

마이크로서비스 요청 추적

마이크로서비스 아키텍처를 사용한 어플리케이션에서 예외 상황이 발생했을 시, 로그를 분석하여 어디에서 예기치 못한 동작이 발생했는지 디버깅을 해야한다. 이때 디버깅을 할 때 어느 컴포넌트에서 문제가 발생했는지, 각 컴포넌트 간 API 호출이 정상적으로 이루어졌는지 등 특히 실시간으로 많은 트래픽이 발생하는 환경에서 단순히 endpoint에서 입력시간을 공유하는 것만으로는 문제의 원인 파악이 힘들다.

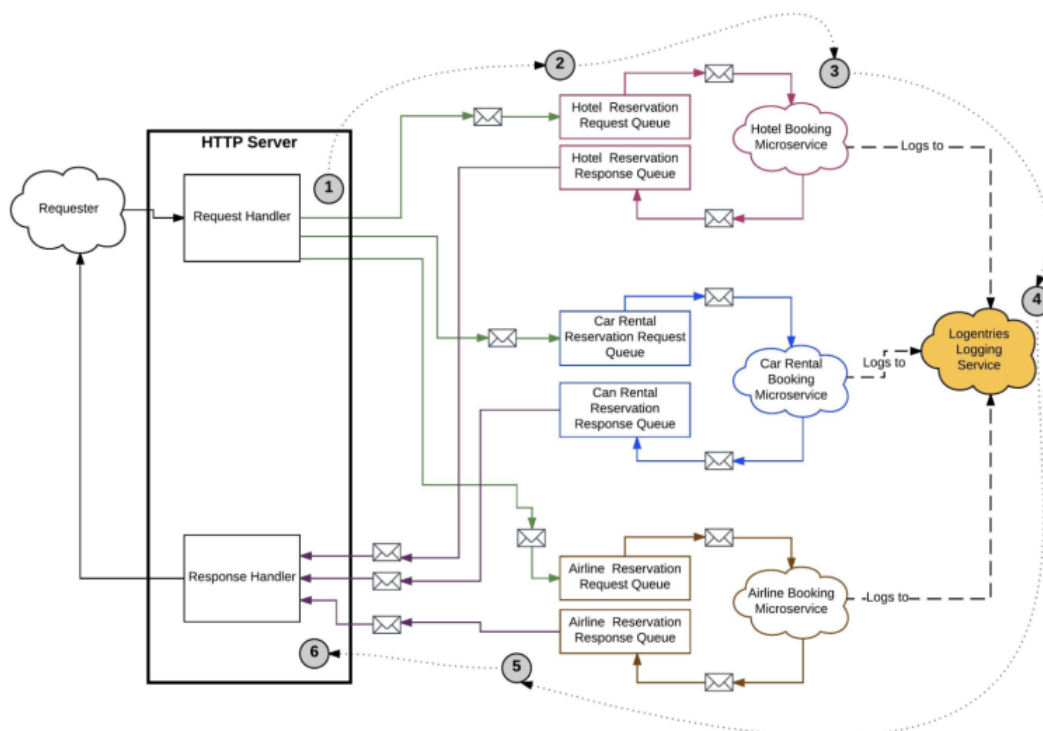
Correlation ID 를 사용하는 방법이 있다.

Correlation ID는 API 호출에 대한 식별자를 정의하고 컴포넌트 간에 식별자를 공유하는 것이다. 각 비즈니스 모델에 맞는 서비스 컴포넌트가 모여 Bounded Context라는 도메인 모델 경계를 이루며 동작하고 있다.

- 컴포넌트 간 API 호출 방식이 서비스 별로 다양
- 동시에 동일한 API 여러번 호출 가능
- 호출시 공유할 식별자를 로그에 기록 → 컴포넌트간 요청과 응답을 API 별로 구분이 가능해 시스템 전반에 걸쳐 일관되게 추적할 수 있음

Correlation ID

transit ID로도 알려져있다. 요청과 메시지에 첨부되는 유니크한 식별자이며 트랜잭션이나 이벤트 체인을 참조할 수 있도록 한다.



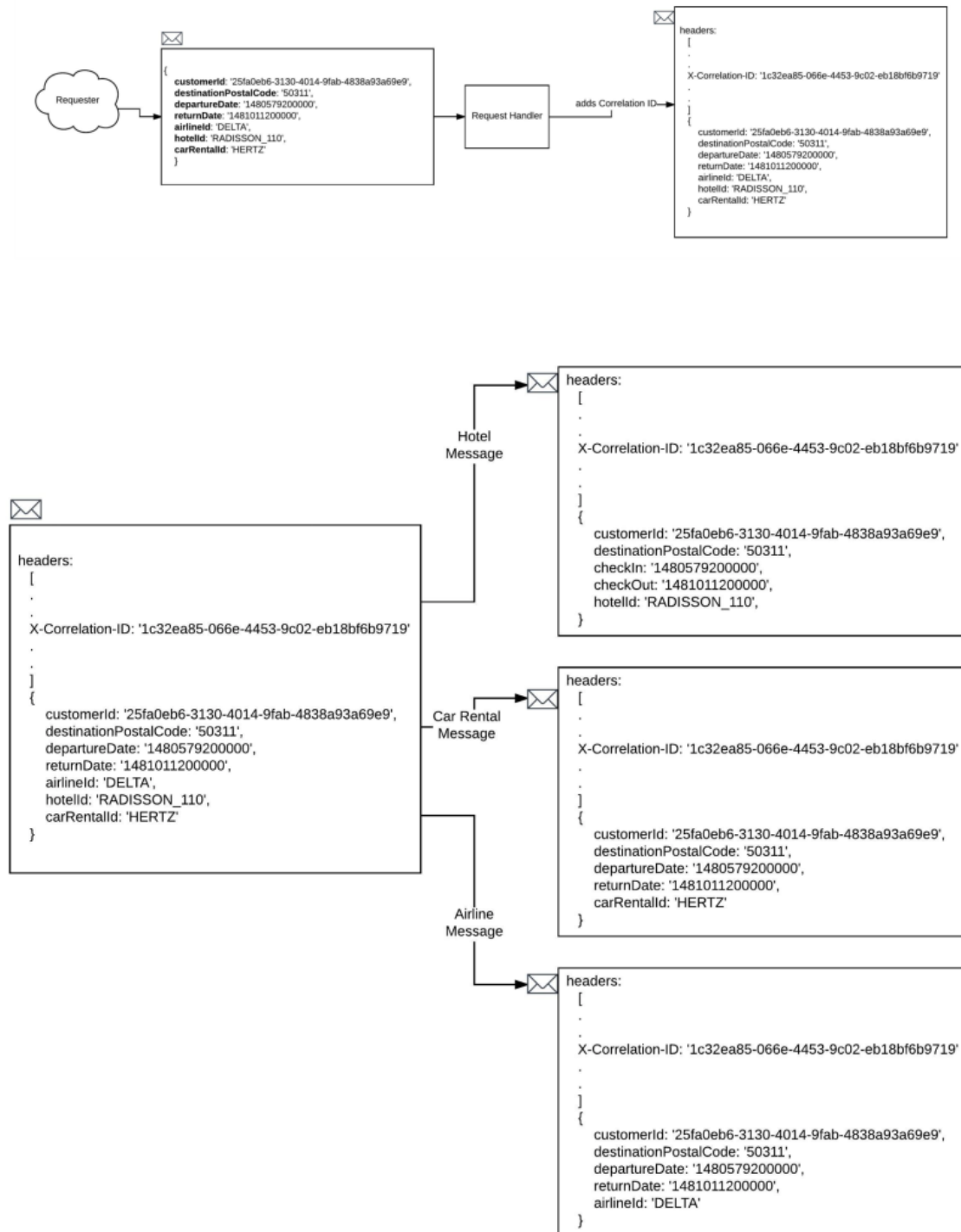
예시)

가상의 여행 예약 시스템으로 호텔, 차, 항공권을 예약할 수 있다.

논리흐름

1. 하나의 요청이 Request Handler에 보내지면 Request Handler는 요청에 있는 정보를 3개의 메시지로 변환한다.

2. 각각의 메시지는 호텔 예약 전용 메시지 큐, 자동차 예약 메시지큐, 항공 예약 메시지 큐로 보내진다.
3. 마이크로서비스는 큐로 입력되는 메시지를 감시하고 있다. 큐로부터 메시지를 가져와서 작업하고 작업 진행 과정들을 로깅 서비스에 정보들을 로깅한다.
4. 마이크로서비스가 작업을 완료하면 응답 큐로 메시지가 보내진다.
5. Response Handler는 보냈던 요청에 대한 3개의 모든 응답 메시지가 회수되면 응답 큐로부터 메시지를 가져온다.
6. Response Handler는 3개 예약 메시지에 있는 정보를 하나의 응답에 모은다.



https://docs.aws.amazon.com/ko_kr/step-functions/latest/dg/getting-started.html

AWS Step Functions