Item 271
git_comments:

1. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, * software distributed under the License is distributed on an * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY * KIND, either express or implied. See the License for the * specific language governing permissions and limitations * under the License.
2. * * @author Stephen Mallette (http://stephen.genoprime.com)
3. * * When set to {@code true}, which is the default, the following methods will be generated to the DSL * implementation of the {@link GraphTraversalSource}: * * * {@link GraphTraversalSource#addV()} * {@link GraphTraversalSource#addV(String)} * {@link GraphTraversalSource#V(Object...)} * {@link GraphTraversalSource#E(Object...)} *

git_commits:

1. **summary:** TINKERPOP-786 Add option for no default method generation on GremlinDsl
   message: TINKERPOP-786 Add option for no default method generation on GremlinDsl

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Patterns for DSL Development
   description: Develop and document the patterns for custom DSLs. The method previously described and under consideration at 3.0.0 seems a bit cumbersome to deal with.

jira_issues_comments:

1. Some discussion on this topic on the mailing list: https://groups.google.com/d/msg/gremlin-users/HW_p51tA74g/FhxduVHvrxUJ
2. **body:** [~dkuppitz] We should work on section of the docs describing this stuff.
   label: documentation
3. Jeremy Hanna was interested in how this will look. Its easy. {code} public SocailTraversal<S,E> extends DefaultTraversal<S,E> { private final GraphTraversal<S,E> rawTraversal = new DefaultGraphTraversal<S,E>(); public SocialTraversal<S,Vertex> know(final String personName) { this.rawTraversal.out("knows").hasLabel("person").has("name",personName); return this; } public SocialTraversal<S,Vertex> olderThan(final String personName) { this.rawTraversal.filter(as("xx").V().has("name",personName").values("age").where(lt("xx"))); return this; } public SocialTraversal<S,Vertex> people() { this.rawTraversal.V().hasLabel("person"); return this; } public SocialTraversal<S,String> name() { this.rawTraversal.values("name"); return this; } public SocialTraversal<S,E> who() { return this; } public SocialTraversal<S,E> thatAre() { return this; } ..... } {code} They will then need a {{TraversalSource}} implementation which we will have a {{DefaultTraversalSource}} to make things easy. And finally, in action: {code} social = graph.traversal(SocialTraversal.class) social.people().who().know("stephen").thatAre().olderThan("daniel").name() {code} TADA!
4. This is now possible with the merging of TINKERPOP-971.

5. **body:** Perhaps we need a tutorial written on this topic at this point as it sounds like the pattern is decided. I'm going to switch the type on this issue to "documentation" and here's some recent discussion on the mailing list for ways DSLs can be implemented: https://groups.google.com/d/msg/gremlin-users/FO3bcXHWTrg/6H64dHrUFgAJ
   label: documentation
6. Unless I'm missing something, this approach doesn't enable extension of an existing traversal DSL. For my use case, and I imagine many others' as well, this ability is critical.
7. It occurs to me that I'm not sure how DSLs will work with the remoting API/GLVs. not sure if anyone has any thoughts on that.
8. **body:** Since all GLVs will probably use some form of code generator which will most likely be based on gremlin-java, I think the best option here would be to make those generators flexible enough so that they can be used with a DSL defined by users. When we define a common interface for the GLV generators, then users can simply call the generators on their own DSL and get the DSL in all GLVs. Maybe the interfaces could look something like this: # {{GraphTraversalGenerator}}: Takes a {{GraphTraversal}} class or a class derived from that as its input # {{GraphTraversalSourceGenerator}}: Takes a {{GraphTraversalSource}} class or a class derived from that as its input # {{EnumGenerator}}: Takes an enum class as its input (like {{T}}) # {{PredicateGenerator}}: Takes a predicate class as its input (like {{P}}) This would probably require some modifications to the existing GLV generators so that they implement those interfaces and also to make them flexible enough to work with a DSL (e.g. the generator should not only expect a {{GraphTraversal}} as the return type for steps but also a type that derives from that).
   label: code-design
9. **body:** That is a good idea [~Florian Hockmann]. That would definitely make it easier to "pump out" GLVs and keep them all unified and easier to maintain.
   label: code-design
10. GitHub user spmallette opened a pull request: https://github.com/apache/tinkerpop/pull/610 TINKERPOP-786 Gremlin DSL Support https://issues.apache.org/jira/browse/TINKERPOP-786 This pull request provides tools, documentation, tests and examples for how to develop DSLs for both Java/Groovy as well Python. Note the addition of the new `gremlin-archetype-dsl` module which demonstrates how the DSL system works with Java/Groovy and see `test_dsl.py` for the recommended method for python. Builds with `docker/build.sh -t -i -n` VOTE +1 You can merge this pull request into a Git repository by running: $ git pull https://github.com/apache/tinkerpop TINKERPOP-786 Alternatively you can review and apply these changes as the patch at: https://github.com/apache/tinkerpop/pull/610.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #610 ---- ----
11. Github user dkuppitz commented on the issue: https://github.com/apache/tinkerpop/pull/610 Slick! It's indeed much easier than I thought. VOTE: +1
12. This looks quite similar to the approach we have been using internally, almost a drop in replacement; very excited to see it, thanks Stephen. I'll take a look and see if it works for our use case; now that there's some existing code here we may be able to contribute some improvements if necessary.
13. Github user okram commented on the issue: https://github.com/apache/tinkerpop/pull/610 VOTE +1.
14. Github user n-tran commented on the issue: https://github.com/apache/tinkerpop/pull/610 +1
15. Github user asfgit closed the pull request at: https://github.com/apache/tinkerpop/pull/610