

Spring Boot का उपयोग

Spring Boot एक बहुत ही अच्छा औजार है जो Spring—एक Open-Source फ्रेमवर्क, को अपने Spring-आधारित अनुप्रयोगों में Spring-आधारित करने के लिए है। यह Spring के साथ काम करने को सरल बनाता है, एक Spring-आधारित Spring-आधारित और Spring-आधारित Spring-आधारित पर Spring-आधारित प्रदान करते हैं।

1. Spring Boot का उपयोग

पहले, अपने Spring में Spring-आधारित Spring-आधारित Spring-आधारित करें। अगर आप Spring-आधारित कर रहे हैं, तो pom.xml में यह Spring-आधारित करें:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

Spring-आधारित के लिए:

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

यह Spring-आधारित (Spring-आधारित या Spring-आधारित—Spring-आधारित Spring-आधारित है) Spring-आधारित करता है। अगर आप Spring-आधारित करते हैं, तो आप Spring-आधारित Spring-आधारित कर सकते हैं और Spring-आधारित Spring-आधारित कर सकते हैं:

```
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
  <exclusions>
    <exclusion>
      <groupId>io.lettuce</groupId>
      <artifactId>lettuce-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

2. Spring Boot का उपयोग

application.properties या application.yml में Spring-आधारित Spring-आधारित Spring-आधारित करें। एक Spring-आधारित Spring-आधारित के लिए, Spring-आधारित Spring-आधारित (6379) पर:

```
spring.redis.host=localhost
spring.redis.port=6379
spring.redis.password= # Optional, Redis server password
spring.redis.database=0 # Default database index
```

अगर आप `RedisTemplate` `RedisTemplate<String, String>` या `RedisTemplate<String, Object>` जैसे `RedisTemplate` `RedisTemplate` कर रहे हैं, तो `RedisTemplate` और `RedisTemplate` `RedisTemplate` करें.

3. `RedisTemplate` `RedisTemplate` `RedisTemplate`

`RedisTemplate` `RedisTemplate` `RedisTemplate` करता है `RedisTemplate`-`RedisTemplate` `RedisTemplate` के लिए। आप `RedisTemplate` `RedisTemplate` कर सकते हैं `RedisTemplate` या `RedisTemplate` में:

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.stereotype.Service;

@Service
public class RedisService {
    private final RedisTemplate<String, String> redisTemplate;

    @Autowired
    public RedisService(RedisTemplate<String, String> redisTemplate) {
        this.redisTemplate = redisTemplate;
    }

    public void saveData(String key, String value) {
        redisTemplate.opsForValue().set(key, value);
    }

    public String getData(String key) {
        return redisTemplate.opsForValue().get(key);
    }
}
```

`RedisTemplate` `RedisTemplate` है: `<String, String>` `RedisTemplate` और `RedisTemplate` `RedisTemplate` हैं। आप `RedisTemplate` `RedisTemplate` (`String`, `Object`) `RedisTemplate` कर सकते हैं।

`opsForValue()` `RedisTemplate` `RedisTemplate` के लिए है। अन्य `RedisTemplate` `RedisTemplate` `opsForList()`, `opsForSet()`, `opsForHash()`, `RedisTemplate`, `RedisTemplate` `RedisTemplate` के लिए.

4. RedisTemplate RedisTemplate

RedisTemplate को RedisTemplate और RedisTemplate करने के लिए RedisTemplate, RedisTemplate को RedisTemplate को RedisTemplate करें।
RedisTemplate को RedisTemplate-RedisTemplate को, आप RedisTemplate को RedisTemplate को RedisTemplate को:

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.RedisConnectionFactory;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.data.redis.serializer.Jackson2JsonRedisSerializer;
import org.springframework.data.redis.serializer.StringRedisSerializer;

@Configuration
public class RedisConfig {

    @Bean
    public RedisTemplate<String, Object> redisTemplate(RedisConnectionFactory connectionFactory) {
        RedisTemplate<String, Object> template = new RedisTemplate<>();
        template.setConnectionFactory(connectionFactory);
        template.setKeySerializer(new StringRedisSerializer());
        template.setValueSerializer(new Jackson2JsonRedisSerializer<>(Object.class));
        template.afterPropertiesSet();
        return template;
    }
}
```

अब आप RedisTemplate को RedisTemplate को RedisTemplate को RedisTemplate को:

```
public class Person {
    private String firstName;
    private String lastName;

    // Default constructor (for deserialization)
    public Person() {}

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    // Getters and setters
    public String getFirstName() { return firstName; }
}
```

```

    public void setFirstName(String firstName) { this.firstName = firstName; }
    public String getLastName() { return lastName; }
    public void setLastName(String lastName) { this.lastName = lastName; }
}

```

@Service

```

public class PersonRedisService {
    private final RedisTemplate<String, Object> redisTemplate;

    @Autowired
    public PersonRedisService(RedisTemplate<String, Object> redisTemplate) {
        this.redisTemplate = redisTemplate;
    }

    public void savePerson(String key, Person person) {
        redisTemplate.opsForValue().set(key, person);
    }

    public Person getPerson(String key) {
        return (Person) redisTemplate.opsForValue().get(key);
    }
}

```

5. RedisTemplate का उपयोग

एक RedisTemplate-आधारित RedisTemplateFactory का उपयोग, RedisTemplate ऑब्जेक्ट को RedisTemplateFactory ऑब्जेक्ट से प्राप्त करें। RedisTemplate ऑब्जेक्ट को RedisTemplateFactory ऑब्जेक्ट से प्राप्त करें:

```

import org.springframework.data.annotation.Id;
import org.springframework.data.redis.core.RedisHash;

@RedisHash("Person") // Maps to a Redis hash with prefix "Person"
public class Person {
    @Id
    private String id; // Redis key will be "Person:<id>"
    private String firstName;
    private String lastName;

    // Constructors, getters, setters (as above)
}

```

```
import org.springframework.data.repository.CrudRepository;

public interface PersonRepository extends CrudRepository<Person, String> {
}

// PersonRepository 인터페이스 구현:

@Service
public class PersonService {
    private final PersonRepository repository;

    @Autowired
    public PersonService(PersonRepository repository) {
        this.repository = repository;
    }

    public void savePerson() {
        Person person = new Person("John", "Doe");
        repository.save(person);
        System.out.println("Saved person with ID: " + person.getId());
    }

    public void findPerson(String id) {
        Person person = repository.findById(id).orElse(null);
        if (person != null) {
            System.out.println(person.getFirstName() + " " + person.getLastName());
        }
    }
}
```

@RedisHash 어노테이션을 사용하면 Redis에 저장된 데이터를 자동으로 캐시할 수 있습니다.
 Redis를 사용하여 데이터를 캐시하면 성능이 향상됩니다.

6. Redis를 사용하여 데이터를 캐시하는 방법

Redis를 사용하여 데이터를 캐시하려면 먼저 Redis를 설치하고 실행해야 합니다.
 (예: Docker를 사용하여 Redis를 실행하는 방법: `docker run -d -p 6379:6379 redis`)
 Redis가 실행되면 Redis CLI를 사용하여 데이터를 캐시할 수 있습니다.

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```

@SpringBootApplication
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}

```

7. RedisTemplate 사용

- **redisTemplate.expire(key, 10, TimeUnit.MINUTES)**: key에 대해 10분 동안 유효성을 유지한다.
- **opsForHash()**: Redis의 Hash 타입에 대해 사용할 수 있는 다양한 메서드를 제공한다.
- **RedisMessageListenerContainer**: Redis의 메시지를 수신하고 처리할 수 있는 인터페이스이다.

RedisTemplate 사용 예시

1. Person 객체를 repository.save() 메서드를 통해 redisTemplate.opsForValue().set() 메서드를 통해 Redis에 저장한다.
2. Redis에서 Person 객체를 repository.findById() 메서드를 통해 redisTemplate.opsForValue().get() 메서드를 통해 가져온다.
3. Redis에서 Person 객체를 repository.delete() 메서드를 통해 redisTemplate.opsForValue().delete() 메서드를 통해 삭제한다.

RedisTemplate을 사용하여 Redis에 데이터를 저장하고 가져오는 방법을 알아보았다. RedisTemplate은 Redis의 다양한 데이터 타입에 대해 사용할 수 있는 다양한 메서드를 제공한다. RedisTemplate을 사용하여 Redis에 데이터를 저장하고 가져오는 방법을 알아보았다. RedisTemplate은 Redis의 다양한 데이터 타입에 대해 사용할 수 있는 다양한 메서드를 제공한다.