

Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

Cannot build minimal Spring Boot HATOEAS 1.0.0.BUILD-SNAPSHOT Example #731

New Issue

Closed spanierm opened this issue on 4 Aug 2018 · 9 comments



spanierm commented on 4 Aug 2018

Hi,

I just wanted to try out some new features of Spring HATEOAS 1.0.0.BUILD-SNAPSHOT. I took a look at <https://github.com/spring-projects/spring-hateoas-examples> and tried to recreate the example from scratch using the newest Spring Boot version and play around a little bit.

Unfortunately, I faced some issues when trying to use the 1.0.0.BUILD-SNAPSHOT version. I cannot build a minimal Spring Boot version (generated with Spring Initializr, Maven, Java, Spring Boot 2.0.4) that includes the 1.0.0.BUILD-SNAPSHOT version of Spring HATEOAS since I always face the following issue:

```
*****
APPLICATION FAILED TO START
*****

Description:

Parameter 0 of method linkDiscoverers in org.springframework.hateoas.config.HateoasConfig:
- entityLinksPluginRegistry: defined by method 'entityLinksPluginRegistry' in class
- relProviderPluginRegistry: defined by method 'relProviderPluginRegistry' in class
- linkDiscovererRegistry: defined in null

Action:

Consider marking one of the beans as @Primary, updating the consumer to accept multiple beans.

2018-08-04 12:41:11.123 ERROR 23639 --- [main] o.s.test.context.TestContextManager :

java.lang.IllegalStateException: Failed to load ApplicationContext
    at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadContext(DefaultCacheAwareContextLoaderDelegate.java:104)
    at org.springframework.test.context.support.DefaultTestContext.getApplicationContext(DefaultTestContext.java:113)
    at org.springframework.test.context.web.ServletTestExecutionListener.setUpRequestContextIfNecessary(ServletTestExecutionListener.java:190)
    at org.springframework.test.context.web.ServletTestExecutionListener.prepareTestInstance(ServletTestExecutionListener.java:142)
    at org.springframework.test.context.TestContextManager.prepareTestInstance(TestContextManager.java:261)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.createTest(SpringJUnit4ClassRunner.java:228)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runTest(SpringJUnit4ClassRunner.java:210)
    at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.methodBlock(SpringJUnit4ClassRunner.java:238)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChild(SpringJUnit4ClassRunner.java:84)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChild(SpringJUnit4ClassRunner.java:84)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
    at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)
    at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
    at org.junit.runners.ParentRunner.access$900(ParentRunner.java:58)
    at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)
    at org.springframework.test.context.junit4.statements.RunBeforeTestClassCallbacks.evaluate(RunBeforeTestClassCallbacks.java:61)
    at org.springframework.test.context.junit4.statements.RunAfterTestClassCallbacks.evaluate(RunAfterTestClassCallbacks.java:61)
    at org.junit.runners.ParentRunner.run(ParentRunner.java:363)
    at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.run(SpringJUnit4ClassRunner.java:143)
    at org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:34)
    at org.apache.maven.surefire.junit4.JUnit4Provider.executeWithRerun(JUnit4Provider.java:158)
    at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:143)
    at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:199)
    at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:187)
    at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:125)
    at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:413)
Caused by: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'org.springframework.hateoas.config.HateoasConfig': Unsatisfied dependency:
    at org.springframework.beans.factory.support.ConstructorResolver.createArgumentArray(ConstructorResolver.java:771)
    at org.springframework.beans.factory.support.ConstructorResolver.instantiateUsingFactoryMethod(ConstructorResolver.java:471)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.instantiateUsingFactoryMethod(AbstractAutowireCapableBeanFactory.java:1168)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:1084)
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:316)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:191)
    at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:861)
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:540)
    at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:762)
    at org.springframework.boot.SpringApplication.refreshContext(SpringApplication.java:100)
    at org.springframework.boot.SpringApplication.run(SpringApplication.java:308)
    at org.springframework.boot.test.context.SpringBootTestContextLoader.loadContext(SpringBootTestContextLoader.java:113)
    at org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate.loadContext(DefaultCacheAwareContextLoaderDelegate.java:138)
    ... 27 common frames omitted
Caused by: org.springframework.beans.factory.NoUniqueBeanDefinitionException: No qualifying beans of type 'org.springframework.hateoas.config.HateoasConfig' found:
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1045)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:954)
    at org.springframework.beans.factory.support.ConstructorResolver.resolveAutowiredArgument(ConstructorResolver.java:855)
    at org.springframework.beans.factory.support.ConstructorResolver.createArgumentArray(ConstructorResolver.java:771)
    ... 45 common frames omitted
```

The minimal project can be found at <https://github.com/spanierm/spring-hateoas-100-minimal>.

Did I miss a configuration? Any hints, why the issue occurs and how to fix it?

Thanks a lot in advance.



spanierm commented on 4 Aug 2018

Author

After submitting the issue, I took another look at the pom.xml file of <https://github.com/spring-projects/spring-hateoas-examples> and found the missing part:

```
<properties>
<spring-plugin.version>2.0.0.BUILD-SNAPSHOT</spring-plugin.version>
</properties>
```

Right after using this snapshot of the spring-plugin, everything worked as expected.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

4 participants



Seems like I was too blind to see it before :(

Anyways, thanks for this great project and the new features in `1.0.0` :)

3

 **spanierm** closed this on 4 Aug 2018



gregturn commented on 4 Aug 2018

Member ...

:)



spanierm commented on 4 Aug 2018

Author ...

@gregturn Btw, is there any roadmap when to release version `1.0.0`? How stable is the current build already?



gregturn commented on 4 Aug 2018

Member ...

I'll have to converse with @olivergierke. We are trying to get some bits into 1.0.0 that would support Affordances in Spring Data REST.



toedter commented on 27 Oct 2018 • edited

...

Great stuff. I am having a similar problem to run a minimal Spring Boot 2.0.6 project with HATEOAS 1.0.0.BUILD-SNAPSHOT and a Gradle Build. @gregturn, could you point me to an example how to use this with Gradle? I tried the 2.1.0.RC1 of the Spring Boot Gradle plugin, but it did not work...



gregturn commented on 27 Oct 2018

Member ...

Do you have spring-plugin set to 2.0.0.BUILD-SNAPSHOT? (NOT the same as the Spring Boot plugin).



toedter commented on 27 Oct 2018 • edited

...

yes, thanks. But I must have screwed up my Java dev environment. Currently I also cannot run the original repo with Maven anymore (getting marshaling errors). Will investigate...



rainer198 mentioned this issue on 28 Mar 2019

Application does not start due to two bean instances of org.springframework.plugin.core.PluginRegistry #966

 Closed



colin-young commented on 20 Jun 2019

Contributor ...

@toedter or @gregturn Do you have an example of exactly what the `spring-plugin` definition would look like in `build.gradle`? I cannot find any references to anything called "spring-plugin" other than a definition like `<spring-plugin.version>2.0.0.BUILD-SNAPSHOT</spring-plugin.version>` in the `pom.xml` of a different project that is building for me. The project I am currently working with is exhibiting the same problem as the original here, but is `build.gradle`-based and so far my attempts to convert it to `pom.xml` to see if I can figure it out from there have failed.



toedter commented on 20 Jun 2019

...

Currently everything is working fine, you find a working Gradle based example at <https://github.com/toedter/movies-demo/blob/master/backend/build.gradle>

2

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

