# Tasks

* Add Jaeger server to the demo project. Update docker compose to run it.
* Add Jaeger Client <https://github.com/jaegertracing/jaeger-client-node> to
  + Frontend service (Tracing in Http)
  + Book service (Tracing in Http)
  + Author service (Tracing in Http)
* Add Redis Span to the Book and Author service to see this call in the Jaeger and perform call via Postman to the localhost: 8083/api/v1/details.
* Create screenshot for request tracing localhost:8083/api/v1/details from Jaeger UI.
* Add delay to the Book service route api/v1/books (any loop can be used like “while”) and perform call via Postman to the localhost: 8083/api/v1/details.
* Create screenshot for request tracing localhost: 8083/api/v1/details from Jaeger UI.
* Add exception throw to the Book service route api/v1/books and perform call via Postman to the localhost: 8083/api/v1/details.
* Create screenshot for request tracing localhost: 8083/api/v1/details from Jaeger UI.
* Create screenshot of dependency graph from Jaeger UI.

# Links

* Instrumenting Node.js for Tracing in Jaeger <https://logz.io/blog/jaeger-tracing-nodejs/>
* Jaeger Client for NodeJS <https://github.com/jaegertracing/jaeger-client-node>
* JavaScript Open tracing library <https://github.com/opentracing/opentracing-javascript>
* Jaeger configuration <https://www.jaegertracing.io/docs/1.8/deployment/>
* Open tracing protocol <https://opentracing.io/docs/overview/>
* NodeJS Open tracing tutorial <https://github.com/yurishkuro/opentracing-tutorial/tree/master/nodejs>
* Open tracing recommendations for NodeJS <https://opentracing.io/registry/nodejs-jaeger-tracer/>

# Result

* Create folder in project repository which is called screenshots and place four screenshots that were created to that folder and push these changes to the source control.
* Create pull request for the changes to the code of each. (one repo for all service). Updated docker-compose has to be commit as well to allow run it locally during review.
* Commit name System-observability-session-1