iBrowser is not working on google platform when google platform upgrade the Java version to java 8.

Investigation: We have conducted meeting with onsite counterparts Sundardas Ramani , Ashish Chokhani and offshore java experts Vishwas Raiborde, Umakant Mohanty, Onsite handed over us the source code and explained in details about the issue and guided us with code walkthrough. After this point the investigation at offshore started. While working on the investigation we found following issues (potential cause of failure).

**Source Code compatibility issue**: Assuming the code was compile on minor version and deployed on java major version java 8 .

**Deprecated Api / Coding issue:** Assuming there has been some coding issues making the application fail at run time

**Internal Server Errors:** Assuming some internal startup conflict is resulting in failure of application once migrated on java 8 configuration.

**Project Build Issues :**  Current project scaffold did not have any build guidelines and has an older project structure .

Approach And Solution:

**Step 1 Modernize Project scaffold and build:** We at offshore successfully migrated the old project scaffold to newer maven based project structure and manage libraries dynamically. There are still some old libraries which were not available in m2 repositories were endorsed by system path.

These can be changed once we get handle on JL repo management to install these libraries as and artifact. After which the entire build can be automated and code base will be lighter.

**Step 2 DataNucleus Enhancement:** In existing code base since we did not have proper enhancement plugin, we have added a data enhancement maven plugin, which enhances the class files during the build process. Also, there is not handle on how to manage JDO classes. there should be a datanucleaus.properties to manage entity classes.

**Step 3 Manage Misconfigure Velocity Logging:** There is no control in the present code base to handle velocity logging and this interferes with the server logging and brings the application down. Velocity logging is now disabled and this configuration is overridden by endorsing the velocity.properties files in classes folder. These configurations should also be handled aptly as a technical debt cleanse.

**Step 4 Code Compliance:** Hard coding was detected in major logical section, such as permission of IP’s , we have not made any code change but will recommend to remove these technical Debts.

**public** **static** **final** List<IPRangeItem> ***IP\_RANGES*** = Arrays.*asList*(

**new** IPRangeItem("193.35.248.1", "193.35.255.254"), // JL IP ranges

**new** IPRangeItem("86.160.18.1", "86.160.18.254"),

**new** IPRangeItem("83.221.179.1", "83.221.179.254"),

**new** IPRangeItem("86.164.216.151", "86.164.216.151"),

**new** IPRangeItem("86.160.16.1", "86.160.16.254"),

**new** IPRangeItem("38.104.182.34", "38.104.182.34"),

**new** IPRangeItem("195.158.26.1", "195.158.26.254"),

**new** IPRangeItem("125.252.68.90", "125.252.68.94"),

**new** IPRangeItem("103.21.170.52", "103.21.170.52"),

**new** IPRangeItem("10.20.34.54", "10.20.34.54"),

**new** IPRangeItem("127.0.0.1", "127.0.0.1")

);

These are major code vulnerabilities and

**public** **class** CronServlet **extends** HttpServlet **implements** GlobalConstants {

**for**(GoogleGroup group: allGroups){

String groupName = group.getName();

**if**(groupName==**null** || "".equals(groupName.trim()) ||

(!groupName.toLowerCase().startsWith("\_JL Branch:".toLowerCase()) &&

!groupName.toLowerCase().startsWith("Branch:".toLowerCase()) &&

!groupName.toLowerCase().startsWith("iBr".toLowerCase()) &&

!groupName.toLowerCase().startsWith("\_JL iBr".toLowerCase()) &&

!groupName.toLowerCase().startsWith("iBrowser".toLowerCase())

)){

**Step 5 Code compilation on java 8:** Velocity Logging was fixed, Modernised code was compiled successfully, then Enhance and then deployed to GCP App Engine (local account)

**Step 7 Remove IP restrictions:** only for testing purpose. If tomorrow we choose to change our IP in network, we will need a code change, a classic example of hard coding. This could be one potential issue of failure as well. I will keep this code for the testing purpose and will remove it once testing is done.

**protected** **boolean** isValidIPRange(HttpServletRequest request) **throws** UnknownHostException {

String clientIPAdderss = request.getHeader("X-FORWARDED-FOR"); //Load balancer sets this header client's IP

logger.log(Level.INFO, "IP from header :"+ clientIPAdderss);

**if** (clientIPAdderss == **null**) {

clientIPAdderss = request.getRemoteAddr();

logger.log(Level.INFO, "IP from remote addr :"+ clientIPAdderss);

**if** ((clientIPAdderss != **null** && clientIPAdderss.contains("."))) {

//if (!IP\_RANGES.isEmpty()) {

**for** (IPRangeItem ipRange : IP\_RANGES) {

**long** ipLower = ipToLong(InetAddress.getByName(ipRange.getLowerRange()));

**long** ipHigher = ipToLong(InetAddress.getByName(ipRange.getHigherRange()));

**long** ipClient = ipToLong(InetAddress.getByName(clientIPAdderss));

**if** ((ipLower <= ipClient && ipClient <= ipHigher)) {

// validIP = true;

logger.log(Level.INFO,"TRUE");

**return** **true**;

}

}

//}

// validIP = true;

// return;

}

}

logger.log(Level.INFO,"FALSE");

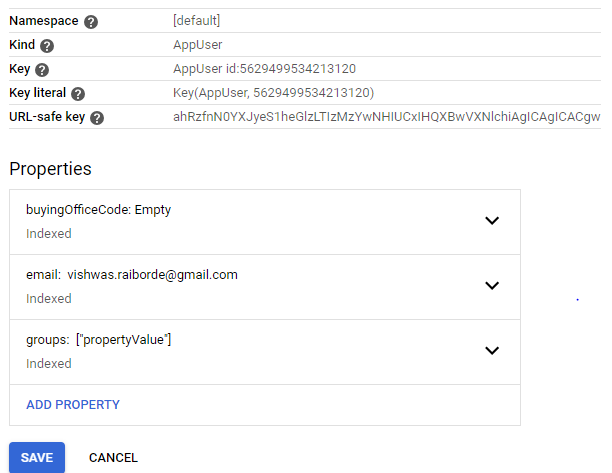
//**TODO** hack to work with locak vishwas return false;

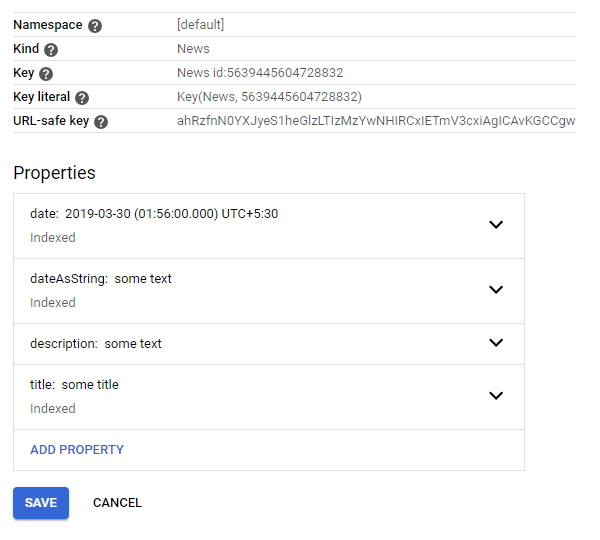
**return** **true**;

}

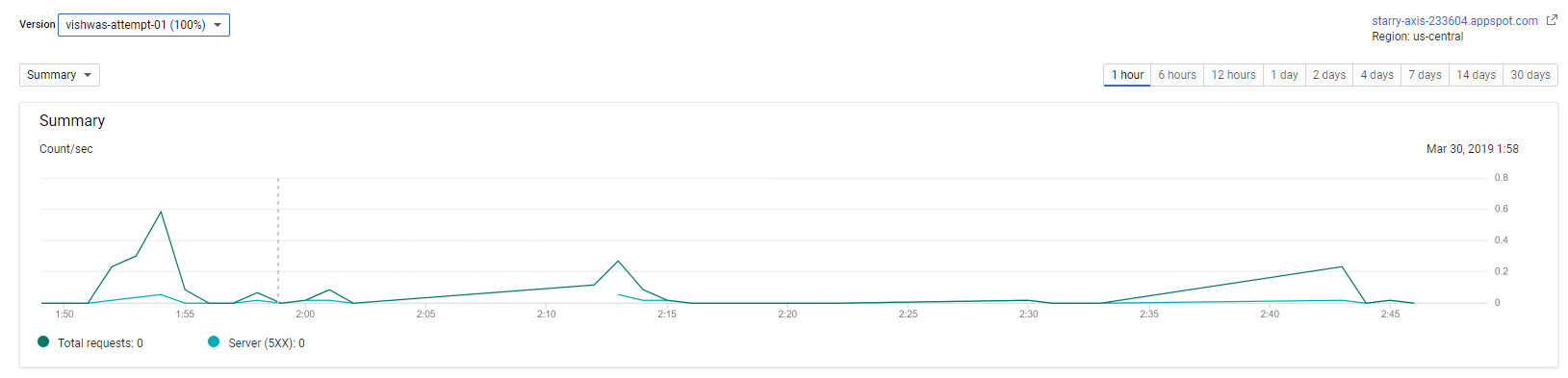
**Step 6 Dummy Entites created for testing:** If the application is successfully deployed the data from these entities should be displayed. This will ensure all the relevant layer of the application are working correctly, this was also needed As I did not have JL google environment and needed some kind of stubbing to test. This can be tested in JL test and confirmed in our next testing phase.

(We will need support in testing as offshore java team in not aware of all functional areas)

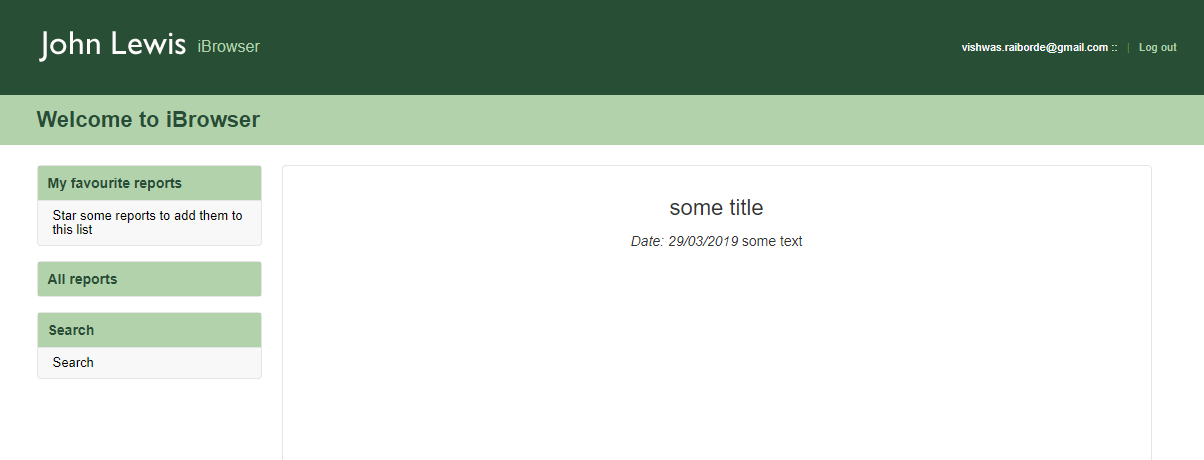




Result: Ibowser application is successfully deployed and working for the test data set up.

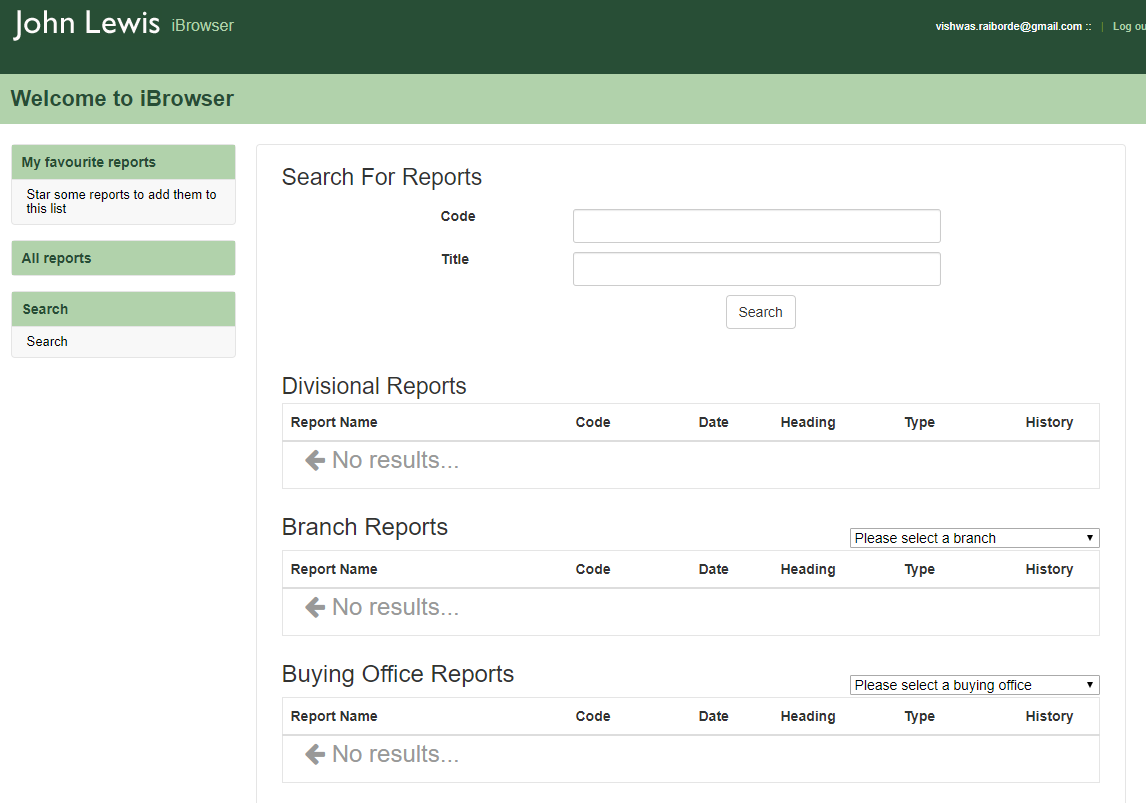


Data Set up In News Entity is displayed

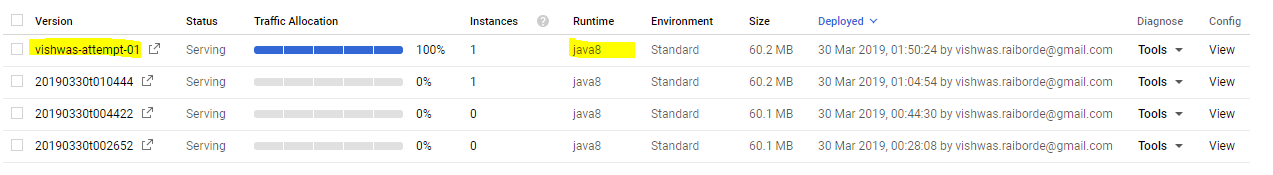


As the bucket and relevant Entities are not populated search Screen is empty,

Though the user [vishwas.raiborde@gmail.com](mailto:vishwas.raiborde@gmail.com) is populated from AppUser entity



Java 8 instance as below



Code changes involve:

* Removing unused imports
* Formatting done for better redability

**Code base with fix:** <https://github.com/VishwasRaiborde/ft-ibrowser-java-8-compatiable.git>

**Access URL:** <https://ibrowser-ptest-495.appspot.com/app/>