Debugging, Logging and Testing

Martin Ledvinka

martin.ledvinka@fel.cvut.cz

Winter Term 2024

Contents

Debugging

Debugging

- No software is perfect
- When an issue occurs, it is necessary to be able to:
  - Identify it
  - Reproduce it
  - Determine its cause
  - Fix it

Debugging

Multiple ways of identifying an issue:

1. Error in browser (Developer console/tools – *F12*)
2. Log entry
3. Exception stack trace
   - It is necessary to be able to determine which part of a stack trace is relevant and which is just the framework infrastructure
   - Enterprise applications → long stack traces
     - Hint: Look for the last `Caused by` entry for the exception
     - Hint: Look for stack trace entries with your application packages

Stack trace

```
05-10-2016 10:14:32.617 [RMI TCP Connection(2)-127.0.0.1] WARN  o.s.w.c.s.AnnotationConfigWebApplicationContext - Exception encountered during context initialization - cancelling refresh attempt
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'reportController': Injection of autowired dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException:
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessPropertyValues(AutowiredAnnotationBeanPostProcessor.java:334)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1202)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:537)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:476)
    at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:303)
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:299)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:755)
    at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:757)
    at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:480)
    at org.springframework.web.context.ContextLoader.configureAndRefreshWebApplicationContext(ContextLoader.java:403)
    at org.springframework.web.context.ContextLoader.initWebApplicationContext(ContextLoader.java:306)
    at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:106)
    at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4726)
    at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5166)
    at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:150)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:725)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:701)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:717)
    at org.apache.catalina.startup.HostConfig.manageApp(HostConfig.java:1618) <4 internal calls>
    at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:300)
    at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)
    at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)
    at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:463)
    at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:413) <4 internal calls>
    at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:300)
    at com.sun.jmx.interceptor.DefaultMBeanServerInterceptor.invoke(DefaultMBeanServerInterceptor.java:819)
    at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)
    at javax.management.remote.rmi.RMIConnectionImpl.doOperation(RMIConnectionImpl.java:1468)
    at javax.management.remote.rmi.RMIConnectionImpl.access$300(RMIConnectionImpl.java:76)
    at javax.management.remote.rmi.RMIConnectionImpl$PrivilegedOperation.run(RMIConnectionImpl.java:1309)
    at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1401)
    at javax.management.remote.rmi.RMIConnectionImpl.invoke(RMIConnectionImpl.java:829) <16 internal calls>
    at java.lang.Thread.run(Thread.java:745)
Caused by: org.springframework.beans.factory.BeanCreationException: Could not autowire field: private cz.cvut.kbss.inbas.reporting.service.ReportBusinessService cz.cvut.kbss.inbas.reporting.rest.ReportController.reportService; n
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:561)
    at org.springframework.beans.factory.annotation.InjectionMetadata.inject(InjectionMetadata.java:88)
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessPropertyValues(AutowiredAnnotationBeanPostProcessor.java:331)
    ... 58 common frames omitted
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'cachingReportBusinessService': Injection of autowired dependencies failed; nested exception is org.springframework.beans.factory.
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessPropertyValues(AutowiredAnnotationBeanPostProcessor.java:334)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1202)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:537)
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:476)
    at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:303)
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:299)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.findAutowireCandidates(DefaultListableBeanFactory.java:1120)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1044)
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:942)
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:533)
    ... 60 common frames omitted
```

Stack trace II

```
java.lang.NullPointerException: null
    at cz.cvut.kbss.inbas.reporting.persistence.dao.BaseDao.persist(BaseDao.java:70)
    at cz.cvut.kbss.inbas.reporting.service.repository.BaseRepositoryService.persist(BaseRepositoryService.java:36)
    at cz.cvut.kbss.inbas.reporting.service.MainReportService.persist(MainReportService.java:80)
    at cz.cvut.kbss.inbas.reporting.service.CachingReportBusinessService.persist(CachingReportBusinessService.java:39)
    at cz.cvut.kbss.inbas.reporting.rest.ReportController.createReport(ReportController.java:43)
    at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204)
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:717)
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:157)
    at org.springframework.security.access.intercept.aopalliance.MethodSecurityInterceptor.invoke(MethodSecurityInterceptor.java:68)
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:179)
    at org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept(CglibAopProxy.java:653)
    at cz.cvut.kbss.inbas.reporting.rest.ReportController$$EnhancerBySpringCGLIB$$8793688a.createReport(<generated>) <14 internal calls>
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:648) <1 internal calls>
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:729)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:291)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:316)
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:126)
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFilter(FilterSecurityInterceptor.java:90)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.access.ExceptionTranslationFilter.doFilter(ExceptionTranslationFilter.java:114)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.session.SessionManagementFilter.doFilter(SessionManagementFilter.java:122)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.authentication.AnonymousAuthenticationFilter.doFilter(AnonymousAuthenticationFilter.java:111)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter.doFilter(SecurityContextHolderAwareRequestFilter.java:169)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.savedrequest.RequestCacheAwareFilter.doFilter(RequestCacheAwareFilter.java:48)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.session.ConcurrentSessionFilter.doFilter(ConcurrentSessionFilter.java:133)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.authentication.AbstractAuthenticationProcessingFilter.doFilter(AbstractAuthenticationProcessingFilter.java:205)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.authentication.logout.LogoutFilter.doFilter(LogoutFilter.java:120)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.header.HeaderWriterFilter.doFilterInternal(HeaderWriterFilter.java:64) <1 internal calls>
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.context.SecurityContextPersistenceFilter.doFilter(SecurityContextPersistenceFilter.java:91)
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter.doFilterInternal(WebAsyncManagerIntegrationFilter.java:53) <1 internal calls>
    at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
    at org.springframework.security.web.FilterChainProxy.doFilterInternal(FilterChainProxy.java:213)
    at org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:176) <2 internal calls>
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
    at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:239)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:219)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:106)
    at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:502)
    at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:142)
    at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:79)
```

Logging

Logging

- Logging provides important auditing information about the application
- Good to log:
  - Exceptions (duh)
  - Application state information
  - Input/output data information
- Configurable levels
  - Minimal logging for production
  - Maximal for debugging/testing

Java - logging frameworks

- **JUL (java.util.logging)**
  - Part of JDK, no external libraries necessary
  - multitude of logging levels
- **Apache Log4j**
  - Leading logging frameworks for years
  - Easy configuration
- **Simple Logging Facade for Java (SLF4J)**
  - Logging facade, requires implementation
  - This separation is favourable – possible to use with JUL, Log4j, logback or others
  - Current logging leader
- **logback**
  - Successor of the original Log4j
  - Today's most common configuration – SLF4J + logback

Logging levels

From SLF4J:

1. Trace
2. Debug
3. Info
4. Warn
5. Error
6. Fatal
7. All

Testing

Enterprise Application Testing

We are...

assuming you already have experience with testing.

Enterprise application testing is:

- Harder in certain aspects
- More about the right configuration
- Application frameworks (like Spring) try to help with testing, e.g., by providing restricted DI container for tests, so that DI still works
- Otherwise similar to the usual:
  - JUnit, TestNG
  - Mockito
  - Cucumber, JBehave
  - WebDriver/Selenium
- Should be automated as much as possible
  - Big applications, not enough testers, not enough time

Code coverage

- How much of the code is covered by tests
- Coverage of lines, branches, classes, methods etc.
- **100% coverage does not mean code without bugs**
- JaCoCo, Cobertura
- Testing and coverage can be configured as part of Maven build

Task

Syncing Your Fork

1. Ensure you have no uncommitted changes in the working tree
   - `git status` → your branch is up to date, nothing to commit
2. Fetch branches and commits from the upstream repository (EAR/B241-eshop)
   - `git fetch upstream`
3. Check out the task branch from **upstream** (one line!)
   - `git checkout -b b241-seminar-06-task upstream/b241-seminar-06-task`
4. Do the task
5. Commit and push the solution to **your** fork
   - `git push -u origin b241-seminar-06-task`

E-shop Tests Tour

Let's go through the tests in our e-shop.

- In-memory database configured
- Spring Boot-based configuration
  - `@SpringBootTest`
  - `@DataJpaTest`
- Transactional with automatic rollback
  - Test runs in a transaction which is rolled back in the end to keep the database clean

Together

- Application won't deploy
- Let's examine the stack trace and fix the issue

Task – 1 point

We have received the following issue reports:

Issues #006

When I attempt to add a product to a category for the first time, the application throws a `NullPointerException`. However, if the product already belongs to some category, no such issue occurs.

Issues #007

When I create an order, all the items remain in the cart. As far as I am concerned, creating an order should move the items from cart to the new order and the cart should be empty.

Task – Cont.

- Write suitable tests reproducing the issues. For issue #006, the tests should cover both cases (i.e., when the product does not belong to any category and when it already belongs to some category), as the behavior needs to be consistent
  - That is, the tests should fail before the fix
- Fix the issues
- Verify that the tests pass
- Note: Make sure to find a suitable spot where to test the behavior. Also look for existing code generating tests data
- **Acceptance criteria:** Tests reproducing the original issues exist (you will need at least three tests). The issues are fixed and the tests pass.

Resources

- Spring Testing
  - `https://docs.spring.io/spring/docs/current/spring-framework-reference/testing.html`
- Spring Boot Testing
  - `https://docs.spring.io/spring-boot/docs/current/reference/html/features.html#features.testing`
- JaCoCo Maven Plugin
  - `https://www.jacoco.org/jacoco/trunk/index.html`
- SJF4J Docs
  - `http://www.slf4j.org/docs.html`