

Debugging, Logging and Testing

Martin Ledvinka

martin.ledvinka@fel.cvut.cz

Winter Term 2018



Contents

- 1 Debugging
- 2 Logging
- 3 Testing
- 4 Task



Debugging



Debugging

- No application 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 [RM1 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:1203)
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:489)
at org.springframework.web.context.ContextLoader.configureAndRefreshWebApplicationContext(ContextLoader.java:463)
at org.springframework.web.context.ContextLoader.initalizeWebApplicationContext(ContextLoader.java:386)
at org.springframework.web.context.ContextLoaderListener.contextInitialized(ContextLoaderListener.java:106)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:4726)
at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5366)
at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:159)
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:380)
at com.sun.jmx.interceptor.DefaultJMXBeanServerInterceptor.invoke(DefaultJMXBeanServerInterceptor.java:819)
at com.sun.jmx.mbeanserver.JMXMBeanServer.invoke(JMXMBeanServer.java:83)
at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:463) <4 internal calls>
at org.apache.catalina.mbeans.MBeanFactory.createStandardContext(MBeanFactory.java:413) <4 internal calls>
at org.apache.tomcat.util.modeler.BaseModelMBean.invoke(BaseModelMBean.java:380)
at com.sun.jmx.interceptor.DefaultJMXBeanServerInterceptor.invoke(DefaultJMXBeanServerInterceptor.java:819)
at com.sun.jmx.mbeanserver.JMXMBeanServer.invoke(JMXMBeanServer.java:83)
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:1399)
at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1481)
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.vut.kbss.inbas.reporting.service.ReportBusinessService cz.vut.kbss.inbas.reporting.rest.ReportController.reportService;
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 'achingReportBusinessService': 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:1203)
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$1.getObject(AbstractBeanFactory.java:303)
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:299)
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.findAutowireCandidates(DefaultListableBeanFactory.java:1126)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1044)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:942)
at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:531)
... 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:39)
    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.scheduling.annotation.AsyncMethodInvoker.invoke(AsyncMethodInvoker.java:58)
    at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204)
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:717)
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:157)
    at org.springframework.security.access.intercept.aopalliance.MethodSecurityInterceptor.invoke(MethodSecurityInterceptor.java:68)
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:179)
    at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeIntercept(CglibAopProxy.java:653)
    at cz.cvut.kbss.inbas.reporting.rest.ReportController$$EnhancerBySpringCGLIB$$5879368Ba.createReport(<generated>) <1 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$VirtualFilterChain.doFilter(FilterChainProxy.java:316)
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:126)
    at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFilter(FilterSecurityInterceptor.java:99)
    at org.springframework.security.web.FilterChainProxy$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:44) <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
- Much more about the right configuration
- Otherwise similar to the usual:
 - JUnit
 - Mockito
 - Cucumber
 - 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



E-shop Tests Tour

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

- In-memory database configured
- Spring-based configuration in common base classes
- Transactional with automatic rollback

Together

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



Task – 1 point

Check out branch `b181-seminar-06-task` of the e-shop project (<https://gitlab.fel.cvut.cz/ear/b181-eshop>). We have received an issue report stating the following:

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.

- Write a suitable test reproducing the issue
- Fix the issue
- Verify that the test passes
- **Acceptance criteria:** A test reproducing the original issue exists. The issue is fixed and the test passes.



Resources

- Spring Testing
 - <https://docs.spring.io/spring/docs/current/spring-framework-reference/testing.html>
- JaCoCo Maven Plugin
 - <http://www.eclemma.org/jacoco/trunk/doc/maven.html>
- SJF4J Docs
 - <http://www.slf4j.org/docs.html>

