Redis

16 Spring Boot Cache
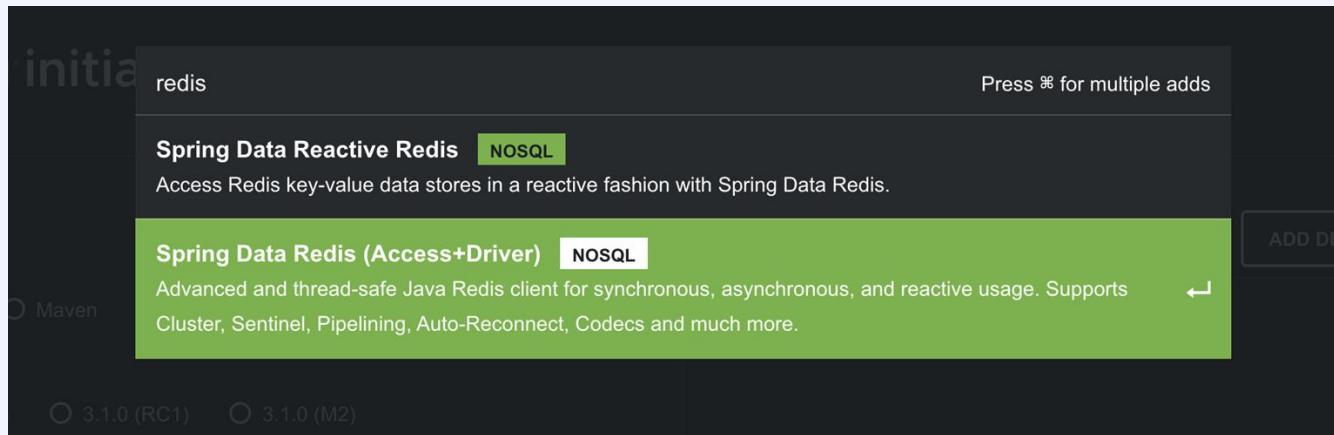
Spring Boot Cache   **Spring Data Redis**

2.

Redis

Spring Boot Cache

Spring Data Redis

Spring Data

https://start.spring.io/

Spring Data <u>JPA</u>
Spring Data <u>Redis</u>
Spring Data <u>MongoDB</u>

.

.

https://start.spring.io/

Spring Boot Cache   **Spring Data Redis**

```xml
<dependencies>

    <!-- other dependency elements omitted -->

    <dependency>
      <groupId>org.springframework.data</groupId>
      <artifactId>spring-data-redis</artifactId>
      <version>3.0.5</version>
    </dependency>

</dependencies>
```

gradle.build

implementation 'org.springframework.boot:spring-boot-starter-data-redis'

Redis Clients
1. **Lettuce**
2. Jedis

RedisTemplate

abstraction
connection
serializer

RedisTemplate

Lettuce

Jedis
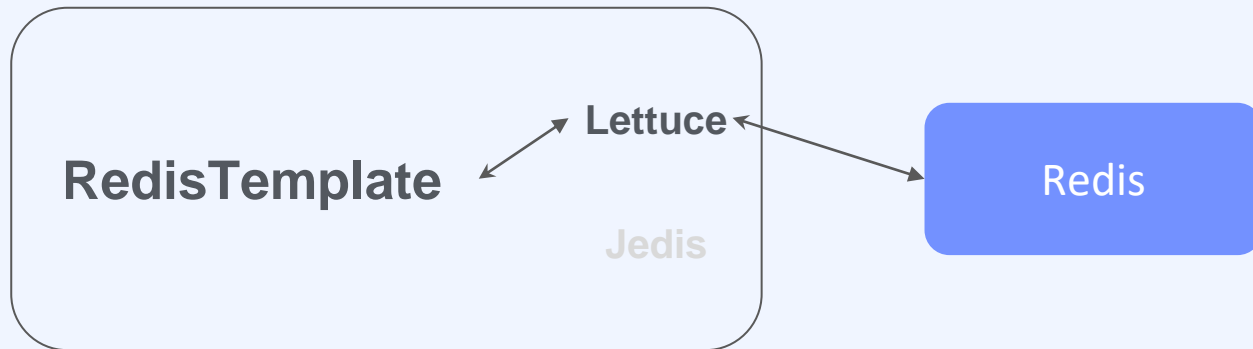
Redis

Spring Boot Cache   **Spring Data Redis**

Application.yaml

```
spring:
  data:
    redis:
      host: 127.0.0.1
      port: 6379
```

Spring Boot Cache   **Spring Data Redis**

```java
@Bean
public StringRedisTemplate stringRedisTemplate(RedisConnectionFactory redisConnectionFactory) {
    StringRedisTemplate template = new StringRedisTemplate();
    template.setConnectionFactory(redisConnectionFactory);
    return template;
}
```

Spring Boot Cache　　　　**Spring Data Redis**

2.

Redis

```java
public String getUserEmail(Long id) {
    var email : String = redisTemplate.opsForValue().get("users:%s:email".formatted(id));
    if (email != null) {
        return email;
    }
    email = userRepository.findById(id).orElse(User.builder().build()).getEmail();
    redisTemplate.opsForValue().set("users:%s:email".formatted(id), email, Duration.ofSeconds(30));
    return email;
}
```

Spring Boot Cache

Spring Data Redis

```java
@Bean
public LettuceConnectionFactory redisConnectionFactory(RedisProperties redisProperties) {
    return new LettuceConnectionFactory(new RedisStandaloneConfiguration(
            redisProperties.getHost(), redisProperties.getPort()
    ));
}
```

Spring Boot Cache   **Spring Data Redis**

Redis Repository

Spring Boot Cache   **Spring Data Redis**

```
@RedisHash("people")
public class Person {

    @Id String id;
    String firstname;
    String lastname;
    Address address;
}
```

Hash data type <-> Object

Spring Data Redis

```java
@RedisHash("people")
public class Person {

    @Id String id;
    String firstname;
    String lastname;
    Address address;
}
```

HMSET people:{id} id xxx firstname yyy …
SADD people {id}

```java
@RedisHash("people")
public class Person {

    @Id String id;
    @Indexed String firstname;
    String lastname;
    Address address;
}
```

SADD people:firstname:{firstname} {id}
SADD people:{id}:idx {firstname}

```java
public interface PersonRepository extends CrudRepository<Person, String> {

}
```

Spring Boot Cache
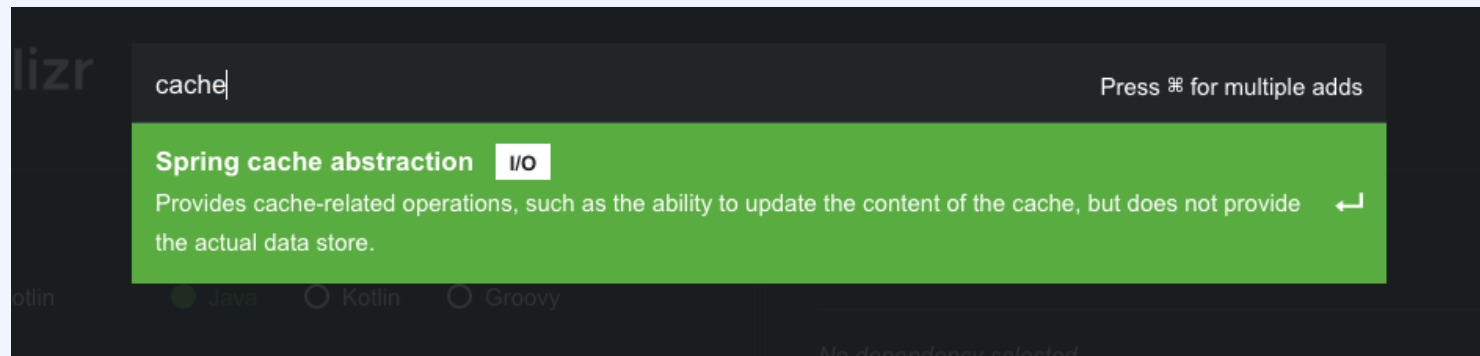
실습

Spring Boot Cache   **Spring Cache**

Spring Cache

Spring Boot Cache   **Spring Cache**

Spring Boot Cache   **Spring Cache**

spring-boot-starter-artemis	Starter for JMS messaging using Apache Artemis
spring-boot-starter-batch	Starter for using Spring Batch
spring-boot-starter-cache	Starter for using Spring Framework's caching support
spring-boot-starter-data-cassandra	Starter for using Cassandra distributed database and Spring Data Cassandra
spring-boot-starter-data-cassandra-reactive	Starter for using Cassandra distributed database and Spring Data

gradle.build

implementation 'org.springframework.boot:spring-boot-starter-cache'

https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using.build-systems.starters

Spring Boot Cache

Spring cache

9. Data
10. Messaging
11. IO
  11.1. Caching
    11.1.1. Supported Cache Providers
      Generic
      JCache (JSR-107)
      Hazelcast
      Infinispan
      Couchbase
      Redis
      Caffeine
      Cache2k
      Simple
      None
  11.2. Hazelcast
  11.3. Quartz Scheduler

Spring Boot

```java
@EnableCaching
@EnableJpaAuditing
@SpringBootApplication
@RequiredArgsConstructor
public class Clip15Application implements ApplicationRunner
    private final UserRepository userRepository;
    private final UserEmailRepository userEmailRepository;
```

```java
    @Cacheable("users")
    public User getUser(Long id) {
        return userRepository.findById(id).orElseThrow();
    }
```

Spring Boot

```
@CacheEvict(cacheNames="cache1", key = "'users:' + #id")
public void evictUser(Long id) {
}
```

Spring Boot Cache

실습