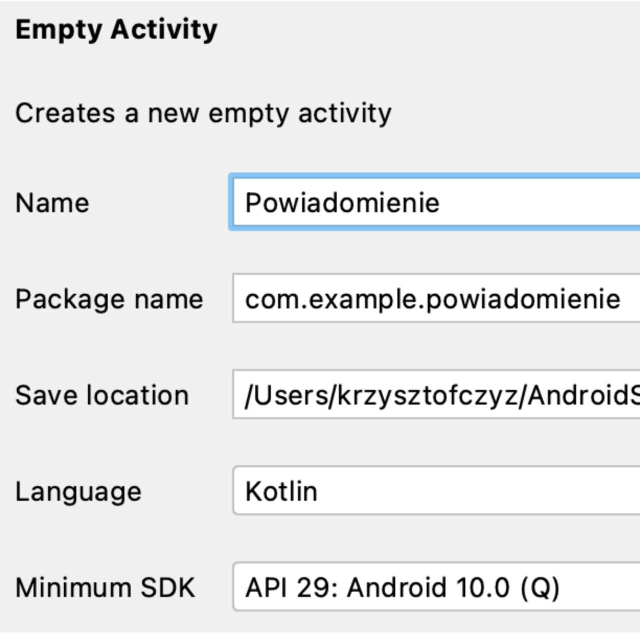
Powiadomienie we własnej aplikacji

Przykładem własnego powiadomienia jest np. aplikacja do gotowania jajek. Ustawiamy czas i datę i otrzymujemy powiadomienie.

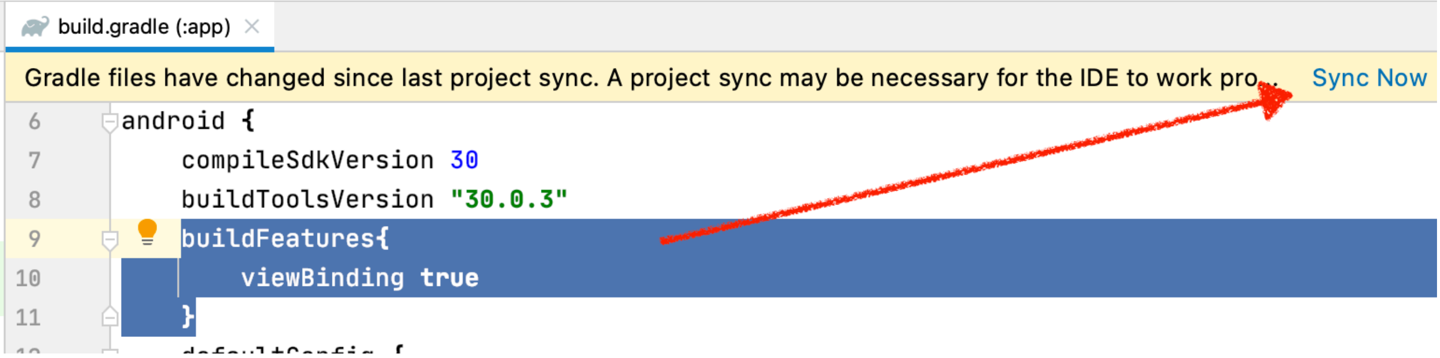
Tworzymy nowy projekt z pustą aktywnością oraz implementujemy viewBinding:



Empty Activity

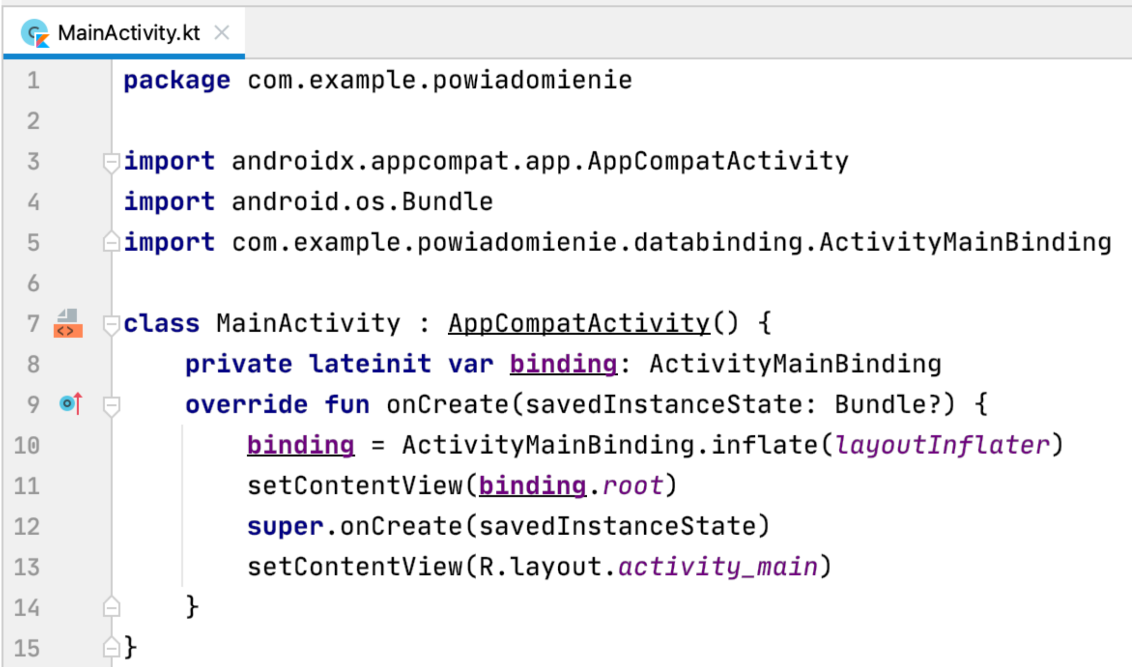
Creates a new empty activity

Name	Powiadomienie
Package name	com.example.powiadomienie
Save location	/Users/krzysztofczyz/AndroidS
Language	Kotlin
Minimum SDK	API 29: Android 10.0 (Q)



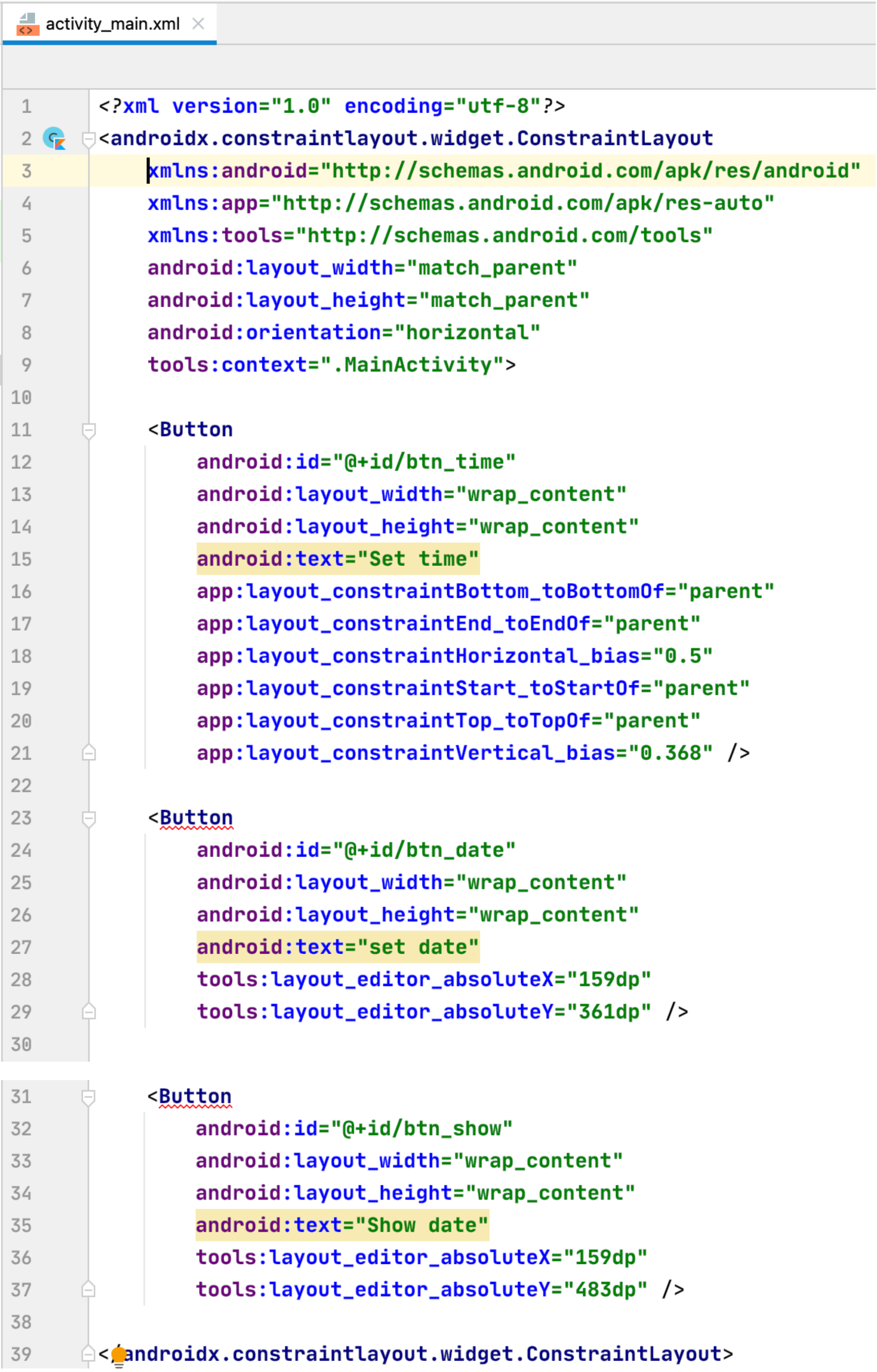
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work pro...

```
android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"
    buildFeatures{
        viewBinding true
    }
    defaultConfig {
```



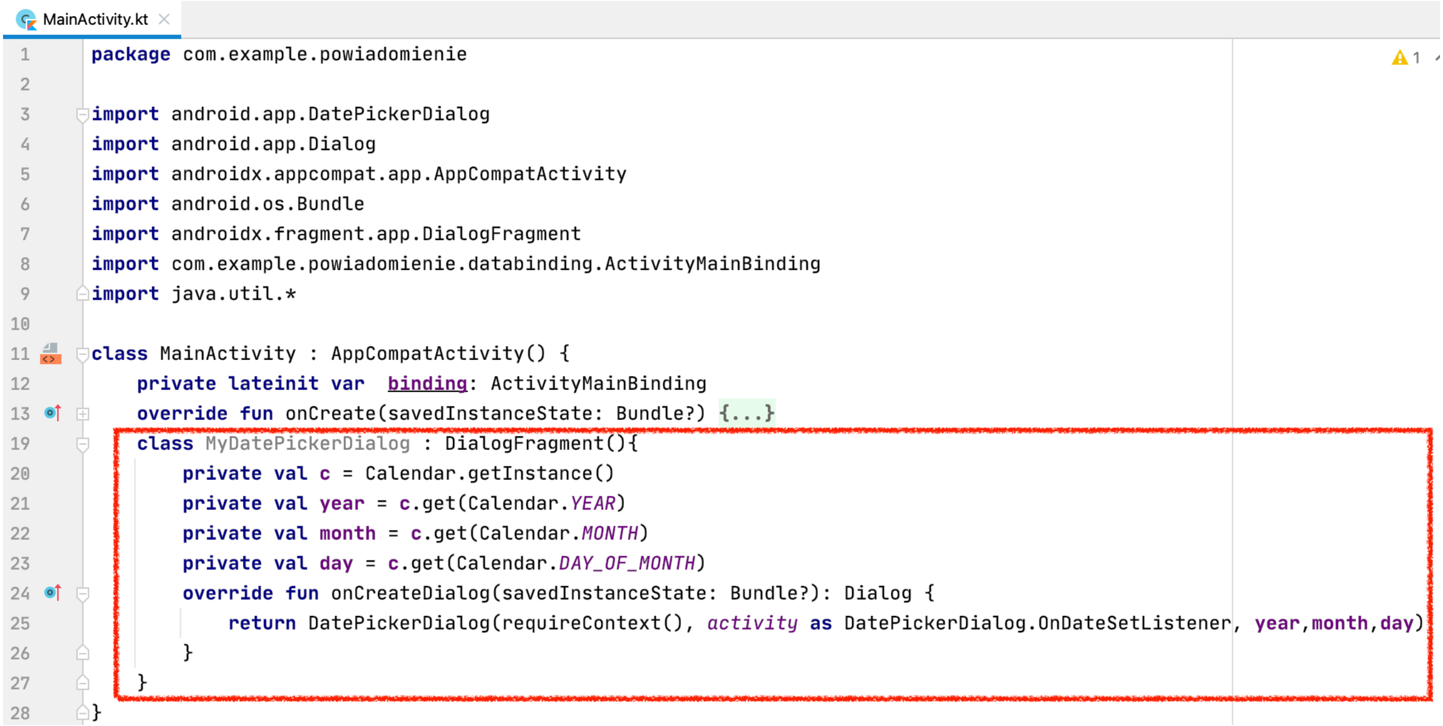
```
package com.example.powiadomienie

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.powiadomienie.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Tworzymy nasz Layout będący interfejsem aplikacji:



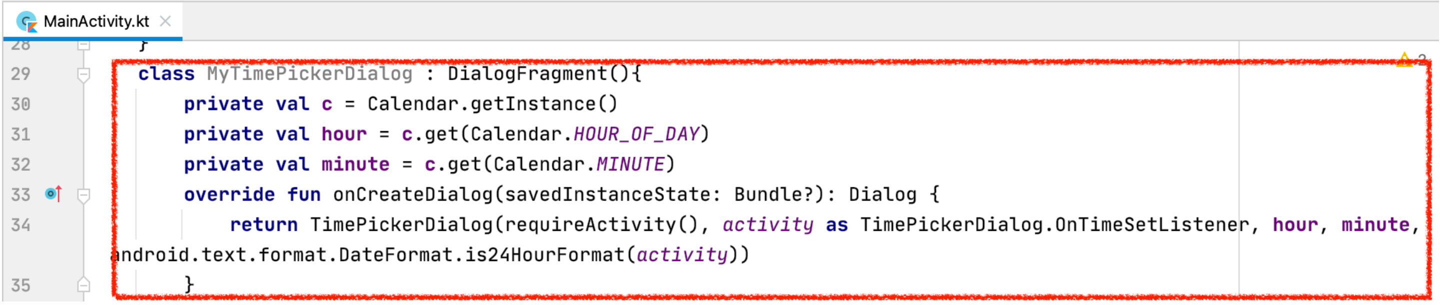
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"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

    <Button
        android:id="@+id/btn_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Set time"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.368" />

    <Button
        android:id="@+id/btn_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="set date"
        tools:layout_editor_absoluteX="159dp"
        tools:layout_editor_absoluteY="361dp" />

    <Button
        android:id="@+id/btn_show"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show date"
        tools:layout_editor_absoluteX="159dp"
        tools:layout_editor_absoluteY="483dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Tworzymy klasy obsługujące datę i godzinę:



MainActivity.kt

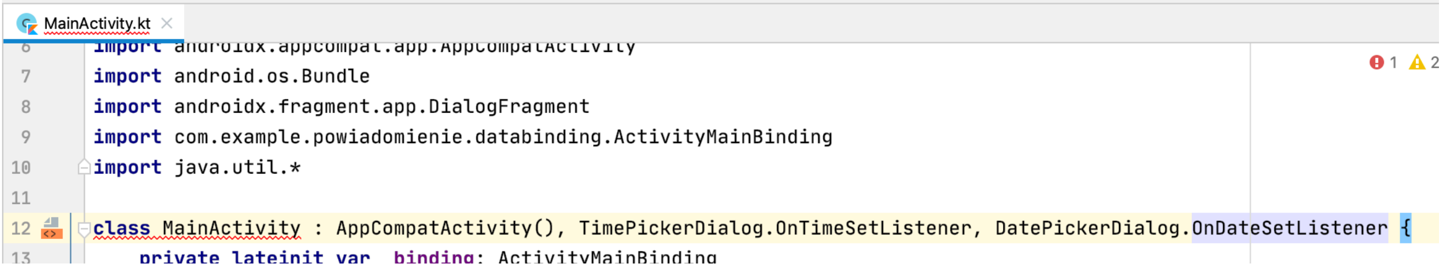
```kotlin
package com.example.powiadomienie

import android.app.DatePickerDialog
import android.app.Dialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.fragment.app.DialogFragment
import com.example.powiadomienie.databinding.ActivityMainBinding
import java.util.*

class MainActivity : AppCompatActivity() {
    private lateinit var  binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {...}
    class MyDatePickerDialog : DialogFragment(){
        private val c = Calendar.getInstance()
        private val year = c.get(Calendar.YEAR)
        private val month = c.get(Calendar.MONTH)
        private val day = c.get(Calendar.DAY_OF_MONTH)
        override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
            return DatePickerDialog(requireContext(), activity as DatePickerDialog.OnDateSetListener, year,month,day)
        }
    }
}
```



MainActivity.kt

```kotlin
    }
    class MyTimePickerDialog : DialogFragment(){
        private val c = Calendar.getInstance()
        private val hour = c.get(Calendar.HOUR_OF_DAY)
        private val minute = c.get(Calendar.MINUTE)
        override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
            return TimePickerDialog(requireActivity(), activity as TimePickerDialog.OnTimeSetListener, hour, minute,
android.text.format.DateFormat.is24HourFormat(activity))
        }
```

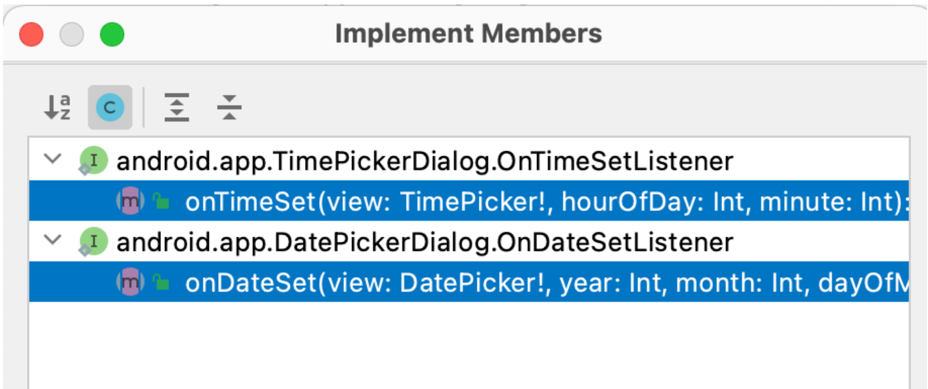
Dodajemy implementację interfejsów:



MainActivity.kt

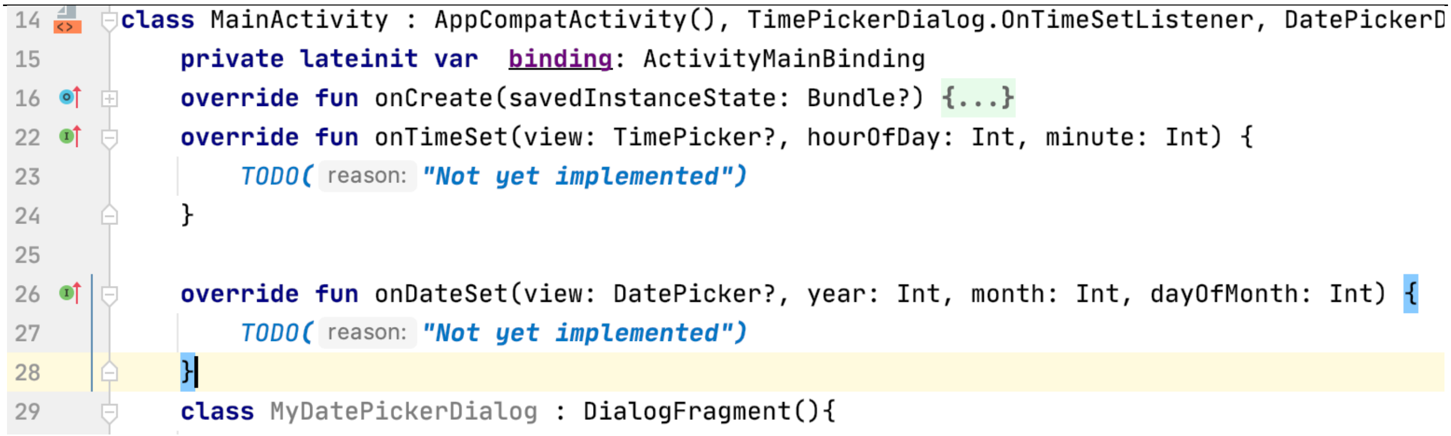
```kotlin
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.fragment.app.DialogFragment
import com.example.powiadomienie.databinding.ActivityMainBinding
import java.util.*

class MainActivity : AppCompatActivity(), TimePickerDialog.OnTimeSetListener, DatePickerDialog.OnDateSetListener {
    private lateinit var  binding: ActivityMainBinding
```

Nadpisujemy dwie metody naszej aktywnoąci:



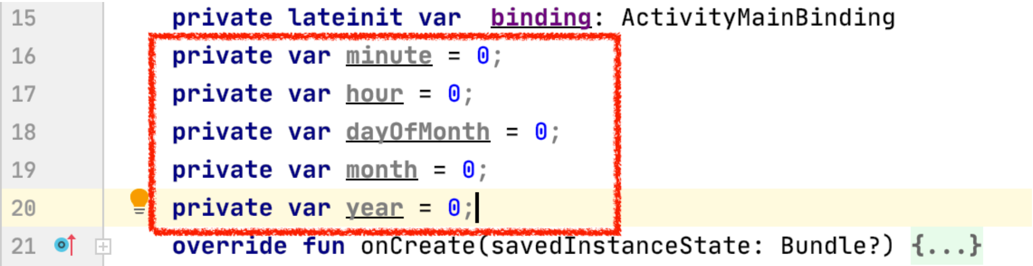
Implement Members

- android.app.TimePickerDialog.OnTimeSetListener
  - onTimeSet(view: TimePicker!, hourOfDay: Int, minute: Int):
- android.app.DatePickerDialog.OnDateSetListener
  - onDateSet(view: DatePicker!, year: Int, month: Int, dayOfM



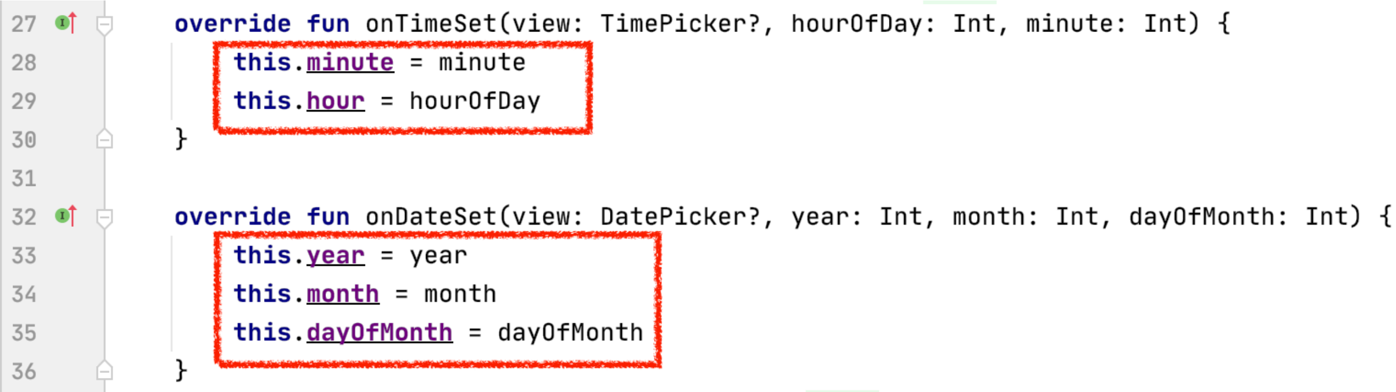
```
class MainActivity : AppCompatActivity(), TimePickerDialog.OnTimeSetListener, DatePickerD
    private lateinit var  binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {...}
    override fun onTimeSet(view: TimePicker?, hourOfDay: Int, minute: Int) {
        TODO( reason: "Not yet implemented")
    }

    override fun onDateSet(view: DatePicker?, year: Int, month: Int, dayOfMonth: Int) {
        TODO( reason: "Not yet implemented")
    }
    class MyDatePickerDialog : DialogFragment(){
```

Dodajemy potrzebne zmienne przechowujące nasze wartości czasowe:



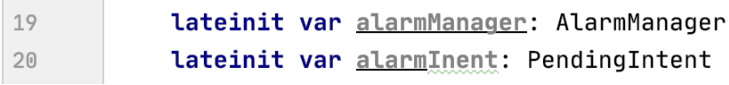
```
    private lateinit var  binding: ActivityMainBinding
    private var minute = 0;
    private var hour = 0;
    private var dayOfMonth = 0;
    private var month = 0;
    private var year = 0;
    override fun onCreate(savedInstanceState: Bundle?) {...}
```

Implementujemy je w klasach:

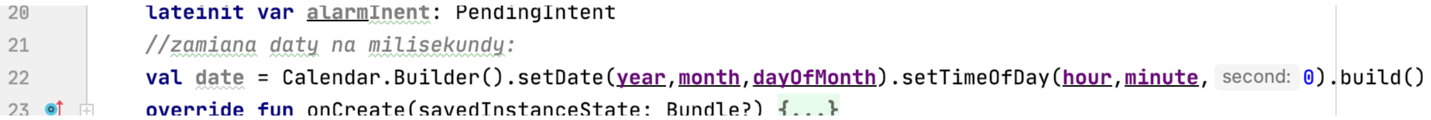


```
    override fun onTimeSet(view: TimePicker?, hourOfDay: Int, minute: Int) {
        this.minute = minute
        this.hour = hourOfDay
    }

    override fun onDateSet(view: DatePicker?, year: Int, month: Int, dayOfMonth: Int) {
        this.year = year
        this.month = month
        this.dayOfMonth = dayOfMonth
    }
```

Tworzymy dodatkowe zmienne przechowujące stan alarmu oraz zamieniające naszą datę na milisekundy:

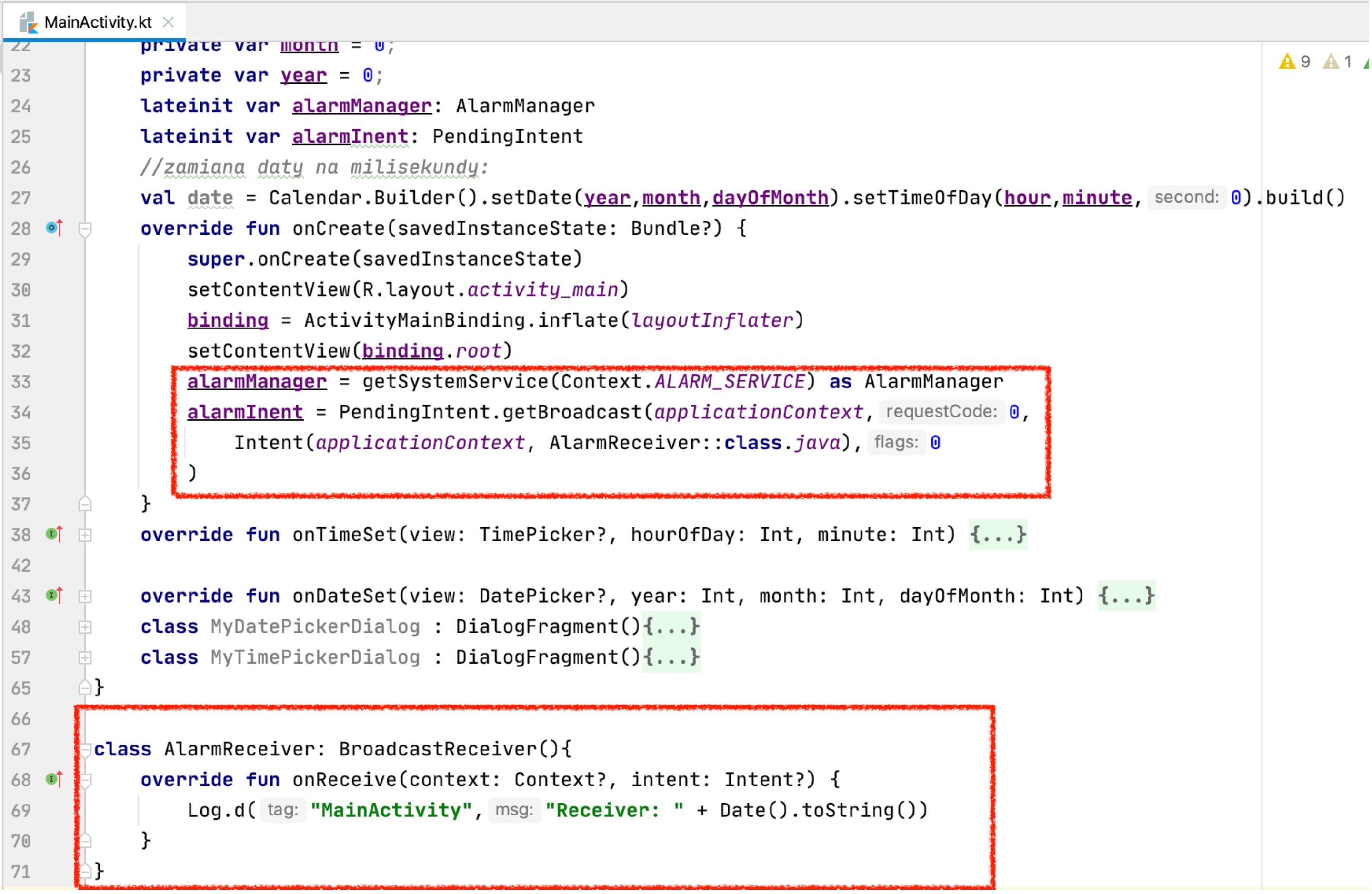


```
    lateinit var alarmManager: AlarmManager
    lateinit var alarmInent: PendingIntent
```



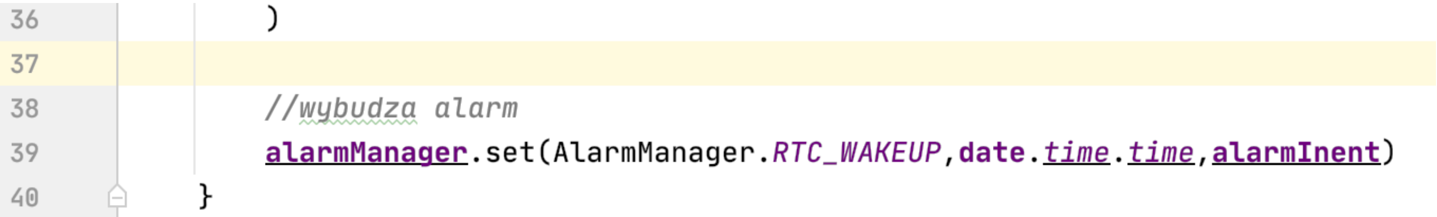
```
    lateinit var alarmInent: PendingIntent
    //zamiana daty na milisekundy:
    val date = Calendar.Builder().setDate(year,month,dayOfMonth).setTimeOfDay(hour,minute, second: 0).build()
    override fun onCreate(savedInstanceState: Bundle?) {...}
```

Tworzymy zmienne oraz klasę pomocniczą



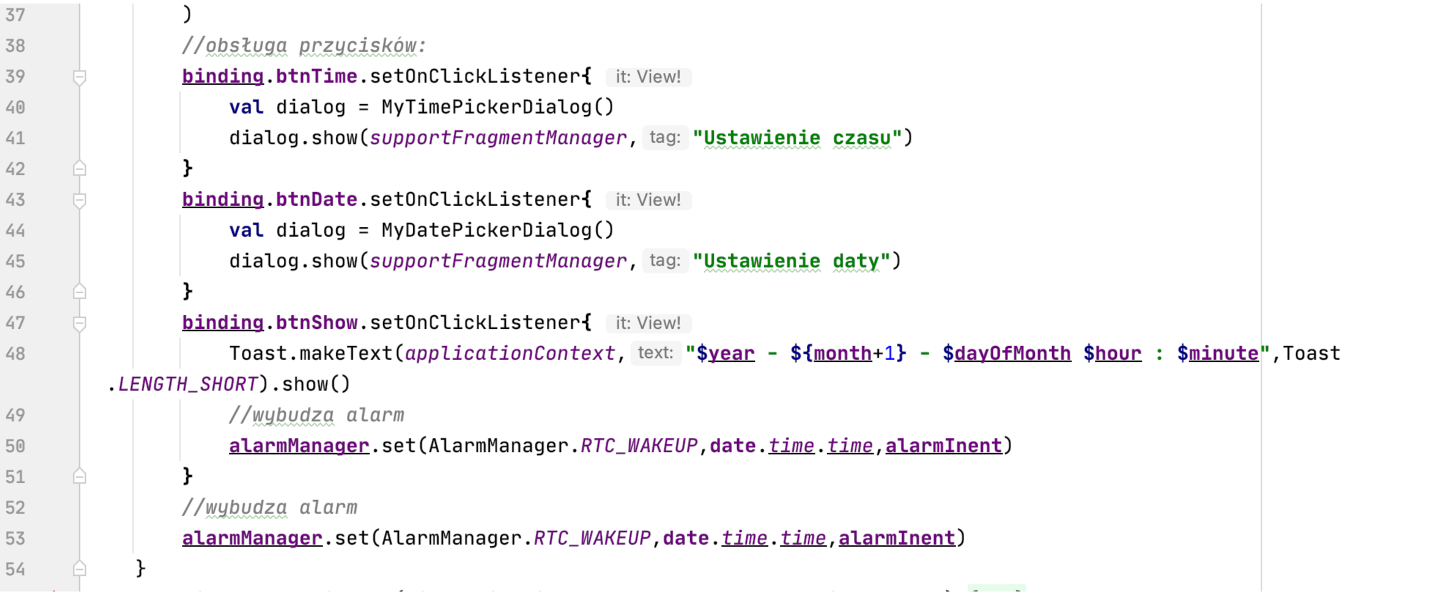
```
    private var month = 0;
    private var year = 0;
    lateinit var alarmManager: AlarmManager
    lateinit var alarmInent: PendingIntent
    //zamiana daty na milisekundy:
    val date = Calendar.Builder().setDate(year,month,dayOfMonth).setTimeOfDay(hour,minute, second: 0).build()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        alarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
        alarmInent = PendingIntent.getBroadcast(applicationContext, requestCode: 0,
            Intent(applicationContext, AlarmReceiver::class.java), flags: 0
        )
    }
    override fun onTimeSet(view: TimePicker?, hourOfDay: Int, minute: Int) {...}

    override fun onDateSet(view: DatePicker?, year: Int, month: Int, dayOfMonth: Int) {...}
    class MyDatePickerDialog : DialogFragment(){...}
    class MyTimePickerDialog : DialogFragment(){...}
}

class AlarmReceiver: BroadcastReceiver(){
    override fun onReceive(context: Context?, intent: Intent?) {
        Log.d( tag: "MainActivity", msg: "Receiver: " + Date().toString())
    }
}
```

Dodajemy wybudzanie alarmu:



```
        )

        //wybudza alarm
        alarmManager.set(AlarmManager.RTC_WAKEUP,date.time.time,alarmInent)
    }
```

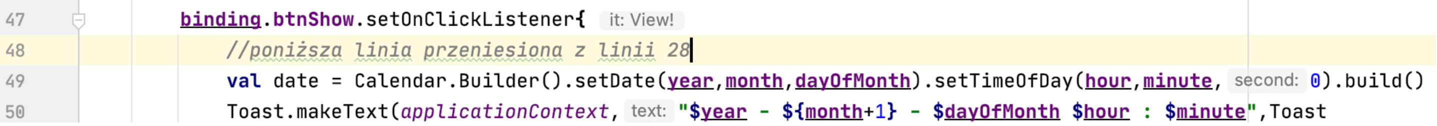
Dodajemy obsługę przycisków:



```
        )
        //obsługa przycisków:
        binding.btnTime.setOnClickListener{  it: View!
            val dialog = MyTimePickerDialog()
            dialog.show(supportFragmentManager, tag: "Ustawienie czasu")
        }
        binding.btnDate.setOnClickListener{  it: View!
            val dialog = MyDatePickerDialog()
            dialog.show(supportFragmentManager, tag: "Ustawienie daty")
        }
        binding.btnShow.setOnClickListener{  it: View!
            Toast.makeText(applicationContext, text: "$year - ${month+1} - $dayOfMonth $hour : $minute",Toast
   .LENGTH_SHORT).show()
            //wybudza alarm
            alarmManager.set(AlarmManager.RTC_WAKEUP,date.time.time,alarmInent)
        }
        //wybudza alarm
        alarmManager.set(AlarmManager.RTC_WAKEUP,date.time.time,alarmInent)
    }
```

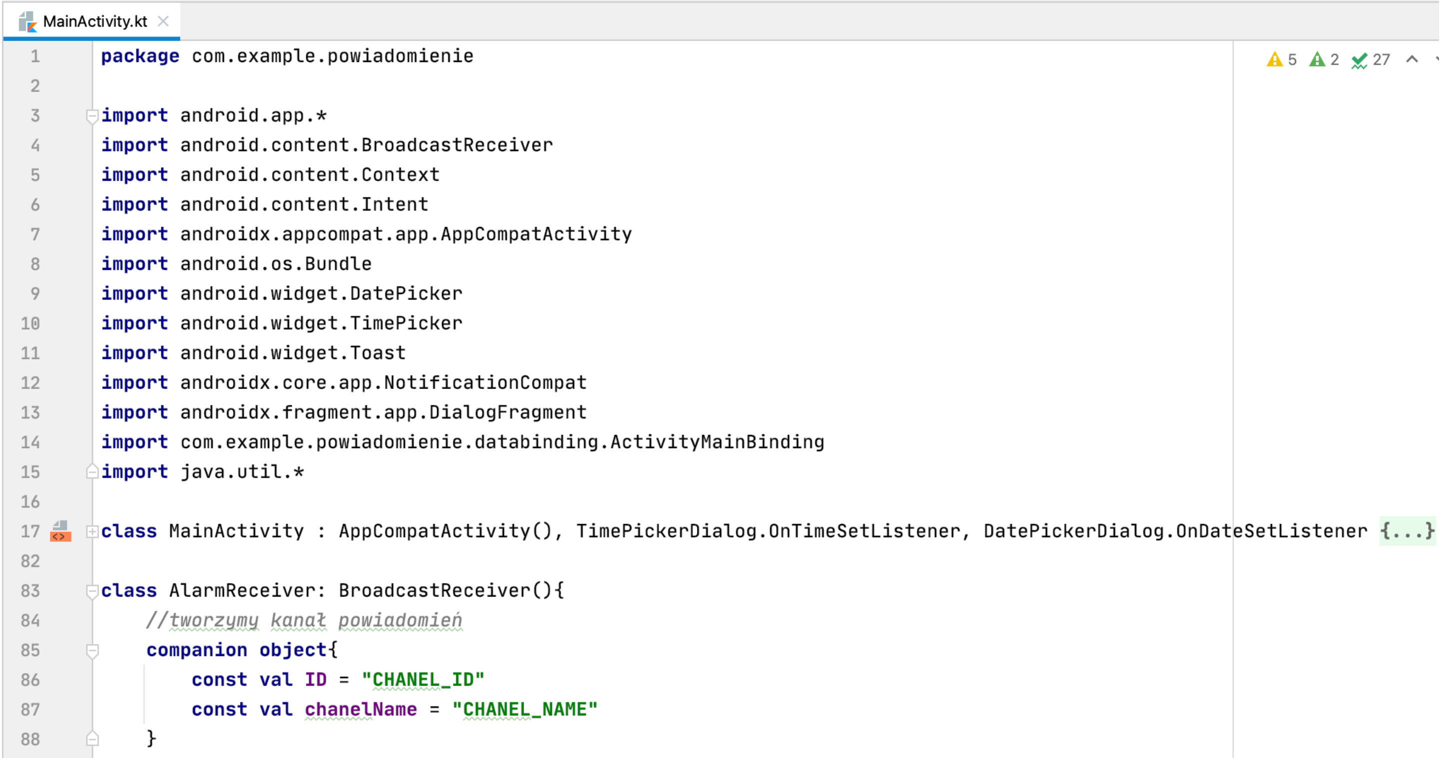
Uzyskaliśmy możliwość wprowadzenia daty i czasu oraz wyświetlenia wpisanych danych.

Przenosimy linię z 28 do obsługi przycisku btn_show:



```
        binding.btnShow.setOnClickListener{ it: View!
            //poniższa linia przeniesiona z linii 28
            val date = Calendar.Builder().setDate(year,month,dayOfMonth).setTimeOfDay(hour,minute, second: 0).build()
            Toast.makeText(applicationContext, text: "$year - ${month+1} - $dayOfMonth $hour : $minute",Toast
```

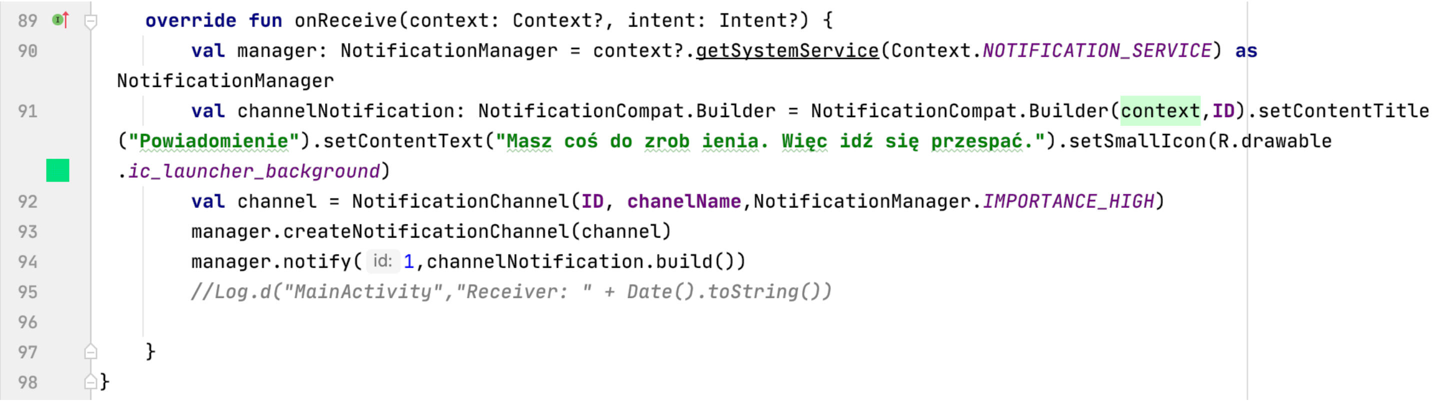
Teraz zadbamy o pojawianie się powiadomień. W klasie AlarmReceiver. Modyfikujemy ją do postaci:



MainActivity.kt

⚠5 ⚠2 ✓27

```
package com.example.powiadomienie

import android.app.*
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.DatePicker
import android.widget.TimePicker
import android.widget.Toast
import androidx.core.app.NotificationCompat
import androidx.fragment.app.DialogFragment
import com.example.powiadomienie.databinding.ActivityMainBinding
import java.util.*

class MainActivity : AppCompatActivity(), TimePickerDialog.OnTimeSetListener, DatePickerDialog.OnDateSetListener {...}

class AlarmReceiver: BroadcastReceiver(){
    //tworzymy kanał powiadomień
    companion object{
        const val ID = "CHANEL_ID"
        const val chanelName = "CHANEL_NAME"
    }
```



```
    override fun onReceive(context: Context?, intent: Intent?) {
        val manager: NotificationManager = context?.getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
        val channelNotification: NotificationCompat.Builder = NotificationCompat.Builder(context,ID).setContentTitle("Powiadomienie").setContentText("Masz coś do zrob ienia. Więc idź się przespać.").setSmallIcon(R.drawable.ic_launcher_background)
        val channel = NotificationChannel(ID, chanelName,NotificationManager.IMPORTANCE_HIGH)
        manager.createNotificationChannel(channel)
        manager.notify( id: 1,channelNotification.build())
        //Log.d("MainActivity","Receiver: " + Date().toString())

    }
}
```

Aplikacja gotowa.