What Could Go Wrong?

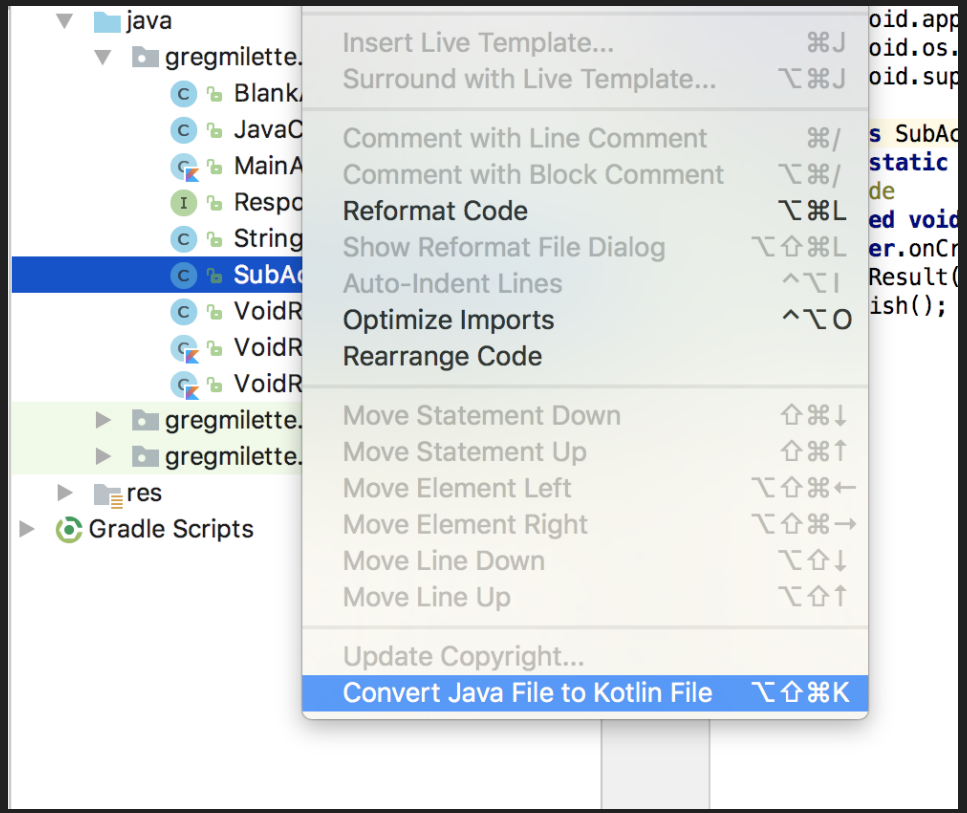
A paranoid developer's guide to Kotlin conversion

Greg Milette | TripAdvisor
@gregmilette


Java

So… what can go wrong?

Initial conversion procedure:

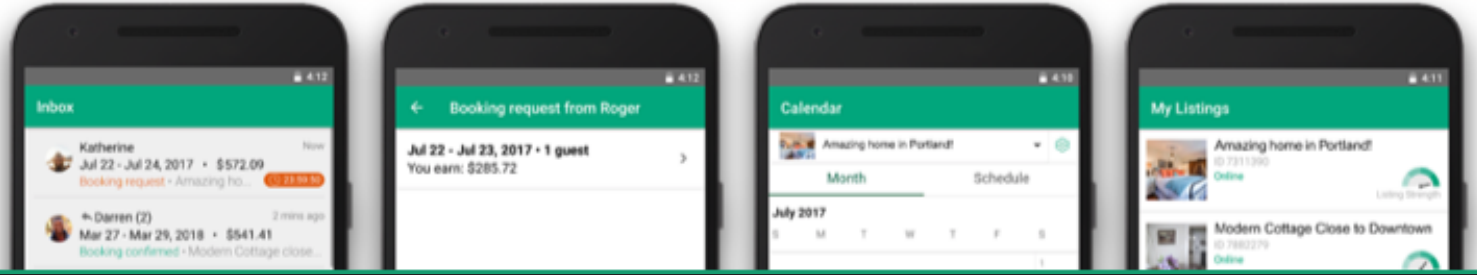


- Convert/write new
- Revise
- Monitor

Achieved: Happy Developers

Our app crashed more than usual after releasing ~20% of the code converted

Now it's crashing less than ever

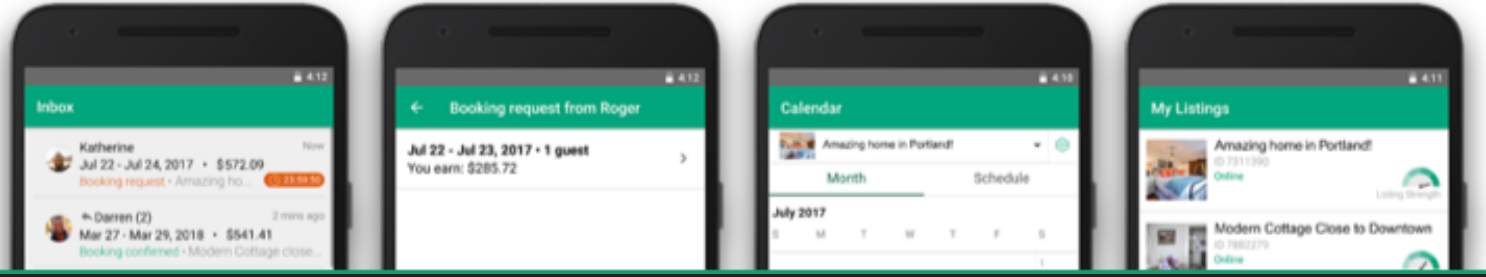


Exceptions:

Parameter specified as non-null is null: method kotlin.jvm.internal.Intrinsics.checkParameterIsNotNull

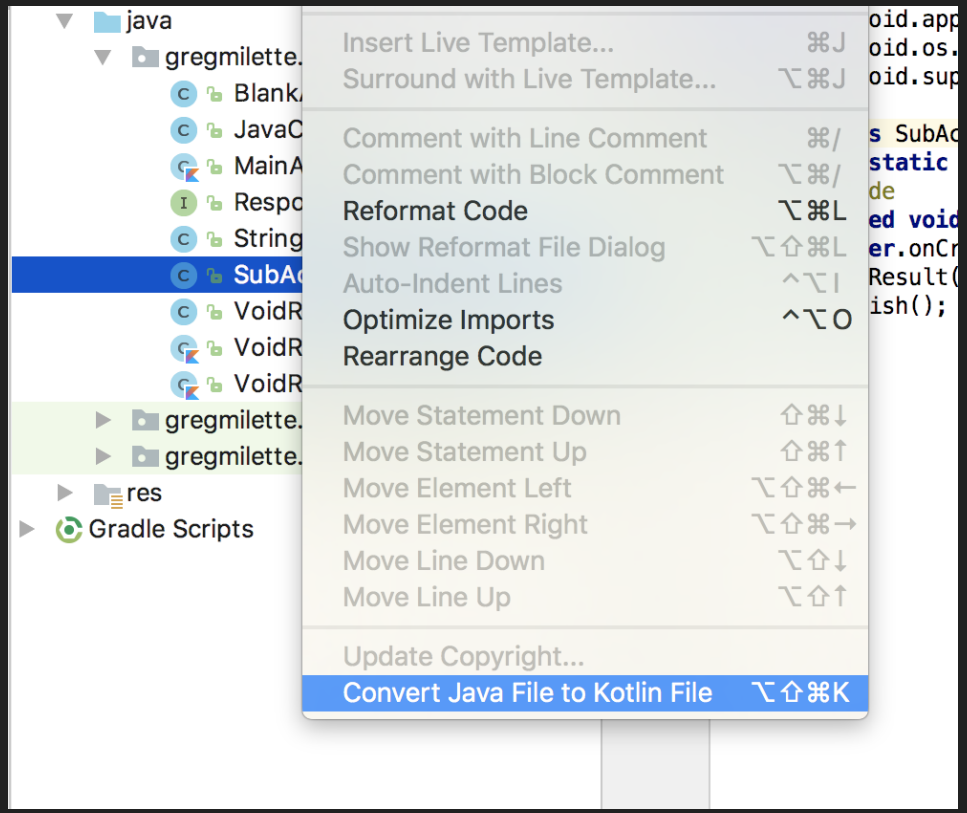
kotlin.UninitializedPropertyAccessException: lateinit property adapter has not been initialized

kotlin.TypeCastException: null cannot be cast to non-null



Check the Wiki Page

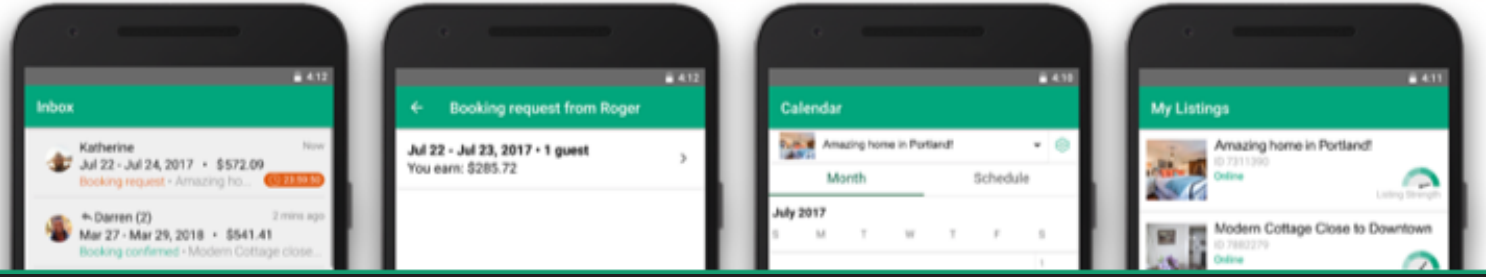
Final Conversion procedure



- Prepare Java
- Convert/write new
- Revise
  - Decide Nullability
    - Review lateinit / lazy
  - Review casts
  - Remove findViewById
  - Consult tools
  - Code review from experienced Kotlin coworker
- Monitor

Exceptions:

Parameter specified as non-null is null: method kotlin.jvm.internal.Intrinsics.checkParameterIsNotNull



Nulls in Java and Kotlin

Null or Non Null

Nullable

NonNullable

Where Java and Kotlin meet, Null Pointers Lurk

```
castle.build(wall);
```


```
fun build(wall: Wall) {

}
```

Where Java and Kotlin Meet, Exceptions Lurk

```
wall = null;
castle.build(wall);
```


```
fun build(wall: Wall) {

}
```

Parameter specified as non-null is null: method kotlin.jvm.internal.Intrinsics.checkParameterIsNotNull

Where Java and Kotlin meet, Null Pointers Lurk

```
wall = null
castle.build(wall);
```


```
// safer
fun build(wall: Wall?) {

}
```

Typical Android code

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
}
```

https://developer.android.com/guide/components/activities/activity-lifecycle.html

Typical Android code

```java
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
}
```

https://developer.android.com/guide/components/activities/activity-lifecycle.html


```fun
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent) {
}
```

Typical Android code

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setResult(RESULT_CODE);
    finish();
}
```

Sets Intent null

Typical Android code

```java
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setResult(RESULT_CODE);
    finish();
}
```

Sets Intent null


```kotlin
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent) {
}
```

Parameter specified as non-null is null: method kotlin.jvm.internal.Intrinsics.checkParameterIsNotNull

@Nullable

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
                                @Nullable Intent data) {

}
```

Add @Nullable

@Nullable

```java
@Override
protected void onActivityResult(int requestCode, int resultCode,
                                @Nullable Intent data) {
}
```

Add @Nullable


```kotlin
override fun onActivityResult(requestCode: Int, resultCode: Int,
                              data: Intent?) {
}
```

Converted Nullable Type

Summary: Java and Kotlin Interaction

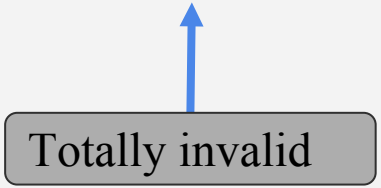
- Add @Nullable annotations before any conversion to avoid errors
- Paranoid thinking: Java could pass a null at any time, especially libraries without proper annotations

Defining a variable

```
var castle:GoodCastle
```

Defining a variable

```
var castle:GoodCastle
```



Should I define this?

```
val castleImmutable = GoodCastle()
```

Should I define this?

```
val castleImmutable = GoodCastle()

var castleNullable: GoodCastle? = null
```

Should I define this?

```
val castleImmutable = GoodCastle()

var castleNullable: GoodCastle? = null

// initialize later
lateinit var castleLate: GoodCastle
// initialize later using block
val castleLazy: GoodCastle by lazy { GoodCastle() }
```

Handling Nullable Types

```
if (castleNullable != null && army.close) {
    castleNullable.shoot()
    castleNullable.closeDrawbridge()
}
```

Handling Nullable Types

```
if (castleNullable != null && army.close) {
    castleNullable.shoot()
    castleNullable.closeDrawbridge()
}

castleNullable?.let {
    if (army.close) {
        it.shoot()
        it.closeDrawbridge()
    }
}
```

Handling Nullable Types

```
castleNullable?.shoot()

title = castleNullable.name ?: DEFAULT_NAME

// ...
```

```java
public void setCastle(GoodCastle castle) {
    this.castleNullable = castle;
}
public void shoot() {
    castleNullable.shoot();
}
```

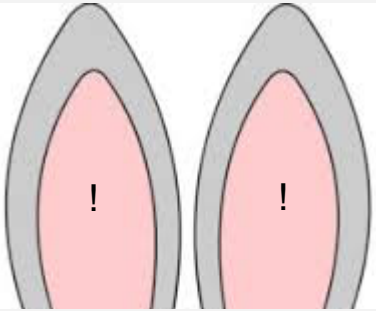
!

```java
public void setCastle(GoodCastle castle) {
    this.castleNullable = castle;
}
public void shoot() {
    castleNullable.shoot();
}
```


```fun
fun setCastle(castle: GoodCastle) {
    this.castleNullable = castle
}

fun shoot() {
    castleNullable!!.shoot()
}
```

```java
public void setCastle(GoodCastle castle) {
    this.castleNullable = castle;
}
public void shoot() {
    castleNullable.shoot();
}
```

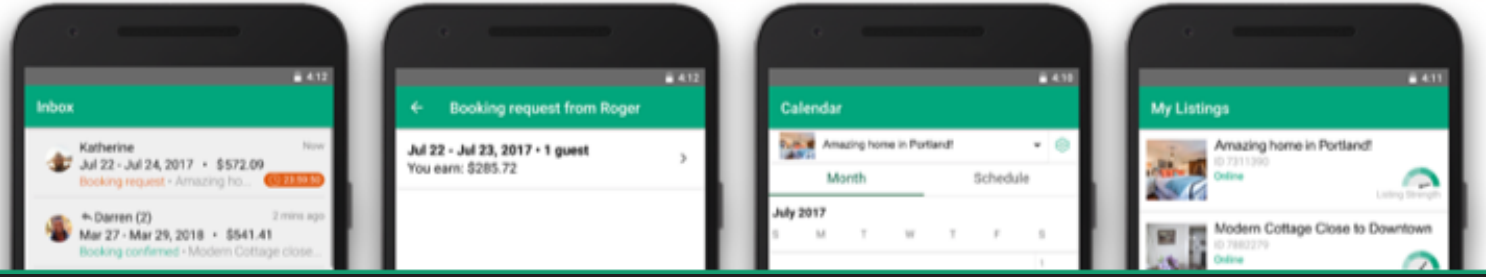

```fun
fun setCastle(castle: GoodCastle) {
    this.castleNullable = castle
}

fun shoot() {
    castleNullable!!.shoot
}
```



Beware of the bunny ears

Exceptions:

kotlin.UninitializedPropertyAccessException: lateinit property adapter has not been initialized



Lateinit

```
lateinit var castleRequester: CastleRequester
lateinit var castleLate: GoodCastle
```

Lateinit

```
lateinit var castleRequester: CastleRequester
lateinit var castleLate: GoodCastle

override fun onCreate(savedInstanceState: Bundle?) {
    castleRequester = CastleRequester(this)

    build_now.setOnClickListener {
        castleLate.build()
    }

    castleRequester.requestCastleAsync() {
        castleLate = GoodCastle()
    }
}
```

Lateinit

```
lateinit var castleRequester: CastleRequester
lateinit var castleLate: GoodCastle

override fun onCreate(savedInstanceState: Bundle?) {
    castleRequester = CastleRequester(this)

    build_now.setOnClickListener {
        castleLate.build()
    }

    castleRequester.requestCastleAsync() {
        castleLate = GoodCastle()
    }
}
```

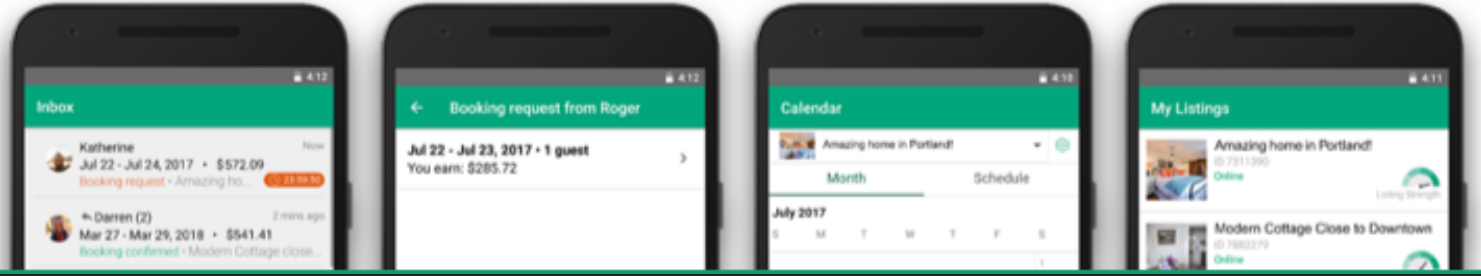
kotlin.UninitializedPropertyAccessException:
lateinit property adapter has not been initialized

Lazy delegate

```
val castleLazy: GoodCastle by lazy { if (army.evil) EvilCastle()
                                     else GoodCastle() }
lateinit var army:Army

override fun onCreate(savedInstanceState: Bundle?) {
   castleLazy.build()
   army = army()
}
```

Exceptions:

kotlin.TypeCastException: null cannot be cast to non-null



Casting

```
if (getIntent() != null) {
    castle =
        (GoodCastle)getIntent().getSerializableExtra(CASTLE_DATA);
}
```

"Throw an exception if the object is the wrong type"

```
castle = intent?.getSerializableExtra(CASTLE_DATA) as GoodCastle
```

"Throw exception if the object is the wrong type or if it is null"

Casting

```java
if (getIntent() != null) {
    castle =
        (GoodCastle)getIntent().getSerializableExtra(CASTLE_DATA);
}
```


```kotlin
castle = intent?.getSerializableExtra(CASTLE_DATA) as GoodCastle
```

kotlin.TypeCastException: null cannot be cast to non-null

Casting Options

```
castle = intent?.getSerializableExtra(CASTLE_DATA) as GoodCastle?
```

“Throw an exception if the object is the wrong type” (like java)

```
castle = intent?.getSerializableExtra(CASTLE_DATA) as? GoodCastle
```

“Return null if the cast fails”

Summary so far

- Java and Kotlin interaction
  - Add @Nullable annotations before any conversion to avoid errors
  - Paranoid thinking: Java could pass a null at any time, especially libraries without proper annotations
- Defining Variables
  - Use kotlin null handling syntax like let
  - Beware of the Bunny Ears (!!)
- Lateinit / by lazy
  - Good for when you are sure it will get initialized before use or the lazy init block will succeed
  - Do not use too many
- Casting
  - Use as Object?

Android Synthetic Views

```
val itemTitle = layout.item_title

val itemTitle = item_title
```

Android Synthetic Views

```java
class MyActivity {
    private View confirmation;
    private ImageView send;
    private TextView tip;
    private ProgressBar progressBar;
    private RecyclerView listView;
    private ViewGroup actionButton;

    private void setupUi() {
        confirmation = findViewById(R.id.confirmation);
        send = (ImageView) findViewById(R.id.send);
        tip = (TextView) findViewById(R.id.tip);
        progressBar = (ProgressBar) findViewById(R.id.progress_bar);
        actionButton = (ViewGroup) findViewById(R.id.action_button);
        listView = (RecyclerView) findViewById(R.id.list_view);
    }
}
```

Android Synthetic Views

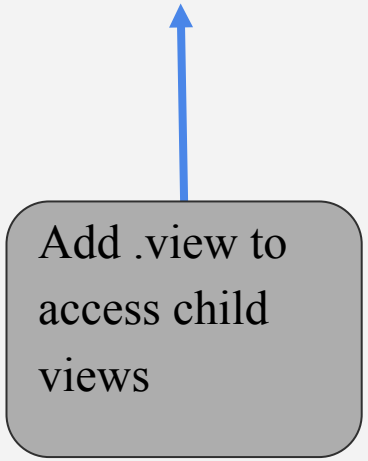
```
import kotlinx.android.synthetic.main.activity_main.*

tip.setText("the title")
```

Layout name

Synthetic view

Android Synthetic Views

```
import kotlinx.android.synthetic.main.castle_item.view.*
```



Android Synthetic Views

```
import kotlinx.android.synthetic.main.castle_item.view.*

val layout =
inflater.inflate(R.layout.castle_item, container, false) as ViewGroup
```

Android Synthetic Views

```
import kotlinx.android.synthetic.main.castle_item.view.*

val layout =
inflater.inflate(R.layout.castle_item, container, false) as ViewGroup

val itemTitle = layout.item_title
```

Android Synthetic Views

```
import kotlinx.android.synthetic.main.castle_item.view.*

val layout =
inflater.inflate(R.layout.castle_item, container, false) as ViewGroup

val itemTitle = layout.item_title

val itemTitle = item_title
```

Android synthetic views error

```
holder.itemView.my
    myCoordinator from inbox_activity.xml for…   Coordinat
    my_listing_divider from inbox_activity.xml for View (A
    my_listing_divider from my_listing_view_item…   Relati
    my_listing_image from my_listing_view_item.xml fo…   I
    my_listing_layout from my_   TrackedRelati
```

Same id in two layouts

Unexpected layout

```
import kotlinx.android.synthetic.main.inbox_activity.view.*
import kotlinx.android.synthetic.main.my_listing_view_item.view.*
```

Summary: Android Extensions

- Access the right view cache
- Check imports, ensure you pick the correct view

Tools

Check the tools: Lint

overlap_booking_modal.xml 2 warnings

two_button_layout.xml 1 warning

Duplicate id @+id/a_button, defined or included mu

Hardcoded reference to /sdcard 2 warnings

Implied default locale in case conversion 1 warning

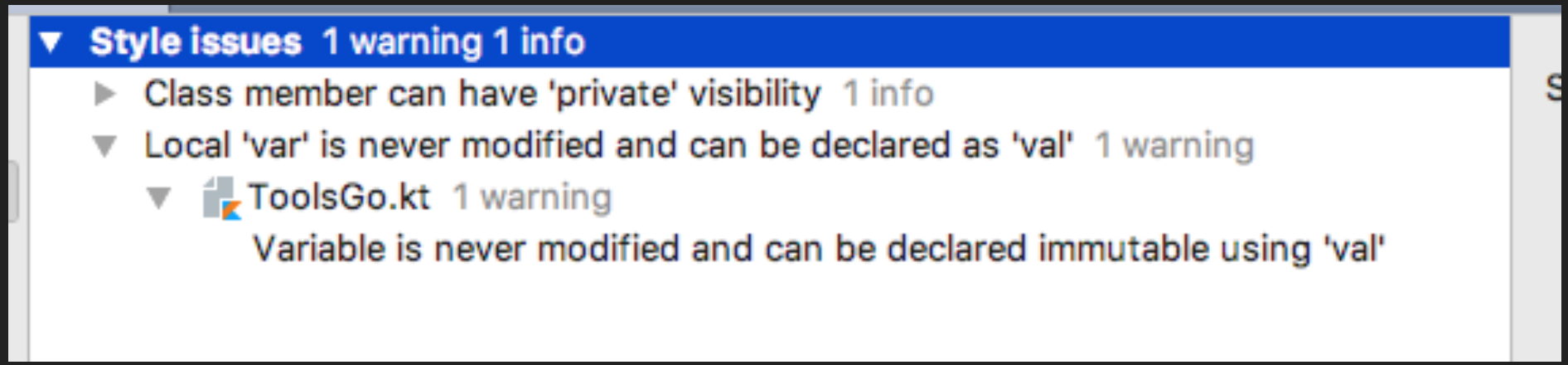
Implied locale in date format 2 warnings

Implied Quantities 6 errors

```
<include layout="@layout/good_button"/>
```

- This layout includes two views with the same id

Check the tools: Lint



- Change a var to val

Check the tools: Android Studio Intention Actions

```
// before
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? {
    for (vehicle in vehicles) {
        if (isTruck(vehicle)) {
            return vehicle
        }
    }
    return null
}
```

Check the tools: Android Studio Intention Actions

```
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? {
    for (vehicle in vehicles) {
```

- Remove braces from 'for' statement
- Replace with 'firstOrNull{}'
- Replace with a 'forEach' function call
- Add indices to 'for' loop

Check the tools: Android Studio Intention Actions

```
// before
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? {
    for (vehicle in vehicles) {
        if (isTruck(vehicle)) {
            return vehicle
        }
    }
    return null
}
// after
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? {
    return vehicles.firstOrNull { isTruck(it) }
}
```

From 8 lines down to 1

```
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? {
    return vehicles.firstOrNull { isTruck(it) }
```

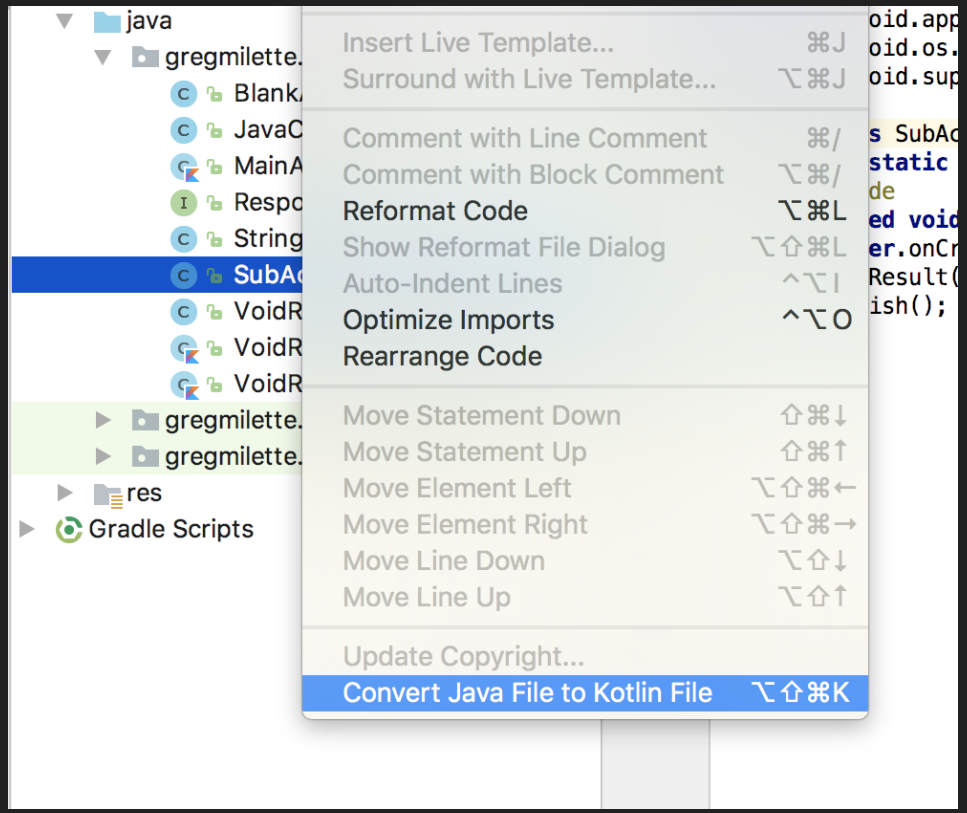
Convert to expression body

```
// after
fun firstTruck(vehicles: Array<Vehicle>): Vehicle? =
    vehicles.firstOrNull { isTruck(it) }
```

Summary: Why consult tools?

- Find potential errors
- Make safe improvements to your code
- Learn Kotlin language

Conversion procedure



- Prepare Java
- Convert/write new
- Revise
  - Decide Nullability
    - Review lateinit / lazy
  - Review casts
  - Remove findViewById
  - Consult tools
  - Code review from experienced Kotlin coworker
- Monitor

Do it!