

# آشنایی با Redis

## Redis چیه؟

Redis مخفف عبارت Remote Dictionary Server است. در واقع ردیس یک نوع ساختمان داده است که در RAM قرار می‌گیرد و اطلاعات به صورت موقتی در آن ذخیره می‌شوند.

ساختار های داده ای یا Data Structure می‌توانند یک رشته ساده و یا یک لیستی از داده‌ها باشند که Redis اونها رو توی RAM برای ما نگه‌داری می‌کنه و می‌توانیم با سرعت بالا اون‌ها رو بازیابی کنیم. از آنجایی که داده‌ها رو ذخیره و بازیابی می‌کند یک نوع دیتابیس است و چون توی RAM نگه‌داری می‌دارد اصطلاحاً in-memory Database نامگذاری می‌شود. اما برای هر داده‌ای که در Redis ذخیره می‌کنیم یک key یا کلید داریم و زمان بازیابی با استفاده از این key به داده خودمون که همان value هست می‌رسیم و بنابر این Redis یک key value database هست.

### مطالعه

برای آشنایی بیشتر با اینکه redis چی هست [این ویدیو](#) رو ببین.

### مطالعه

برای مقایسه‌ی redis با یکی دیگه از in memory cache های معروف (memcache) [این لینک](#) رو مطالعه کن.

## کامندهای ساده GET و SET و TTL؟

برای ست کردن یک کلید با یک مقدار از دستور "`set <key> <value>`" استفاده می‌کنیم و برای گرفتن مقدار ست شده هم از "`get <key>`" استفاده می‌کنیم. همچنین برای اینکه به ردیس بگیم چقدر نیاز داریم تا این مقدار رو در حافظه نگه‌داریم میتوانیم از `ttl` استفاده کنیم که مخفف `time to live` است استفاده کنیم.

## تمرین ۱

فرض کنید سرویس فروشگاهی ای داریم و برای هر کدام از وندور های قرار است یک صورت حساب ایجاد کنیم که فرایند پیچیده ای دارد و ما نیاز داریم این مقدار را یک روز در کش نگهداری کنیم تا در منابع صرفه جویی کنیم. فرض کنید دیتا ها به این شکل هستند:

Ali => 1000

Alex => 2000

Hamid => 1356

Hirad => 10180

این دیتا های را با ttl برابر با یک روز روی redis بریزید و بعد سعی کنید مقدار های ست شده را بخوانید.

## پیاده سازی لاک با ردیس

هدف لاک این است که اطمینان حاصل شود که در بین چند اینستنس کد که ممکن است سعی کنند کار مشابهی را انجام دهند، فقط یکی از آن ها آن کار را در واقع انجام دهد (حداقل فقط یکی در یک زمان). این کار ممکن است نوشتن برخی از داده ها در یک سیستم ذخیره سازی مشترک، انجام برخی محاسبات، فراخوانی برخی از API های خارجی یا موارد مشابه باشد. از طرفی باید این نکته را در نظر داشت که ردیس یک دیتابیس single core است و در هر لحظه فقط میتواند یک کامند رو اجرا کند. ما میخواهیم از این ویژگی ردیس برای پیاده سازی تمرین زیر استفاده کنیم.

## تمرین ۲

فرض کنید یک سرویس داریم که اینستنس های مختلفی از آن در حال اجراست. یک درخواست مشابه به همه ی پادها ارسال میشود که دیتابیس رو آپدیت کنند، حال میخواهیم با استفاده از ردیس کاری کنیم که اگر پادها درخواست های مشابه ایی همزمان دریافت کردند فقط یکی از پاد کار را انجام دهد. با جست و جو وب راه حل مناسب رو پیدا کنید.

## ساختمان داده های دیگر

برای مقادیر value ها در redis میتوان از ساختمان داده های متنوعی استفاده کرد که ساده ترین ان ها string است. در ردیس ساختمان داده های پیشرفته تری نیز وجود دارند که از جمله ی ان ها میتوان به hash, sortedset, set, list اشاره کرد.

### مطالعه

برای آشنایی مقدماتی با ساختار داده های ردیس [این آدرس](#) را مطالعه کنید.

### مطالعه

برای مثال با استفاده از sortedset ها میتوان یک لیدر بورد بزرگ با سطر های زیاد ایجاد کرد و با سرعت بالا به دیتا هایش دسترسی داشت. برای مطالعه ی بیشتر [این لینک](#) رو مطالعه کنید.

و از طرفی می توان با استفاده از GeoSpatial به طور مثال دیتا های چندین هزار رستوران رو ذخیره کنیم و با سرعت بسیار بالایی به آن دسترسی داشته باشیم. برای مطالعه ی بیشتر میتوانید [این لینک](#) را مطالعه کنید.

## lua scripts & Pipeline , transactions

Pipeline، Transactions و Lua Script سه روش برای اجرای گروهی کامند ها در ردیس هستند. این کامند شباهت ها و تفاوت هایی دارند و هر کدام استفاده مخصوص به خود دارد.

### مطالعه

برای آشنایی مقدماتی با این مفاهیم [این مقاله](#) را مطالعه کنید.

### تمرین ۳

فرض کنید میخواهیم ۱۰۰۰ کاربر جدید در ثانیه در ردیس بسازیم. برای این کار اگر به صورت عادی این کار را انجام دهیم به ازای هر ریکوئست ۱ میلی ثانیه زمان صرف می شود و در کل این کار ۱۰s طول خواهد کشید اما اگر این کار را به یکباره و با transaction یا pipeline ها انجام دهیم زمان بسیار کمتری

صرف خواهد شد. به این صورت که این بار لیست کامند ها را به ردیس می دهیم و ردیس کامند ها را به ترتیب انجام میدهد. فرض کنید ساختار کاربران به این شکل است و سعی کنید کد این برنامه را پیاده کنید.

```
User-12: hash (  
    Name: Ali  
    LastName: Alavi  
)
```

### مطالعه

برای مطالعه ی این موضوع که چطور pipeline پرفورمنس رو بهتر میکند [این لینک](#) رو مشاهده کنید.

### مطالعه ی بیشتر

[Redis Security](#)