The Spring Data Redis (SDR) framework makes it easy to write Spring applications that use the Redis as a store for Java objects (POJOs) by eliminating the redundant tasks and boilerplate code required for interacting with the store through Spring’s excellent infrastructure support.

This tutorial is **an introduction to Spring Data Redis**, which provides the abstractions of the Spring Data platform to [Redis](http://redis.io/) — the popular in-memory data structure store.

Redis is driven by a keystore-based data structure to persist data and can be used as a database, cache, message broker, etc.

We'll be able to use the common patterns of Spring Data (templates, etc.) while also having the traditional simplicity of all Spring Data projects.

Redis as database :

Download radis :

<https://github.com/microsoftarchive/redis/releases/tag/win-3.2.100>

pom.xml :

<dependency>

<groupId>redis.clients</groupId>

<artifactId>jedis</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-redis</artifactId>

</dependency>



Redis as cache :

@CachePut(key = "#id",value = "Product")

@Cacheable(key = "#id",value = "Product",unless = "#result.price < 1000")

@CacheEvict(key = "#id",value = "Product")

