

Initial Observations (not set in stone, can be changed later):

The app will use the following layers:

- REST
- Client
- Cache
- Kafka
- Scheduled Job
- Service
- Exception (possibly sub-package)

Technology possibilities

- REST -> rest controller
- Client -> Spring Rest client
- Cache -> Caffeine
- Kafka -> Spring Kafka

Possible paths:

Call to rest endpoint -> retrieve or add to cache

Scheduled job every 10 seconds -> service is executed -> call to external API via client -> publish retrieved message to Kafka topic

Possible considerations:

Using a database and Redis for storing event data

Using concurrency with possibly Reactive programming or Java concurrency (Java Threads, CompletableFuture etc)

Testing:

- Integration test in app using SpringBootTest
- Unit test coverage > 95%
- Incorporate swagger to test manually

Notes:

Could use leadership for a scheduled task

Consider backup using AOF or snapshotting

Consider JPA repositories and/or a Separate database

Possibly use spring cache annotations (Cacheable)

Final Observations:

In the end, I decided to go with Redis repositories. It was easier to schedule the job after making the design change decision.

The following are some considerations on possible changes:

- Validation is done at the entry to the rest controller, but possibly having a validation component that tests any incoming request that it is valid, instead of just on rest controller entry. For example, the app also received message updates from a kafka topic and use the same validation for the incoming message, so the domain requirements are correct across the app.
- Use a String instead for the event status, and convert, would have a more detailed error message
- The rescontroller is returning a EventDataRequest, but would use a custom object if time allowed for errors and also add more swagger documentation

AI assistance:

- IntelliJ AI assistant and CoPilot - used to come up with suggestions when the first word was typed
- AI provided code snippets from previous development, which reduced the time to code considerably.
- An example would be a repeatable configuration used, such as Kafka configuration and components, such as RestControllers
- Next AI step is to be able to automate obvious unit tests