* Plumbing
  + Add retry logic, and timeouts
  + Plan out all the various states
  + Plan out return codes
  + Add “Names” for events for easy visualization?
  + Learn Cassandra partitioning
  + Security considerations
  + Application Status vs System status
* 3 versions:
  + Desktop (for prototyping)
  + Synch (http calls)
  + A synch (Kafka?)
* Add "registered" services:
  + Docker
  + Kubernetes
* Add Admin GUI
  + Logs/Traces
  + Replay logic
  + Analyze replay to see how to keep idempotent
* Add analytics
  + Does the engine have to assign UIDs, or can the local library?
* Client side libraries:
  + Java
  + Python
  + Node
  + Etc.
* Create best practices paper
  + On idempotent services
  + Process domains (APIs, Adapters, Process, etc.)
* How to tell if root is complete?

Project to do:

* Research the need
  + Research orchestration vs conductor

Eclipse steps for R2FileApp

* Create a new “Dynamic Web Project”
* Right click on project, Configure, Convert to Maven project
* Added dependencies to pom.xlm

<dependencies>

<dependency>

<groupId>com.datastax.cassandra</groupId>

<artifactId>cassandra-driver-core</artifactId>

<version>3.10.1</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.8.6</version>

</dependency>

<dependency>

<groupId>org.json</groupId>

<artifactId>json</artifactId>

<version>20090211</version>

</dependency>

</dependencies>

* Added plugin to pom.xlm

<plugin>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-maven-plugin</artifactId>

<version>9.4.30.v20200611</version>

</plugin>

* Right Click on src, new Servlet (I named it r2fileapp, first class R2FileAPI)
* Copied the RtoosLib.java into the r2fileapi package (for now simply copy the java for the client lib)
* Copied a web.xml into the WebContent/WEB-INF directory

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>RtoosEvent</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<description> </description>

<servlet-name>R2FileAPI</servlet-name>

<servlet-class>r2fileapp.R2FileAPI</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>R2FileAPI</servlet-name>

<url-pattern>/R2FileAPI.html</url-pattern>

</servlet-mapping>

</web-app>

* If the servers view is not showing on the bottom, go to menu Windows/Show View/Servers. You can drag this to bottom of eclipse if you wish
* Drag R2FileApp project down and drop on the J2EE Preview at localhost line in the servers tab
* To add servlets, right click on r2fileapp and new servlet, add the servlet to the web.xml

NOTE: after any code change you have to stop and start the server by right clicking on it, not sure why but restart does not work on my machine