



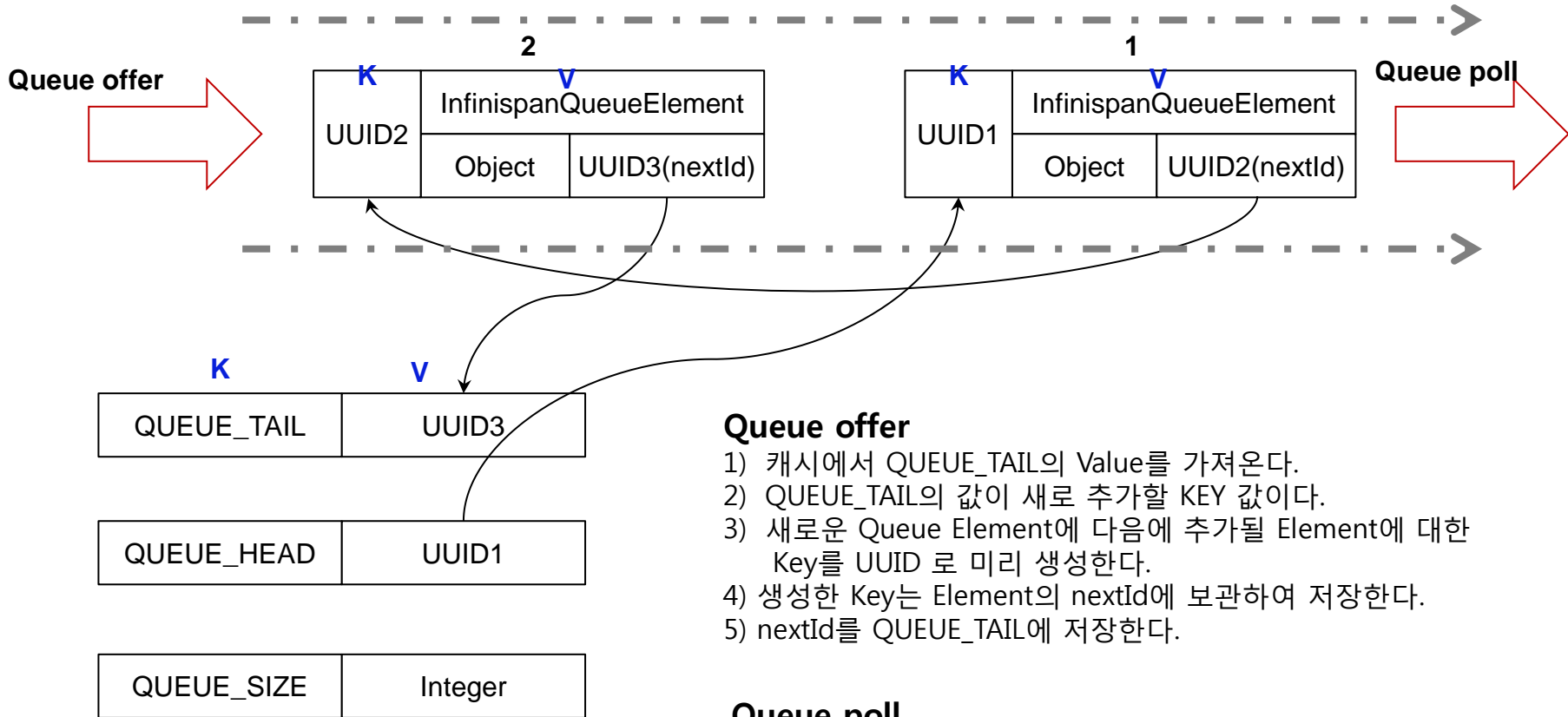
opennaru

# Infinispan Queue 구현 방법



# Queue 구현 방안

- 데이터 그리드는 Hashmap 구조로 현재 Queue 자료구조를 지원하지 않음
- Key / Value만으로 Queue 자료구조를 구현하기 위한 방안이 필요함
- **구현 방안**
  - Linked List 구조를 만들기 위해 Value 객체에 연결된 다음 객체의 Key를 보관함
  - Queue의 Head와 Tail에 대한 포인터 Key를 별도 캐시에 보관
  - 맨 마지막 Element는 앞으로 추가할 Key를 미리 생성하여 QUEUE\_TAIL과 Element의 Next Key에 보관함.
  - Queue에서 Element를 꺼내고 캐시에서 제거한 후, Element의 Next Key를 별도 키 (QUEUE\_HEAD)에 보관함.
  - 모든 Key는 UUID(Universally Unique Identifier)로 랜덤 키로 생성함.



## Queue offer

- 1) 캐시에서 QUEUE\_TAIL의 Value를 가져온다.
- 2) QUEUE\_TAIL의 값이 새로 추가할 KEY 값이다.
- 3) 새로운 Queue Element에 다음에 추가될 Element에 대한 Key를 UUID 로 미리 생성한다.
- 4) 생성한 Key는 Element의 nextId에 보관하여 저장한다.
- 5) nextId를 QUEUE\_TAIL에 저장한다.

## Queue poll

- 1) 캐시에서 QUEUE\_HEAD의 Value를 가져온다.
- 2) QUEUE\_HEAD의 값 UUID가 꺼낼 Queue의 KEY 값이다.
- 3) GET한 Element의 nextID를 QUEUE\_HEAD에 보관한다.

<https://github.com/nameislocus/infinispan-queue>

A close-up photograph of several lavender flower spikes. The central spike is in sharp focus, showing the dark purple buds and the emerging light purple flowers. The background is a soft, out-of-focus green, suggesting foliage.

Opennaru, Inc.  
<http://www.opennaru.com/>