# 

[**Guava**](#_6or3ar9r994d) **1**

[**🗆com.google.common.collect.Ordering (This is a common example of extension, they won’t do anything about it, they even use the example in their documentation)**](#_t84zzy88r63v) **1**

[**Hibernate**](#_p8az0gxg5ke0) **2**

[✅ SQLite Dialect (already requested as pull request)](#_czy9y7xpn7iz) 2

[✅ H2Dialect (already posted)](#_exy5o0q9yty9) 2

[✅ HsqlDialect (already posted and fixed)](#_yru8wpacj2p6) 2

[🗆IdentifyGenerator (posted)](#_utuxognmvcsu) 3

[🗆SnakeCaseNamingStrategy (reported HIBERNATE-166)](#_5gb75mbw4g6f) 3

[**Jackson**](#_cozlieuqwxcf) **5**

[🗆BSON (posted https://github.com/FasterXML/jackson-core/issues/544):](#_v7k7vtrfmntn) 5

[**Junit**](#_4qo8zlwqfyff) **6**

[🗆TestSuite (This code was actually added 7 years ago, they are just using a very old version, and copy pasted the code themselves)](#_hs648h842itk) 6

[**Log4j**](#_cb1x57k3jrk) **8**

[✅ JSON Layout](#_9z83v0mzbwwi) 8

[✅ Add custom info at header and](#_iqwlv1zfb51f) 8

# 

# Guava

## 🗆~~com.google.common.collect.Ordering~~ (This is a common example of extension, they won’t do anything about it, they even use the example in their documentation)

Would it be possible to add a compare method to <https://github.com/google/guava/blob/master/guava/src/com/google/common/collect/Ordering.java> that returns an order that compares the length of two strings?

Example:

private final class OrderingByLenght extends Ordering<String> {

@Override

public final int compare(final String s1, final String s2) {

return Ints.compare(s1.length(), s2.length());

}

}

Taken from: org.baeldung.guava GuavaOrderingExamplesUnitTest

Example of this type of code on their wiki :<https://github.com/google/guava/wiki/OrderingExplained>

# Hibernate

## **✅** SQLite Dialect (already requested as pull request)

Project that will be pulled in: <https://github.com/gwenn/sqlite-dialect/issues/7>

Multiple projects are currently implementing their own SQLite Dialects, why is this not already standard?

Projects that do this:

|  |
| --- |
| 52North/SOS/core/sqlite-config/src/main/java/org/n52/sos/config/sqlite/hibernate/HibernateSQLiteDialect.java |
| xnx3/wangmarket/src/main/java/com/xnx3/j2ee/dialect/SQLiteDialect.java |
| gavinking/query-validator/src/main/java/org/hibernate/query/validator/GenericDialect.java |
| GoogleCloudPlatform/google-cloud-spanner-hibernate/google-cloud-spanner-hibernate-dialect/src/main/java/com/google/cloud/spanner/hibernate/SpannerDialect.java |
| tiagohs/spring-javafx-material-design-admin/src/main/java/com/tiagohs/dialect/SQLiteDialect.java |
| juniorware/sqlite-jpa/sqlite-jpa/src/main/java/com/blogspot/thedevside/util/SQLiteDialect.java |

# 

## **✅** H2Dialect (already posted)

Found bug <https://hibernate.atlassian.net/browse/HHH-9861>

In project xetys/jhipster-uaa-setup/app2/src/main/java/com/mycompany/myapp/domain/util/FixedH2Dialect.java. They had to workaround the column type float, the issues has already been presented on the Hibernate forums, with no fix in sight (2015)

## **✅** HsqlDialect (already posted and fixed)

Found bug <https://hibernate.atlassian.net/browse/HHH-1598>

Issue has since been allegedly resolved and ticket is closed. Still considered a workaround by people before it was closed

Studied in:

/raytrend/utopiaframework/utopiaframework-core/src/main/java/cn/raytrend/utopiaframework/core/orm/hibernate3/dialect/CustomHSQLDialect.java

/BroadleafCommerce/BroadleafCommerce/common/src/main/java/org/broadleafcommerce/common/dialect/DemoHSQLDialect.java

/42BV/jarb/src/main/java/nl/\_42/jarb/utils/orm/hibernate/dialect/ImprovedHsqlDialect.java

/spring-projects/spring-data-jpa/src/test/java/org/springframework/data/jpa/repository/CustomHsqlHibernateJpaVendorAdaptor.java

/spot-next/core-next/spot-core/src/main/java/io/spotnext/core/persistence/hibernate/impl/HsqlDialect.java

## 🗆IdentifyGenerator (posted)

Primary key needs to be unquoted else it breaks on Postgres:

Posted to the PostGres forums <https://www.postgresql.org/message-id/1323270625199-5055944.post%40n5.nabble.com>

Problem can also be found here:

<https://stackoverflow.com/questions/2308543/composite-id-in-hibernatepostgres-breaks-due-to-returned-column-order>

Hacked solution can be found here:

<https://github.com/ManyDesigns/Portofino/blob/master/portofino-database/src/main/java/com/manydesigns/portofino/persistence/hibernate/PortofinoIdentityGenerator.java>

Posted to StackOverflow as JBoss does not allow for “Bugs” to be posted directly on their Jira by non-commiters.

“I found a solution that seems to work here:

https://github.com/ManyDesigns/Portofino/blob/master/portofino-database/src/main/java/com/manydesigns/portofino/persistence/hibernate/PortofinoIdentityGenerator.java

I realize this question is quite old, but this appears to be a working workaround, it doesn't appear as though this was ever officially reported as a bug, but if anyone needs it here it is.”

## 🗆SnakeCaseNamingStrategy (reported [HIBERNATE-166](https://issues.jboss.org/browse/HIBERNATE-166))

<https://github.com/vladmihalcea/hibernate-types/blob/master/hibernate-types-4/src/main/java/com/vladmihalcea/hibernate/type/util/CamelCaseToSnakeCaseNamingStrategy.java>

The guy even has a blog post about it...

# Jackson

## 🗆BSON (posted):

Hi, would it be possible to add some basic BSON support for the Tree Traversing Parser?

This project: <https://github.com/mongojack/mongojack/blob/700f3fd5506e1df8d7786fb16aa0d350515a5b39/src/main/java/org/mongojack/internal/object/BsonObjectTraversingParser.java>

<https://github.com/mongojack/mongojack/blob/700f3fd5506e1df8d7786fb16aa0d350515a5b39/src/main/java/org/mongojack/internal/object/document/DocumentObjectTraversingParser.java>

Is currently using a workaround for your API by copying your classes and adapting it for BSON Objects. Would it be possible to integrate some of this functionality directly into Jacksons API so that everyone could have access to this functionality without using another library like mongojack?

Found potential solution here too in: /michel-kraemer/bson4jackson/src/main/java/de/undercouch/bson4jackson/BsonParser.java

/immutables/immutables/mongo/src/org/immutables/mongo/bson4jackson/BsonParser.java

# Junit

## 🗆TestSuite (This code was actually added 7 years ago, they are just using a very old version, and copy pasted the code themselves)

The eclipse foundation also uses a similar method in their performance testing suite:

<https://github.com/eclipse/eclipse.jdt.ui/blob/655861083de9498320b598976207621d9613e16b/org.eclipse.jdt.text.tests/src/org/eclipse/jdt/text/tests/performance/PerfTestSuite.java>

|  |  |
| --- | --- |
| private void addPerformanceTestMethod(Method m, Vector<String> names, Class<?> theClass) { |  |
|  | String name= m.getName(); |
|  | if (names.contains(name)) |
|  | return; |
|  | if (!isPublicMeasureMethod(m)) { |
|  | if (isMeasureMethod(m)) |
|  | addTest(warning("Test method isn't public: "+m.getName())); //$NON-NLS-1$ |
|  | return; |
|  | } |
|  | names.addElement(name); |
|  | addTest(createPerformanceTest(theClass, name)); |
|  | } |

And the following project also uses a similar method: <https://github.com/sebastianbenz/Jnario/blob/master/plugins/org.jnario.lib/src/org/jnario/junit3/NamedJnarioTestSuite.java>

private void addTestMethod(Method m, List<String> names, Class<?> theClass) {

String name= m.getName();

if (names.contains(name))

return;

if (! isPublicTestMethod(m)) {

if (isTestMethod(m))

addTest(warning("Test method isn't public: "+ m.getName() + "(" + theClass.getCanonicalName() + return;

}

names.add(name);

addTest(createTest(theClass, name));

}

This method can be found both in org.eclipse.jdt.text.tests.performance and org.jnario.junit3, it would likely be functionality that others could use.

# 

# Log4j

## **✅** JSON Layout

At least 3 projects are implementing their own JSON Layout, probably this would be a good idea to add

A library already does this, and the feature was integrated into Log4j2

Studied in:

/logstash/log4j-jsonevent-layout/src/main/java/net/logstash/log4j/JSONEventLayoutV0.java

/logstash/log4j-jsonevent-layout/src/main/java/net/logstash/log4j/JSONEventLayoutV1.java

/xiancloud/xian/xian-log/xian-gelf-common/src/main/java/biz/paluch/logging/gelf/log4j/GelfLayout.java

/kdgregory/log4j-aws-appenders/log4j1-appenders/src/main/java/com/kdgregory/log4j/aws/JsonLayout.java

/kdgregory/log4j-aws-appenders/log4j1-appenders/src/test/java/com/kdgregory/log4j/testhelpers/ThrowingLayout.java

/mp911de/logstash-gelf/src/main/java/biz/paluch/logging/gelf/log4j/GelfLayout.java

## **✅** Add custom info at header and

This was integrated into Log4j2 as a configuration setup, stack overflow post can be found here:

<https://stackoverflow.com/questions/20819376/log4j2-rollingfile-appender-add-custom-info-at-the-start-of-each-logfile/20979314#20979314>

xiancloud/xian/xian-log/xian-log4j2/src/main/java/info/xiancloud/log4j2/AbstractCustomConfigurationFactoryTemplate.java

cloudera/logredactor/src/test/java/org/cloudera/log4j/redactor/Log4j2RedactorTest.java