Лекция 15

ПСКП, ПОИТ-3

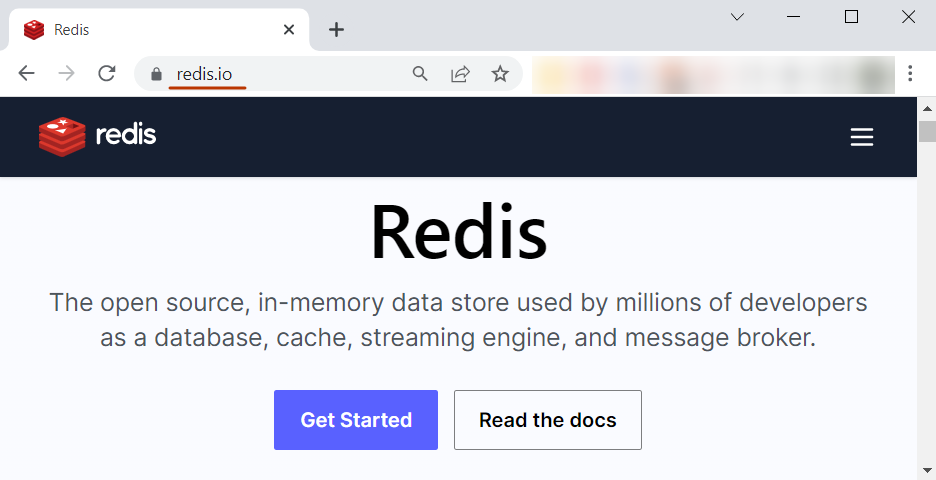
**Node.js: Redis**

1. **Redis** (**RE**mote **DI**ctionary **S**erver)**:** noSQL СУБД с открытым кодом (BSD-лицензия), Redis Labs, Сальваторе Санфилиппо:

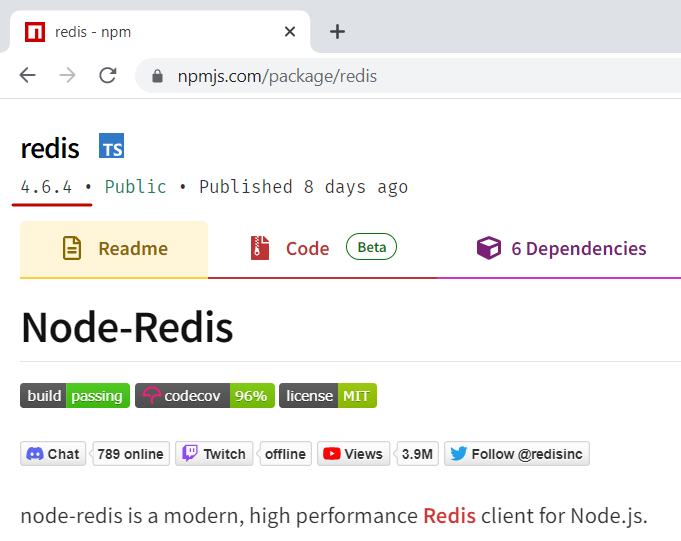
* хранилище данных в оперативной памяти (таким образом объем данных ограничен объемом оперативной памяти);
* написана на ANSI С;
* используется в качестве БД, кэша или брокера сообщений;
* структуры данных: строки, списки, наборы, отсортированные наборы, хэш-таблицы, потоки, растровые изображения, битовые поля, геопространственные индексы, HyperLogLog;
* есть возможность указать время жизни данных (проверка помощью TTL);
* СУБД ориентирована на быстрое выполнение атомарных операций (выполн. одной операции порядка 0,00001 сек.);
* поддерживает транзакции и пакетную обработку команд;
* есть механизм издателль/подписчик;
* умеет масштабироваться путем репликации;
* есть механизмы обеспечения персистенности (сохранность состояния после завершения процесса): 1) механизм RDB-снимков для асинхронного сохранения (с потерями); 2) механизм упреждающей записи;
* последняя стабильная версия: 7**;**
* API: C, C++, C#, Java, JavaScript, Python, … ;
* нет официальной версии для Windows;
* установка: Windows(WSL2, OpenServer, служба, Docker), Linux(apt), MacOS(brew);
* есть Redis CLI.

HyperLogLog - это алгоритм для задачи, связанной с подсчетом, аппроксимирующий количество различных элементов в мультимножестве. Для вычисления точного количества элементов мультимножества требуется объем памяти, пропорциональный количеству элементов, что нецелесообразно для очень больших наборов данных**.**

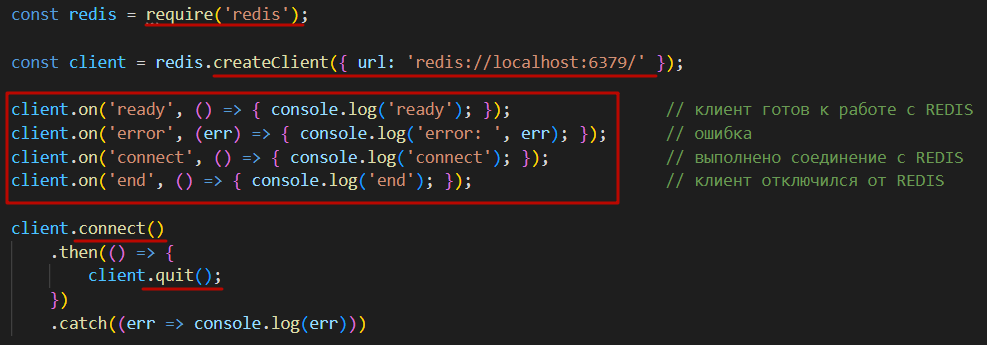
1. **Redis:**

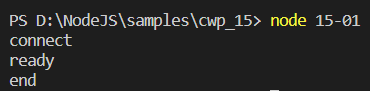


1. **Redis:** облачное хранилище <https://redis.com/try-free/>
2. **Redis:** локальный сервер
3. **Redis:** информация по установке в зависимости от ОС <https://redis.io/docs/getting-started/installation/>
4. **Redis:** как вариант,можно развернуть Docker-container (<https://hub.docker.com/_/redis>) либо скачать и установить службу Redis (<https://github.com/microsoftarchive/redis/releases>, ***самый простой вариант для Windows***)
5. **Redis:** npm redis

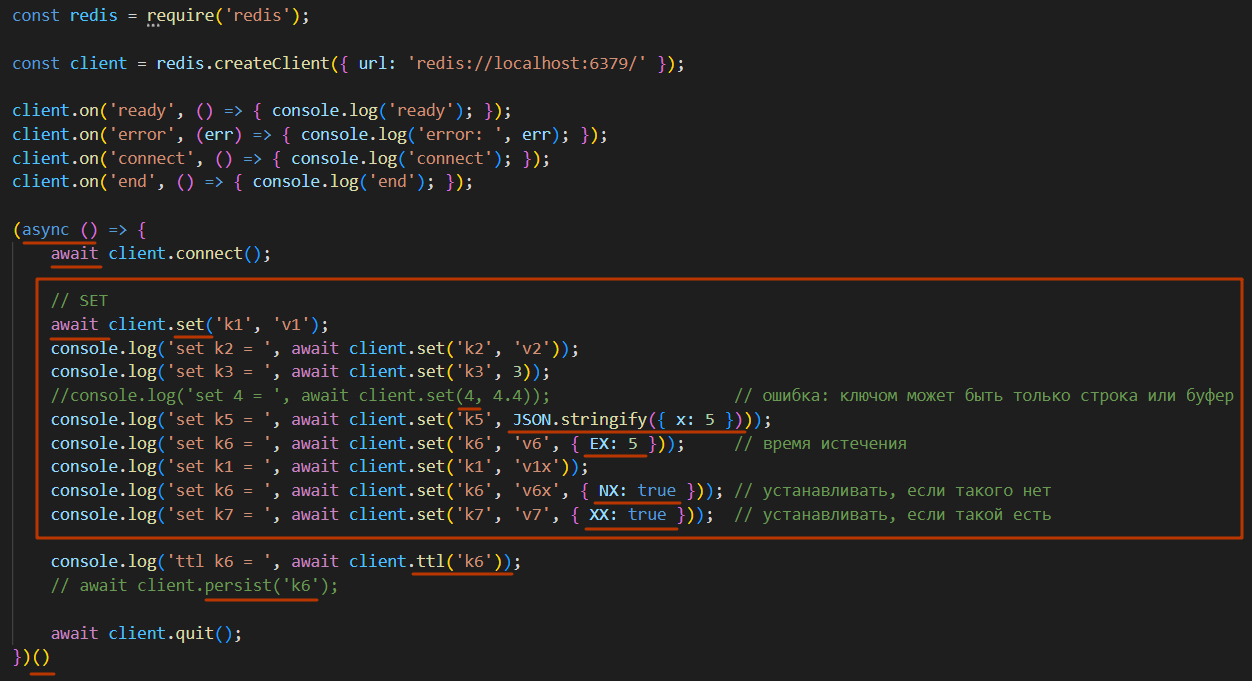


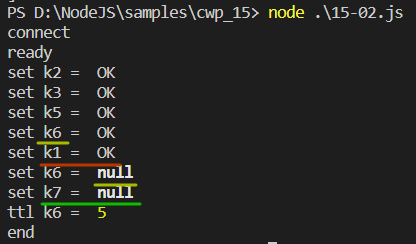
1. **Redis:** npm redis,соединение, завершение, события



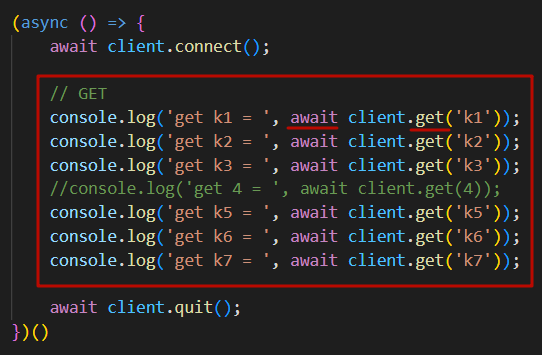
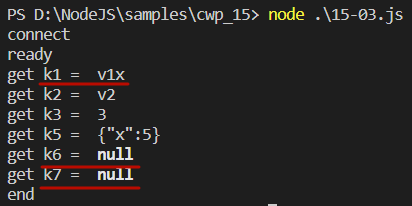


1. **Redis:** set, NX, XX

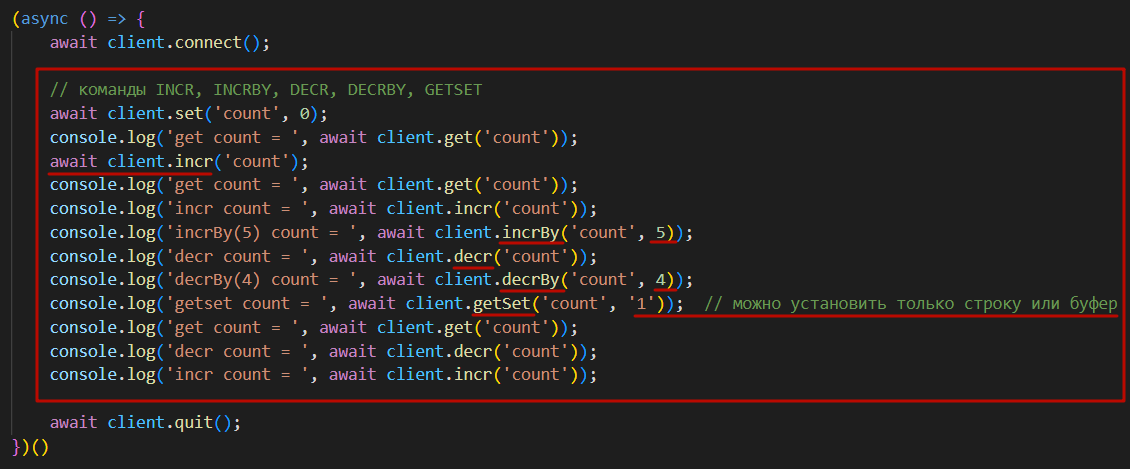


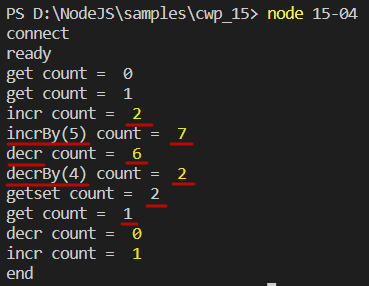


1. **Redis:** с помощью .ttl(key) можно проверить оставшееся время жизни
2. **Redis:** при необходимости убрать установленное время жизни можно воспользоваться методом .persist(key)
3. **Redis:** get

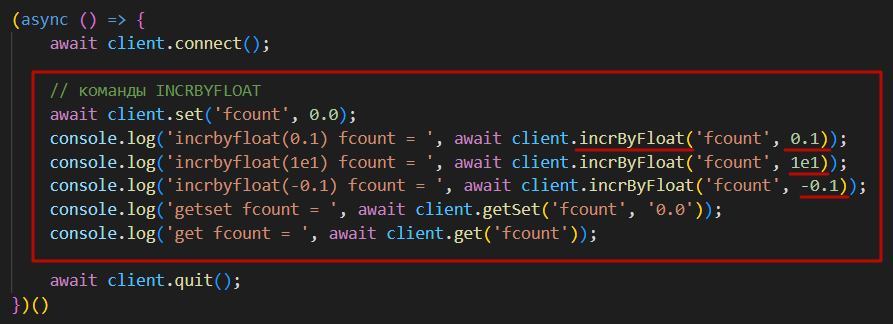
 

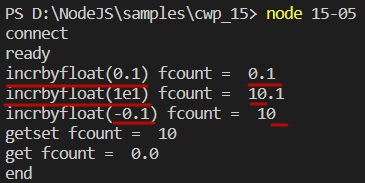
1. **Redis:** incr, incrby, decr, decrby, getset



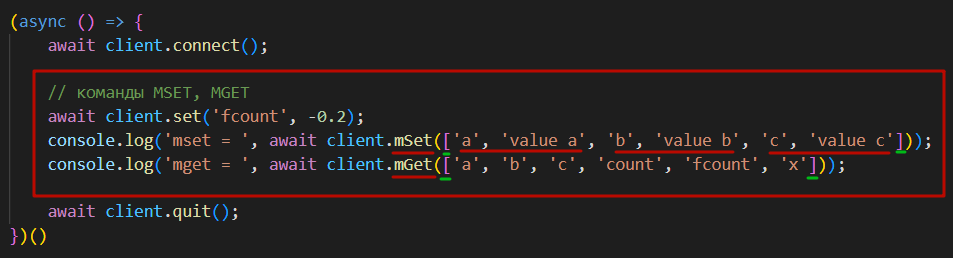


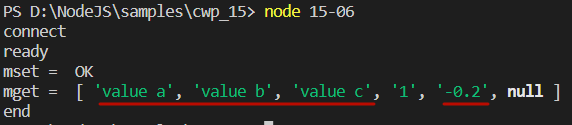
1. **Redis:** incrbyfloat, getset



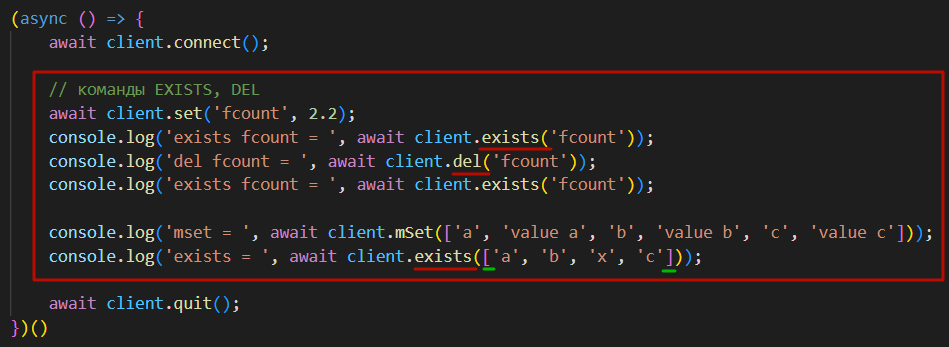


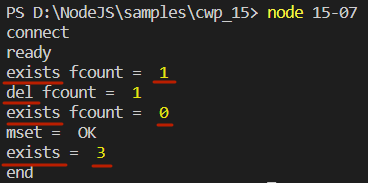
1. **Redis:** mset, mget



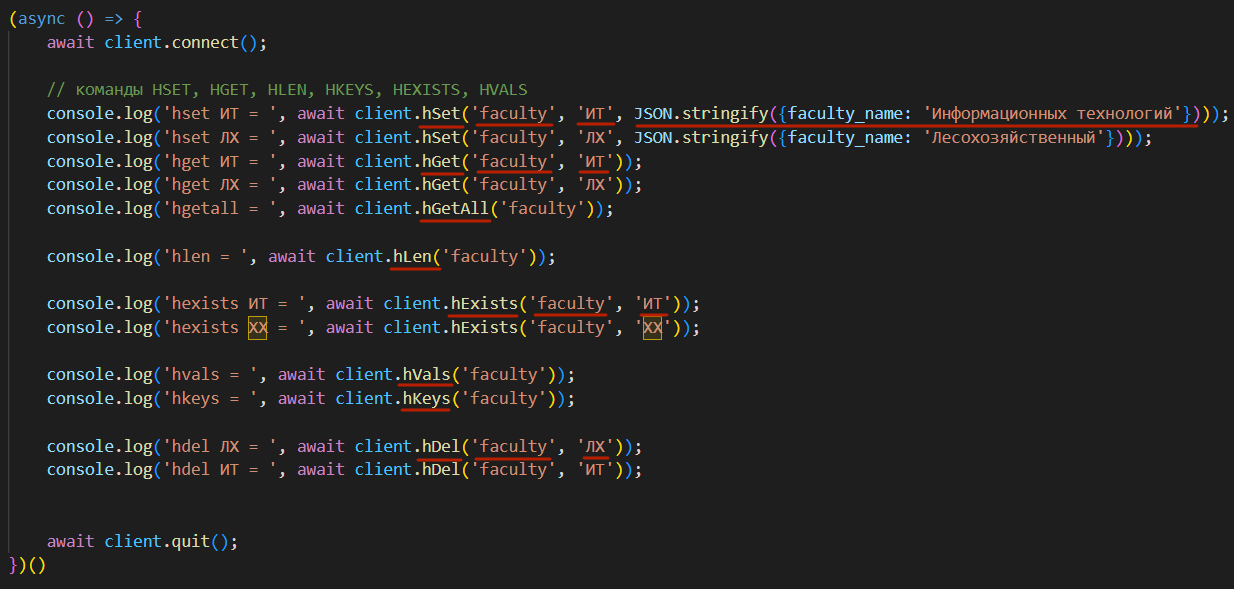


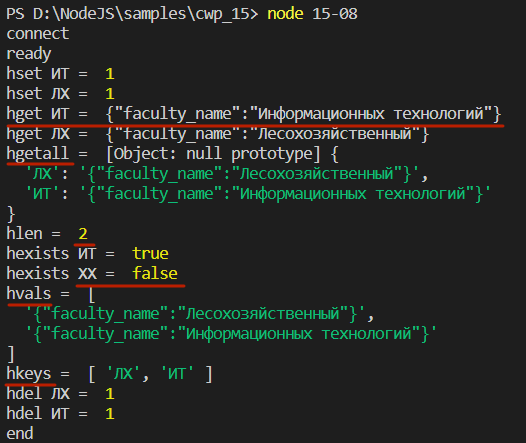
1. **Redis:** del, exists



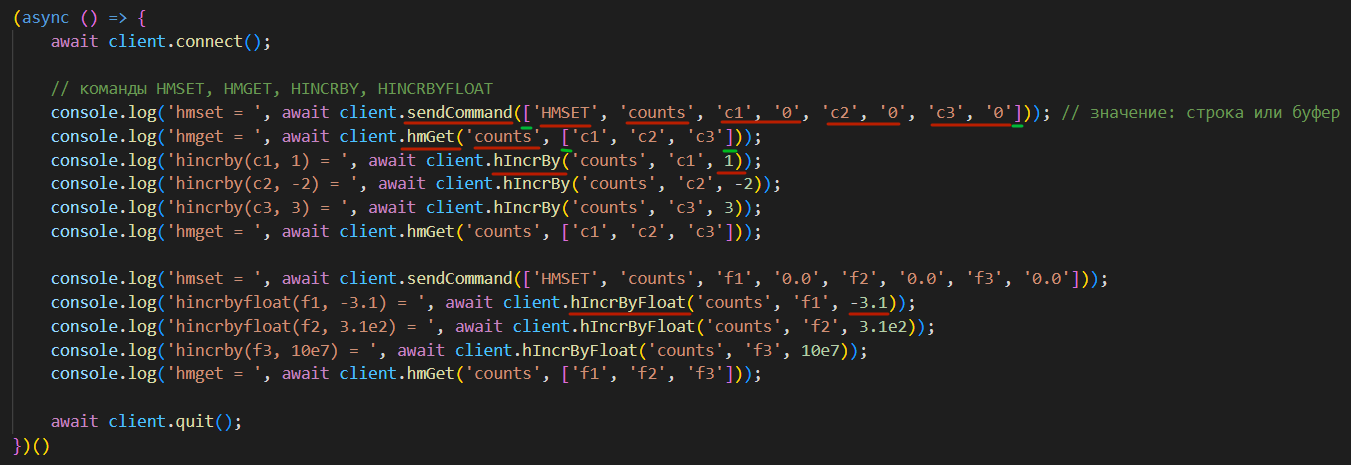


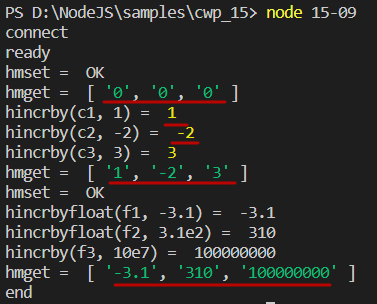
1. **Redis:** hset, hget, hlen, hdel, hexists, hgetall, hvals



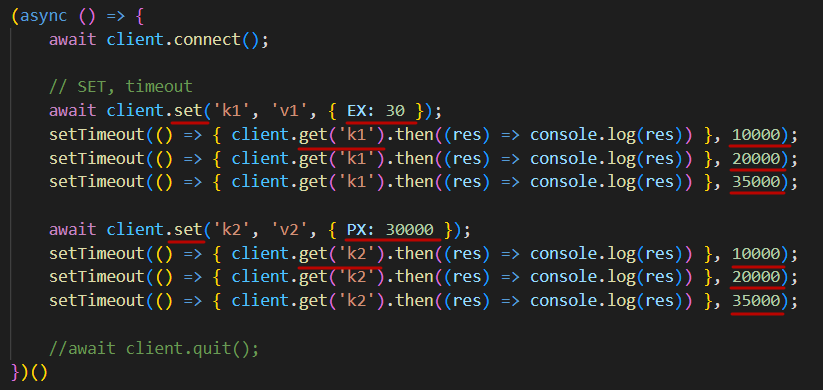


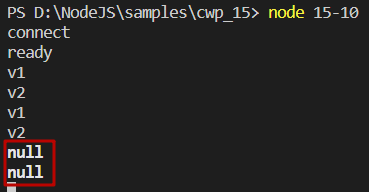
1. **Redis:** hmset, hmget, hincrby, hincrbyfloat



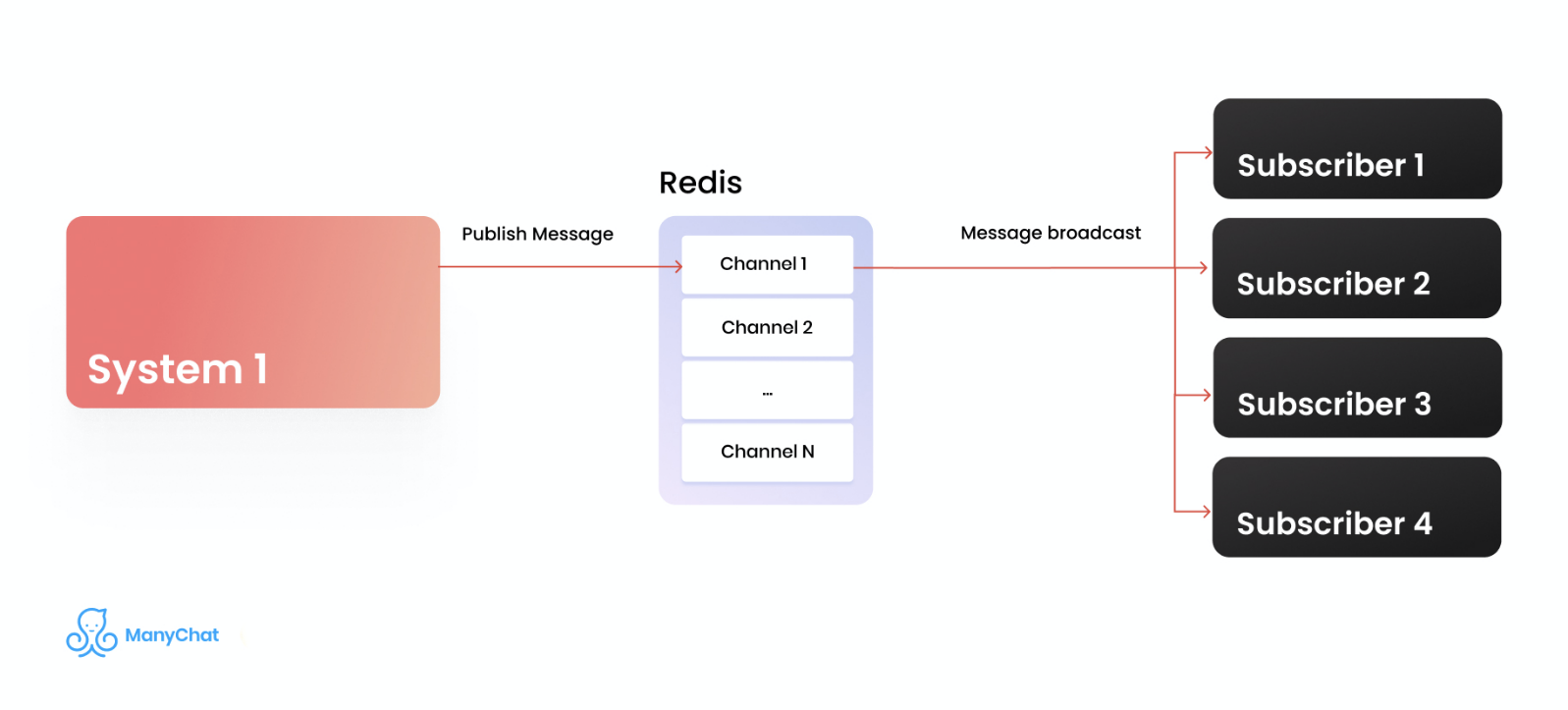


1. **Redis:** set, get, timeout

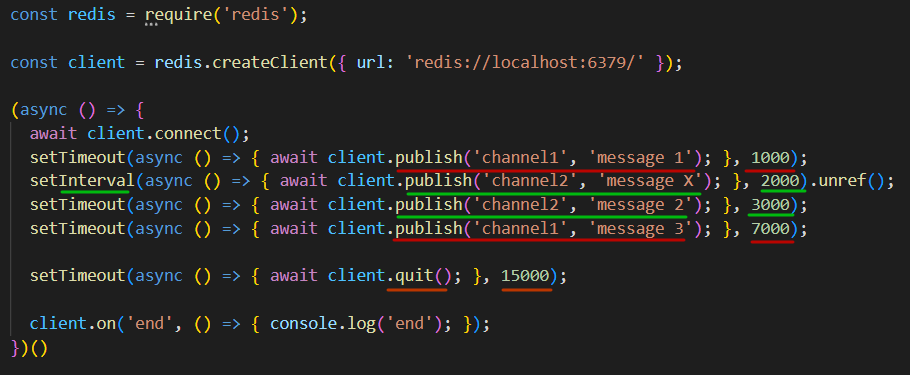




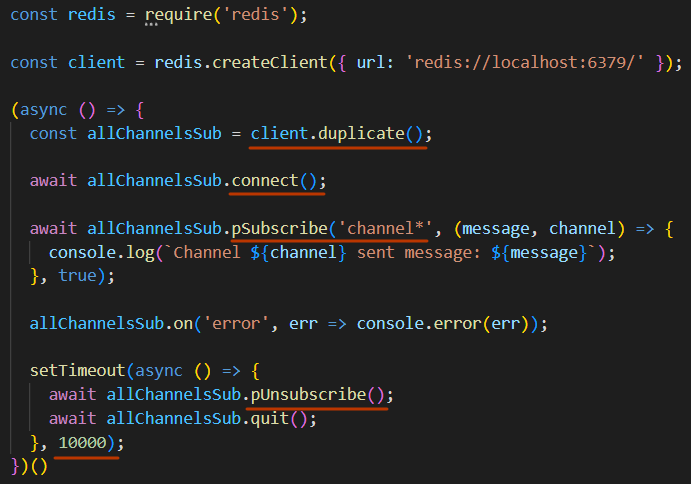
1. **Redis:** канал (channel) – это абстракция, используемая для быстрого обмена данными между издателем и подписчиком.

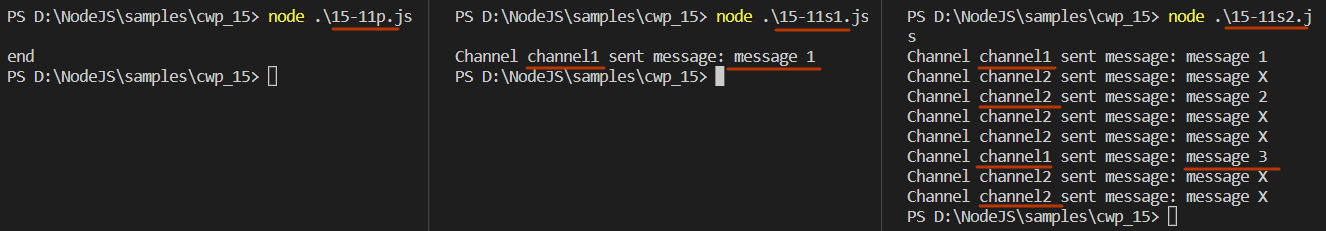


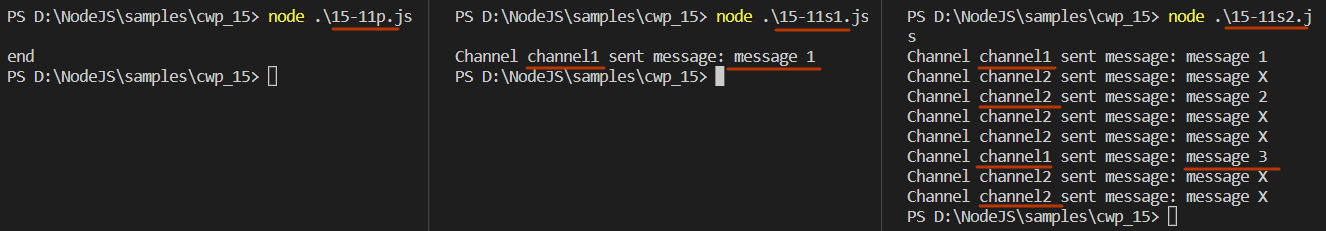
1. **Redis:** publish, subscribe, psubscribe





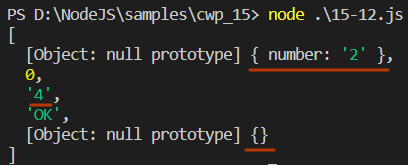






1. **Redis:** транзакция**,** multi, exec, disconnect





1. **Redis:** для *изолированной* транзакции можно отслеживать ключи, используя .watch(). Транзакция будет прервана, если какой-либо из отслеживаемых ключей изменится.
2. **Redis:** list, sorted set

