Annotation Processing with Android II

Andrew Grosner

@agrosner/github.com/agrosner

Android Developer

Raizlabs

Breakdown

- Getting Started

- Best Practices

- Demo -> **@IntentBuilder!**

- Limitations + Gotchas

Getting Started

Create New Project
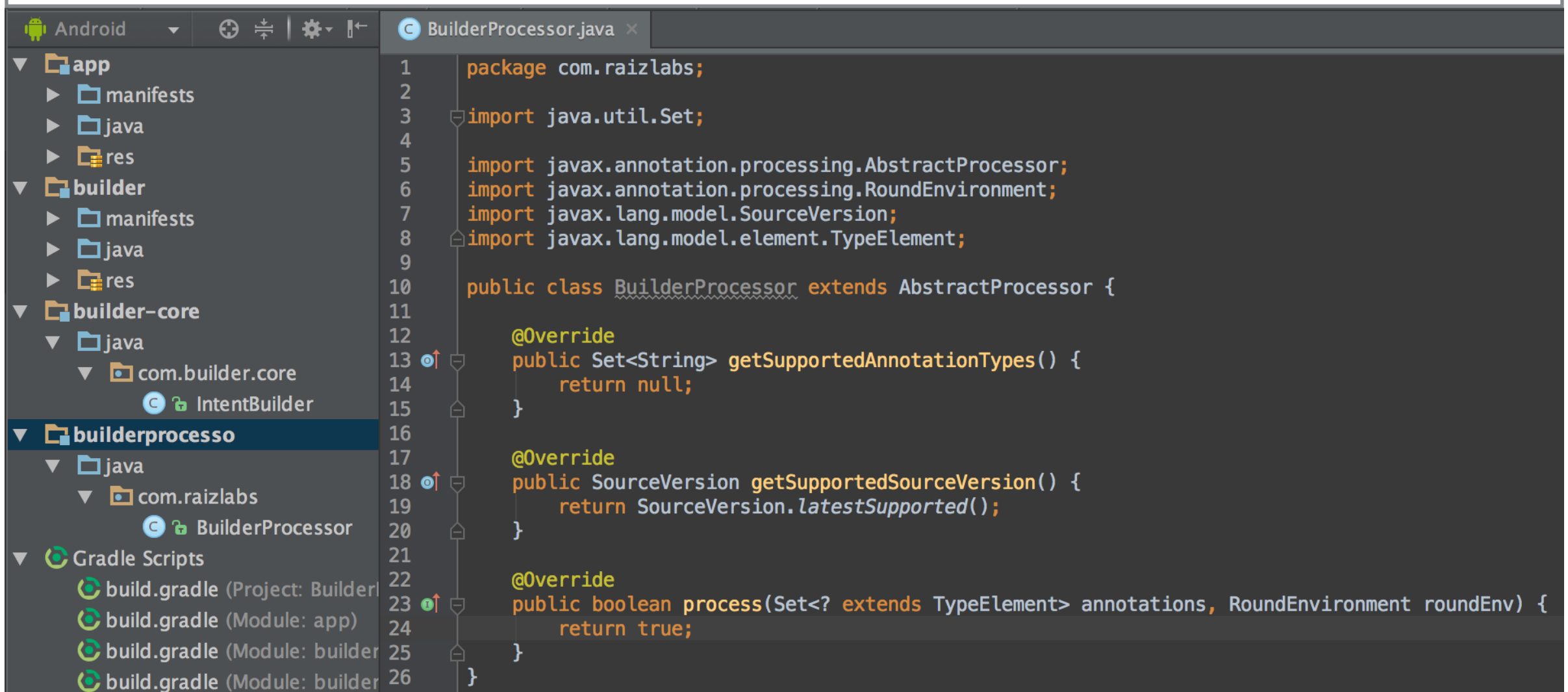
Add Projects
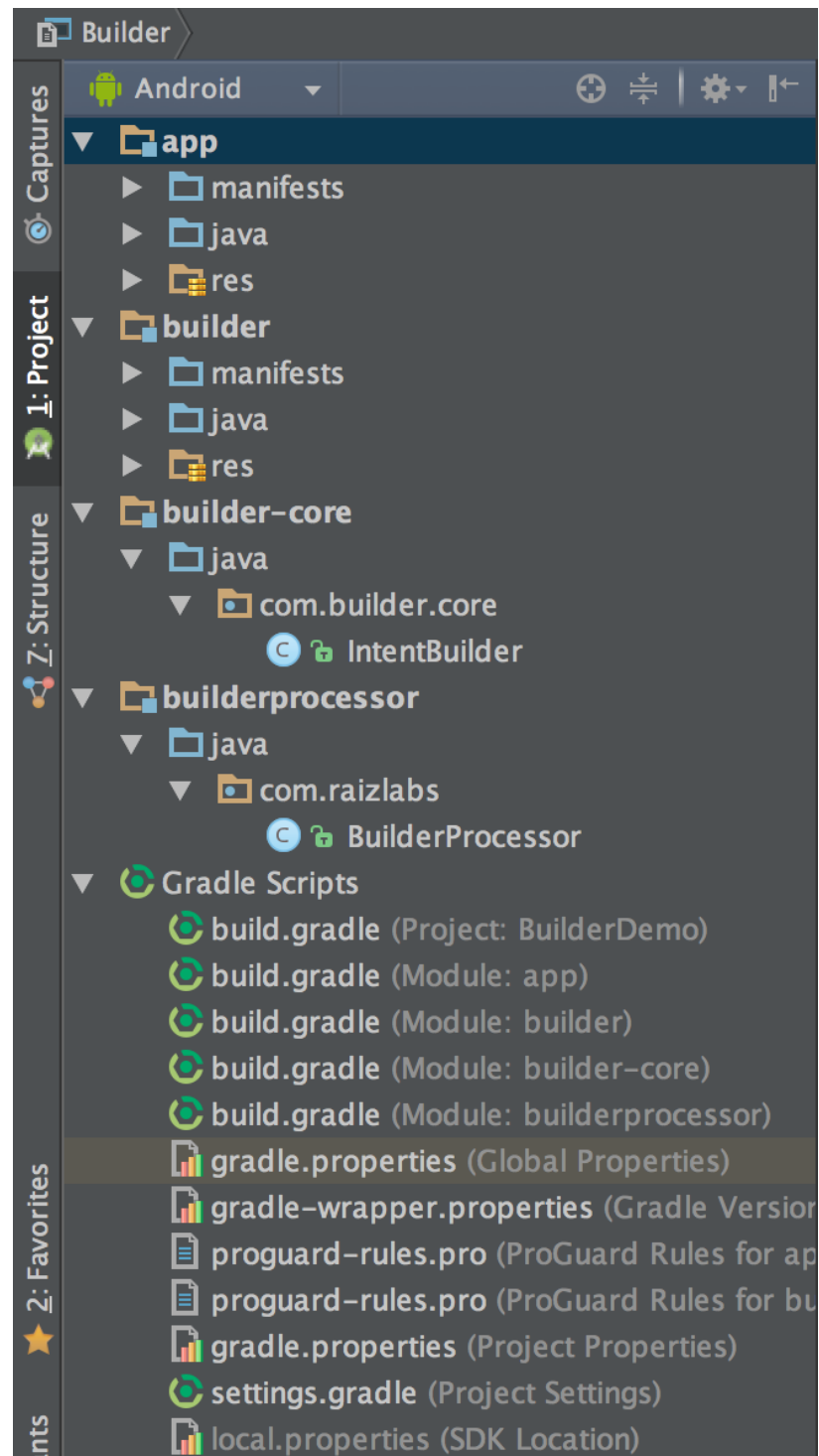
Create the Processor


```java
package com.raizlabs;

import java.util.Set;

import javax.annotation.processing.AbstractProcessor;
import javax.annotation.processing.RoundEnvironment;
import javax.lang.model.SourceVersion;
import javax.lang.model.element.TypeElement;

public class BuilderProcessor extends AbstractProcessor {

    @Override
    public Set<String> getSupportedAnnotationTypes() {
        return null;
    }

    @Override
    public SourceVersion getSupportedSourceVersion() {
        return SourceVersion.latestSupported();
    }

    @Override
    public boolean process(Set<? extends TypeElement> annotations, RoundEnvironment roundEnv) {
        return true;
    }
}
```

Bring Your Dependencies

```
BuilderProcessor.java ×        builderprocessor ×

1     apply plugin: 'java'
2
3     dependencies {
4         compile project(':builder-core')
5
6         compile 'com.squareup:javapoet:1.1.0'
7         compile 'com.google.auto.service:auto-service:1.0-rc2'
8     }
```

JavaPoet (https://github.com/square/javapoet)
Google Auto-Service (https://github.com/google/auto/service)

```
@AutoService(Processor.class)
public class BuilderProcessor extends AbstractProcessor {
```

Best Practices

Best Practices

- Associate the processing steps with objects

  - The manager

  - Annotation Handlers

  - Definitions/Writers

- Use libraries like JavaPoet

- Keep as much code out of the processor as you can!

Demo

Questions?

Feedback

Please take a moment to fill out the class feedback form via the app. Paper feedback forms are also available in the back of the room.

eventmobi.com/adcboston

Raizlabs

Thank You.