

# Programiranje za suvremene procesore

---

*Programiranje za Android*

*TkoMeSmeta*

Tin Babić i Krešimir Sinković

## Uvod

Kad god na mobitelu dobijemo poziv s telefonskog broja koji se nalazi u mobitelovom imeniku, mobitel ispisuje ime osobe koja nas zove. No ako broj nije u imeniku, tada uskače naša aplikacija pokrećući pretraživanje telefonskog imenika dostupnog na internetu. Osim što ispisuje pronađeno ime pozivatelja u trenutku poziva, aplikacija također može to ime upisati u mobitelov imenik (ako to odobrimo). Također, aplikacija može u bilo kojem drugom trenutku pronaći podatke o telefonskom broju koji upišemo preko virtualne tipkovnice i isto tako pronađene podatke upisati u imenik ako to želimo. Aplikacija može pogledati kroz povijest poziva i prepoznati sve nepoznate brojeve te pretražiti online imenik.

## Kako koristiti

Pri prvom pokretanju aplikacije pokreće se *BroadcastReceiver* koji u pozadini sluša i pokreće se kada imamo dolazni poziv. Kada dođe do dolaznog poziva aplikacija se spaja s online imenikom i traži broj. Kada je pretraga završena pojavljuje se *Notification* s primljenim podacima. Pritiskom na *Notification* pokreće se aplikacija i daje nam mogućnost upisa pronađenog broja u kontakte.

Glavni zaslon aplikacije nam daje mogućnost unosa broja kojeg želimo pretražiti. Pritiskom na „*Find online*“ gumb traži se na online imeniku, a pritiskom na „*Contacts*“ gumb traži se u internom imeniku mobitela.

Ako kliknemo gumb „*Call list*“ prikazuje se lista svih nepoznatih dolaznih i propuštenih poziva. Pritiskom na jedan od elemenata liste pretražuje se online imenik i ako je pronađen *AlertDialog* nam nudi upis u interni imenik.

## Implementacija

### Contact.java

Ovo je pomoćna klasa koja nudi korisniku pretraživanje imenika (online i internog) te upisivanje u interni imenik. Konstruktor traži kontekst i jedan Handler. Handler nam treba kako bi napravili nešto kad poziv funkcija pretraživanja završe s radom jer se odvijaju u zasebnim dretvama.

```
private Handler handler = new Handler() {  
  
    @Override  
    public void handleMessage(Message msg) {  
        switch (msg.what) {  
            case 0:  
                // Radi nesto dok završi funkcija findOnline  
                break;  
            case 1:  
                //Radi nesto dok završi funkcija findContact  
                break;  
        }  
    }  
};
```

### Funkcija findOnline

Funkcija koristi vanjsku biblioteku *Jsoup* koja nam služi za lako pretraživanje html stranica. Stranicu koju smo izabrali za pretraživanje je *imenik.hr*. Sa *Jsoup.connect()* se spajamo na stranicu, a sa *doc.select()* biramo element na kojem bi trebalo biti zapisano ime kontakta.

```
public void findOnline(String broj){  
    if (broj.contains("+")) broj = "0" + broj.substring(4, broj.length());  
    final String br = broj;  
  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            try {  
                Document doc =  
Jsoup.connect("http://www.imenik.hr/imenik/trazi/1/"+br+".html").get();  
                Elements ele = doc.select("div.telefon > a");  
  
                if(ele.isEmpty()) {  
                    _kontakt=_context.getResources().getString(R.string.Not_found);  
                }  
                else {  
                    _kontakt=ele.first().attr("title").toString();  
                }  
  
            } catch (Exception e) {  
                _kontakt=_context.getResources().getString(R.string.Not_found);  
            }  
            _handler.sendMessage(0);  
        }  
    }).start();  
}
```

## MyCallReceiver.java

Ovo je *BroadcastReceiver* koji sluša intente i kada registrira intent koji govori da je trenutno u stanju dolaznog poziva pretražuje online dolazni broj i nakon toga generira *Notification* s primljenim podacima.

```
@Override
public void onReceive(Context context, Intent intent) {
    String stateStr = intent.getExtras().getString(TelephonyManager.EXTRA_STATE);
    final String number =
intent.getExtras().getString(TelephonyManager.EXTRA_INCOMING_NUMBER);
    final Context ctx = context;
    if (stateStr.equals(TelephonyManager.EXTRA_STATE_RINGING)) {
        ContentResolver cr = context.getContentResolver();
        Uri uri = Uri.withAppendedPath(ContactsContract.PhoneLookup.CONTENT_FILTER_URI,
Uri.encode(number));
        Cursor cursor = cr.query(uri, new String[]{ContactsContract.PhoneLookup.DISPLAY_NAME},
null, null, null);
        if (cursor == null) {
            return;
        }
        if (!cursor.moveToFirst()) {

            Handler nadji = new Handler() {

                @Override
                public void handleMessage(Message msg) {
                    switch (msg.what) {
                        case 0:
                            Intent notificationIntent = new Intent(ctx, MainActivity.class);
                            Bundle bundle = new Bundle();
                            bundle.putString("number", number);
                            bundle.putString("name", pretraga._kontakt);
                            notificationIntent.putExtras(bundle);
                            PendingIntent contentIntent = PendingIntent.getActivity(ctx,
                                0, notificationIntent,
                                PendingIntent.FLAG_CANCEL_CURRENT);

                            NotificationCompat.Builder builder = new
NotificationCompat.Builder(

                                ctx)
                                    .setContentIntent(contentIntent)
                                    .setSmallIcon(R.mipmap.ic_launcher)

                            .setContentTitle(ctx.getResources().getString(R.string.Number_to_find) + number)
                                    .setContentText(pretraga._kontakt);

                            NotificationManager notificationmanager = (NotificationManager)
ctx

                                    .getSystemService(Context.NOTIFICATION_SERVICE);
                            notificationmanager.notify(0, builder.build());
                            break;
                        case 1:
                            //Radi nesto dok završi funkcija findContact
                            break;
                    }
                }
            };
            pretraga = new Contact(context, nadji);
            pretraga.findOnline(number);
        }
    }
}
```

## MainActivity.java

Glavna aktivnost koja se aktivira pokretanjem aplikacije.

### Funkcija onResume

Ovo je funkcija koja prima podatke iz *Notificationa* kojeg je naš *BroadcastReceiver* poslao i daje nam opciju upisa tog broja u interni imenik.

```
@Override
public void onResume() {
    super.onResume();
    Bundle b = getIntent().getExtras();
    if (b != null) {
        if (b.containsKey("number")) {
            number = b.getString("number");
            kontakt = b.getString("name");
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            if (!kontakt.equals(getResources().getString(R.string.Not_found))) {
                builder.setPositiveButton(R.string.Ok, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        contact.addContact(kontakt, number);
                    }
                });
            }
            builder.setTitle(R.string.Add_contact)
                .setMessage(getResources().getString(R.string.Number_to_add) + ": " +
                    number + "\n" + getResources().getString(R.string.Name) + ": " + kontakt)
                .setNegativeButton(R.string.Cancel, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                    }
                });
            AlertDialog alert = builder.create();
            alert.show();
        }
    }
}
```