



Desafio de Desenvolvimento - Aquiris

Objetivo

[Redis](#) é um banco de dados NoSQL em memória que suporta operações - ou comandos - sobre diversas estruturas de dados, tais como conjuntos, listas e hashes. Seu objetivo é implementar um serviço que contemple um subconjunto de comandos do Redis, ou seja, você deverá implementar um "mini Redis".

O exercício contém, assim, duas partes: em primeiro lugar, a implementação dos comandos do Redis e estruturas de dados necessárias para executá-los, em segundo lugar (opcionalmente), suporte para executar TODOS os comandos do seu "mini Redis" através da rede.

Conforme você iniciar o trabalho no desafio, tente concluir a primeira parte complementamente antes de seguir para a segunda etapa.

Parte 1: Núcleo da aplicação

Requisitos

Comandos

Você deverá implementar os seguintes comandos. A definição de cada um deles pode ser encontrada na página principal da documentação do Redis (<http://redis.io/commands>), onde você também poderá testar os parâmetros de entrada e resultados das execuções dos mesmos.

1. [SET](#) key value
2. [SET](#) key value EX seconds (você não precisa implementar as outras opções do comando SET)
3. [GET](#) key
4. [DEL](#) key [key ...] (você deve implementar a exclusão de múltiplas chaves)
5. [DBSIZE](#)
6. [INCR](#) key
7. [ZADD](#) key score member
8. [ZCARD](#) key
9. [ZRANK](#) key member



10. [ZRANGE](#) key start stop (nenhuma implementação adicional é necessária, apenas a padrão)

Atomicidade

Um dos principais benefícios que o Redis garante é o acesso atômico e ordenado aos dados. Sua implementação deverá oferecer a mesma garantia, isto é, o acesso por múltiplas threads (ou clientes) deverá ser realizado de forma segura.

Pacote de entrega

Quando você tiver concluído essa parte do desafio deverá ter uma implementação dos comandos de acordo com a especificação acima e um ou mais métodos - ou pontos de entrada - que possam invocar os mesmos, isto é, seu pacote de entrega deverá conter scripts de testes ou testes unitários que possam demonstrar a funcionalidade dos comandos implementados.

Parte 2 (Opcional): comunicação em rede

Nessa parte do desafio você incluirá uma forma de acesso ao seu "mini Redis" através da rede, aceitando múltiplas conexões de forma simultânea.

Requisitos

Quando executada, sua aplicação deve esperar por conexões HTTP na porta 8080. Você não precisa se preocupar em implementar o protocolo do Redis; ao invés disso, faremos algumas modificações radicais para simplificar seu trabalho:

1. Chaves e valores são representados por caracteres do conjunto [a-zA-Z0-9_].
2. Comandos são strings ASCII com parâmetros delimitados por espaços.
3. Respostas da sua aplicação devem ser strings ASCII com valores delimitados por espaços (de acordo com a necessidade).

Outros requisitos de rede

1. Seu servidor deverá suportar múltiplas conexões simultâneas, assim como o Redis real.
2. Deve ser possível conectar ao seu servidor com qualquer cliente HTTP, conforme exemplo abaixo. O exemplo apresenta apenas alguns comandos: seu servidor deverá suportar todas as operações implementadas na parte 1.



Exemplo

```
curl localhost:8080/?cmd=SET%20mykey%20cool-value      OK
curl localhost:8080/?cmd=GET%20mykey                   cool-value
curl localhost:8080/?cmd=DEL%20mykey                   OK
curl localhost:8080/?cmd=GET%20mykey                   (nil)
```

Opcionalmente você pode iterar na sua implementação e utilizar o padrão REST para seus comandos, conforme exemplo abaixo. No exemplo apenas alguns comandos são apresentados, no entanto a sua aplicação deverá expor todos os comandos disponíveis. A escolha de verbos e caminhos adequados para cada um deles é de sua responsabilidade.

Exemplo

```
curl -d "cool-value" -X PUT localhost:8080/mykey        OK
curl localhost:8080/mykey                               cool-value
curl -X DELETE localhost:8080/mykey                    OK
curl localhost:8080/mykey                               (nil)
```

Orientações Gerais

Linguagens de programação e Frameworks

Você pode utilizar qualquer linguagem de programação que queira, bem como frameworks com os quais se sinta confortável para completar o desafio. O uso de uma base de dados Redis como parte da implementação invalida o teste.

Suporte de Bibliotecas

Para completar o desafio você deve se concentrar em completar as funcionalidades do núcleo principal do Redis, incluindo as estruturas de dados necessárias para suportar os comandos solicitados. Você pode utilizar bibliotecas que apresentem bom desempenho para implementação de servidores HTTP como ponto de partida para o desenvolvimento da sua camada de rede.



Recursos Adicionais

Você pode consultar qualquer material além deste guia, incluindo recursos online que você considere úteis para o desenvolvimento desse desafio, mas esteja preparado para explicar o desenvolvimento de sua solução, bem como as decisões de arquitetura, escolhas de estruturas de dados, organização dos módulos, etc.

Submissão

Ao concluir o desenvolvimento do desafio, crie um arquivo compactado (**ZIP**) e o envie por email para o entrevistador que lhe solicitou o teste.