- Library Resources

Quiz 9: Event Driven Architecture Results for Yufu Liao

Correct answers will be available on Nov 19 at 12am.
Score for this quiz: 0 out of 100 *
* Some questions not yet graded
Submitted Nov 16 at 4:52pm
This attempt took 23 minutes.
Move To... This element is a more accessible alternative to drag & drop reordering. Press Enter or Space to move this question.
Question 1
Not yet graded / 20 pts

What are the four principles of event driven architecture?

Your Answer:

• Push architecture:
– Don’t wait for event consumers to pull
• Timeliness
– Client responds immediately
• Asynchronous
– Notification is “fire-and-forget”
• Command-free
– Notification is a report, not a request for specific
action

Move To... This element is a more accessible alternative to drag & drop reordering. Press Enter or Space to move this question.
Question 2
Not yet graded / 20 pts

1. What are the two principles of eventual consistency?
2. How does eventual consistency differ from strong consistency?

Your Answer:

1. What are the two principles of eventual consistency?

– Consistent ordering: Updates are done in same order on all replicas
– Total propagation: Updates are performed on all replicas eventually

2. How does eventual consistency differ from strong consistency?

The update of strong consistency can be visible by any operation, but eventual consistency have an order of replicas

Move To... This element is a more accessible alternative to drag & drop reordering. Press Enter or Space to move this question.
Question 3
Not yet graded / 20 pts

What are the benefits of CQRS?

Your Answer:

• Horizontal scaling
– Different for command and query services
• Optimize state representations
– Appropriate to the service
• Read-side failover availability
– Handle queries with event log down
• Scalability of event-sourced systems

Move To... This element is a more accessible alternative to drag & drop reordering. Press Enter or Space to move this question.

Question 4
Not yet graded / 20 pts

Explain (in one sentence) how Kafka differs from Message-Oriented Middleware in each of the following respects:

1. Message retention
2. Per client message filtering
3. Message replay

Your Answer:

1. Message retention

Traditional Messaging is Reference count based message retention, Kafka Messaging is Time/size based message retention

2. Per client message filtering

Kafka Messaging is smart at client, Traditional Messaging is smart at broker

3.Message replay

Traditional Messaging is Per consumer filtering, kafka is Message stream replay

Move To... This element is a more accessible alternative to drag & drop reordering. Press Enter or Space to move this question.

Question 5
Not yet graded / 20 pts

The following describes an event producer that generates events during the execution of a game, and an event consumer that reacts to events that involve points being scored. The annotation is to distinguish the kinds of point-scoring events the consumer is interested in. Complete this code with the CDI annotations it requires.

```
@Qualifier
@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.METHOD, ElementType.FIELD, ElementType.PARAMETER})
public @interface ScoreEvent {}

@RequestScoped
public class EventProducer {
   private Event<GameEventDto> gameEvent;
   public void notifyPointsScored(GameEventDto gameEventDto) {
      gameEvent.fireAsync(gameEventDto);
   }
}

@RequestScoped
public class ScoreObserver {
```

```
    public void onEvent(Event<GameEventDto> gameEvent) {
       GameEventDto gameEventDto = gameEvent.getEventMessage();
    }
}
```

Your Answer:

```
@Qualifier
@Retention(RetentionPolicy.RUNTIME)
@Target({ElementType.METHOD, ElementType.FIELD, ElementType.PARAMETER})
public @interface ScoreEvent {}

@RequestScoped
public class EventProducer {
   @Inject @GameEvent
   private Event<GameEventDto> gameEvent;
   public void notifyPointsScored(GameEventDto gameEventDto) {
      gameEvent.select(new Point-scoringEventQualifier()).fireAsync(gameEventDto);
   }
}

@RequestScoped
public class ScoreObserver {
   public void onEvent(@Observes Event<GameEventDto> gameEvent) {
      GameEventDto gameEventDto = gameEvent.getEventMessage();
   }
}
```

Quiz Score: 0 out of 100
* Some questions not yet graded

Quiz Submissions

- *This score is pending review, and may change* Attempt 1: 0

Yufu Liao has no attempts left

Back to Quiz