

Features/Properties	Immutables 2.0	AutoValue 1.1	FreeBuilder 1.0
Open sourced since	Jul-13	Aug-13	Jan-15
Engine	Annotation processing	Annotation processing	Annotation processing
Model source	Abstract Value Type	Abstract Value Type and Builder API	Abstract Value Type
Minimum Java version	1.7 compiled under 1.7 could run on 1.6	1.6	1.6
Android support	Yes Pro guard recommended	Yes	Yes probably
Java 8 support	Yes Default methods, Optional*, Some support for type annotations	No	No
Definition boilerplate	Minimal	Moderate	Minimal
Generated code prettiness	Excellent	Fair	Good
Builder	Yes	Yes	Yes
Constructor support	Yes Dedicated support	Yes Only builder or only constructor	No DIY factory method may use builder
Immutable from abstract class	Yes	Yes	Yes
Immutable from Interface	Yes	No	Yes
Immutable from Annotation	Yes	Yes Limited support using @AutoAnnotation	No
Guava usage	Yes. Optional Guava impl is repackaged and minimized. Guava in the codebase is detected.	Minimal. Optional Guava impl is repackaged and minimized. Guava in the codebase could be used if declared	Yes. Mandatory Guava is embedded without repackaging.
Collection support	Yes Include sugar to set attributes	Minimal Adding abstract builder helper methods	Yes Guava only, Including sugar to set attributes
Default Attributes	Yes Dedicated support	DIY Ad hoc/DIY support, Works only for builders	DIY Ad hoc in builder constructor
Derived Attributes	Yes Dedicated support	No You can compute and cache manually, but it is not the same as support	No You can compute and cache manually, but it is not the same as support
Lazy attributes	Yes	DIY	DIY
Implementation hiding	Yes Possible by declaring abstract all builder methods	Mandatory Forces to declare all builder methods	Yes Possible by overriding all builder methods
Style Flexibility	Yes Full flexibility using styles, structural features and naming patterns	Basic Set/get vs no prefix detection	No not yet developed
Precondition Check/Validation	Yes Dedicated support	DIY Ad hoc/DIY buy overriding build/factory, dedicated support is deprecated	DIY Ad hoc/DIY buy overriding build
Null safety, Immutable collection conversion	Yes Dedicated support	Yes By adding special helpers builder	Yes Dedicated support
Optional attributes support	Yes Dedicated support including guava and java 8	No	Yes Dedicated support, guava
Nullable support	Yes	Yes	Yes
IDE Support. Eclipse And IDEA	Yes Complex cases work	Yes Works for simple cases	Yes Works for simple cases
Refactoring stability	Basic	Moderate	Basic
Generics support	Limited Instantiation only	Yes Works for simple cases	No
Trait inheritance	Yes	No	No
Nullness annotations propagations	Yes If jsr 305 or android @Nullable is present	n/a may not be needed if present in API	No
Modify by copy	Yes Structural sharing	No Only using builder	No Only using builder
Modify by copy fill builder	Yes	Yes	Yes
Singleton instances	Yes	No	No
Instance interning	Yes Strong interning and "ordinal-based domains"	No	No
Precompute hash code	Yes	No DIY cached hashCode possible	No DIY cached hashCode possible
Precompute hash code	Yes	No DIY cached hashCode possible	No DIY cached hashCode possible
JSON support	Yes Easy Jackson mapping and GSON type adapter generation	DIY Complex annotation mapping or custom type adapters	DIY Complex annotation mapping or custom type adapters
Java binary serialization support	Yes Preserves singleton	DIY No control beyond adding Serializable	DIY No control beyond adding Serializable
Customize toString, hashCode and equals	Yes	Yes	Yes
Auxiliary attributes (excluded from generated equals, hashCode)	Yes	No	No
Conclusion	Unfortunately you have to try to use all the solutions to see the real differences ;)		