις πραγματικές διαδικτυακές διευθύνσεις των εικόνων που θέλετε να χρησιμοποιήσετε.

3. Για να συμπεριλάβετε τη βιβλιοθήκη Picasso στην εφαρμογή, πρέπει να προσθέσετε την αντίστοιχη εξάρτηση (dependency) στο αρχείο **build.gradle** του project.

- Άνοιξε το αρχείο `build.gradle` που βρίσκεται στον κύριο φάκελο του project (όχι στο φάκελο `app`).

- Στην ενότητα dependencies, προσθέστε την ακόλουθη γραμμή για να προσθέσετε την εξάρτηση της βιβλιοθήκης Picasso:

```
implementation 'com.squareup.picasso:picasso:2.5.2'
```

Δεν ξεχνάμε να επιλέξουμε Sync Now.

## Βασικές αλλαγές-προσθήκες για να έχω λήψη στοιχείων από μια εξωτερική υπηρεσία - βάση δεδομένων.

Για να κάνετε λήψη δεδομένων από μια εξωτερική υπηρεσία στο Android με χρήση τη βιβλιοθήκη OkHttp, πρέπει να ενημερώσετε το Android Manifest και να δημιουργήσετε ένα αρχείο network\_security\_config.xml. Εδώ είναι ένα βήμα προς βήμα παράδειγμα:

### 1. Ενημέρωση του Android Manifest:

Προσθέστε την άδεια INTERNET στο αρχείο AndroidManifest.xml στο σημείο πριν το **application**:

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

<application
    android:allowBackup="true"
```

ενώ τελευταία γραμμή πριν το activity να εμφανίζονται οι γραμμές:

```
    android:networkSecurityConfig="@xml/network_security_config"
    android:usesCleartextTraffic="true">

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

    <activity
```

### 2. Δημιουργία του αρχείου network\_security\_config.xml:

Δημιουργήστε ένα νέο φάκελο με όνομα res/xml στον φάκελο του προγράμματος (New- → Folder → XML Resources Folder). Εν συνεχεία, δημιουργήστε ένα αρχείο network\_security\_config.xml μέσα σε αυτόν τον φάκελο. Το περιεχόμενο του αρχείου θα είναι κάτι παρόμοιο με αυτό (η IP θα είναι αυτή που έχετε στον υπολογιστή σας):

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

```
<domain includeSubdomains="true">192.168.1.104</domain>
</domain-config>
</network-security-config>
```

Αυτό το αρχείο ρυθμίζει την πολιτική ασφαλείας του δικτύου για την εφαρμογή σας.

### 3. Παράδειγμα με το OkHttp:

Εδώ είναι ένα παράδειγμα κώδικα που χρησιμοποιεί τη βιβλιοθήκη OkHttp για να κάνει ένα απλό GET αίτημα:

```
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();
        try {
            // Δημιουργεί ένα JSONObject αναλύοντας τα δεδομένα απόκρισης
            // που ανακτήθηκαν.
            JSONObject json = new JSONObject(data);

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

            // Επαναλαμβάνεται πάνω από κάθε κλειδί στο JSONObject.
            while(keys.hasNext()) {
                // Ανακτά το επόμενο κλειδί από τον επαναλήπτη,
                // που αντιπροσωπεύει την επωνυμία.
                String brand = keys.next();

                // Ανακτά την αντίστοιχη τιμή για το κλειδί επωνυμίας από το
```

```

        // JSONObject και τη μετατρέπει σε συμβολοσειρά.
        String models = json.get(brand).toString();

        // Δημιουργεί ένα νέο αντικείμενο CarBrand χρησιμοποιώντας την
        // επωνυμία και μοντελοποιεί τιμές και το προσθέτει στη cbList.
        cbList.add(new CarBrand(brand, models));
    }
} catch (JSONException e) {
    e.printStackTrace();
}

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

```

4. Βεβαιωθείτε ότι έχετε προσθέσει τις απαραίτητες εξαρτήσεις στο αρχείο **build.gradle ((Module App))** για να χρησιμοποιήσετε τη βιβλιοθήκη OkHttp.

```

dependencies {
    implementation 'com.squareup.okhttp3:okhttp:4.9.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
    ...
}

```

Δεν ξεχνάμε να επιλέξουμε Sync Now.

## Παράδειγμα για να γεμίσει ένα RadioGroup με RadioButtons τα οποία παίρνει από μια βάση δεδομένων μέσω webservice από τον <http://195.251.511.64> (κλάση OkHttpHandler) και JSON parsing

Για να συνδεθείτε σε μια απομακρυσμένη υπηρεσία βάσης δεδομένων και να ανακτήσετε δεδομένα για να τα εμφανίσετε σε ένα RadioGroup στην εφαρμογή σας, θα χρειαστείτε να κάνετε τις εξής βασικές αλλαγές και να προσθέσετε αρχεία:



## 1. Προσθήκη Δικτυακής Άδειας (Network Permission):

Προσθέστε τη δικτυακή άδεια στο αρχείο AndroidManifest.xml, ώστε η εφαρμογή σας να έχει τη δυνατότητα να επικοινωνεί με το διακομιστή:

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

## 2. Εισαγωγή Βιβλιοθήκης OkHttp:

Για τις αιτήσεις δικτύου, μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη OkHttp. Προσθέστε την εξάρτηση του OkHttp στο αρχείο build.gradle:

```
implementation 'com.squareup.okhttp3:okhttp:4.9.1'
```

Για να επιτύχετε το επιθυμητό αποτέλεσμα, πρέπει να δημιουργήσετε τα παρακάτω αρχεία:

### 1. activity\_main.xml: Το layout αρχείο που θα περιέχει το RadioGroup.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:orientation="vertical">

    </RadioGroup>

</RelativeLayout>
```

### 2. MainActivity.java: Το κύριο αρχείο Java που θα υλοποιεί τη λειτουργία της εφαρμογής.

Βεβαιωθείτε ότι έχετε προσθέσει τη βιβλιοθήκη OkHttp στο αρχείο build.gradle πριν χρησιμοποιήσετε τον παρακάτω κώδικα:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.widget.RadioGroup;
import android.widget.RadioButton;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONException;
```

```

import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private RadioGroup radioGroup;
    private OkHttpClientHandler okHttpClientHandler;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        radioGroup = findViewById(R.id.radioGroup);
        okHttpClientHandler = new OkHttpClientHandler();

        // Κάνε την ασύγχρονη αίτηση για τα δεδομένα JSON
        okHttpClientHandler.execute("http://195.251.511.64");
    }

    private class OkHttpClientHandler extends AsyncTask<String, Void, String> {
        private final OkHttpClient client = new OkHttpClient();

        @Override
        protected String doInBackground(String... urls) {
            String responseString = "";
            try {
                okhttp3.Request request = new okhttp3.Request.Builder()
                    .url(urls[0])
                    .build();

                okhttp3.Response response = client.newCall(request).execute();
                responseString = response.body().string();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return responseString;
        }

        @Override
        protected void onPostExecute(String response) {
            super.onPostExecute(response);

            try {
                List<String> radioButtonLabels = parseJSON(response);
                createRadioButtons(radioGroup, radioButtonLabels);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private List<String> parseJSON(String json) throws JSONException {
    List<String> radioButtonLabels = new ArrayList<>();
    JSONArray jsonArray = new JSONArray(json);

    // Λήψη των ετικετών "label" από το JSON και προσθήκη τους στη λίστα
    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject jsonObject = jsonArray.getJSONObject(i);
        String label = jsonObject.optString("label");
        radioButtonLabels.add(label);
    }

    return radioButtonLabels;
}

private void createRadioButtons(RadioGroup rg, List<String> mList) {
    int count = mList.size();
    RadioButton rb[] = new RadioButton[count];

    // Δημιουργία RadioButtons με τα λεκτικά από τη λίστα
    for (int i = 0; i < mList.size(); i++) {
        rb[i] = new RadioButton(this);
        rb[i].setText(mList.get(i));
        rb[i].setId(i + 100);
        rg.addView(rb[i]);
    }

    // Ορισμός του ακροατή για την αλλαγή επιλεγμένου RadioButton
    rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup radioGroup, int checkedId) {
            RadioButton rb = findViewById(checkedId);
            // Εμφάνιση Toast με το επιλεγμένο κείμενο του RadioButton
            Toast myToast = Toast.makeText(
                getApplicationContext(),
                "Selected: " + rb.getText(),
                Toast.LENGTH_LONG
            );
            myToast.show();
        }
    });
}
}

```

Αυτό το παράδειγμα υλοποιεί την επιθυμητή λειτουργία, συνδέοντας το RadioGroup με το διακομιστή μέσω του OkHttpClientHandler για την ασύγχρονη ανάκτηση των δεδομένων JSON. Έπειτα, χρησιμοποιεί τη μέθοδο createRadioButtons για τη δημιουργία των RadioButtons με τα κατάλληλα λεκτικά. Το επιλεγμένο RadioButton εμφανίζεται σε ένα Toast όταν γίνει επιλογή. Φυσικά, προσαρμόστε το JSON parsing σύμφωνα με το πραγματικό δομή των δεδομένων που θα λαμβάνετε από τον διακομιστή σας.

# Ποιες είναι οι απαραίτητες αλλαγές και βασικά αρχεία για σύνδεση σε μια απομακρυσμένη υπηρεσία βάσης δεδομένων πχ <http://195.251.211.64/> για ανάγνωση και εισαγωγή σε ένα spinner;

Για να συνδεθείτε σε μια απομακρυσμένη υπηρεσία βάσης δεδομένων και να ανακτήσετε δεδομένα για να τα εμφανίσετε σε ένα Spinner στην εφαρμογή σας, θα χρειαστείτε να κάνετε τις εξής βασικές αλλαγές και να προσθέσετε αρχεία:

## 1. Προσθήκη Δικτυακής Άδειας (Network Permission):

Προσθέστε τη δικτυακή άδεια στο αρχείο AndroidManifest.xml, ώστε η εφαρμογή σας να έχει τη δυνατότητα να επικοινωνεί με το διακομιστή:

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

## 2. Εισαγωγή Βιβλιοθήκης OkHttp:

Για τις αιτήσεις δικτύου, μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη OkHttp. Προσθέστε την εξάρτηση του OkHttp στο αρχείο build.gradle:

```
implementation 'com.squareup.okhttp3:okhttp:4.9.1'
```

## 3. Layout (activity\_main.xml):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp" />

</LinearLayout>
```

## 4. Java Code (MainActivity.java):

```

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import androidx.appcompat.app.AppCompatActivity;
import com.squareup.okhttp3.OkHttpClient;
import com.squareup.okhttp3.Request;
import com.squareup.okhttp3.Response;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private Spinner spinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        spinner = findViewById(R.id.spinner);

        // Καλέστε τη μέθοδο OkHttpClientHandler για να ανακτήσετε
        // τα δεδομένα από την υπηρεσία
        OkHttpClientHandler okHttpClientHandler = new OkHttpClientHandler();
        okHttpClientHandler.execute("http://195.251.211.64/cities/getCities.php");
    }

    private class OkHttpClientHandler extends AsyncTask<String, Void, String> {
        private OkHttpClient client = new OkHttpClient();

        @Override
        protected String doInBackground(String... params) {
            Request.Builder builder = new Request.Builder();
            builder.url(params[0]);
            Request request = builder.build();

            try {
                Response response = client.newCall(request).execute();
                return response.body().string();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }

        @Override
        protected void onPostExecute(String response) {
            super.onPostExecute(response);

```

```

        if (response != null) {
            List<String> cityNames = parseJSON(response);
            ArrayAdapter<String> adapter = new ArrayAdapter<>(MainActivity.this,
                android.R.layout.simple_spinner_item, cityNames);
            adapter.setDropDownViewResource(android.
                R.layout.simple_spinner_dropdown_item);

            spinner.setAdapter(adapter);
        }
    }

    private List<String> parseJSON(String json) {
        List<String> cityNames = new ArrayList<>();
        try {
            JSONArray jsonArray = new JSONArray(json);
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String cityName = jsonObject.getString("name");
                cityNames.add(cityName);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return cityNames;
    }
}

```

Σε αυτό το παράδειγμα, το RadioGroup γεμίζει με επιλογές από τα δεδομένα της υπηρεσίας. Όταν επιλέγεται ένας RadioButton, εμφανίζεται ένα Toast μήνυμα με την επιλογή της πόλης. Η κλάση OkHttpClientHandler χρησιμοποιείται για να αναλάβει την ασύγχρονη αίτηση HTTP με τη χρήση του OkHttp. Παρακαλώ, βεβαιωθείτε ότι έχετε προσθέσει την εξάρτηση του OkHttp στο αρχείο build.gradle.

**Ένα παράδειγμα ενός spinner το οποίο γεμίζει από μια απομακρυσμένη υπηρεσία (πχ <http://IP/cities/getCities.php>) η οποία επιστρέφει σε μορφή JSON αρχείο τα περιεχόμενα ενός πίνακα Cities (name, monument, country, image). Αφού γεμίσει, επιλέγω και πατάω ένα κουμπί. Μετά από αυτό εμφανίζει τι επέλεξα στην οθόνη.**

Παρακάτω είναι ένα παράδειγμα κώδικα που χρησιμοποιεί έναν Spinner για να γεμίσει επιλογές από μια απομακρυσμένη υπηρεσία που επιστρέφει δεδομένα σε μορφή JSON. Όταν επιλέγετε μια επιλογή και πατάτε το κουμπί, εμφανίζεται η επιλεγμένη επιλογή στην οθόνη:

1. Layout (activity\_main.xml):

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/showResultButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Result" />

    <TextView
        android:id="@+id/resultTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Selected Item: " />

</LinearLayout>

```

2. Java Code (MainActivity.java): activity\_main.xml Για αυτό το παράδειγμα, θα χρησιμοποιήσουμε τη βιβλιοθήκη OkHttp για να κάνουμε το αίτημα HTTP και τη βιβλιοθήκη Gson για να αναλύσουμε τα δεδομένα JSON.

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import com.google.gson.Gson;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class MainActivity extends AppCompatActivity {

```



```

private Spinner spinner;
private Button showResultButton;
private TextView resultTextView;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    spinner = findViewById(R.id.spinner);
    showResultButton = findViewById(R.id.showResultButton);
    resultTextView = findViewById(R.id.resultTextView);

    // Κάλεσε τη μέθοδο για να ανακτήσεις τα δεδομένα JSON
    // από την απομακρυσμένη υπηρεσία
    fetchDataFromRemoteService();

    // Ενέργεια κλικ κουμπιού
    showResultButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String selectedOption = spinner.getSelectedItem().toString();
            resultTextView.setText("Selected Item: " + selectedOption);
        }
    });
}

// Υποθετική μέθοδος για ανάκτηση δεδομένων JSON από την απομακρυσμένη υπηρεσία
private void fetchDataFromRemoteService() {
    OkHttpClient client = new OkHttpClient();
    Request request = new Request.Builder()
        .url("http://195.251.211.64/cities/getCities.php")
        .build();

    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            e.printStackTrace();
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            if (response.isSuccessful()) {
                String responseData = response.body().string();
                processJsonData(responseData);
            }
        }
    });
}

// Υποθετική μέθοδος για επεξεργασία των δεδομένων JSON και γέμισμα του Spinner
private void processJsonData(String jsonData) {

```

```

Gson gson = new Gson();
City[] cities = gson.fromJson(jsonData, City[].class);

List<String> cityNames = new ArrayList<>();
for (City city : cities) {
    cityNames.add(city.name);
}

runOnUiThread(new Runnable() {
    @Override
    public void run() {
        ArrayAdapter<String> adapter = new ArrayAdapter<>(MainActivity.this,
            android.R.layout.simple_spinner_dropdown_item, cityNames);
        spinner.setAdapter(adapter);
    }
});
}

// Υποκλάση για αναπαράσταση των δεδομένων JSON
private static class City {
    String name;
    String monument;
    String country;
    String image;
}
}

```

Σε αυτό το παράδειγμα, η μέθοδος **fetchDataFromRemoteService()** καλείται για να ανακτήσει τα δεδομένα JSON από την απομακρυσμένη υπηρεσία. Τα δεδομένα JSON αναλύονται με τη βοήθεια της βιβλιοθήκης Gson, και η λίστα των ονομάτων των πόλεων γεμίζει στον Spinner.

Θυμηθείτε ότι πρέπει να προσθέσετε τις απαραίτητες εξαρτήσεις στο αρχείο build.gradle για τη χρήση του OkHttp και του Gson.

## Κώδικες που πρέπει να γνωρίζω

**Δημιουργία ενός RadioGroup με RadioButtons.** Η μέθοδος δέχεται το RadioGroup και μια λίστα και παίρνει στοιχεία της λίστας και τα προσθέτει στο RadioGroup ως RadioButton.

```

// Μέθοδος που θα χρησιμοποιήσω παραπάνω.
// Δέχεται το radiobutton και την λίστα που θέλουμε να προσθέσουμε
private void createRadioButtons(RadioGroup rg, List<String> mList) {

    // Παίρνω το μέγεθος της λίστας. Αυτό μπορεί να μπει και κατευθείαν στην κάτω
    int count = mList.size();

    // Δημιουργώ ένα radio button (μέσα στην αγγύλη μπορώ

```

```

// να βάλω κατευθείαν το μέγεθος
RadioButton rb[] = new RadioButton[count];

// Κατασκευή του radiobutton-radiogroup
for (int i=0 ; i<mList.size(); i++){
    // Φτιάχνω ένα radio button ανάλογα με τι μέγεθος θα έχει η λίστα
    rb[i] = new RadioButton(this);

    // Προσθέτω το κείμενο που πήρα από την λίστα ως κείμενο σε radiobutton
    rb[i].setText(mList.get(i));

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

```

## Κώδικας δημιουργίας spinner

Παρακάτω ο κώδικας αρχικοποιεί μια CarBrandList, ορίζει την προβολή περιεχομένου της δραστηριότητας, δημιουργεί ένα Spinner, το συμπληρώνει με επωνυμίες χρησιμοποιώντας έναν ArrayAdapter και προετοιμάζει μια RadioGroup. Αυτό ρυθμίζει τα στοιχεία διεπαφής χρήστη για τη δραστηριότητα, επιτρέποντας στον χρήστη να επιλέξει μια μάρκα αυτοκινήτου από το Spinner και να αλληλεπιδράσει με το RadioGroup.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    // Στο αντικείμενο cbl περνάω το αρχείο με τα brands με την getAssets()
    cbl = new CarBrandList(myIP);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Δημιουργούμε το Spinner και το γεμίζουμε με τα brands
    Spinner dropDown = (Spinner) findViewById(R.id.cars);
    // Δημιουργεί έναν ArrayAdapter που θα χρησιμοποιηθεί για τη συμπλήρωση
    // του Spinner με δεδομένα. Ο ArrayAdapter αρχικοποιείται με την τρέχουσα
    // δραστηριότητα (this), τον πόρο διάταξης για κάθε στοιχείο Spinner
    // και τη λίστα των brands που ελήφθησαν από το CarBrandList cbl
    // χρησιμοποιώντας τη μέθοδο getAllBrands.
    ArrayAdapter<String> arrayAdapter =
        new ArrayAdapter<String>(this,
            android.R.layout.simple_spinner_dropdown_item,
            cbl.getAllBrands());
    dropDown.setAdapter(arrayAdapter);

    // Αρχικοποιεί ένα γραφικό στοιχείο RadioGroup που ονομάζεται rg
    // βρίσκοντας την προβολή με το αναγνωριστικό rg από τη διάταξη
    // της δραστηριότητας
    rg = (RadioGroup) findViewById(R.id.rg);
}

```

## Μια μέθοδος που κάνει μια ενέργεια όταν πατηθεί το κουμπί μετά την επιλογή του RadioButton.

Στον παρακάτω κώδικα δημιουργείται το RadioGroup με τα RadioButtons και με το που επιλέγεται κάποιο και πατιέται ένα κουμπί, το εμφανίζει στην οθόνη.

```
// 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(); // το αδειάζουμε

    // Κατασκευή του radiobutton (όπως ο παραπάνω κώδικας).
    for(int i=0;i<allModels.size();i++) {

        // Φτιάχνω ένα radio button ανάλογα με τι μέγεθος θα έχει η λίστα
        RadioButton rb = new RadioButton(this);

        // Προσθέτω το κείμενο που πήρα από την λίστα ως κείμενο σε radiobutton
        rb.setText(allModels.get(i));

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

Αμέσως παρακάτω, μπορώ να λάβω την επιλογή από το RadioGroup και να το εμφανίσω στην οθόνη.

```
// Φτιάχνω έναν Listener που θα τον υλοποιήσω εδώ μέσα
rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup radioGroup, int checkedId) {
        // Φτιάχνω ένα αντικείμενο Radio button για να βρεί τι επέλεξα
        RadioButton rb = (RadioButton) findViewById(checkedId);
        // εμφανίζει ένα toast με το μοντέλο που επιλέχθηκε
        Toast.makeText(getApplicationContext(),
            "The selected model is " + brand + ": " + rb.getText(),
            Toast.LENGTH_SHORT).show();
    }
}
```

```
}); // Εδώ κλείνει ο Listener  
} // Εδώ κλείνει η μεθοδος
```

Όταν έχω να λάβω δεδομένα από μια υπηρεσία, κάνω τις αλλαγές που αναφέρθηκαν παραπάνω. Δηλαδή:

- Ενημέρωση του Android Manifest:  
Προσθέστε την άδεια INTERNET στο αρχείο AndroidManifest.xml στο σημείο πριν το **application**:

```
<uses-permission android:name="android.permission.INTERNET" />  
  
<application  
    android:allowBackup="true"
```

ενώ τελευταία γραμμή πριν το activity να εμφανίζονται οι γραμμές:

```
    android:networkSecurityConfig="@xml/network_security_config"  
    android:usesCleartextTraffic="true">  
  
    <meta-data  
        android:name="com.google.android.actions"  
        android:resource="@xml/network_security_config" />  
  
    <activity
```

- Δημιουργία του αρχείου **network\_security\_config.xml**:

Δημιουργήστε ένα νέο φάκελο με όνομα res/xml στον φάκελο του προγράμματος (New- → Folder → XML Resources Folder). Εν συνεχεία, δημιουργήστε ένα αρχείο network\_security\_config.xml μέσα σε αυτόν τον φάκελο. Το περιεχόμενο του αρχείου θα είναι κάτι παρόμοιο με αυτό (η IP θα είναι αυτή που έχετε στον υπολογιστή σας):

```
<?xml version="1.0" encoding="utf-8"?>  
<network-security-config>  
    <domain-config cleartextTrafficPermitted="true">  
        <domain includeSubdomains="true">192.168.1.104</domain>  
    </domain-config>  
</network-security-config>
```

Αυτό το αρχείο ρυθμίζει την πολιτική ασφαλείας του δικτύου για την εφαρμογή σας.

- Βεβαιωθείτε ότι έχετε προσθέσει τις απαραίτητες εξαρτήσεις στο αρχείο **build.gradle ((Module App))** για να χρησιμοποιήσετε τη βιβλιοθήκη **OkHttp**.

```
dependencies {  
    implementation 'com.squareup.okhttp3:okhttp:4.9.0'  
    implementation 'androidx.appcompat:appcompat:1.4.1'  
    ...  
}
```

- Και για εικόνες για να συμπεριλάβετε τη βιβλιοθήκη Picasso στην εφαρμογή, πρέπει να προσθέσετε την αντίστοιχη εξάρτηση (dependency) στο αρχείο **build.gradle ((Module App))** του project.

- Άνοιξε το αρχείο build.gradle που βρίσκεται στον κύριο φάκελο του project (όχι στο φάκελο app).

- Στην ενότητα dependencies, προσθέστε την ακόλουθη γραμμή για να προσθέσετε την εξάρτηση της βιβλιοθήκης Picasso:

```
implementation 'com.squareup.picasso:picasso:2.5.2'
```

Δεν ξεχνάμε να επιλέξουμε Sync Now.

1. Αρχεία που πρέπει να έχω:

- Πρέπει να δημιουργήσουμε μια κλάση **OkHttpHandler**(OkHttpHandler.java).

Εδώ είναι ένα παράδειγμα κώδικα που χρησιμοποιεί τη βιβλιοθήκη OkHttp για να κάνει ένα απλό GET αίτημα:

```
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();
try {
    // Δημιουργεί ένα JSONObject αναλύοντας τα δεδομένα απόκρισης
    // που ανακτήθηκαν.
    JSONObject json = new JSONObject(data);

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

    // Επαναλαμβάνεται πάνω από κάθε κλειδί στο JSONObject.
    while(keys.hasNext()) {
        // Ανακτά το επόμενο κλειδί από τον επαναλήπτη,
        // που αντιπροσωπεύει την επωνυμία.
        String brand = keys.next();

        // Ανακτά την αντίστοιχη τιμή για το κλειδί επωνυμίας από το
        // JSONObject και τη μετατρέπει σε συμβολοσειρά.
        String models = json.get(brand).toString();

        // Δημιουργεί ένα νέο αντικείμενο CarBrand χρησιμοποιώντας την
        // επωνυμία και μοντελοποιεί τιμές και το προσθέτει στη cbList.
        cbList.add(new CarBrand(brand, models));
    }
} catch (JSONException e) {
    e.printStackTrace();
}

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

```

- Ακολουθεί το αρχείο MainActivity.java από το παραπάνω OkHttpClientHandler.java.

```

package com.example.carpicker;

import androidx.appcompat.app.*;
import android.os.*;
import android.view.*;

```



```

import android.widget.*;
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;

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

    // 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(); // το αδειάζουμε

        // Κατασκευή του radiobutton-radiogroup
        for(int i=0;i<allModels.size();i++) {
            RadioButton rb = new RadioButton(this); // Φτιάχνω ένα radio button ανάλογα
            rb.setText(allModels.get(i)); // Προσθέτω το κείμενο που πήρα από την λίστα
            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 + "/carsDBServices/logHistory.php?brand="
        + brand + "&model=" + rb.getText().toString() + "&timestamp="
        + new Date(System.currentTimeMillis()).toString();

        try {
            // Δημιουργεί αντικείμενο μιας προσαρμοσμένης κλάσης OkHttpHandler
            // η οποία φαίνεται να χειρίζεται αιτήματα HTTP χρησιμοποιώντας
            // τη βιβλιοθήκη OkHttp.
            OkHttpHandler okHttpHandler = new OkHttpHandler();

            // Επικαλείται τη μέθοδο logHistory του αντικειμένου OkHttpHandler
            // περνώντας τη συμβολοσειρά URL που δημιουργήθηκε ως παράμετρο.
            // Αυτό υποδηλώνει ότι η μέθοδος logHistory είναι υπεύθυνη
            // για την αποστολή ενός αιτήματος HTTP στην καθορισμένη διεύθυνση
            // URL, πιθανόν για την καταγραφή της επιλεγμένης επωνυμίας,
            // μοντέλου και χρονικής σφραγίδας.
            okHttpHandler.logHistory(url);

            // Εμφανίζει ένα μήνυμα toast μικρής διάρκειας που υποδεικνύει
            // ότι η επιλογή καταγράφηκε με επιτυχία.
            Toast.makeText(getApplicationContext(), "Selection Logged",
            Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

}

// Παρακάτω ο κώδικας αρχικοποιεί μια CarBrandList, ορίζει την προβολή περιεχομένου
// της δραστηριότητας, δημιουργεί ένα Spinner, το συμπληρώνει με επωνυμίες
// χρησιμοποιώντας έναν ArrayAdapter και προετοιμάζει μια RadioGroup.
// Αυτό ρυθμίζει τα στοιχεία διεπαφής χρήστη για τη δραστηριότητα, επιτρέποντας
// στον χρήστη να επιλέξει μια μάρκα αυτοκινήτου από το Spinner και να
// αλληλεπιδράσει με το RadioGroup.

@Override
protected void onCreate(Bundle savedInstanceState) {
    // Στο αντικείμενο cbl περνάω το αρχείο με τα brands με την getAssets()
    cbl = new CarBrandList(myIP);
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Δημιουργούμε το Spinner και το γεμίζουμε με τα brands
    Spinner dropDown = (Spinner) findViewById(R.id.cars);

```

```
// Δημιουργεί έναν ArrayAdapter που θα χρησιμοποιηθεί για τη συμπλήρωση
// του Spinner με δεδομένα. Ο ArrayAdapter αρχικοποιείται με την τρέχουσα
// δραστηριότητα (this), τον πόρο διάταξης για κάθε στοιχείο Spinner και
// τη λίστα των brands που ελήφθησαν από το CarBrandList cbl
// χρησιμοποιώντας τη μέθοδο getAllBrands.
```

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

```
// Αρχικοποιεί ένα γραφικό στοιχείο RadioGroup που ονομάζεται rg
// βρίσκοντας την προβολή με το αναγνωριστικό rg από τη διάταξη της
// δραστηριότητας
```

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

```
}
```

```
}
```