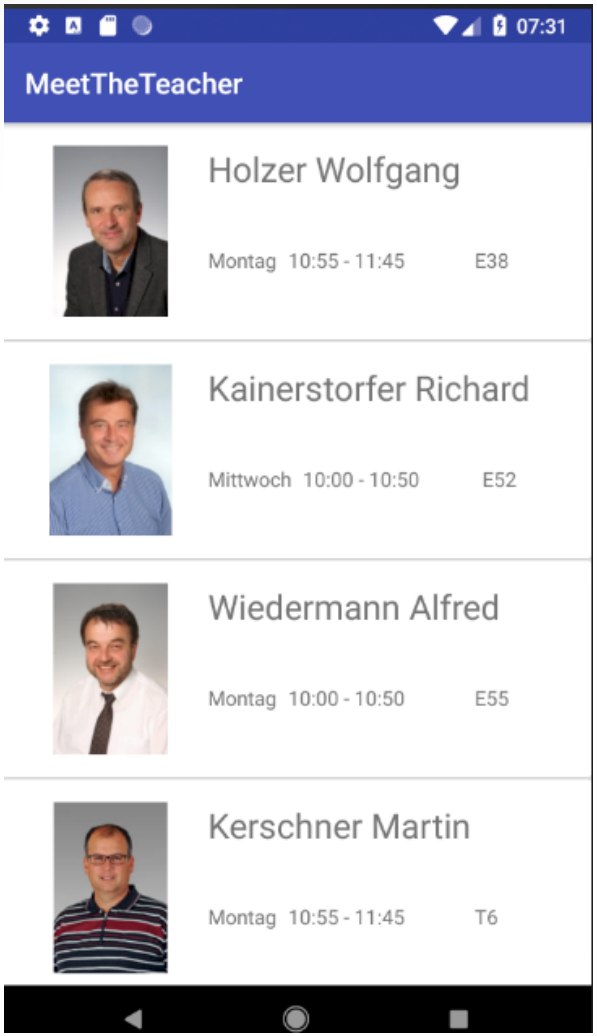
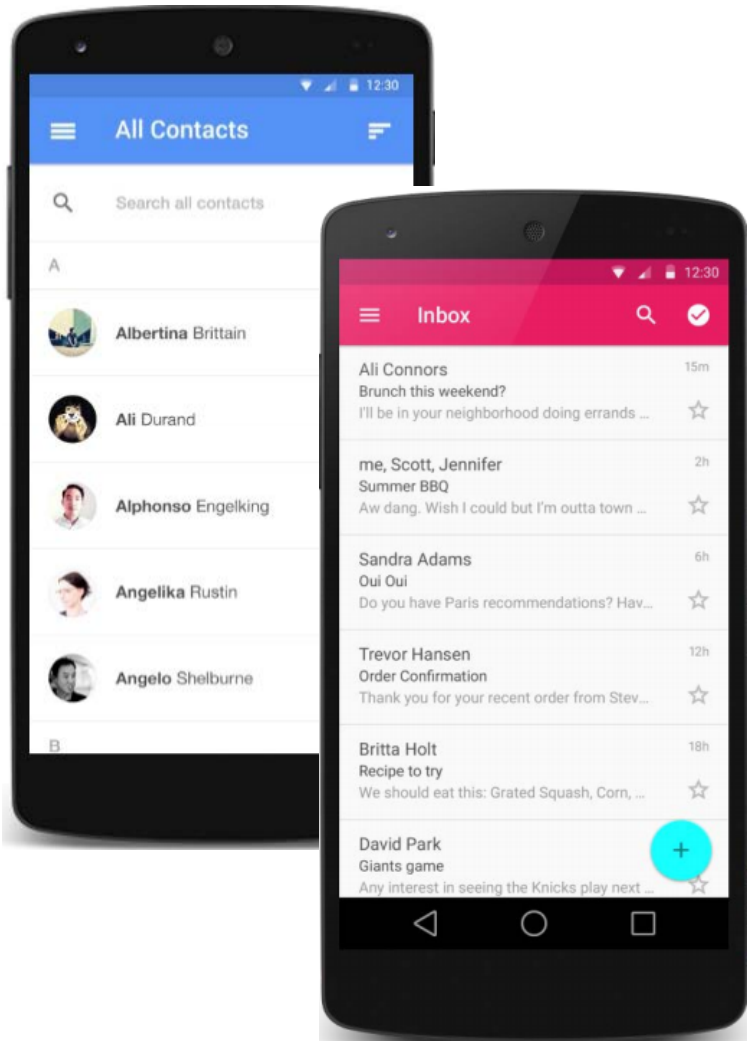
RecyclerView



HTL LEONDING

Was ist eine RecyclerView - Beispiele

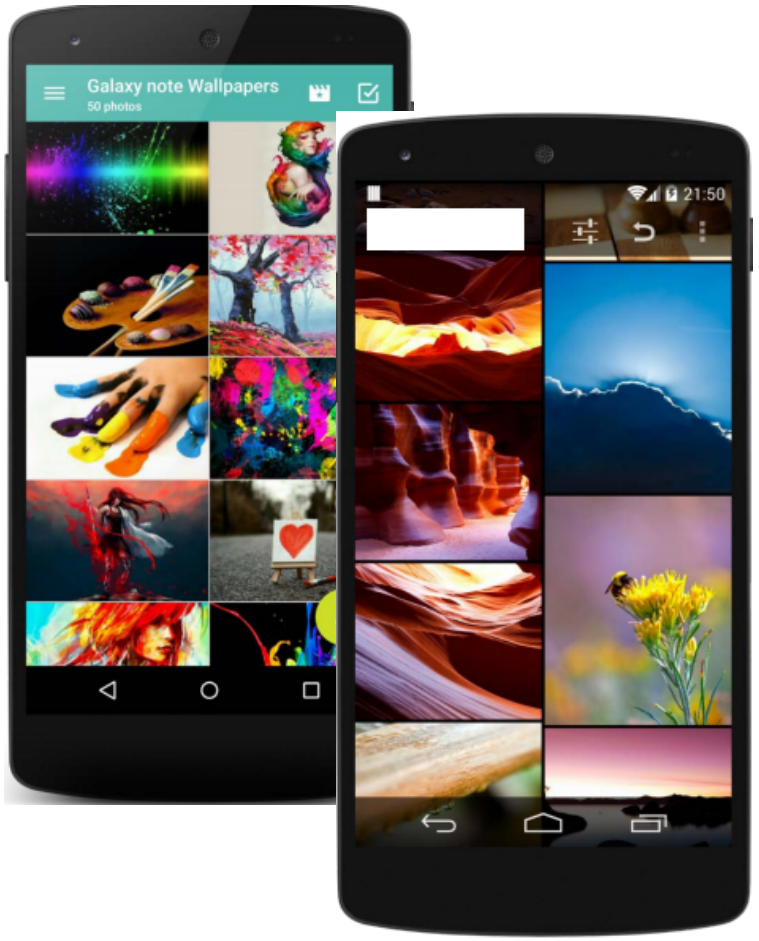
- Effiziente Listendarstellung großer Datenmengen
  - Eingebautes „Recycling“ derListenelemente
- Erweiterte und effizientere ListView
  - Unterschiedliche Layouts



HTL LEONDING

Gridlayout

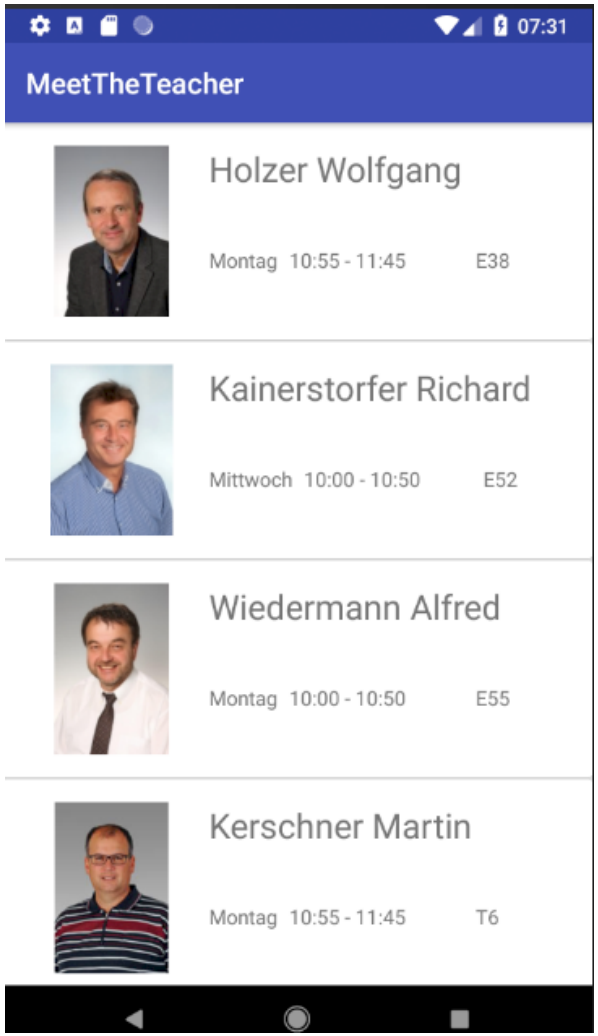
- Horizontale oder vertikale ScrollList
  - Horizontal ist bei ListView nicht möglich
- StaggeredLayout-Manager ➔ ZickZack-Design
- Defaultanimationen



HTL LEONDING

Benutzerdefiniertes Listenlayout

- Früher Listactivity mit eigenem Layout für Zeile
  - Ausgelagert in XML-View
- ListView ist nicht sehr performant
  - Pro Zeile wird eine View erzeugt
  - Lange Liste ➔ Speicher, Laufzeit



MeetTheTeacher – RecyclerView Anfang

HTL LEONDING

Datenklasse anlegen

- Dummy-Factory-Method liefert 4 Lehrer
- Bilder liegen im drawable-Ordner
  - Von Schulhomepage kopieren
  - Referenz vorläufig als int

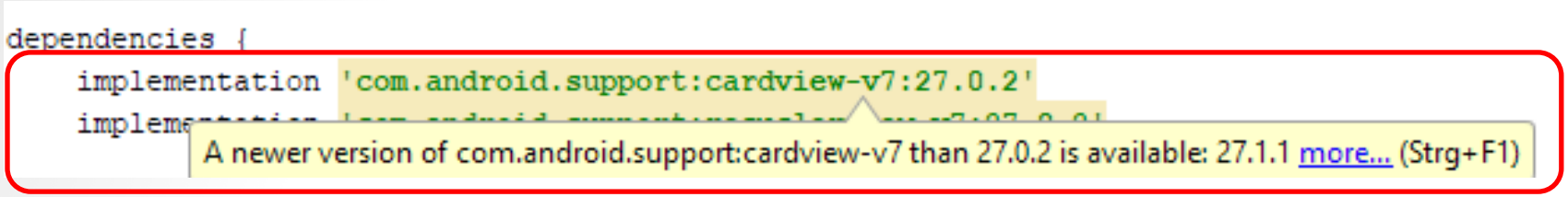
```
package at.htl.meettheteacher

data class Teacher (val name: String, val day: String, val time: String, val room : String, val detailsId : Int?, val imageUri: Int)

public fun getDummyTeachers(){
    val teachers = listOf<Teacher>(
            Teacher( name: "Holzer Wolfgang", room = "E38", day = "Montag", time = "10:55 - 11:45", detailsId = 534, imageUri = R.drawable.holzer),
            Teacher( name: "Kainerstorfer Richard", room = "E52", day = "Mittwoch", time = "10:00 - 10:50", detailsId = 2450, imageUri = R.drawable.kainerstorfer),
            Teacher( name: "Wiedermann Alfred", room = "E55", day = "Montag", time = "10:00 - 10:50", detailsId = 2453, imageUri = R.drawable.wiedermann),
            Teacher( name: "Kerschner Martin", room = "T6", day = "Montag", time = "10:55 - 11:45", detailsId = 2251, imageUri = R.drawable.kerschner)
    )
}
```

Dependencies in Gradle eintragen

- In build.gradle (Module: app)
- Aktuelle Version „erarbeiten“



- Synchronisieren

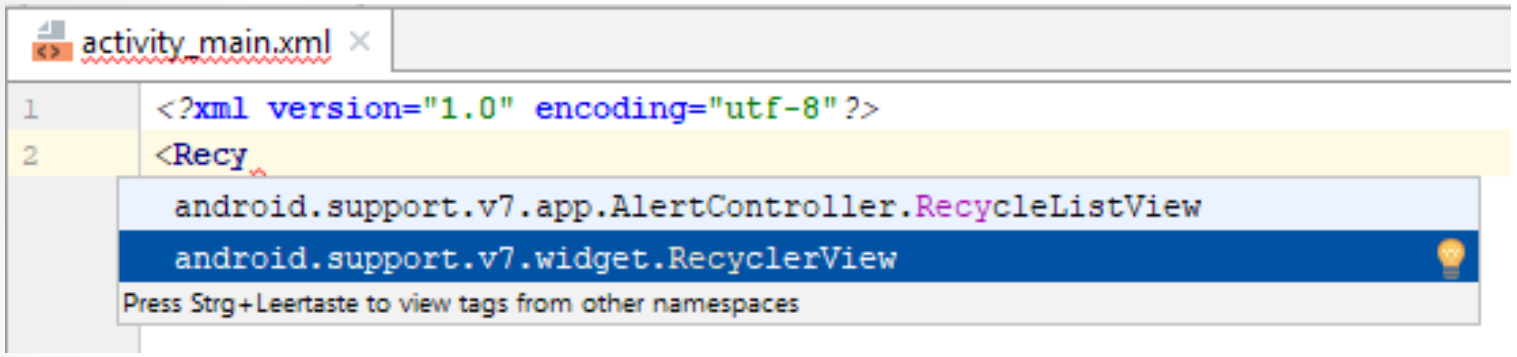
- Alternativ einfach per Designer in Layout ziehen

Schritte zur Implementierung

- Layout für Datenelement als CardView definieren
  - teacher_card.xml
- RecyclerView als Container
- Benutzerspezifischen Adapter definieren
  - TeacherRecyclerAdapter.java
  - Innere ViewHolder-Klasse
- RecyclerView konfigurieren
  - Mit Adapter verknüpfen
  - LayoutManager konfigurieren
  - Animationen

activity_main.xml

- ConstraintLayout ➔ RecyclerView



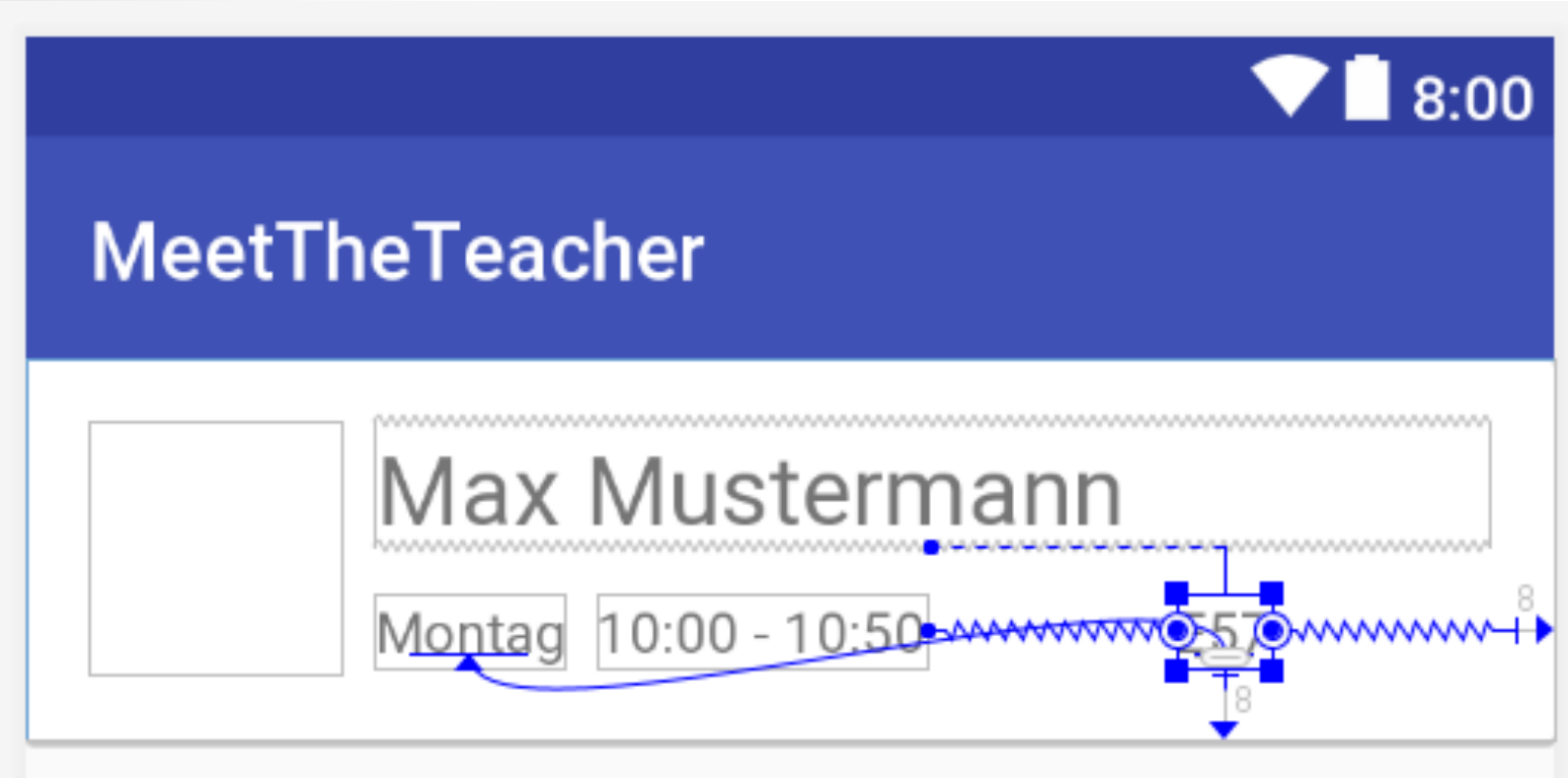
HTL LEONDING

RecyclerView

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.v7.widget.RecyclerView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:id="@+id/rv_teachers"
5      xmlns:tools="http://schemas.android.com/tools"
6      tools:context=".MainActivity"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent">
9
10 </android.support.v7.widget.RecyclerView>
```

teacher_card.xml



HTL LEONDING

MainActivity – RecyclerView anlegen

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        rv_teachers.setHasFixedSize(true)
        rv_teachers.layoutManager=LinearLayoutManager( context: this)
        rv_teachers.adapter=TeachersAdapter(getDummyTeachers())
    }
}
```

- Create abstract function 'TeachersAdapter'
- Create class 'TeachersAdapter'
- Create function 'TeachersAdapter'
- Rename reference

Adaptergerüst wird angelegt

- Übergebene Daten als Property verwenden

TeachersAdapter.kt

```
1  package at.htl.meettheteacher
2
3  import android.support.v7.widget.RecyclerView
4
5  class TeachersAdapter(dummyTeachers: Unit) : RecyclerView.Adapter<*>() {
```

Noch nicht alles

- Übergebene Daten typisieren
  - Mit val als Property definieren
- ViewHolder muss noch angelegt werden

TeachersAdapter.kt

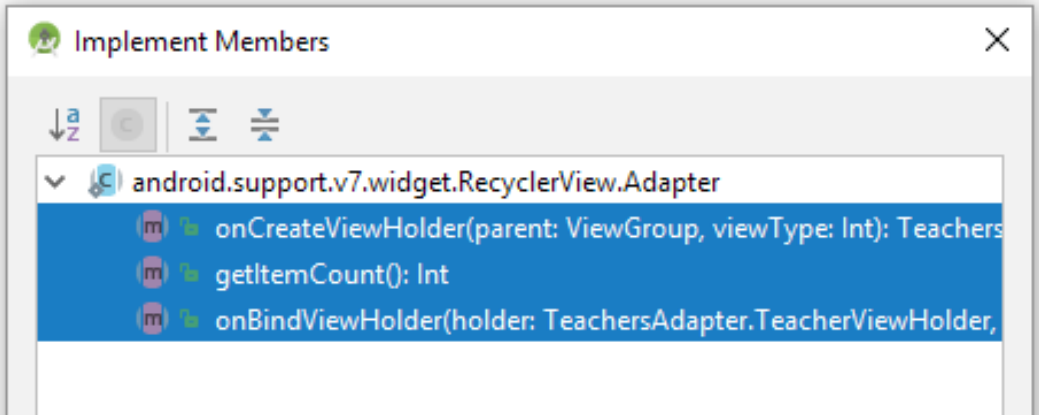
```
1  package at.htl.meettheteacher
2
3  import android.support.v7.widget.RecyclerView
4
5  class TeachersAdapter(teachers: List<Teacher>) : RecyclerView.Adapter<TeacherViewHolder>() {
6
7  }
```

Unresolved reference: TeacherViewHolder

Methoden anlegen lassen

```
class TeachersAdapter(teachers: List<Teacher>) : RecyclerView.Adapter<TeachersAdapter.TeacherViewHolder>() {
    class TeacherViewHolder(iv : View) : RecyclerView.ViewHolder(iv)

}
```



HTL LEONDING

Methoden implementieren

- card ist in TeacherViewHolder ein Property

```
class TeachersAdapter(private val teachers: List<Teacher>)
                : RecyclerView.Adapter<TeachersAdapter.TeacherViewHolder>() {
    val LOG_TAG = TeachersAdapter::class.java.simpleName

    class TeacherViewHolder(val card : View) : RecyclerView.ViewHolder(card)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int)
                    : TeacherViewHolder {...}

    override fun getItemCount() = teachers.size

    override fun onBindViewHolder(viewHolder: TeachersAdapter.TeacherViewHolder, position: Int) {...}

}
```

ViewHolder erzeugen

- Anzahl der erzeugten ViewHolder hängt von Anzahl der Cards am Schirm ab
  - Maximal 2 in Reserve
- Ausgabe der Erzeugung mit Hashcode zur Identifizierung

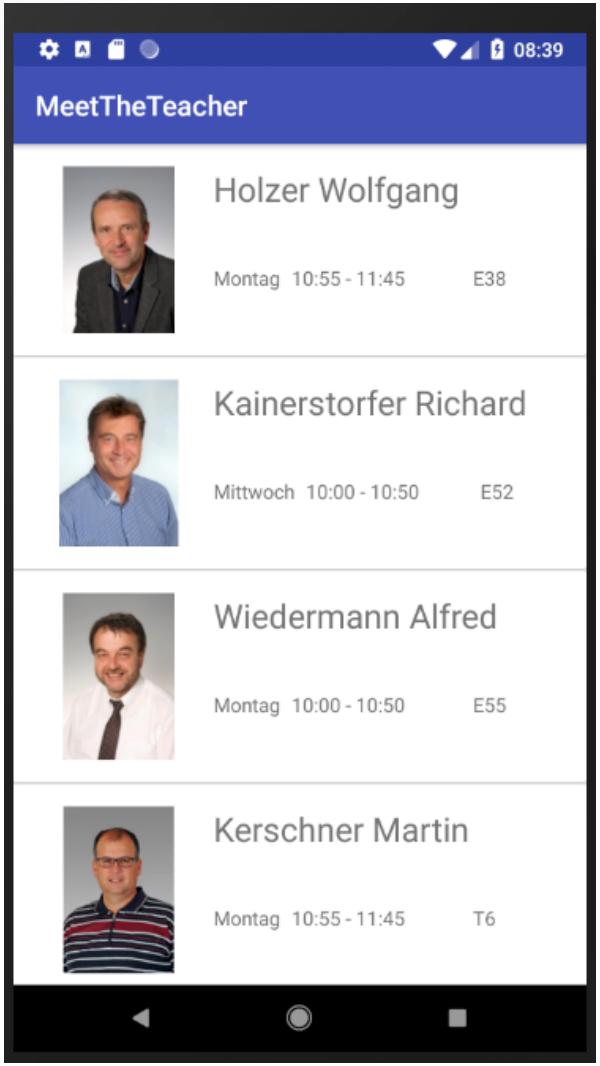
```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int)
                : TeacherViewHolder {
    val view = LayoutInflater.from(parent.context)
                .inflate(R.layout.teacher_card, parent,  attachToRoot: false)
    val viewHolder = TeacherViewHolder(view)
    Log.d(LOG_TAG,  msg: "onCreateViewHolder() ${viewHolder.hashCode()}")
    return viewHolder
}
```

ViewHolder an Daten binden

- Beim Scrollen werden Viewholder „recycelt“
- Verwendung der inline-Function with()

```
override fun onBindViewHolder(viewHolder: TeachersAdapter.TeacherViewHolder, position: Int) {
    val teacher = teachers[position]
    Log.d(LOG_TAG,  msg: "onBindViewHolder() ${viewHolder.hashCode()} to ${teacher.name}")
    with(viewHolder.card){  this: View
        tv_name.text = teacher.name
        tv_day.text = teacher.day
        tv_time.text = teacher.time
        tv_room.text = teacher.room
        iv_teacher.setImageResource(teacher.imageUri)
    }
}
```

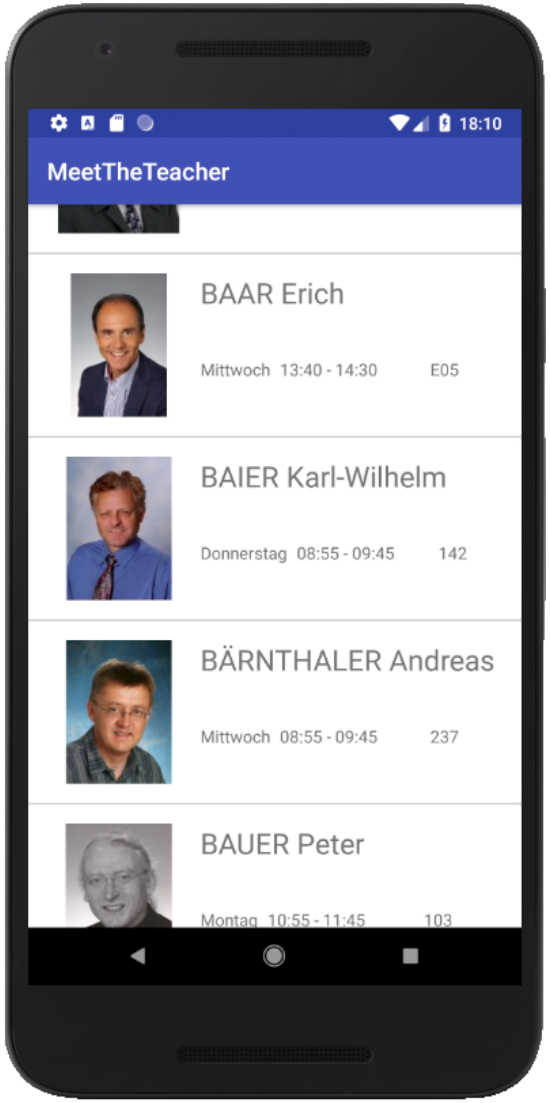
Ergebnis



HTL LEONDING

MeetTheTeacher – Livedaten von Homepage

- Einsatz von einfachem Multithreading
  - AsyncTask
  - Später ➔ Service
- Textdaten von der Homepage herunterladen
- Bilder im Hintergrund laden



HTL LEONDING

Verwendung der Klasse AsyncTask

- Vereinfacht Kopplung UI-Thread mit Hintergrundthread
  - Ergebnisübermittlung an UI-Thread ohne offensichtliche Threads/Handler
  - Vergleichbar mit SwingWorker, BackgroundWorker
  - Generische Parameter
    - Start des Hintergrundthreads
    - Meldung des Fortschritts
    - Ergebnis des Hintergrundthreads

AsyncTask's generic types	
Params:	the type of the parameters sent to the task upon execution.
Progress:	the type of the progress units published during the background computation.
Result:	the type of the result of the background computation.

AsyncTask - Überblick

```
private class VerySlowTask extends AsyncTask<String, Long, Void> {

    // Begin - can use UI thread here
    protected void onPreExecute() {

    }

    // this is the SLOW background thread taking care of heavy tasks
    // cannot directly change UI
    protected Void doInBackground(final String... args) {
    ... publishProgress((Long) someLongValue);
    }

    // periodic updates - it is OK to change UI
    @Override
    protected void onProgressUpdate(Long... value) {

    }

    // End - can use UI thread here
    protected void onPostExecute(final Void unused) {

    }
}
```

Daten per AsyncTask laden

- AsyncTask ohne onProgressUpdate()
- Übergabe einer CallbackRoutine, mit der Ergebnis an MainActivity übertragen wird

```
class LoadTeachersAsyncTask(private val resultCallback: (List<Teacher>) -> Unit)
                                    : AsyncTask<String, Void, List<Teacher>>() {
    private val LOG_TAG = LoadTeachersAsyncTask::class.java.simpleName

    override fun doInBackground(vararg urls: String): List<Teacher>? {...}
    override fun onPostExecute(teachers: List<Teacher>) {...}

    private fun getTeachersFromHtml(htmlString: String): List<Teacher> {...}

    /** Liefert für den Inhalt des HTML-Tags. ...*/
    private fun getTagContent(line: String): String {...}
}
```

HTML-Seite in List<Teacher> parsen

- Verschiedene Wege
  - Regex
  - Jsoup

```
private fun getTeachersFromHtml(htmlString: String): List<Teacher> {
    val lines = htmlString.split("\\n".toRegex()).dropLastWhile { it.isEmpty() }.toTypedArray()
    val teachers = ArrayList<Teacher>()
    var lineNumber = 1  // Header überlesen
    // Überschrift überlesen
    while (lineNumber < lines.size && !lines[lineNumber].contains( other: "Raum")) {
        lineNumber++
    }
    while (lineNumber < lines.size) {...}
    return teachers
}
```

Liste an MainActivity zurückgeben

- onPostExecute() läuft wieder im UI-Thread

```
override fun onPostExecute(teachers: List<Teacher>) {
    super.onPostExecute(teachers)
    resultCallback(teachers)
}
```

MainActivity – Aufruf ➔ asynchron

```
val loadTeachersAsyncTask = LoadTeachersAsyncTask{ it: List<Teacher>
    rv_teachers.adapter=TeachersAdapter(it)
    it.forEach { it: Teacher
        if(it.detailsId == 0){
            it.bitmap = BitmapFactory.decodeResource(resources, R.drawable.anonym)
        }else{
            val loadTeachersImageAsyncTask = LoadTeachersImageAsyncTask()
            try {
                loadTeachersImageAsyncTask.execute(it)
            } catch (e: Exception) {
                e.printStackTrace()
                Log.e(LOG_TAG, msg: "getImage() load image, exception: " + e.message)
            }
        }
    }
}
try {
    loadTeachersAsyncTask.execute(url)
    Log.e(LOG_TAG, msg: "asynctask started")
} catch (e: Exception) {
    e.printStackTrace()
    Log.e(LOG_TAG, msg: "Loading teachers failed: " + e.message)
}
```

Lehrerdaten im Hintergrund laden

- https://www.htl-leonding.at/index.php?id=1626

```
D/LoadTeachersAsyncTask: getTeachersFromHtml(): FEILMAIR Ewald
D/LoadTeachersAsyncTask: getTeachersFromHtml(): FELSNER Anja
D/LoadTeachersAsyncTask: getTeachersFromHtml(): FUERLINGER Josef
D/LoadTeachersAsyncTask: getTeachersFromHtml(): GALLISTL Andreas
D/LoadTeachersAsyncTask: getTeachersFromHtml(): GIRITZHOFER Rudolf
D/LoadTeachersAsyncTask: getTeachersFromHtml(): GRUBER Erhard
D/LoadTeachersAsyncTask: getTeachersFromHtml(): GÜNTHER Harald
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HACKENBUCHNER Franz
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HAMMER Hans-Christian
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HANNESSCHLÄGER Jürgen
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HASLINGER Klaus
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HAUNSCHMID Wilfried
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HOFER Gerhard
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HÖFER Gerhard
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HOLZER Wolfgang
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HOLZMANN Michael
D/LoadTeachersAsyncTask: getTeachersFromHtml(): HUBER Gerhard
```

LoadTeachersImageAsyncTask

- Bild asynchron von Homepage laden und in übergebenen Teacher abspeichern
- Ausgeblendet ist das Laden der Detailseite

```
override fun doInBackground(vararg teachers: Teacher): Int {
    val teacher = teachers[0]
    try {
...
        val pattern = Pattern.compile( regex: "src=\"uploads/(.*?)\"")
        val matcher = pattern.matcher(htmlText)
        if(matcher.find()){
            val pictureName = matcher.group( group: 1)
            val imageUrl = URL( spec: imagePath + pictureName)
            Log.d(LOG_TAG, msg: "LoadTeachersImageAsyncTask, Imageurl: ${imageUrl}")
            val httpImageURLConnection = imageUrl.openConnection() as HttpURLConnection
            teacher.bitmap = BitmapFactory.decodeStream(httpImageURLConnection.inputStream)
        }
    } catch (e: Exception) {
        Log.e(LOG_TAG, msg: "LoadTeachersImageAsyncTask, doInBackground(), exception: " + e.message)
        e.printStackTrace()
    }
    return 0
}
```

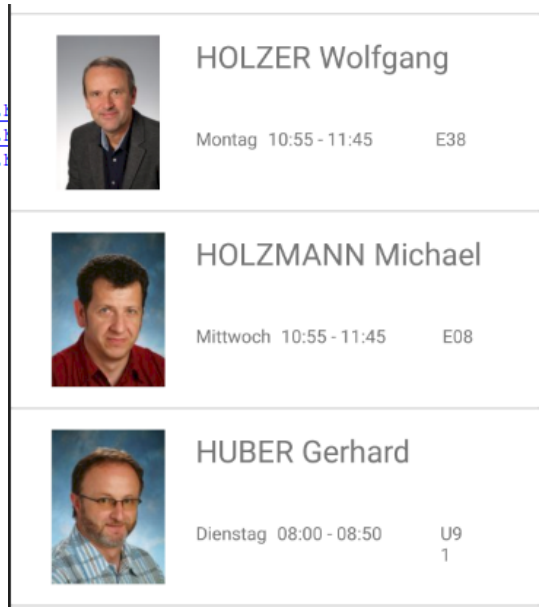
LogCat

```
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/aistleitner.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/aitenbichler.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/appolonio.JPG
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/auberger.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/auer.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/auernig.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/baar.JPG
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/baier.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/baernthaler.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/bauer.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/billinger.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/brenn.JPG
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/bucek.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/anonym.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/denkmair.jpg
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/draxlbauer.JPG
D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/ehrenmueller.jpg
```

Spezialistenaufgabe

- Datenfehler bei Prof. Holzmann

```
04-19 07:34:22.288 23930-23995/at.htl.meettheteacher D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/holzer.JPG
04-19 07:34:23.834 23930-23999/at.htl.meettheteacher D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.htl-leonding.at/uploads/holzmann.jpg
04-19 07:34:23.890 23930-23999/at.htl.meettheteacher E/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, doInBackground(), exception: http://www.htl-leonding.at/uploads/holzmann.jpg
04-19 07:34:23.890 23930-23999/at.htl.meettheteacher W/System.err: java.io.FileNotFoundException: http://www.htl-leonding.at/uploads/holzmann.jpg
        at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getInputStream(HttpURLConnectionImpl.java:251)
        at at.htl.meettheteacher.LoadTeachersImageAsyncTask.doInBackground(LoadTeachersImageAsyncTask.kt:37)
04-19 07:34:23.891 23930-23999/at.htl.meettheteacher W/System.err:     at at.htl.meettheteacher.LoadTeachersImageAsyncTask.doInBackground(LoadTeachersImageAsyncTask.kt:12)
        at android.os.AsyncTask$2.call(AsyncTask.java:333)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at android.os.AsyncTask$SerialExecutor$1.run(AsyncTask.java:245)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1162)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:636)
04-19 07:34:23.892 23930-23999/at.htl.meettheteacher W/System.err:     at java.lang.Thread.run(Thread.java:764)
04-19 07:34:34.963 23930-23999/at.htl.meettheteacher D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.h
04-19 07:34:35.850 23930-23999/at.htl.meettheteacher D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.h
04-19 07:34:36.945 23930-23999/at.htl.meettheteacher D/LoadTeachersImageAsyncTask: LoadTeachersImageAsyncTask, Imageurl: http://www.h
```

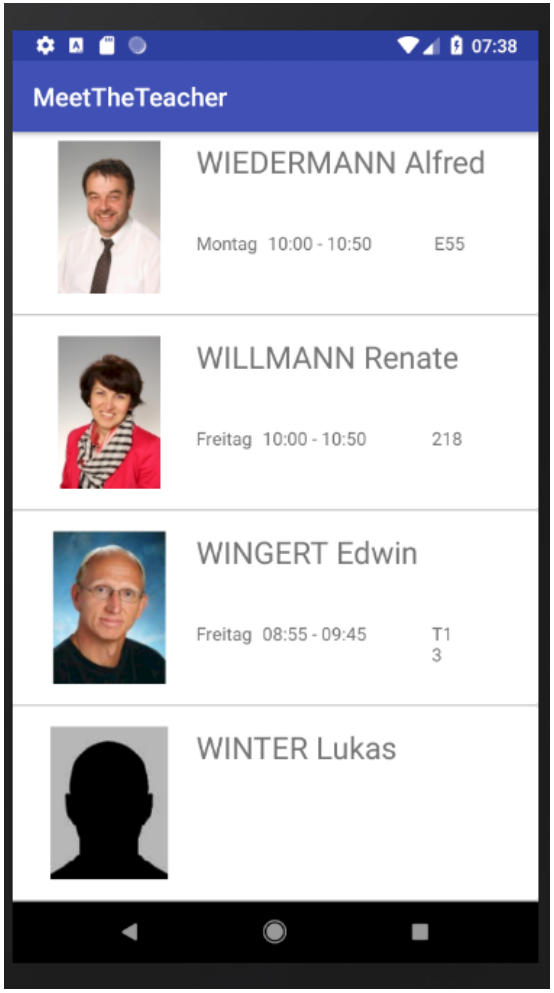


- Falsches Bild wird angezeigt
- Anonymes Bild anzeigen

HANNESSCHLÄGER
Jürgen

HTL LEONDING

Ergebnis



HTL LEONDING