



**„Coffee & Friends Make The Perfect Blend“**

NEAT LITTLE JAVA HELPERS – A CURATED COLLECTION

**LESS CODE =  
LESS BUGS**

*„Honey, I Shrunk The Boilerplate“*



# Project Lombok (projectlombok.org)

## ✓ Adding lombok to your pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.20</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

# Project Lombok (projectlombok.org)

## @ToString

```
import lombok.ToString;

@ToString
public class ToStringExample {
    private static final int STATIC_VAR = 10;
    private String name;
    private Shape shape = new Square(5, 10);
    private String[] tags;
    @ToString.Exclude private int id;

    public String getName() {
        return this.name;
    }

    @ToString(callSuper=true, includeFieldNames=true)
    public static class Square extends Shape {
        private final int width, height;

        public Square(int width, int height) {
            this.width = width;
            this.height = height;
        }
    }
}
```

## @Builder

```
import lombok.Builder;
import lombok.Singular;
import java.util.Set;

@Builder
public class BuilderExample {
    @Builder.Default private long created =
        System.currentTimeMillis();
    private String name;
    private int age;
    @Singular private Set<String> occupations;
}
```

# Project Lombok (projectlombok.org)

## @Getter / @Setter

```
import lombok.AccessLevel;
import lombok.Getter;
import lombok.Setter;

public class GetterSetterExample {
    @Getter @Setter private int age = 10;

    @Setter(AccessLevel.PROTECTED) private String name;
}
```

## @No- / @All- / @RequiredArgsConstructor

```
import lombok.AccessLevel;
import lombok.RequiredArgsConstructor;
import lombok.AllArgsConstructor;
import lombok.NonNull;

@RequiredArgsConstructor(staticName = "of")
@AllArgsConstructor(access = AccessLevel.PROTECTED)
public class ConstructorExample<T> {
    private int x, y;
    @NonNull private T description;

    @NoArgsConstructor
    public static class NoArgsExample {
        @NonNull private String field;
    }
}
```

# MapStruct (mapstruct.org)

## ✓ Adding MapStruct to your pom.xml

```
...
<properties>
  <org.mapstruct.version>1.4.2.Final</org.mapstruct.version>
</properties>
...
<dependencies>
  <dependency>
    <groupId>org.mapstruct</groupId>
    <artifactId>mapstruct</artifactId>
    <version>${org.mapstruct.version}</version>
  </dependency>
</dependencies>
...
```

# MapStruct (mapstruct.org)

## ✓ Adding MapStruct to your build

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
      <configuration>
        <source>1.8</source> <!-- depending on your project -->
        <target>1.8</target> <!-- depending on your project -->
        <annotationProcessorPaths>
          <path>
            <groupId>org.mapstruct</groupId>
            <artifactId>mapstruct-processor</artifactId>
            <version>${org.mapstruct.version}</version>
          </path>
          <!-- other annotation processors -->
        </annotationProcessorPaths>
      </configuration>
    </plugin>
  </plugins>
</build>
```

# MapStruct (mapstruct.org)

## Car.java

```
public class Car {  
  
    private String make;  
    private int numberOfSeats;  
    private CarType type;  
  
    //constructor, getters, setters etc.  
}
```

## CarMapper.java

```
@Mapper  
public interface CarMapper {  
  
    CarMapper INSTANCE = Mappers.getMapper(CarMapper.class);  
  
    @Mapping(source = "numberOfSeats", target = "seatCount")  
    CarDto carToCarDto(Car car);  
}
```

## CarDto.java

```
public class CarDto {  
  
    private String make;  
    private int seatCount;  
    private String type;  
  
    //constructor, getters, setters etc.  
}
```

## Somewhere else...

```
...  
Car car = new Car( "Morris", 5, CarType.SEDAN );  
  
CarDto carDto = CarMapper.INSTANCE.carToCarDto( car );  
...
```



# THERE IS NO BARISTA WITHOUT A SPOON



Everyday tools to make  
your daily dose of java even  
more enjoyable



# Google Guava ([github.com/google/guava](https://github.com/google/guava))

## ✓ Adding Guava to your pom.xml

```
<dependency>  
  <groupId>com.google.guava</groupId>  
  <artifactId>guava</artifactId>  
  <version>30.1.1-jre</version>  
</dependency>
```

# Google Guava ([github.com/google/guava](https://github.com/google/guava))

## Immutable Collections

```
public static final ImmutableSet<String> COLOR_NAMES =
    ImmutableSet.of(
        "red",
        "orange",
        "yellow",
        "green",
        "blue",
        "purple");

class Foo {
    final ImmutableSet<Bar> bars;
    Foo(Set<Bar> bars) {
        this.bars = ImmutableSet.copyOf(bars);
    }
}
```

## Graphs

```
ImmutableGraph<Country> countryAdjacencyGraph =
    GraphBuilder.undirected()
        .<Country>immutable()
        .putEdge(FRANCE, GERMANY)
        .putEdge(FRANCE, BELGIUM)
        .putEdge(GERMANY, BELGIUM)
        .addNode(ICELAND)
        .build();

MutableValueGraph<String, Double> graph = ValueGraphBuilder
    .undirected()
    .allowsSelfLoops(true)
    .build();
graph.putEdgeValue("San Francisco", "San Francisco", 0.0);
graph.putEdgeValue("San Jose", "San Jose", 0.0);
graph.putEdgeValue("San Francisco", "San Jose", 48.4);
```

# Google Guava ([github.com/google/guava](https://github.com/google/guava))

## Caches

```
LoadingCache<Key, Graph> graphs = CacheBuilder
    .newBuilder()
    .maximumSize(1000)
    .expireAfterWrite(10, TimeUnit.MINUTES)
    .removalListener(MY_LISTENER)
    .build(
        new CacheLoader<Key, Graph>() {
            @Override
            public Graph load(Key key) {
                return createExpensiveGraph(key);
            }
        });

...
try {
    return graphs.get(key);
} catch (ExecutionException e) {
    throw new OtherException(e.getCause());
}
```

## EventBus

```
class EventBusChangeRecorder {
    @Subscribe
    public void recordCustomerChange(ChangeEvent e) {
        recordChange(e.getChange());
    }
}

// somewhere during initialization
eventBus.register(new EventBusChangeRecorder());

// much later
public void changeCustomer()
    ChangeEvent event = getChangeEvent();
    eventBus.post(event);
}
```

# FasterXML Jackson ([github.com/FasterXML](https://github.com/FasterXML))

## ✓ Adding Guava to your pom.xml

```
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-core</artifactId>  
  <version>${jackson.version.core}</version>  
</dependency>
```

Jackson is a suite of data-processing tools for Java (and the JVM platform), it supports:

Avro, BSON, CBOR, CSV, JSON, Smile, (Java) Properties, Protobuf, TOML, XML or YAML

# FasterXML Jackson ([github.com/FasterXML](https://github.com/FasterXML))

## Changing names

```
public class MyBean {
    private String _name;

    // without annotation, we'd get "theName", but we
    // want "name":
    @JsonProperty("name")
    public String getTheName() { return _name; }

    // note: it is enough to add annotation on just
    // getter OR setter;
    // so we can omit it here
    public void setTheName(String n) { _name = n; }
}
```

## Ignoring properties

```
// means that if we see "foo" or "bar" in JSON, they will be
// quietly skipped
// regardless of whether POJO has such properties
@JsonIgnoreProperties({ "foo", "bar" })
public class MyBean
{
    // will not be written as JSON; nor assigned from JSON:
    @JsonIgnore
    public String internal;

    // no annotation, public field is read/written normally
    public String external;

    @JsonIgnore
    public void setCode(int c) { _code = c; }

    // note: will also be ignored because setter has
    // annotation!
    public int getCode() { return _code; }
}
```

# FasterXML Jackson ([github.com/FasterXML](https://github.com/FasterXML))

## Using custom constructors

```
public class CtorBean
{
    public final String name;
    public final int age;

    // constructor can be public, private, whatever
    @JsonCreator
    private CtorBean(@JsonProperty("name") String name,
        @JsonProperty("age") int age)
    {
        this.name = name;
        this.age = age;
    }
}
```

## Factory Methods

```
public class FactoryBean
{
    // fields etc omitted for brevity

    @JsonCreator
    public static FactoryBean create(
        @JsonProperty("name") String name) {
        // construct and return an instance
    }
}
```

# Thank You

## CONTACT:

 Christian Scharr

 [contact@minus16.com](mailto:contact@minus16.com)

