




Refactor InternalAbstractGraphDatabase





Merged

 tinwelint merged 3 commits into neo4j:2.3 from rickardoberg:refactoriagdb2 Apr 27, 2015

 Conversation 39

 Commits 3

 Files changed 155

+4,440 -3,414    



rickardoberg commented Apr 20, 2015

Collaborator

Split IAGDB into a factory and a facade. The factory instantiates 3 modules (platform, edition, datasource), and subclasses then specifies how each edition is instantiated.



rickardoberg added **kernel** **2.3** labels Apr 20, 2015



thobe commented Apr 20, 2015

Owner

I have a feeling that the refactorings.patch file should not have been added.



chrisvest commented Apr 20, 2015

Collaborator

The data/dbms/auth file also looks like it was added by mistake.



rickardoberg commented Apr 20, 2015

Collaborator

@thobe @chrisvest fixed both, thanks for catching that!



tinwelint self-assigned this Apr 20, 2015



thobe commented on the diff Apr 20, 2015

...rg/neo4j/kernel/impl/factory/GraphDatabaseFacade.java

[View full changes](#)



((236 lines not shown))

```
236 +     try ( Statement statement = threadToTransactionBridge.get() )
237 +     {
238 +         if ( !statement.readOperations().relationshipExists( id ) )
239 +         {
240 +             throw new NotFoundException( format( "Relationship %d not found", id ) );
241 +             new EntityNotFoundException( EntityType.RELATIONSHIP, id );
242 +         }
243 +     }
244 +     return nodeManager.newRelationshipProxy( id );
245 + }
246 +
247 +
248 + @Override
249 + public IndexManager index()
250 + {
251 +     // TODO: txManager.assertInUncommittedTransaction();
```



thobe added a note Apr 20, 2015

Owner

Why is this not done?



rickardoberg added a note Apr 20, 2015

Collaborator

Labels

2.3

kernel

Milestone

No milestone

Assignee

tinwelint

6 participants



Good question, I don't know. All the method implementations are just copied verbatim from IAGDB. That line was from a commit made by Jakub in July last year. Maybe we should add it? Not sure why it was commented out, and if we can maybe safely add it back.

Add a line note

 **thobe** commented on the diff Apr 20, 2015

...rg/neo4j/kernel/impl/factory/GraphDatabaseFacade.java

[View full changes](#)



((91 lines not shown))

```
91 +
92 +/**
93 + * Implementation of the GraphDatabaseService/GraphDatabaseService interface
94 + * services created by the {@link org.neo4j.kernel.impl.factory.GraphDatabaseFacade}
95 + *
96 + * To make a custom GraphDatabaseFacade, the best option is to subclass an
97 + * alternative, used by legacy database implementations, is to subclass the
98 + * {@link org.neo4j.kernel.impl.factory.GraphDatabaseFacadeFactory#newFacade}
99 + */
100 +public class GraphDatabaseFacade
101 +    implements GraphDatabaseAPI
102 +{
103 +    private static final long MAX_NODE_ID = IdType.NODE.getMaxValue();
104 +    private static final long MAX_RELATIONSHIP_ID = IdType.RELATIONSHIP.getMaxValue();
105 +
106 +    private boolean initialized = false;
```



thobe added a note Apr 20, 2015

Owner

I can't quite see this flag serving a purpose. It is only assigned in the constructor, so it could be final. But the only cases when that assignment wouldn't take place are cases when the object wouldn't be published (because of exceptions in the constructor) so it does in fact serve no purpose.

Perhaps it should be traded for a flag that helps making shutdown idempotent?



rickardoberg added a note Apr 20, 2015

Collaborator

This is another one that was simply copied. It is not set in the constructor though, but rather the `init()` method. In 3.0 I hope we can change this to be constructor though, which would be better than the `init()` method.

But replacing it with shutdown flag is definitely an option.



thobe added a note Apr 20, 2015

Owner

No. I'm silly. that's not a constructor, that's a method called `init` ... I'm still wondering if `shutdown()` shouldn't set this field to false.

Add a line note

 **thobe** commented on the diff Apr 20, 2015

...rg/neo4j/kernel/impl/factory/GraphDatabaseFacade.java

[View full changes](#)

		((112 lines not shown))
112	+	<code>private AvailabilityGuard availabilityGuard;</code>
113	+	<code>private Log msgLog;</code>
114	+	<code>private LifeSupport life;</code>
115	+	<code>private Supplier<KernelAPI> kernel;</code>
116	+	<code>private Supplier<QueryExecutionEngine> queryExecutor;</code>
117	+	<code>private KernelEventHandlers kernelEventHandlers;</code>
118	+	<code>private TransactionEventHandlers transactionEventHandlers;</code>
119	+	
120	+	<code>private long transactionStartTimeout;</code>
121	+	<code>private DependencyResolver dependencies;</code>
122	+	<code>private Supplier<StoreId> storeId;</code>
123	+	<code>protected String storeDir;</code>
124	+	
125	+	<code>public PlatformModule platformModule;</code>
126	+	<code>public EditionModule editionModule;</code>
127	+	<code>public DataSourceModule dataSourceModule;</code>



thobe added a note Apr 20, 2015

Owner

These module fields doesn't appear to be used (except for platform module which is used for accessing the `config` , but that could be extracted to a separate field in the init method just like the other components of these objects).

I'd propose removing these fields from here. The module types still serve a purpose, as vectors of their respective fields, but after that purpose have been served, I don't see the point of keeping those objects around.



rickardoberg added a note Apr 20, 2015

Collaborator

The main reason I left them is if we want to have tests access them instead of using `DependencyResolver`. In general it would be nice if we could cut down on `DependencyResolver` usage, as it is a bit overused as it is. With the module fields it will be much easier to do "Find usages" when we make refactorings of services and whatnot. Makes sense?



rickardoberg added a note Apr 20, 2015

Collaborator

We could also opt for removing all the other fields, and have accesses go through module refs, as there should be no performance overhead for doing so.



thobe added a note Apr 20, 2015

Owner

Makes sense. I didn't notice that they were public.

We could have all access go through the module refs, although I do think there would be a (tiny) performance overhead of doing so, since it is one additional indirection.



lutowich added a note Apr 23, 2015

Collaborator

Imho public fields are very bad and replacing hackish `DependencyResolver` (which we at least have control over) with those is not a good idea.



rickardoberg added a note Apr 24, 2015

Collaborator

The reason is because with `DependencyResolver` it is non-trivial to know what services are used by what code. With the public fields we can do "Find usages" during refactorings. Making them final is a good point though.

Add a line note



cleishm commented Apr 23, 2015

Owner

Squash?



cleishm commented Apr 23, 2015

Owner

I was referring to the "Merge pull request #2" commit. Seems odd that a PR would contain a merge from a wholly different repo, and that'll look very weird in our history.

...



rickardoberg commented Apr 23, 2015

Collaborator

@cleishm yup, understand what you mean, fixed



cleishm commented on an outdated diff Apr 23, 2015

Show outdated diff



cleishm commented on an outdated diff Apr 23, 2015

Show outdated diff



tinwelint commented Apr 23, 2015

Collaborator

You seem to have gotten another unrelated commit into this PR (related to what @cleishm mentioned before). This one: Fixed masking of startup exception with shutdown exception in IAGD should not be in here. It seems to come from #4458



lutovich commented on the diff Apr 23, 2015

...java/org/neo4j/kernel/impl/factory/EditionModule.java

[View full changes](#)



((27 lines not shown))

```
27 +import org.neo4j.kernel.impl.api.SchemaWriteGuard;
28 +import org.neo4j.kernel.impl.core.LabelTokenHolder;
29 +import org.neo4j.kernel.impl.core.PropertyKeyTokenHolder;
30 +import org.neo4j.kernel.impl.core.RelationshipTypeTokenHolder;
31 +import org.neo4j.kernel.impl.locking.Locks;
32 +import org.neo4j.kernel.impl.storemigration.UpgradeConfiguration;
33 +import org.neo4j.kernel.impl.transaction.TransactionHeaderInformationFactor
34 +import org.neo4j.kernel.info.DiagnosticsManager;
35 +
36 +/**
37 + * Edition module for {@link org.neo4j.kernel.impl.factory.GraphDatabaseFac
38 + * need to create all the services that would be specific for a particular
39 + */
40 +public abstract class EditionModule
41 +{
42 +    public IdGeneratorFactory idGeneratorFactory;
```



lutovich added a note Apr 23, 2015

Collaborator

Public non-final fields???



tinwelint added a note Apr 23, 2015

Collaborator

I don't think that's as bad as it looks, since that's internal classes. Compare that to a setter and a getter which effectively is the same thing. The difference is that you can control access to it, although since this is internal stuff we can change this at any time and the compiler will help us complete such a change.



cleishm added a note Apr 23, 2015

Owner

There shouldn't be a setter for that - if there is, then something is wrong in the way code is structured. It should only be set from within the class, preferably only during construction.

Having a getter is nice for controlling access and later refactoring.





rickardoberg added a note Apr 24, 2015

Collaborator

I can add final, that's fine. Adding getters wouldn't add anything beyond that.

Add a line note

 **lutovich** commented on an outdated diff Apr 23, 2015


 Show outdated diff




tinwelint commented Apr 23, 2015


Collaborator

Personally I think the amount of introduced checkstyle violations is a bit on the high side. How can we improve this moving further? I'm not militant about fixing those before merge, but it'd make me happier.

 **tinwelint** commented on an outdated diff Apr 23, 2015

 Show outdated diff

 **tinwelint** commented on an outdated diff Apr 23, 2015

 Show outdated diff

 **tinwelint** commented on the diff Apr 23, 2015


...a/org/neo4j/kernel/impl/factory/DataSourceModule.java

[View full changes](#)



((316 lines not shown))

```
316 + protected NodeProxy.NodeActions createNodeActions(final GraphDatabaseSe
317 +
318 +
319 + {
320 +     return new NodeProxy.NodeActions()
321 +     {
322 +         @Override
323 +         public Statement statement()
324 +         {
325 +             return threadToStatementContextBridge.get();
326 +         }
327 +
328 +         @Override
329 +         public GraphDatabaseService getGraphDatabase()
330 +         {
331 +             // TODO This should be wrapped as well
```

 **tinwelint** added a note Apr 23, 2015

Collaborator

Should it be wrapped or is it an old comment?

 **rickardoberg** added a note Apr 24, 2015

Collaborator


Most of this is copied verbatim from IAGDB. If we have issues with IAGDB code, then lets deal with that after this is merged.

Add a line note


 **cleishm** commented on the diff Apr 23, 2015

...j/kernel/impl/factory/GraphDatabaseFacadeFactory.java


[View full changes](#)

 ((97 lines not shown))

97	+	*/
98	+	public GraphDatabaseFacade newFacade(Map<String, String> params, final
99	+	{
100	+	return newFacade(params, dependencies, new GraphDatabaseFacade())
101	+	}
102	+	
103	+	/**
104	+	* Instantiate a graph database given configuration, dependencies, and
105	+	*
106	+	* @param params
107	+	* @param dependencies
108	+	* @param graphDatabaseFacade
109	+	* @return
110	+	*/
111	+	public GraphDatabaseFacade newFacade(Map<String, String> params, final
112	+	final GraphDatabaseFacade graphDa



 **cleishm** added a note Apr 23, 2015 Owner

Is this meant to be private? (Factory methods don't take already constructed objects of the type they produce)

 **rickardoberg** added a note Apr 24, 2015 Collaborator


The only reason this method is there at all is because of EmbeddedGraphDatabase and other IAGDB subclasses, which (as per devteam email) need to remain until 3.0 for compatibility reasons. We can remove this later.

Add a line note


  **cleishm** commented on an outdated diff Apr 23, 2015 Show outdated diff

  **cleishm** commented on the diff Apr 23, 2015


...j/kernel/impl/factory/GraphDatabaseFacadeFactory.java View full changes

 ((114 lines not shown))

114	+	PlatformModule platform = createPlatform(params, dependencies, gra
115	+	EditionModule edition = createEdition(platform);
116	+	final DataSourceModule dataSource = createDataSource(dependencies,
117	+	
118	+	// Start it
119	+	graphDatabaseFacade.init(platform, edition, dataSource);
120	+	
121	+	Throwable error = null;
122	+	try
123	+	{
124	+	enableAvailabilityLogging(platform.availabilityGuard, platform
125	+	// "database is now unavailable"
126	+	
127	+	platform.life.start();
128	+	}
129	+	catch (final Throwable throwable)

 **cleishm** added a note Apr 23, 2015 Owner

Should this really catch `Throwable` ? That seems excessive.

 **rickardoberg** added a note Apr 24, 2015 Collaborator

This is copied verbatim from IAGDB. If we have issues with that, let's change after the PR is done. This is mainly a refactoring, not a refactoring and touchup.

Add a line note



cleishm commented Apr 23, 2015

Owner

This doesn't seem to address `HighlyAvailableGraphDatabaseFactory`



rickardoberg commented Apr 24, 2015

Collaborator

@**cleishm** as per devteam email, `HighlyAvailableGraphDatabase` needs to remain, and since that now uses this mechanism, it is addressed. Once HAGD and other IAGDB subclasses are removed, we can change HAGDBF as well.



rickardoberg and others added some commits Apr 14, 2015



Refactor InternalAbstractGraphDatabase ...

7cd76d0



Clean up how we reference JUnit+Hamcrest+Mockito in POM files. Make s... ...

240fd66



Fixed masking of startup exception with shutdown exception in IAGD ...

7b5471d



tinwelint merged commit **2149725** into `neo4j:2.3` Apr 27, 2015

[View details](#)

1 check passed

[Sign up for free](#)

to join this conversation on [GitHub](#). Already have an account? [Sign in to comment](#)

