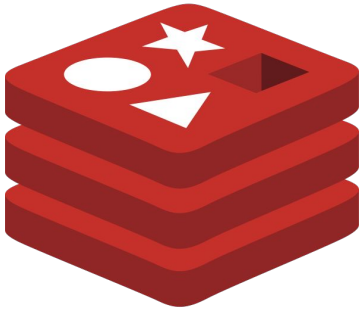


Redis

Diego Pacheco

About me...

- ❑ Cat's Father
- ❑ Principal Software Architect
- ❑ Agile Coach
- ❑ SOA/Microservices Expert
- ❑ DevOps Practitioner
- ❑ Speaker
- ❑ Author

diegopacheco

@diego_pacheco

http://diego-pacheco.blogspot.com.br/

https://diegopacheco.github.io/

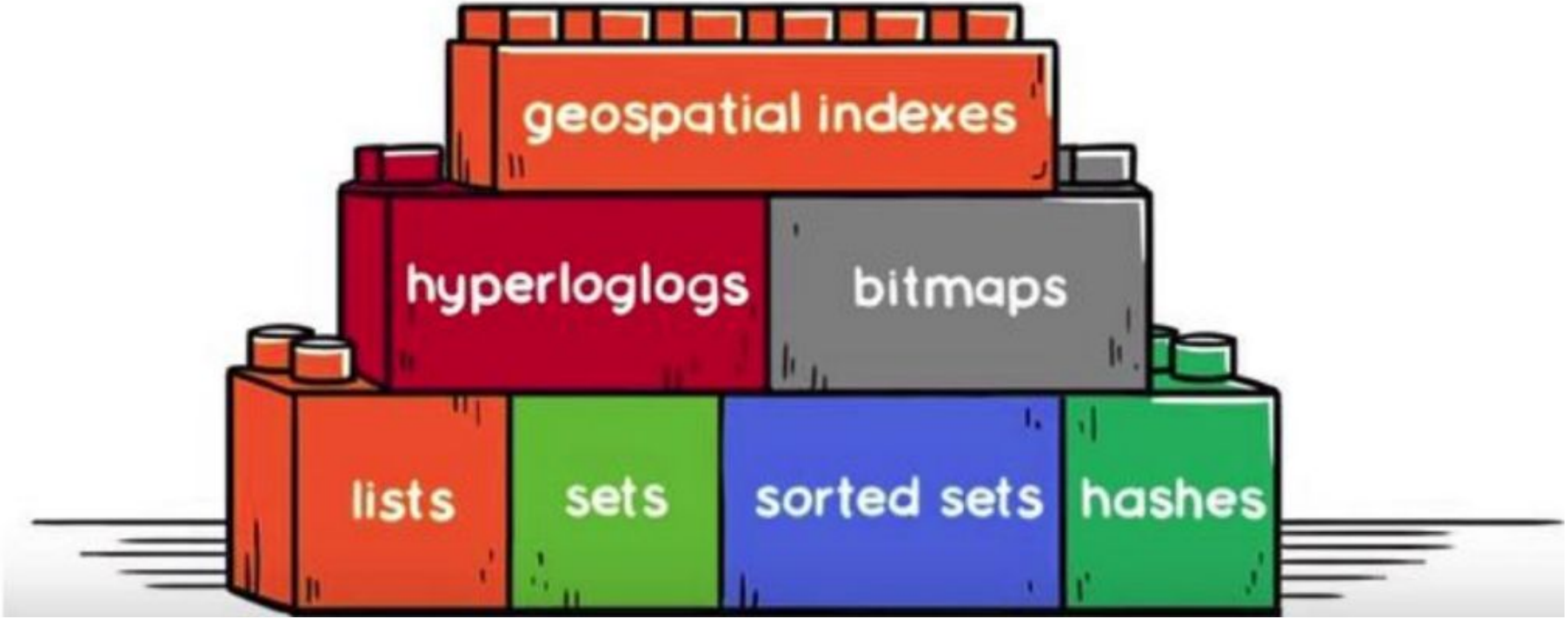
Redis

- FOSS
- In Memory K/V Store written in C
- Create for: Caching, Session Store, Queue, Analytics
- Specific Commands per Data Structures: Strings, Hash, sets
- Keys TTL
- Bring your own Data Structure
- Fast, Low Latency, Battle tested everybody. :D
- Single Thread :(

Redis data strata

v1.0	Strings	v2.2	Bit arrays
	Lists	v2.8.9	HyperLogLog
	Sets	v3.2	Geo Sets
v1.2	Sorted Sets		Bit fields
v2.0	Hashes	v4	Streams (?)
		V5	**MODULES!**

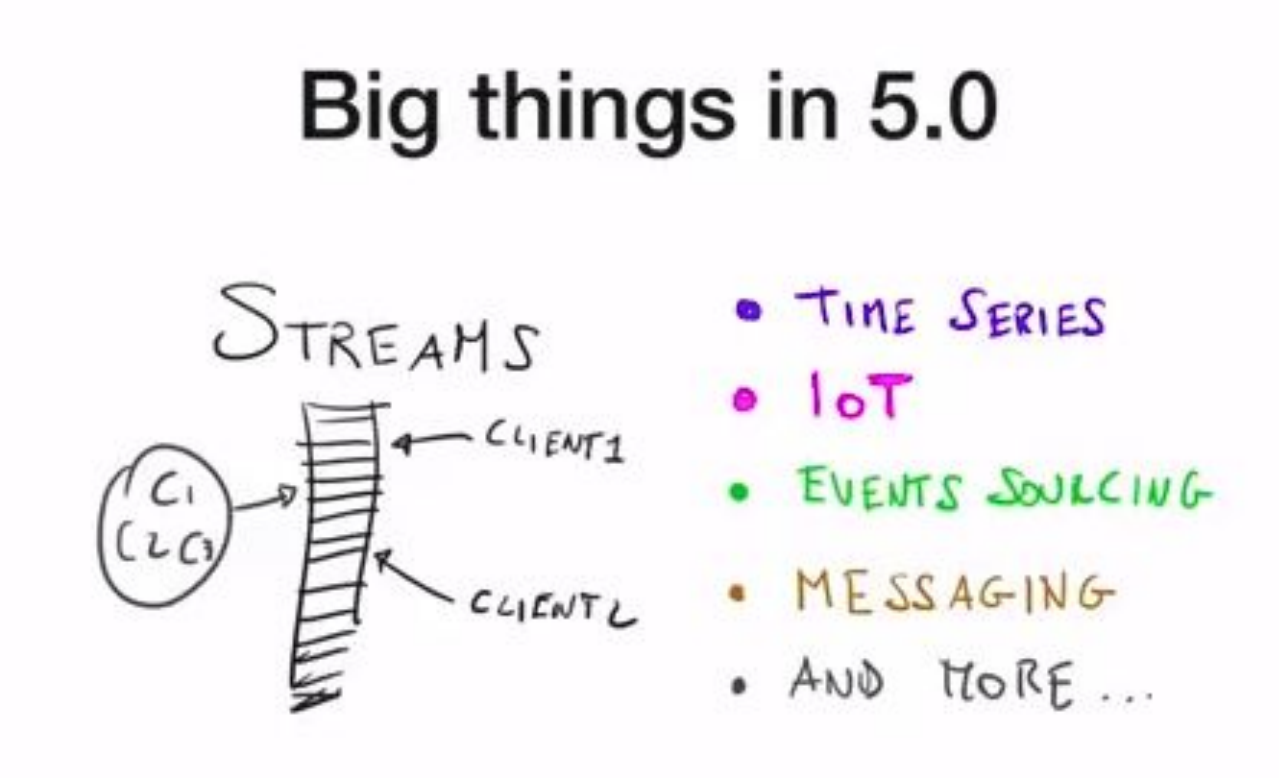
Redis - Data Structures



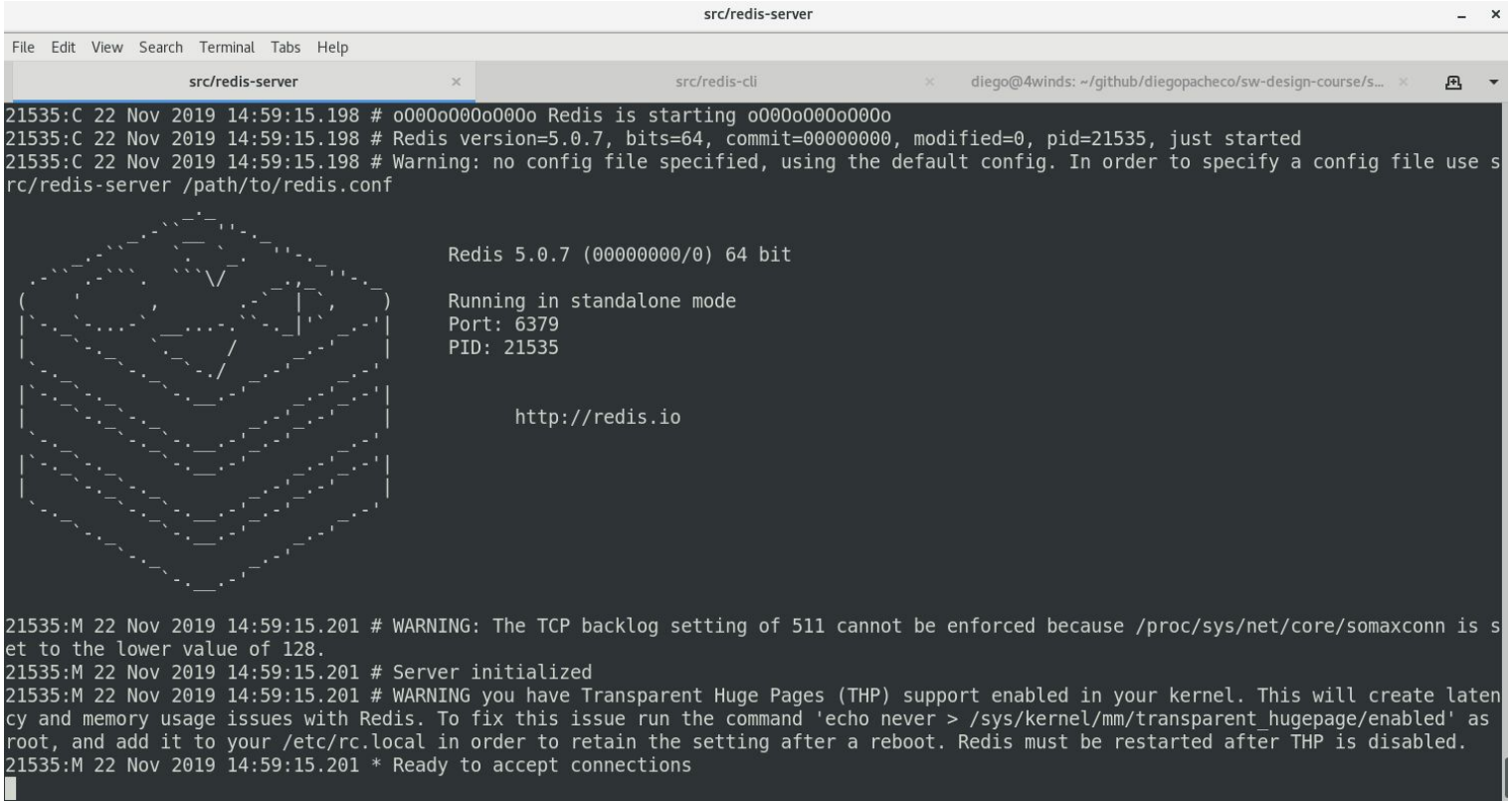
Redis >= 4.x

- Redis-Modules
  - It's not Lua Scripting
  - Built in C
  - Low Latency / Embedded in Redis
  - Use cases: Extends Redis / New Capabilities DSs

Redis >= 5.x

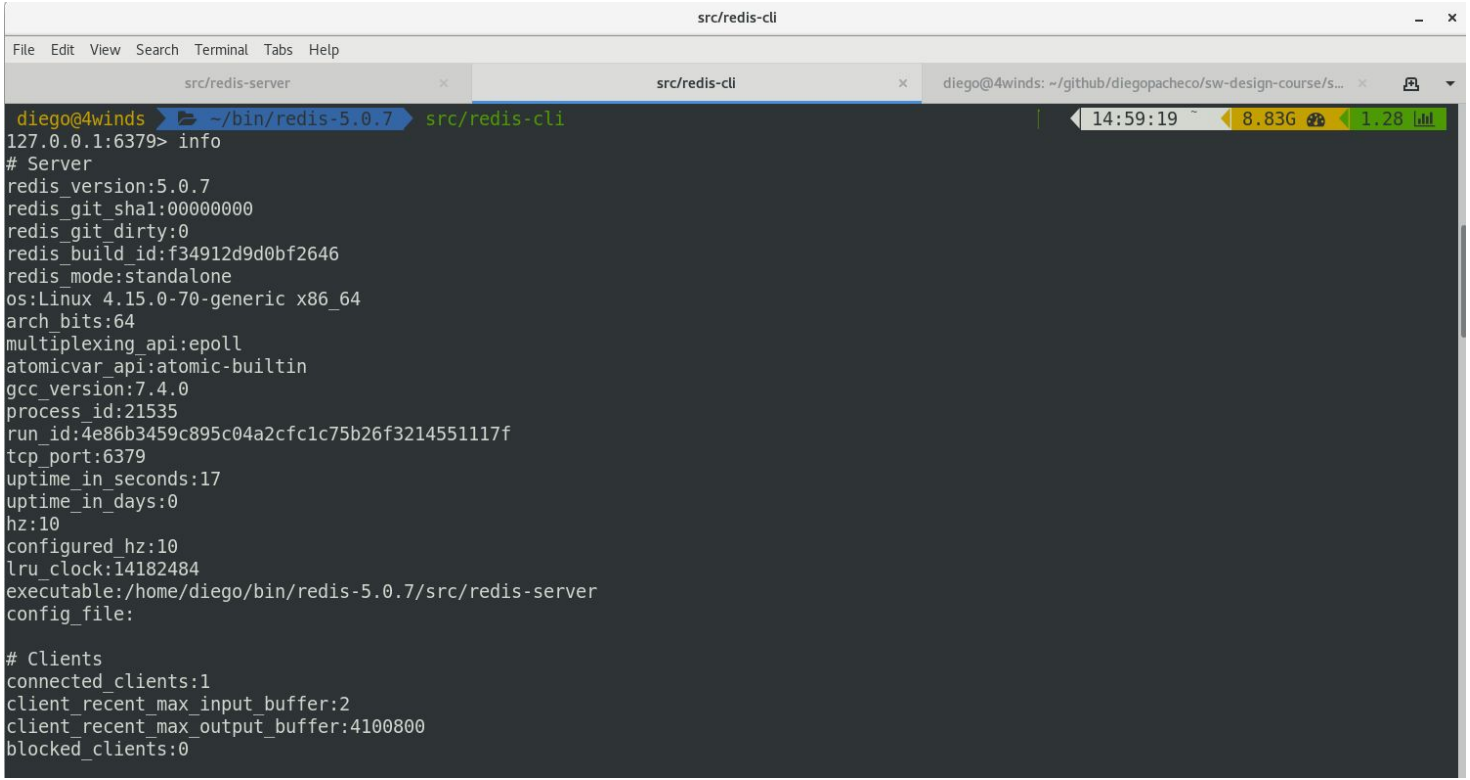

Big things in 5.0
STREAMS
C1
C2
C3
CLIENT1
CLIENTL
• TIME SERIES
• IoT
• EVENTS SOURCING
• MESSAGING
• AND MORE ...

Redis: make ; src/redis-server



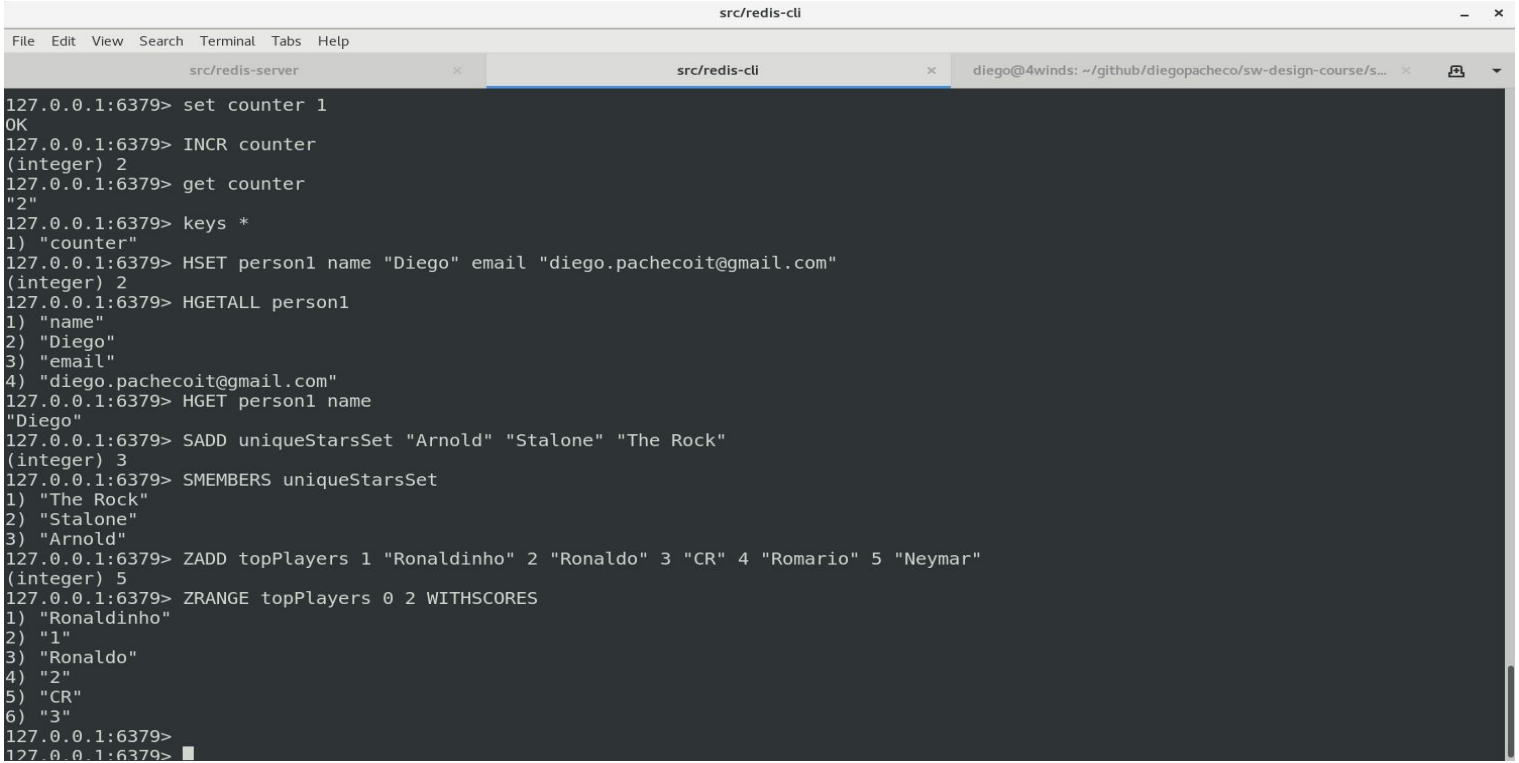
https://redis.io/download

Redis: src/redis-cli



https://redis.io/download

Redis: src/redis-cli



https://redis.io/commands/

Redis & Java: Lettuce


```java
import io.lettuce.core.RedisClient;
import io.lettuce.core.api.StatefulRedisConnection;

public class SimpleLettuceMain {
    Run | Debug
    public static void main(String[] args) {
        RedisClient redisClient = RedisClient.create("redis://localhost/0");
        StatefulRedisConnection<String, String> connection = redisClient.connect();

        System.out.println("Connected to Redis");
        connection.sync().set("k1", "Hello World");
        System.out.println("Key from redis k1: " + connection.sync().get("k1"));

        connection.close();
        redisClient.shutdown();
    }
}
```

```
nov 22, 2019 3:24:23 PM io.lettuce.core.EpollProvider <clinit>
INFO: Starting without optional epoll library
nov 22, 2019 3:24:23 PM io.lettuce.core.KqueueProvider <clinit>
INFO: Starting without optional kqueue library
Connected to Redis
Key from redis k1: Hello World
```

Redis & Java: Spring Data + Lettuce

src/redis-cli

File Edit View Search Terminal Tabs Help

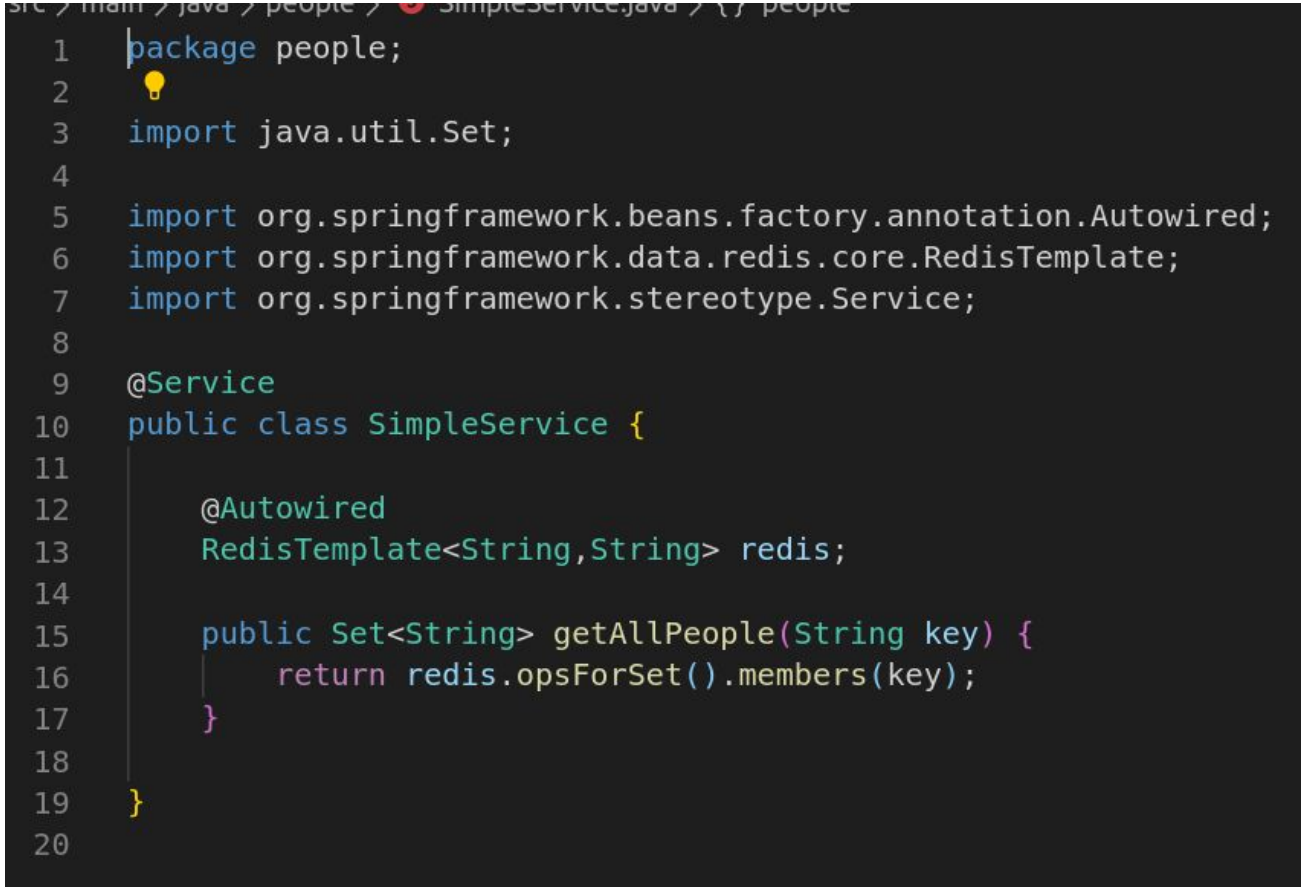
src/redis-cli × src/redis-server ×

```
127.0.0.1:6379> sadd people "diego" "melina" "gandalfy"
(integer) 0
127.0.0.1:6379> SMEMBERS people
1) "melina"
2) "gandalfy"
3) "diego"
127.0.0.1:6379>
```

Redis & Java: Spring Data + Lettuce


```java
package people;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.RedisStandaloneConfiguration;
import org.springframework.data.redis.connection.lettuce.LettuceConnectionFactory;
import org.springframework.data.redis.core.RedisTemplate;

@Configuration
class AppConfig {

  @Bean
  public LettuceConnectionFactory redisConnectionFactory() {
    return new LettuceConnectionFactory(new RedisStandaloneConfiguration("127.0.0.1", 6379));
  }

  @Bean
  public RedisTemplate<?, ?> redisTemplate() {
      RedisTemplate<byte[], byte[]> template = new RedisTemplate<>();
      template.setConnectionFactory(redisConnectionFactory());
      return template;
  }

}

```

Redis & Java: Spring Data + Lettuce



```java
package people;

import java.util.Set;

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

@Service
public class SimpleService {

    @Autowired
    RedisTemplate<String,String> redis;

    public Set<String> getAllPeople(String key) {
        return redis.opsForSet().members(key);
    }

}
```

Redis & Java: Spring Data + Lettuce


```java
package people;

import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class SpringDataRedisMain {

    Run | Debug
    public static void main(String[] args) {
        SpringApplication.run(SpringDataRedisMain.class, args);
    }

    @Bean
    public CommandLineRunner commandLineRunner(ApplicationContext ctx) {
        return args -> {

            System.out.println("Running Spring Boot application in Console... ");

            SimpleService service = ctx.getBean(SimpleService.class);
            System.out.println("All People :" + service.getAllPeople("people"));

        };
    }
}

```

Redis & Java: Spring Data + Lettuce

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Java Process Console

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.1.RELEASE)

2019-11-22 16:10:14.823  INFO 18965 --- [           main] people.SpringDataRedisMain               : Starting Spring
DataRedisMain on 4winds with PID 18965 (/home/diego/github/diegopacheco/sw-design-course/src/java/java-spring-data-r
edis-fun/bin/main started by diego in /home/diego/github/diegopacheco/sw-design-course/src/java/java-spring-data-red
is-fun)
2019-11-22 16:10:14.827  INFO 18965 --- [           main] people.SpringDataRedisMain               : No active profi
le set, falling back to default profiles: default
2019-11-22 16:10:15.469  INFO 18965 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring
 Data modules found, entering strict repository configuration mode!
2019-11-22 16:10:15.473  INFO 18965 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping S
pring Data repositories in DEFAULT mode.
2019-11-22 16:10:15.520  INFO 18965 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring
 Data repository scanning in 24ms. Found 0 repository interfaces.
2019-11-22 16:10:16.573  INFO 18965 --- [           main] people.SpringDataRedisMain               : Started SpringD
ataRedisMain in 2.161 seconds (JVM running for 2.568)
Running Spring Boot application in Console...
2019-11-22 16:10:16.791  INFO 18965 --- [           main] io.lettuce.core.EpollProvider            : Starting withou
t optional epoll library
2019-11-22 16:10:16.792  INFO 18965 --- [           main] io.lettuce.core.KqueueProvider           : Starting withou
t optional kqueue library
All People :[melina, gandalfy, diego]
```

Redis & Java: Spring Data + Lettuce : Hash Mapping


```java
package people;

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

@RedisHash("players")
public class Person{
    @Id String id;
    String name;

    public Person(){}

    public Person(String id,String name){
        this.id=id;
        this.name=name;
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    @Override
    public String toString() {
        return "id: " + id + " name: " + name;
```

Redis & Java: Spring Data + Lettuce : Hash Mapping


```
1   package people;
2
3   import org.springframework.data.repository.CrudRepository;
4
5   public interface PersonRepository extends CrudRepository<Person, String> {}
6
```

Redis & Java: Spring Data + Lettuce : Hash Mapping


```java
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;

@SpringBootApplication
public class SpringDataRedisRepositoryMain{

    Run | Debug
    public static void main(String[] args) {
        SpringApplication.run(SpringDataRedisRepositoryMain.class, args);
    }

    @Bean
    public CommandLineRunner commandLineRunnerMapper(ApplicationContext ctx) {
        return args -> {

            System.out.println("Running Spring Boot application in Console... ");

            PersonRepository repo = ctx.getBean(PersonRepository.class);
            Person p = new Person("diegopacheco", "diego.pacheco.it@gmail.com");
            repo.save(p);
            System.out.println("Persons: " + repo.findAll());
            System.out.println("Count: " + repo.count());
            repo.delete(p);
        };
    }

}
```

Redis & Java: Spring Data + Lettuce : Hash Mapping

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Java Process Console

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.1.RELEASE)

2019-11-22 16:45:46.449  INFO 16552 --- [           main] people.SpringDataRedisRepositoryMain     : Starting Spring
DataRedisRepositoryMain on 4winds with PID 16552 (/home/diego/github/diegopacheco/sw-design-course/src/java/java-spr
ing-data-redis-fun/bin/main started by diego in /home/diego/github/diegopacheco/sw-design-course/src/java/java-sprin
g-data-redis-fun)
2019-11-22 16:45:46.453  INFO 16552 --- [           main] people.SpringDataRedisRepositoryMain     : No active profi
le set, falling back to default profiles: default
2019-11-22 16:45:46.962  INFO 16552 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring
 Data modules found, entering strict repository configuration mode!
2019-11-22 16:45:46.965  INFO 16552 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping S
pring Data repositories in DEFAULT mode.
2019-11-22 16:45:47.185  INFO 16552 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring
 Data repository scanning in 199ms. Found 1 repository interfaces.
2019-11-22 16:45:48.274  INFO 16552 --- [           main] people.SpringDataRedisRepositoryMain     : Started SpringD
ataRedisRepositoryMain in 2.226 seconds (JVM running for 3.039)
Running Spring Boot application in Console...
2019-11-22 16:45:48.491  INFO 16552 --- [           main] io.lettuce.core.EpollProvider            : Starting withou
t optional epoll library
2019-11-22 16:45:48.493  INFO 16552 --- [           main] io.lettuce.core.KqueueProvider           : Starting withou
t optional kqueue library
All People :[melina, gandalfy, diego]
Running Spring Boot application in Console...
Persons: [id: diegopacheco name: diego.pacheco.it@gmail.com]
Count: 1
```

In-memory Persistence - Redis as Source of Truth = Dynomite

https://github.com/Netflix/dynomite
http://diego-pacheco.blogspot.com/2016/05/june-1st-2016-i-will-be-speaking-about.html
https://www.youtube.com/watch?v=Z4_rzsZd70o&feature=youtu.be
http://diego-pacheco.blogspot.com/search?q=dynomite

Q & A

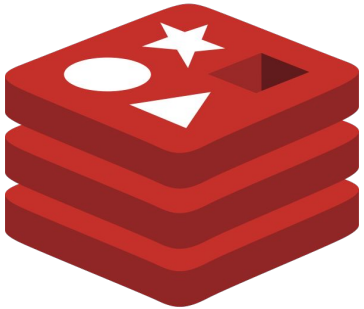
Exercises

Constraints

You MUST code this exercises with Java.
You can use any framework you like. UI is not required.
You need to run Redis 5.x (nativily, docker, does not matter).
Tip: Think about data structures first, test on the CLI them code in Java,

1. Run redis-cli and play with the commands: SET, GET, HSET, HGET, HGETALL, STRLEN
2. Create a Service that count how many times a Movie was watched.
3. Create a simple Cache Service where you have the operations: GET and SET and you receive an string as key and String as value. Serialize the value using Jackson.
4. Create a User Profile Service, where the user should have: name, address, dateOfBirth, twitterHandler, email and operations to change any of this fields based on the email as key.
5. Create a News Feed Service where you will have a Timeline per user and the links of news prioritized by relevance(tip sorted sets). You should have multiple data Structures in Redis.

https://github.com/diegopacheco/redis-training.git



Redis

Diego Pacheco