



Aggiornamenti da Google I/O 2018



Jetpack



- Un completo framework per semplificare lo sviluppo di app
 - Su Android si usano spessissimo pattern di programmazione comuni e pezzetti di codice “idiomatici”
 - In pratica: copia&incolla da developer.android.com o da Stack Overflow
- **Jetpack** raccoglie questi pattern in componenti con API semplificata
 - Fondazionali, Architetture, Comportamentali, UI



Jetpack



<u>Foundation</u>	<u>Architecture</u>	<u>Behaviour</u>	<u>UI</u>
AppCompat	Data Binding	Download manager	Animations & transitions
Android KTX	Lifecycles	Media & playback	Auto
Multidex	LiveData	Notifications	Emoji
Test	Navigation	Permissions	Fragment
	Paging	Sharing	Layout
	Room	Slices	Palette
	ViewModel		TV
			Wear OS



Jetpack: qualche esempio



- Data Binding Library

- Permette di associare dichiarativamente valori nel codice a viste nella UI

- Anziché scrivere, a codice

```
TextView textView = findViewById(R.id.sample_text);  
textView.setText(viewModel.getUserName());
```

possiamo scrivere, nel layout XML

```
<TextView android:text="@{viewModel.userName}" />
```



Jetpack: qualche esempio



Sviluppo Applicazioni Mobili
V. Gervasi – a.a. 2017/18

- Room – relational-object bridging

```
@Entity
public class User {

    @PrimaryKey
    private int uid;

    @ColumnInfo(name = "first_name")
    private String firstName;

    @ColumnInfo(name = "last_name")
    private String lastName;

    ...

}
```

```
@Dao
public interface UserDao {

    @Query("SELECT * FROM user")
    List<User> getAll();

    @Query("SELECT * FROM user WHERE uid IN (:userIds)")
    List<User> loadAllByIds(int[] userIds);

    @Query("SELECT * FROM user WHERE "
        + "first_name LIKE :first AND "
        + "last_name LIKE :last LIMIT 1")
    User findByName(String first, String last);

    ...

}
```

```
@Database(entities = {User.class}, version = 1)
public abstract class MyDB extends RoomDatabase {
    public abstract UserDao userDao();
}
```

```
MyDB db =
    Room.databaseBuilder(appctx,
        MyDB.class, "db-name").build();
```

App Bundle

- Nuovo metodo per la distribuzione delle app
 - Lo sviluppatore invia al Play Store un **app bundle** con tutti i componenti della sua app
 - Codice, risorse, metadati
 - Al momento dell'installazione, il Play Store genera “al volo” un APK contenente i componenti *giusti* per l'ambiente di installazione
 - Per esempio: non vengono incluse le risorse grafiche con qualificatori incompatibili con lo schermo del dispositivo
 - Eventualmente, moduli aggiuntivi possono essere inviati dopo, on demand

App Bundle



Legacy APK



Dynamic Delivery