Adopt-a-JSR workshop: JCache

Contents

Chapter 1

Introduction (10 mins)

1.1 JCache

JCache, apart from being one of the longest running JSRs (about 13 years from 2001 to 2014), is a Java API that provides a unified mechanism for interacting with various caching implementations. The operations provided by the API allow for a uniform way to access, update, create and remove entries from a cache.

1.2 Payara

Payara is a Java application server derived from the Glassfish code base. Support for JCache is provided to Payara by means of the Hazelcast JCache provider.

Chapter 2

Prerequisites (20 mins)

2.1 Setup your IDE for Java EE development

Download Eclipse, IntelliJ or NetBeans with JavaEE support:

- Eclipse for Java EE developers package can be downloaded from http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/mars2;
- IntelliJ community edition can be downloaded from https://www.jetbrains.com/idea/download (unless you have license for the Ultimate edition that provides better support for Java EE development);
- NetBeans with Java EE support can be downloaded from https://netbeans.org/downloads/

2.2 Setup Payara server

1. Download Payara server full (latest version) from http://www.payara.fish/downloads
2. Extract the Payara server zip to a proper directory (e.g. D:\Software)
3. Run the Java DB RDBMS that comes with Payara by executing the D:\Software\payara-4.1.1.161.1\payara41\javadb\bin\startNetw script. Note that Java DB (also known as Apache Derby) is also shipped with Glassfish distributions. By default the Java DB server accepts connections on port 1527.
4. Start Payara server using the asadmin script as follows:
   ---- D:\Software\payara-4.1.1.161.1\payara41\bin\asadmin start-domain ----
5. Verify that server is started by logging in the admin interface at http://localhost:4848/ (Payara uses a modified version of the Glassfish admin interface) If the admin interface does not load the check the Payara logs at D:\software\payara-4.1.1.161.1\payara41\glassfish\domains\domain1\logs errors during initialization.

Chapter 3

Guide (60 mins)

3.1 Setup demo project: Guestbook

1. Clone the Guestbook project to a proper location:
   ---- git clone https://github.com/bgjug/jcache-workshop.git guestbook ----
2. Import the project as a Maven project in your IDE of choice (e.g. for Eclipse: File → Import → Existing Maven Projects)
3. Add Payara (Glassfish 4) deployment support for in our IDE. For Eclipse add a new server from the Servers view in Eclipse (Window → Show View → Servers) by right clicking in the view and selecting **Glassfish 4** from the **Glassfish** category. As a name specify **Payara**, as a location specify **D:/software/payara-4.1.1.161.1/payara41/glassfish** and as a Java Development Kit specify JDK 8 (must be installed on your system and added as a Runtime environment from Window → Preferences in Eclipse). After you finish creating the server right click on it from the server view and select *Add or Remove*, select the *guestbook* project for deployment and click **Finish**.
4. Verify that when the project is synchronized with the server you are able to launch it by navigating to the **Applications** tab in the Payara admin panel and clicking the *Launch* action next to the deployed **guestbook** application.

3.2 Enable JCache

- Cache::put
- Cache::get
- Cache::remove

3.3 Utilize additional JCache APIs

- EntryProcessor
- CacheEntryListeners
- ExpiryPolicy
- CacheWriter / CacheReader?/CacheLoader

3.4 Refactor project to use CDI

3.5 Summary

How many times faster is the application with JCache ?

Chapter 4

References

4.1 JCache overview

[1] JSR 107: JCache - Java Temporary Caching API: https://jcp.org/en/jsr/detail?id=107

[2] Introduction to JCache JSR 107: https://dzone.com/articles/introduction-jcache-jsr-107

[3] Sneak peek into the JCache API: https://www.javacodegeeks.com/2015/02/sneak-peek-jcache-api-jsr-107.html

[4] JCache, why and how ?: https://vaadin.com/blog/-/blogs/jcache-why-and-how-

[5] JCache is Final! I Repeat: JCache is Final!

[6] Java Caching: Strategies and the JCache API

[7] How to speed up your application using JCache: https://www.jfokus.se/jfokus16/preso/How-to-Speed-Up-Your-Application-using-JCache.pdf

[8] After 13 years, JCache specification is finally complete: http://sdtimes.com/13-years-jcache-specification-finally-complete/

4.2 JCache support

[9] Hazelcast blogs (JCache category): http://blog.hazelcast.com/category/jcache/

[10] Hazelcast JCache implementation: http://docs.hazelcast.org/docs/3.3/manual/html-single/hazelcast-documentation.html#hazelcast-jcache-implementation

[11] Hazelcast 3.5 Manual: Introduction to the JCache API: http://docs.hazelcast.org/docs/3.5/manual/html/jcache-api.html

[12] Infinispan JCache support: http://infinispan.org/docs/7.0.x/user_guide/user_guide.html#_using_infinispan_as_a_jsr10

[13] Infinispan JCache example: http://infinispan.org/tutorials/simple/jcache/

[14] Oracle Coherence JCache support: https://docs.oracle.com/middleware/1213/coherence/develop-applications/-jcache_intro.htm#COHDG5778

[15] Ehcache JCache support: https://github.com/ehcache/ehcache-jcache

[16] Apache Ignite JCache provider: https://ignite.apache.org/use-cases/caching/jcache-provider.html

[17] Google App Engine support for JCache: https://cloud.google.com/appengine/docs/java/memcache/usingjcache

[18] Couchbase JCache Implementation Developer Preview 2: http://blog.couchbase.com/jcache-dp2

[19] Couchbase JCache implementation: https://github.com/couchbaselabs/couchbase-java-cache

[20] JCache (Payara 4.1.153): https://github.com/payara/Payara/wiki/JCache-(Payara-4.1.153)

[21] Spring JCache annotations support: https://spring.io/blog/2014/04/14/cache-abstraction-jcache-jsr-107-annotations-support

4.3 JCache & CDI

[22] Using JCache with CDI: http://www.tomitribe.com/blog/2015/06/using-jcache-with-cdi/

[23] High Performace Java EE with JCache and CDI: http://www.slideshare.net/Payara1/high-performance-java-ee-with-jcache-and-cdi

[24] Using the JCache API with CDI on Payara server: http://blog.payara.fish/using-the-jcache-api-with-cdi-on-payara-server