16 NOVEMBER 2017 / RAILS

# Use find_each instead of all.each in Rails

In Rails, sometimes we need to iterate over all the records from a model. To achieve this people prefer to use `all.each` on a model. This can lead to usage of memory if there are millions (huge number of) records in the table.

Let's say we have a model `User`. We will compare the usage of `all.each` and `find_each` to see which the differences and when to use `find_each` or `all.each`

## all.each

When we perform all.each on a model in Rails, it loads all the records from the table in memory and then iterates over those records.

```
User.all.each do |user|
  puts user.name
end
```

When we call `User.all`, it will query the database to fetch all the user records into memory. The query fired when we execute `User.all` is shown below.

```
User Load (1.2ms)  SELECT "users".* FROM "users"
```

This can lead to a lot of memory being used as there is no `limit` or `offset` is used when querying database.

## find_each

The find_each approach internally uses batches to query and get records in memory.

Here is the Rails source code for the find_each method.

```
def find_each(start: nil, finish: nil, batch_size: 1000, error_o
  if block_given?
    find_in_batches(start: start, finish: finish, batch_size: ba
      records.each { |record| yield record }
    end
  else
    enum_for(:find_each, start: start, finish: finish, batch_siz
      relation = self
      apply_limits(relation, start, finish).size
    end
  end
end
```

`find_each` method accepts an hash with following options.

- `batch_size` : Specifies the size of the batch. `batch_size` is defaulted to 1000
- `start` : This is a value of primary key to start records from for the query. Inclusive of the value.
- `finish` : This is a value of primary key to finish records from for the query. Inclusive of the value.

specify if an error should be raised when an order is present in the relation.

We can see that `find_each` internally makes use of `find_in_batches` which queries database in batches.

Even though we do not pass parameters it makes sure that query is done in batches of size 1000 (default `batch_size` )

Thus, even though we feel that the table has less number of records, we can make use of `find_each` as a good practice. `find_each` will take care of using `batches` if number of records are greater than `batch_size` . It also gives additional benefit of not hogging up application memory.

Feel free to comment on with your opinions.



**Akshay Mohite**
Hi there! I am a Rails Enthusiast, building some cool products at DTree Labs.

**Ashwin Vidiyala** • 2 years ago

Really good explanation. Thanks dude!

2 • Reply • Share ›

**Akshay Mohite** Mod Ashwin Vidiyala • 2 years ago

Welcome :)

— Ruby in Rails —

Rails

Rails routes difference between resource and resources

How to get integer value from enum in Rails?

Rails test jbuilder json response with rspec

See all 86 posts →

RUBY

## Safe navigation operator in Ruby

AKSHAY MOHITE 1 MIN READ

# Array comparison in ruby

1 MIN READ