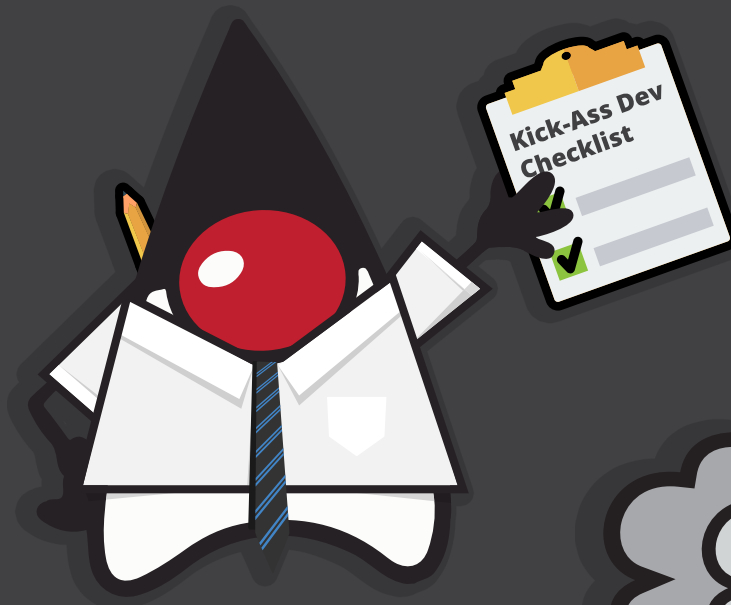


ROCKET-POWERED JAVA DEVELOPMENT

Prepare for Launch



Introduction

Java is now over 20 years old, and it is one of the most popular programming languages in the world. Even so, Java programming comes with many challenges. Luckily, we are dedicated to providing revolutionary developer tools to help you write better code, faster. If you're a Java developer, follow along with this paper to learn how to get back one full work-month per year on average with only a 5 minute investment.

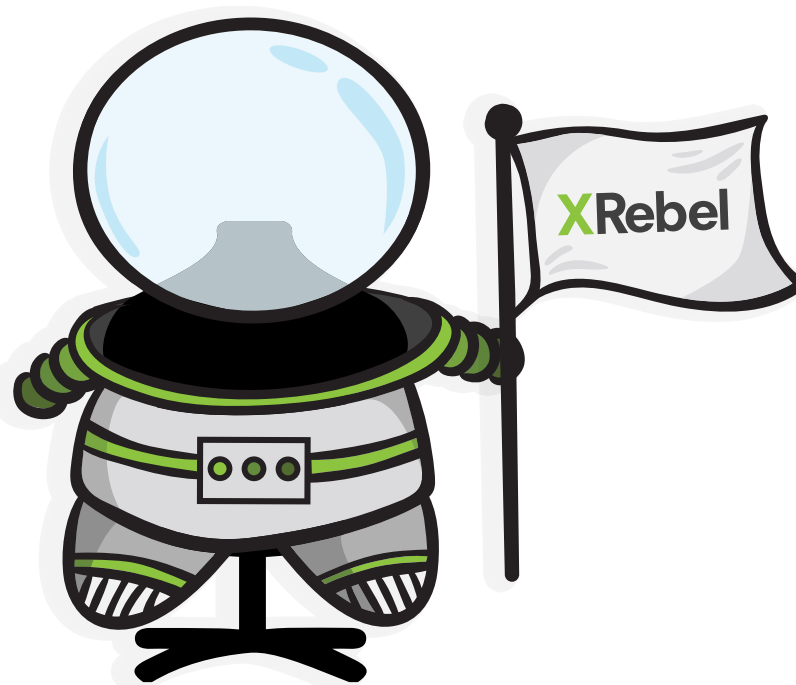
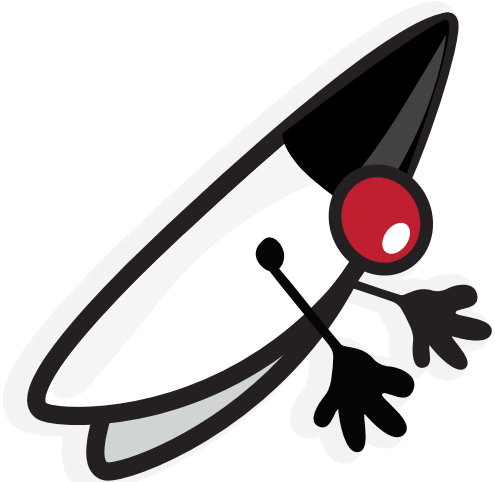


Over 65,000 Java developers are using JRebel today to help them code 17% faster, on average. And now, with the addition of our newest developer productivity tool, XRebel, we can also help you write better code, by finding and fixing issues along the way. Don't you want to save time and write better code too?

JRebel, Java's famed redeploy killer, allows you to quickly make changes and see the impact by reloading those changes immediately. XRebel, the lightweight Java profiler, provides the ability to find and fix issues as you code. These two tools used together have a huge positive impact on how you spend your day coding.

JRebel and XRebel are revolutionizing the way development teams work with Java. Over half of the Fortune 100 use JRebel to turn the laborious build/test/deploy phase of application development into a light-speed cycle as fast and iterative as Python or PHP. XRebel, the lightweight Java profiler, helps developers notice and address potential performance problems as they code. So in the same amount of time, you can write more, better code. Sound good?

This paper will review some of the advantages that you will get by using JRebel and XRebel. Prepare for takeoff with this checklist of techniques for rocket-powered Java development.



Rocket-powered Java Development Booster #1

SEE ALL CHANGES IMMEDIATELY

Does your code smell? Find out really quickly and run your own mini-tests throughout the day. Remove bad code before it goes to QA and they make fun of you.

With JRebel, you get to reload your changes immediately after compiling them. When the IDE builds classes automatically, the process is as fast as using a scripting language or testing your latest HTML changes. In fact, we urge you to try compiling the following code snippet, start it up with JRebel enabled, and change the “Hello World” part to something else while the program is running.

```
public class TestLoop {
    public static void main(String... args) throws Exception {
        while (true) {
            saySomething();
            Thread.sleep(1000);
        }
    }
    private static void saySomething() {
        System.out.println("Hello World");
    }
}
```

For us, the console output following the JRebel console output was (yes, we’re fast typers):

```
Hello World
Hello World
Hello World
JRebel: Reloading class 'TestLoop'.
Goodbye Redeploys
Goodbye Redeploys
```

This opens up a world of possibilities that have never before been available for Java developers. It combines the power of Java with the fast turnaround of dynamic scripting languages. You don’t have to queue up a gazillion changes before redeploying. You can code by making small incremental changes and alt-tabbing to your browser to verify the results.

With JRebel, you get more time to meet your deadlines, research & learn, and *gasp* communicate with people. According to a recent report on developer stress, it came out that developers really do care about making deadlines and their level of expertise, wishing they had more time to spend on education.

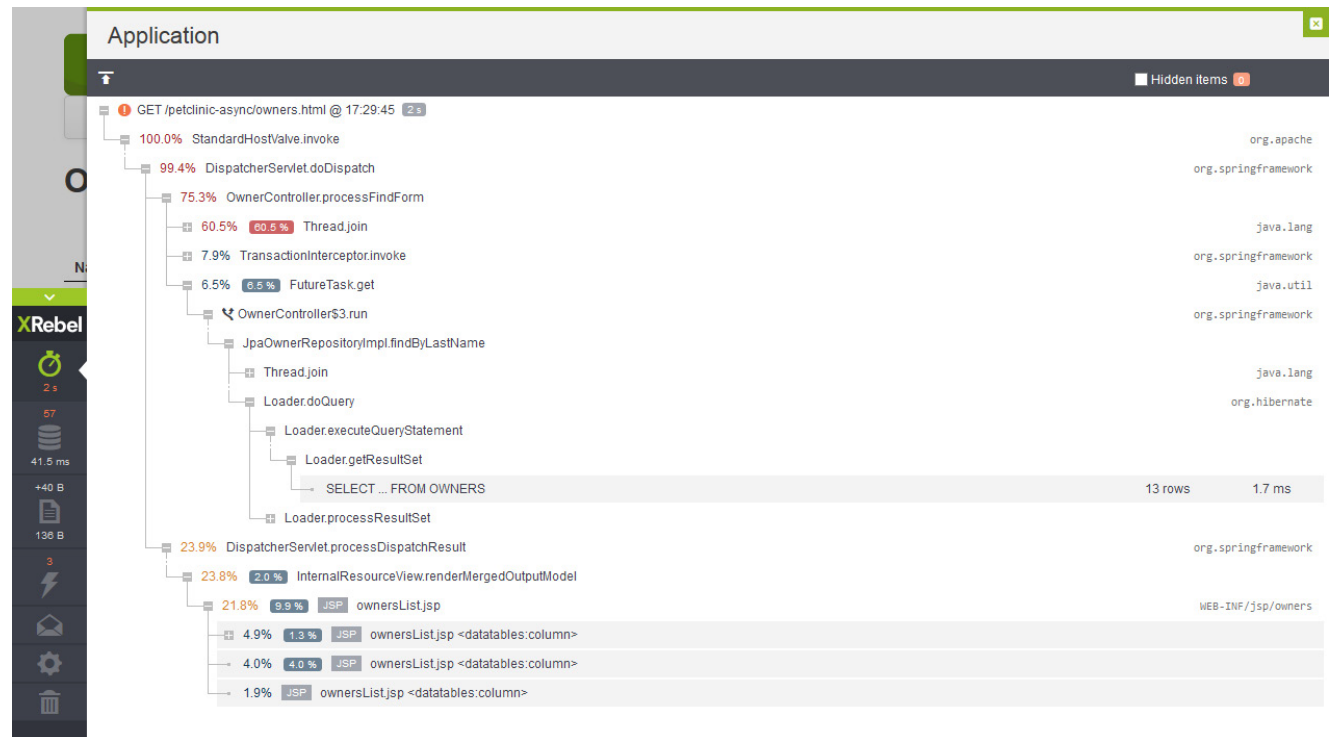
Rocket-powered Java Development Booster #2

APPLICATION PROFILING *AS YOU CODE* (WOOT!)

How performant is your code? With XRebel, you can get an understanding of your application performance *as you code!* Unlike typical profilers and APM tools, XRebel provides alerts in a widget in your web app in real-time as you write your application.

XRebel gives you instant feedback for time spent on each individual request in your application so you can identify the slowest methods. If you see that something takes an unusual amount of time, you can dig right into it and fix your code. You already learned how you can make instant changes to your code without reloading with JRebel — leveraging both tools means that you can find and fix issues and see improvements *immediately*. Pretty cool, eh?

How much time does it take to learn *yet another* development tool? An intuitive, hierarchical view makes it easy to see which layers of your application are involved as your application executes. This makes it easy to troubleshoot and figure out what is really causing the problem, even for the most junior developers.



Pretty nice view into your app, isn't it?

Rocket-powered Java Development Booster #3

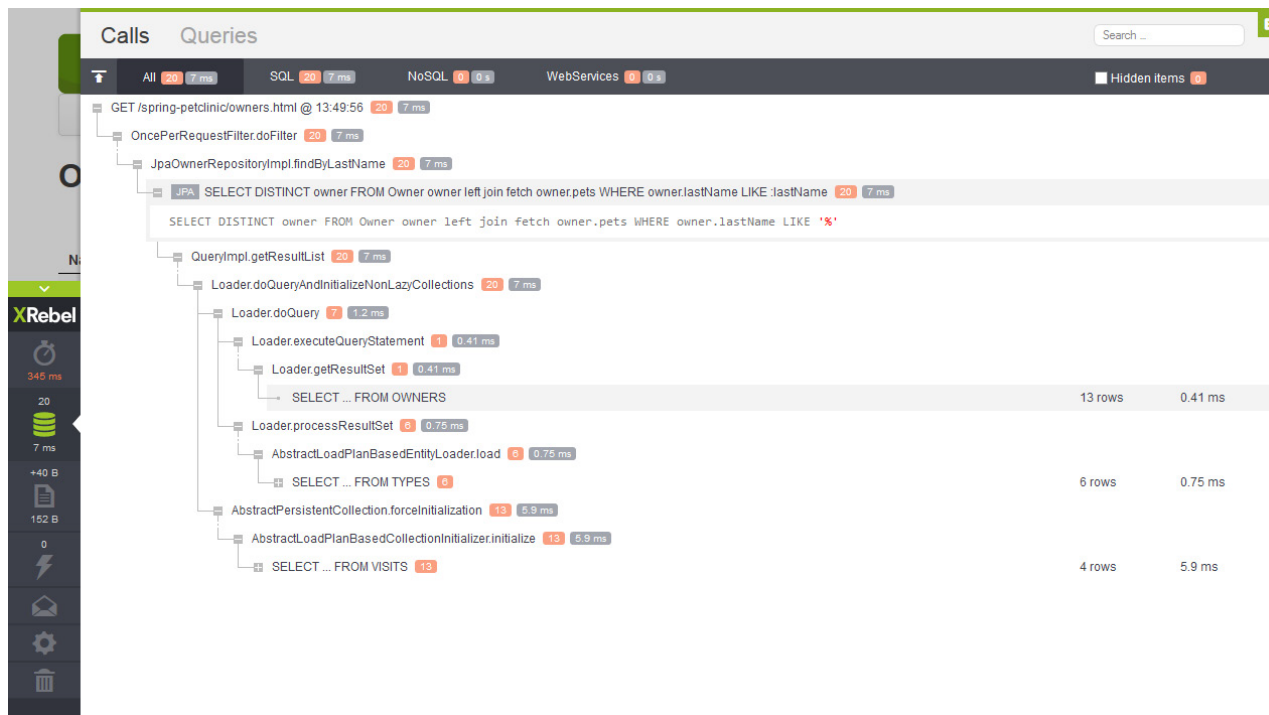
PROFILE JDBC AND NOSQL DATABASES

Have you ever had something wonky going on with your database, but you aren't sure what? XRebel can help. While you are coding, you can see if weird things are happening with your database queries. XRebel identifies excessive database calls, unusual NoSQL queries, and questionable JPA queries. MongoDB, Cassandra, HBase, and Neo4J - we got it covered!

The call tree in the Calls tab on the XRebel UI provides a trace of all queries from the start of the execution leading to the IO event. This gives the user insight, as to how the query was executed. You will be able to quickly get insight into the number of queries and also the time taken for the queries to be rendered. With this information you will be able to easily detect common issues like N+1 issues. Once you find issues, fix them easily and view changes without having to redeploy using JRebel.

To summarize, you can drill into your data-related queries using the simple user interface to locate the root cause and connect your queries with a view of the call stack. Fix it up, refresh your browser, and huzzah!!! you solved it!

With XRebel you also have the ability to hide items that you don't care about by simply clicking on the hide button. You can also unhide them if you choose to. This makes it easy for you to analyze the data and focus on the parts that you really care about. This feature will come in handy when you are profiling large Java enterprise applications.



Get a deep dive into the actual queries run against the database.

Rocket-powered Java Development Booster #4

FIND WEB SERVICES BOTTLENECKS

With XRebel, you can also get a detailed overview of the executed HTTP invocations and inspect web service calls for REST, SOAP, XML-RPC and even plain old `HttpURLConnection`.

Ever been frustrated with how your app can slow down due to extra HTTP requests? The HTTP client may execute multiple requests instead of just one request and you might not even notice! HTTP redirects are the cause of the extra requests. Also, it is not easy to spot those extra requests if you're not specifically looking for them to appear.

From a Java application developer's point of view, HTTP redirects are supposed to be straightforward – with no special handling required. But the underlying library is what handles the communication, including HTTP redirects. You can get notified when an application has slow HTTP requests, based on configurable thresholds. This can help you detect and fix slow HTTP calls. You can then update your code in real-time and see if the problem is now resolved. Fastest feedback loop ever!

The screenshot shows the XRebel tool interface. At the top, a summary bar indicates 'Target.makeRequest' with a red '2' icon and a '212 ms' duration. Below this, two HTTP requests are listed:

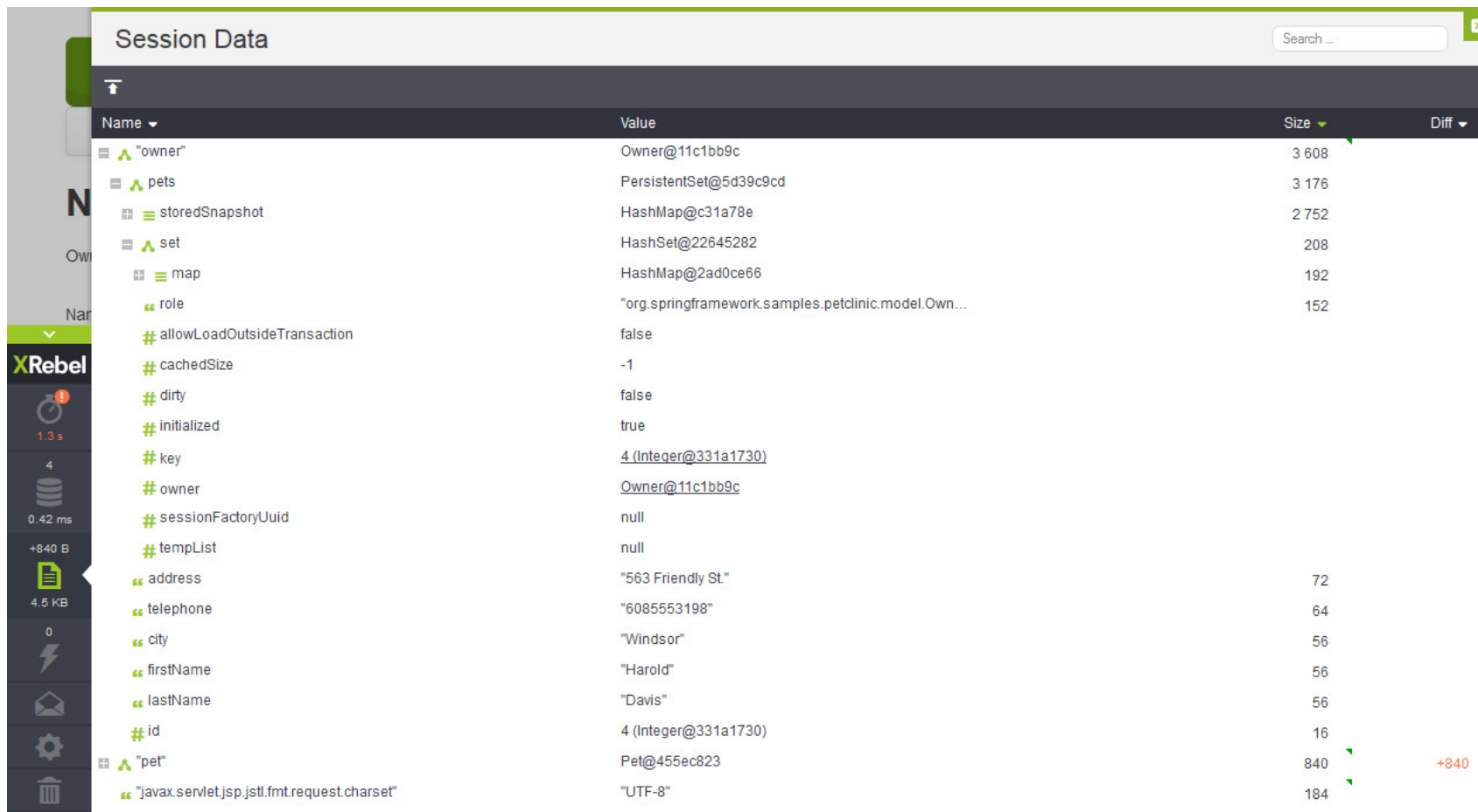
- Request 1:** GET `https://www.google.com/` with status `302 Found` and duration `138.5 ms`. The response headers are expanded, showing:
 - `Location: https://www.google.ee/?q=XRebel&gws_rd=cr&ei=FggQVeDxF4muPN_BgegN` (highlighted with a red box)
 - `Cache-Control: private`
 - `Content-Type: text/html; charset=UTF-8`
 - `Server: gws`
 - `Content-Length: 270`
 - `X-XSS-Protection: 1; mode=block`
 - `X-Frame-Options: SAMEORIGIN`
 - `Alternate-Protocol: 443:quic,p=0.5`
- Request 2:** GET `https://www.google.ee/` with status `200 OK` and duration `73.5 ms`. The request URL is highlighted with a red box: `https://www.google.ee/?q=XRebel&gws_rd=cr&ei=FggQVeDxF4muPN_BgegN`.

XRebel can easily visualize what happened during this request to give you a sense for what is going on under the hood. This is done by tracing the invocations inside the client library. Given you are not aware that HTTP redirects are involved, this might be costing you twice as much as it should be. XRebel can reveal such situations early, before your application goes live!

Rocket-powered Java Development Booster #5

UNDERSTAND SESSION FOOTPRINT

XRebel can help you find session hogs. You can drill into the session object graph and single out bulky objects. You can also track changes in a specific session and find the biggest objects added in the latest request. When you make code changes to eliminate session hogs, you can reload your changes immediately with JRebel and see the impact via XRebel. You can verify code changes against your existing session, thanks to JRebel maintaining state and avoiding the redeploy.

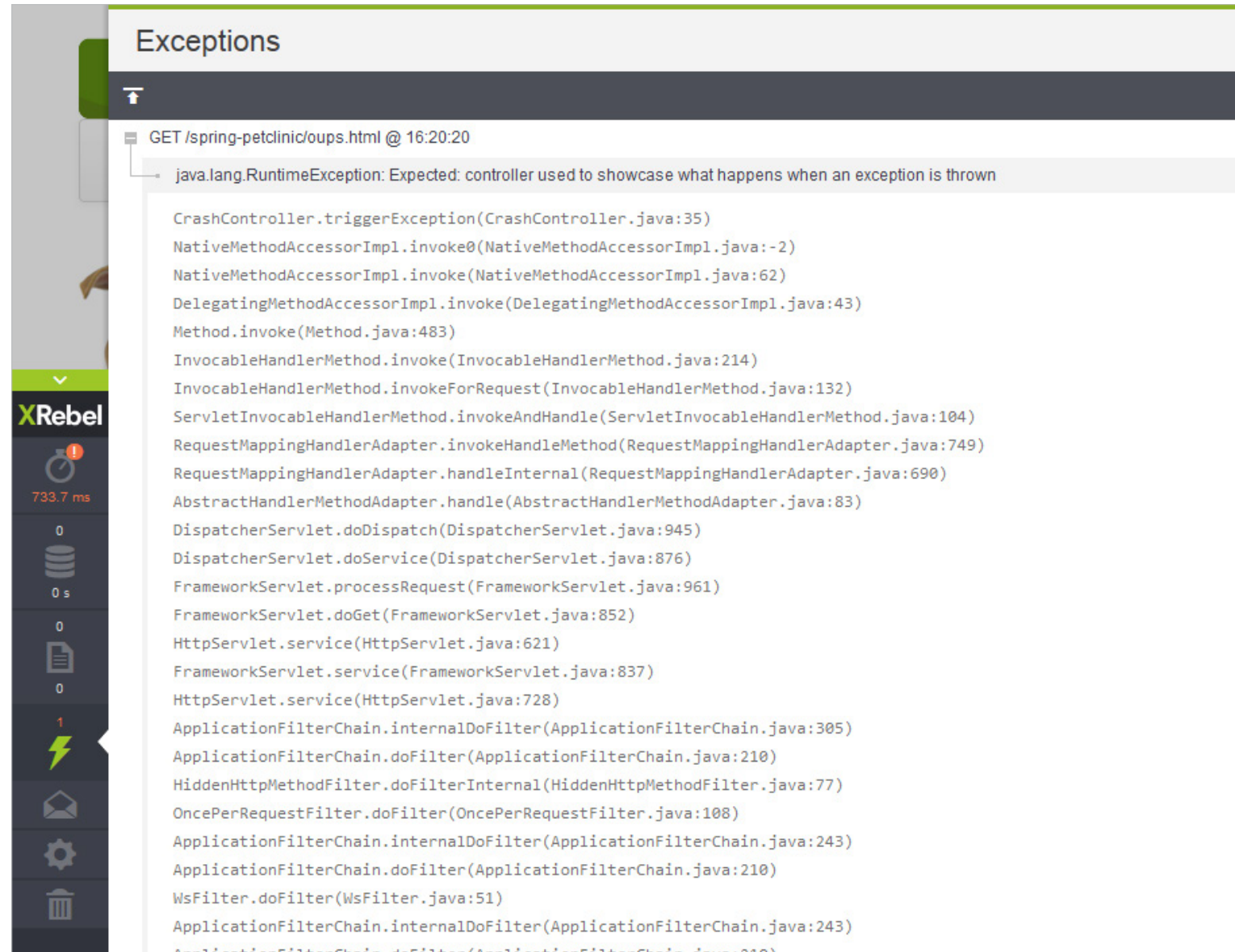


Name	Value	Size	Diff
"owner"	Owner@11c1bb9c	3 608	
pets	PersistentSet@5d39c9cd	3 176	
storedSnapshot	HashMap@c31a78e	2 752	
set	HashSet@22645282	208	
map	HashMap@2ad0ce66	192	
role	"org.springframework.samples.petclinic.model.Own...	152	
allowLoadOutsideTransaction	false		
cachedSize	-1		
dirty	false		
initialized	true		
key	4 (Integer@331a1730)		
owner	Owner@11c1bb9c		
sessionFactoryUuid	null		
tempList	null		
address	"563 Friendly St."	72	
telephone	"6085553198"	64	
city	"Windsor"	56	
firstName	"Harold"	56	
lastName	"Davis"	56	
id	4 (Integer@331a1730)	16	
"pet"	Pet@455ec823	840	+840
"javax.servlet.jsp.jstl.fmt.request.charset"	"UTF-8"	184	

Rocket-powered Java Development Booster #6

TRACK DOWN HIDDEN EXCEPTIONS

XRebel also provides the capability to uncover hidden exceptions. You can use the exceptions view to reverse Java stacktraces and find buried exceptions. Argh!!! You can drill in to find the root cause of the exception. Then, you can quickly iterate on your code and see your results immediately with JRebel.



The screenshot shows the XRebel interface with the 'Exceptions' view open. The view displays a list of exceptions, with the selected one being a `java.lang.RuntimeException: Expected: controller used to showcase what happens when an exception is thrown`. The stack trace for this exception is visible, starting from `CrashController.triggerException(CrashController.java:35)` and going down to `ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)`. The XRebel sidebar on the left shows various performance metrics, including a timer icon with a red exclamation mark and a value of 733.7 ms.

```
Exceptions
GET /spring-petclinic/oups.html @ 16:20:20
  java.lang.RuntimeException: Expected: controller used to showcase what happens when an exception is thrown
    CrashController.triggerException(CrashController.java:35)
    NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:-2)
    NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    Method.invoke(Method.java:483)
    InvocableHandlerMethod.invoke(InvocableHandlerMethod.java:214)
    InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:132)
    ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:104)
    RequestMappingHandlerAdapter.invokeHandleMethod(RequestMappingHandlerAdapter.java:749)
    RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:690)
    AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:83)
    DispatcherServlet.doDispatch(DispatcherServlet.java:945)
    DispatcherServlet.doService(DispatcherServlet.java:876)
    FrameworkServlet.processRequest(FrameworkServlet.java:961)
    FrameworkServlet.doGet(FrameworkServlet.java:852)
    HttpServlet.service(HttpServlet.java:621)
    FrameworkServlet.service(FrameworkServlet.java:837)
    HttpServlet.service(HttpServlet.java:728)
    ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:305)
    ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
    HiddenHttpMethodFilter.doFilterInternal(HiddenHttpMethodFilter.java:77)
    OncePerRequestFilter.doFilter(OncePerRequestFilter.java:108)
    ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:243)
    ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
    WsFilter.doFilter(WsFilter.java:51)
    ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:243)
    ApplicationFilterChain.doFilter(ApplicationFilterChain.java:210)
```

Rocket-powered Java Development Booster #7

FINE-TUNING THROUGH INCREMENTAL CHANGES

This works best when you have to implement some logic that is tricky to get just right. Imagine you have to add validation to your form. With JRebel, you can add the rules field by field and play around with the way errors should be displayed until satisfied by the results. You can even add some new fields to your controller or validator with JRebel as well. JRebel preserves all application state, which means it does not matter how you reached the page you're modifying. Even if you have a really complicated state required for the validator, create it once, then keep using it!

JRebel actually goes way beyond "simple" class reloading. Applications today take advantage of frameworks and Java technologies, such as Java EE, Hibernate, Spring Framework, JSF, Seam, Struts, etc. The thing with frameworks and technology implementations is that they initialize themselves when the application starts up and generate lots of metadata. It is usually a good thing since nobody wants Hibernate to re-initialize itself for every HTTP request.

JRebel has addressed this by having explicit support for 100+ popular frameworks and application containers. This means when you make a change to a "special" resource such as adding a new EJB, changing JPA mappings (either via xml or annotations), or adding new Spring Beans (again, either via xml or annotations), JRebel updates the metadata to reflect those changes. Your job is to press F5 (or CMD+R for our Mac friends) in the browser.

JRebel also makes remote debugging more useful than ever. Imagine you are using it to hunt down some difficult bug. Now you can hit a breakpoint, investigate what's going on, make a change, resume execution, and see the result almost immediately! The new code kicks in right after compiling, next time when the affected method is invoked. You don't even have to worry about getting the code to the JVM. You can use this approach to fix discovered bugs immediately or to quickly add some debugging statements in the code (just remember to avoid committing them). You can even drop to a parent frame and check the effect immediately!

Rocket-powered Java Development Booster #8

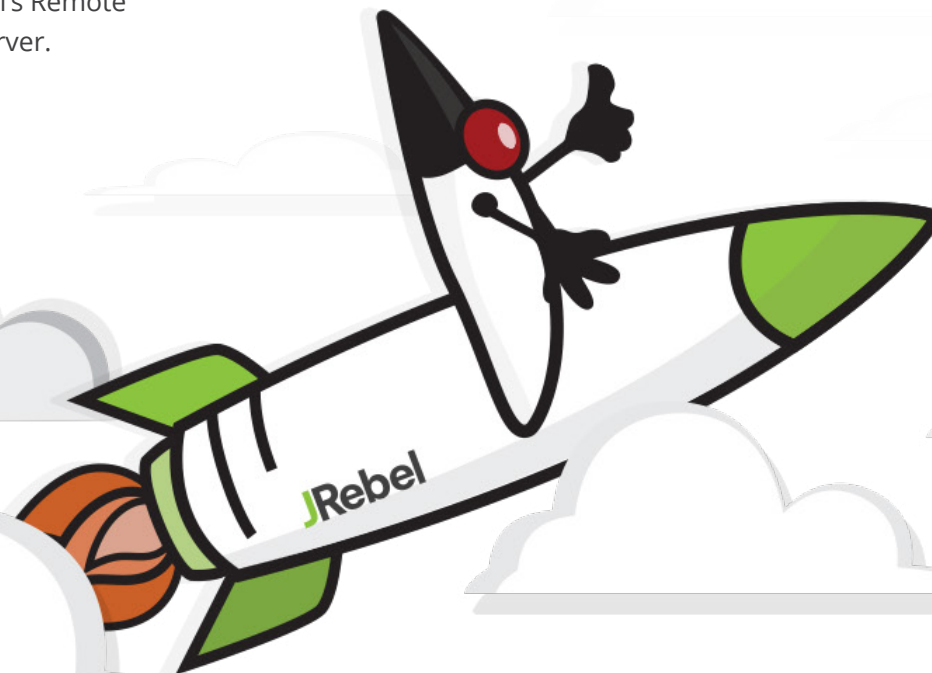
DEPLOYING CODE LOCALLY OR REMOTELY

It's great when a developer can check out code from a repository, compile it, and deploy the thing on his own machine. Sometimes this is not possible.

It can be due to performance reasons, or the whole system is simply too complex. Sometimes the system has to run in a different network than the developer's, which cannot be easily simulated. Some developers may be using low-powered laptops. Your project may be deployed onto a cloud computing platform. JRebel is integrated with leading cloud platforms, including Amazon Web Services, Red Hat OpenShift, IBM Bluemix and SAP HANA Cloud Platform. Whatever the reason, sometimes running the app on the same machine is not an option.

JRebel can help even in this case and the capability to make a system deployable on a developer's machine is not crucial anymore. JRebel's Remote lets you push changes from a developer's machine to a remote server.

You can even let multiple developers loose on a single application instance on a remote machine. By default, JRebel Remote reverts back to the initial codebase when the app server is restarted. You can set up an application on the remote machine, have a developer have his way with it, restart it (ok, there's one redeploy, sorry), and allow another developer do his magic on the app. This is the case when both developers are making changes to the same application. When there are several apps deployed to the remote server, developers can bombard them without stepping on each other's toes.



Final Words

If you want to develop at lightspeed, JRebel and XRebel together work hand in hand to enable you to be nimble and fast – the performance, scalability and robustness of Java, as well as the FTL-turnaround and the feeling of lightness, which are typically associated with 100 line guestbook applications, not with big mature systems. JRebel and XRebel automagically work to give you the right stuff – you don't have to make changes to your code, architecture or sysops. They adapt to your environment

and requirements. All you have to do is start using them. You get to keep all the effort and investments already made to your infrastructure.

We've heard from thousands of developers directly about how JRebel and XRebel have taken their coding to the next level. We hear not only about how productive they work – saving more than 5 full work weeks of time otherwise lost to Java restarts each year – but the way in which they actually write the code, how they discover

faults and test iteratively in order to minimize service requests and bugs.

The joy we provide is achieved in a sensible way, not via some dark magic or something “on rails”. Nothing in your entire app (frameworks included) is vague, ambiguous or hidden from you. All we do is some good engineering with JVMs and popular OSS frameworks with no runtime penalties or lingering dependencies. Download [JRebel](#) and [XRebel](#) and try it for yourself!



 **ZERO TURNAROUND**

REVOLUTIONARY
DEVELOPER TOOLS



REVOLUTIONARY DEVELOPER TOOLS



Contact Us

Twitter: [@zeroturnaround](#)

Web: <http://zeroturnaround.com>

Email: info@zeroturnaround.com

Estonia

Ülikooli 2, 4th floor
Tartu, Estonia, 51003
Phone: +372 653 6099

USA

399 Boylston Street,
Suite 300, Boston,
MA, USA, 02116
Phone: +1 857 277-1199

Czech Republic

Jankovcova 1037/49
Building C, 5th floor,
170 00 Prague 7, Czech Republic
Phone: +420 227 020 130