**Лабораторная работа №2 «Redis. Tutorial»**

Цель: получить навыки работы с протоколом Redis.

1. Заходим на сайт Redis в tutorial <https://try.redis.io/> (рисунок 1).



Рисунок 1. Начальная страница Tutorial

1. Redis - это так называемое хранилище ключей, которое часто называют базой данных NoSQL. Суть хранилища ключей - возможность хранить некоторые данные, называемые значением, внутри ключа. Эти данные могут быть впоследствии получены только в том случае, если мы знаем точный ключ, используемый для его хранения. Мы можем использовать команду SET для сохранения значения «fido» в ключе «server: name» (рисунок 2).

Redis будет хранить наши данные на постоянной основе, поэтому мы можем спросить: «Какое значение хранится с ключом «server: name» и Redis ответит «fido» (рисунок 2).

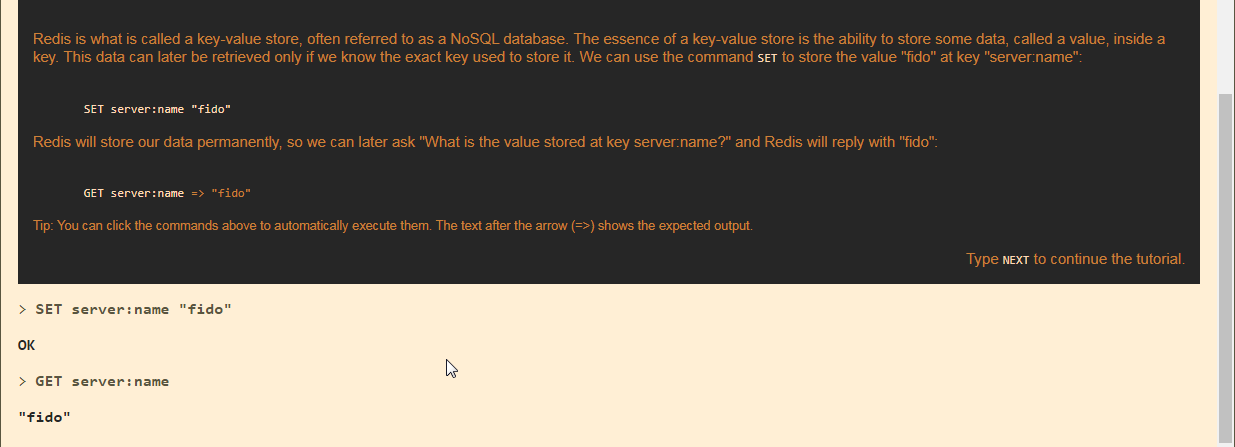


Рисунок 2. Set и Get - запросы в Redis

1. Другими распространенными операциями являются DEL для удаления заданного ключа и связанного с ним значения, SETNX, который устанавливает ключ, только если он еще не существует, и INCR для атомарного увеличения числа, сохранённого в заданном ключе. На рисунке 3 прописаны примеры запросов.

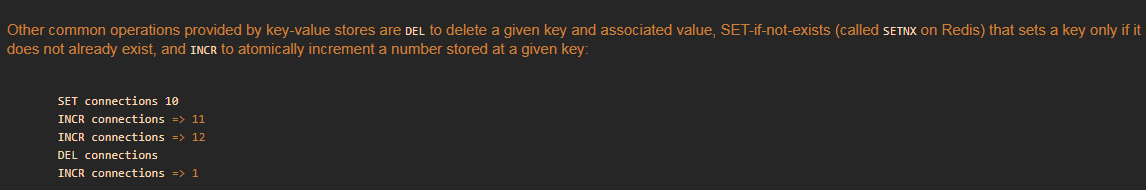


Рисунок 3. Примеры запросов INCR, DEL.

1. В Redis можно задать значение времени с помощью команды EXPIRE, в течение которого ключ может существовать. Команда TTL возвращает значение оставшегося времени (рисунок 4).

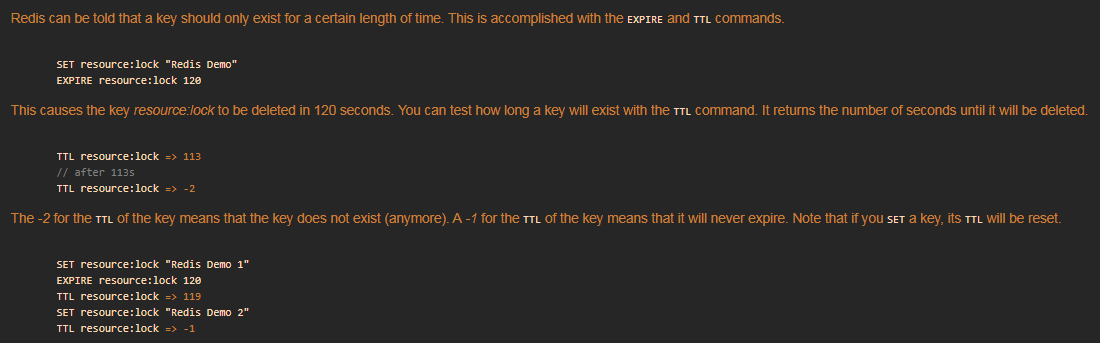


Рисунок 4. Выполнение команд EXPIRE и TTL

1. Redis также поддерживает несколько более сложных структур данных. Первый, который мы рассмотрим, - это список. Список представляет собой серию упорядоченных значений. Некоторые из важных команд для взаимодействия со списками - RPUSH, LPUSH, LLEN, LRANGE, LPOP и RPOP. Вы можете сразу начать работу с ключом в виде списка, если он еще не существует как другой тип.

* RPUSH добавляет новое значение в конец списка
* LPUSH ставит новое значение в начале списка
* LRANGE дает подмножество списка.

LRANGE принимает индекс первого элемента, который вы хотите получить в качестве его первого параметра, и индекс последнего элемента, который вы хотите получить в качестве второго параметра. Значение -1 для второго параметра означает получение элементов до конца списка (рисунок 5).

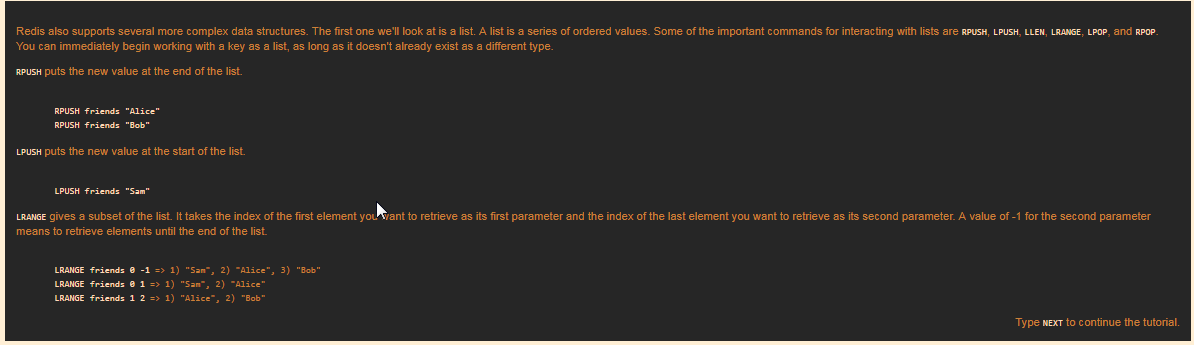


Рисунок 5. Работа со списком командами RPUSH, LPUSH, LRANGE

* LPOP удаляет первый элемент из списка и возвращает его
* RPOP удаляет последний элемент из списка и возвращает его
* LLEN возвращает текущую длину списка

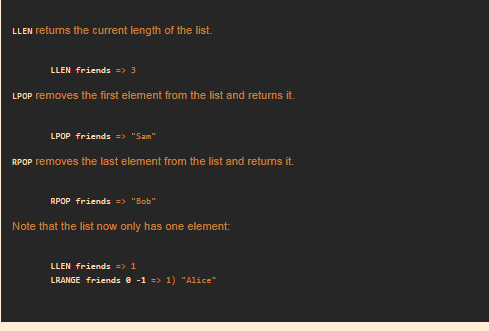


Рисунок 6. Работа со списком командами LLEN, LPOP и RPOP

1. Следующая структура данных, на которую мы будем смотреть, - это набор. Набор похож на список, за исключением того, что он не имеет определенного порядка, и каждый элемент может появляться только один раз. Некоторые из важных команд при работе с наборами - SADD, SREM, SISMEMBER, SMEMBERS и SUNION.

* SADD добавляет заданное значение в набор
* SREM удаляет заданное значение из набора

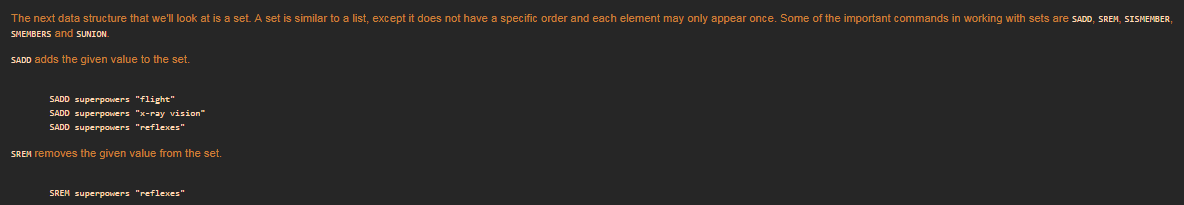


Рисунок 7. Работа с набором командами SADD, SREM

* SISMEMBER проверяет, находится ли заданное значение в наборе. Он возвращает 1, если значение есть, а 0 - нет.
* SMEMBERS возвращает список всех членов этого набора.
* SUNION объединяет два или более набора и возвращает список всех элементов.

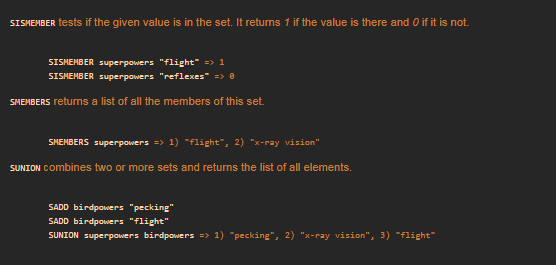


Рисунок 8. Работа с набором командами SISMEMBER, SMEMBERS, SUNION

Наборы - очень удобный тип данных, но поскольку они не отсортированы, поэтому с рядом проблем они хорошо не работают. Вот почему Redis 1.2 представил отсортированные наборы. Сортированный набор похож на обычный набор, но теперь каждое значение имеет связанный счет.

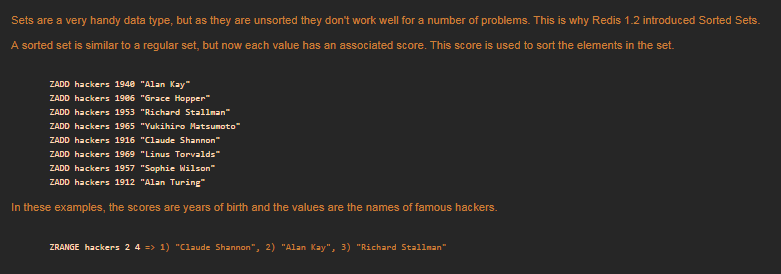


Рисунок 9. Работа с сортированными наборами

1. Хэши - это карты между строковыми полями и строковыми значениями, поэтому они являются идеальным типом данных для представления объектов (например: Пользователь с несколькими полями, такими как имя, фамилия, возраст и т. Д.).

* Чтобы вернуть сохраненные данные, используйте HGETALL
* Чтобы установить сразу несколько полей – HMSET
* Получить одно значение поля – HGET

На рисунке 10 отображена работы с хэшами.

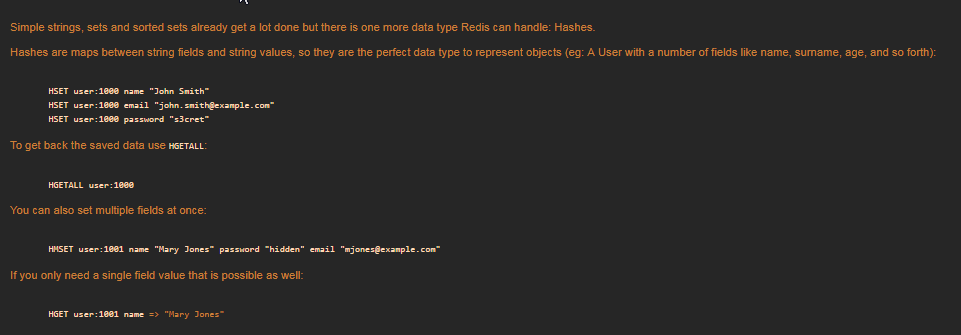


Рисунок 10. Работа с хэщ-полями

1. Числовые значения в хэш-полях обрабатываются точно так же, как и в простых строках, и есть операции для увеличения этого значения в атомном режиме (рисунок 11).

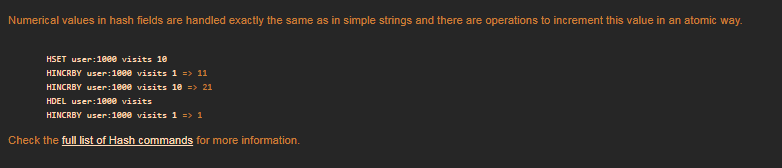


Рисунок 11. Обработка хэш-полей