1. Keeping scanBasePackages = {"org.aggregation"} with @SpringBootApplication to read all spring components/beans , else AggregationServiceApplication file could have kept under org.aggregation, then no need to keep scanBasePackages.
2. Wrapping controller response with ResponseEntity, so that we have full control to setup status code and other data.
3. Interface based implementation, created AggregationServiceImpl class and AggregationService interface, to have loose coupling.
4. Using @Autowire to inject service in to controller.
5. Keeping base-url in application.properties file, so that any change in configuration values need not any new build.
6. restTemplate which is being called from backend client, needs to keep under try/catch to handle service unavailable 503 scenario.
7. Used CompletableFuture to asynchronously run all backend calls and retrieve the results when completes.
8. Used ExecutorService to execute backend calls on threads asynchronously. Also defined pool size using newFixedThreadPool.
9. Used CountDownLatch so that response will include result of all threads.
10. Used @Slf4j to add logs.