

Introducing Mutator Layer in Rails

By Philip Lambok

Skinny Controller – Fat Model

Branch: master ▾ discourse / app / models / topic.rb


 **tgxworld** Annotate models.


44 contributors 

1557 lines (1298 sloc) | 50.2 KB

```
1 require_dependency 'slug'
```



Branch: master ▾ discourse / app / models / user.rb

 **eviltrout** FIX: Deleting Users should work nicely with Reviewable

71 contributors 

1490 lines (1211 sloc) | 43.9 KB

```
4 require_dependency 'base/has'
```

  github.com/discourse/discourse/blob/master/app/models/topic.rb

■ Personal

```
367 def self.for_digest(user, since, opts = nil)
368   opts = opts || {}
369   score = "#{ListController.best_period_for(since)}_score"
370
371   topics = Topic
```

Studi Kasus

PT MRT Jakarta

Nama Kereta

TX-007

*Fitur Pertama

Tambahkan Kereta

Ketentuan:

- Kereta yang di-input otomatis masuk ke daftar antrian.
- Simpan Log-nya penambahan keretanya.

Daftar Antrian Kereta

Urutan Kereta	Nama Kereta	Action
1	PX-223	hapus
2	IO-123	hapus
3	OO-251	hapus
4	IW-211	hapus

*Fitur Kedua

Ketentuan:

- (Penambahan) Jika ada kereta baru di-input maka langsung masuk ke posisi terakhir.
- (Penghapusan) Jika ada penghapusan antrian kereta, maka urutan antrian kereta sebelumnya akan naik & Simpan log penghapusan antriannya.



```
class Trains::QueueController
  def create
    TrainQueueService::Create.new(train).perform
  end
end

module TrainQueueService
  class Create
    def initialize(train)
      @train = train
    end
    def perform
      number = get_current_number
      Train::Queue.create!(number: number, train: @train)
      create_log(train)
    end

    private

    def get_current_number # last_number_train + 1

    def create_log(train) # Kereta dengan tipe X masuk antrian 2
    end
  end
end
```

Train

name kolom	tipe data
id	integer
name	string

Train::Queue

name kolom	tipe data
id	integer
number	integer
train_id	integer

Log

name kolom	tipe data
id	integer
description	string



```
class Trains::QueueController
  def destroy
    TrainQueueService::Destroy.new(train).perform
  end
end
```

```
module TrainQueueService
  class Destroy
    def initialize(train)
      @train = train
    end

    def perform
      decrease_number_behind_of(@train)
      destroy_train_of_queue(train)
      create_log(train)
    end

    private

    def decrease_number_behind_of(train)

    def destroy_train_of_queue(train)

    def create_log(train) #> "Kereta dengan tipe X terhapus dari antrian 3"
    end
  end
end
```

Nama Kereta	Action
TX-991	hapus
PI-112	hapus
IO-111	hapus
KN -821	hapus

*Fitur terakhir

Fitur untuk menghapus kereta.
Ketika kereta dihapus, antrian yang dikereta
juga ikut terhapus

Log yang ditulis:
Kereta PI-112 dihapus dari daftar kereta.

Output:

Log: Kereta dengan tipe PI-112 terhapus dari antrian 8

Log: kereta PI-112 terhapus dari daftar kereta

```
class TrainsController
  def destroy
    TrainService::Destroy.new(train).perform
  end
end
```

```
module TrainService
  class Destroy
    def new(train)
      @train = train
    end

    def perform
      TrainQueueService::Destroy.new(@train).perform
      _train = @train
      @train.destroy
      create_log(_train)
    end

    private

    def create_log(train)
      Log.create(description: "Kereta #{train.name} terhapus dari daftar kereta")
    end
  end
end
```



```
module TrainQueueService
  class Remove
    def initialize(train)
      @train = train
    end

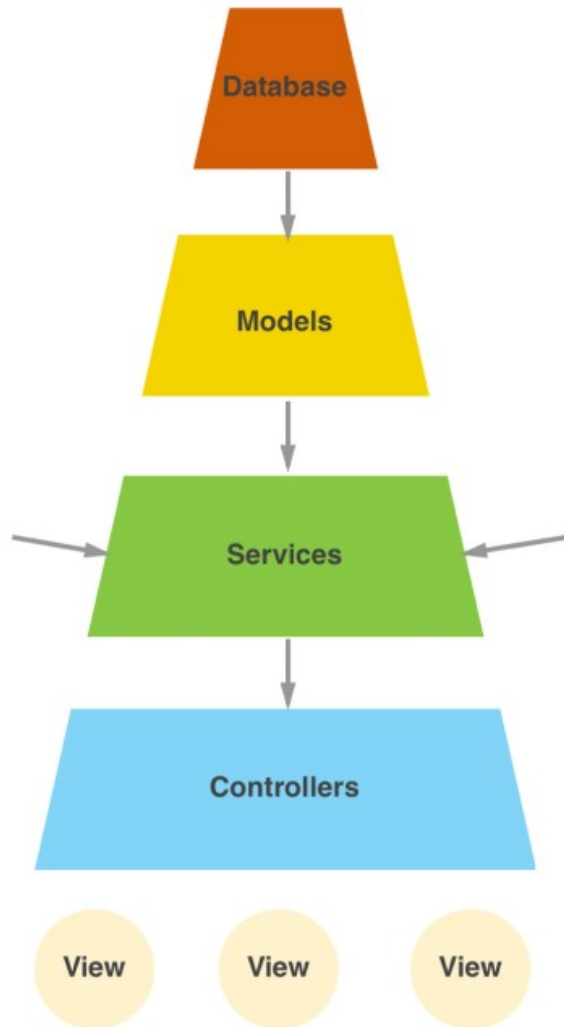
    def perform
      name_of_train = @train.name
      TrainQueueService::Destroy.new(name_of_train).perform
      log_desc = "Kereta dengan tipe #{name_of_train} terhapus dari antrian #{current_number}"
      Log.create!(description: log_desc)
    end
  end
end
```

```
module TrainQueueService
  class Destroy
    def initialize(train)
      @train = train
    end

    def perform
      name_of_train = @train.name
      queue = TrainQueue.find_by(train: @train)
      behind_the_queue = TrainQueue.find_behind_queue_of(queue)
      queue.destroy!
      behid_the_queue.each(&:increase_number!)
    end
  end
end
```

Output:

Log: kereta PI-112 terhapus dari daftar kereta



Skinny Controller
Skinny Model
Fat Service



```
module TrainQueueService
  class Destroy
    def initialize(train)
      @train = train
    end

    def perform
      _train = @train
      TrainQueueMutator.destroy(@train)
      create_log(_train)
    end

    private

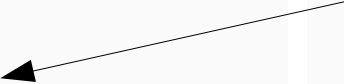
    def create_log(train)
    end
  end

  class TrainQueueMutator
    def self.destroy(train)
      name_of_train = @train.name
      queue = TrainQueue.find_by(train: @train)
      behind_the_queue = TrainQueue.find_behind_queue_of(queue)
      queue.destroy!
      behid_the_queue.each(&:increase_number!)
    end
  end
end
```

Kode	Scenario
Student.create	hanya menambahkan data user (1 perintah)
StudentMutator.create(student)	menambahkan data student + mengatur nomer induknya(before_create)

```
module TrainQueueService
  class Destroy
    def initialize(train)
      @train = train
    end

    def perform
      name_of_train = @train.name
      queue = TrainQueue.find_by(train: @train)
      behind_the_queue = TrainQueue.find_behind_queue_of(queue)
      queue.destroy!
      behid_the_queue.each(&:increase_number!)
    end
  end
end
```



```

class TrainQueueMutator
  def self.create(train)
    number = Train::Queue.find_last_number
    current_number = number + 1
    Train::Queue.create!(number: current_number, train: train)
  end
end

module TrainQueueService
  class Create
    def initialize(train)
      @train = train
    end
    def perform
      TrainQueueMutator.create(train)
      log_desc = "Kereta dengan tipe #{@train.name} masuk ke antrian"
      Log.create!(description: log_desc)
    end
  end
end

```

```

class Trains::QueueController
  def create
    TrainQueueService::Create.new(train).perform
  end
end

module TrainQueueService
  class Create
    def initialize(train)
      @train = train
    end
    def perform
      number = get_current_number
      Train::Queue.create!(number: number, train: @train)
      create_log(train)
    end

    private

    def get_current_number # last_number_train + 1

    def create_log(train) # Kereta dengan tipe X masuk antrian 2
    end
  end
end

```

Mutator (is. Replacement of Rails callback.)

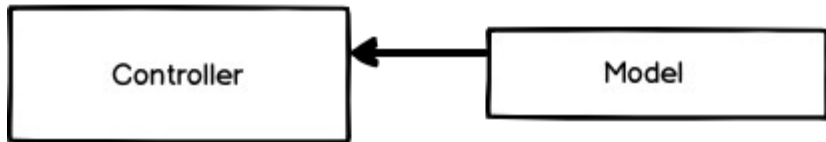
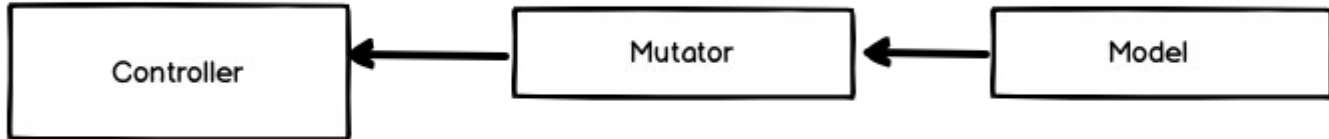
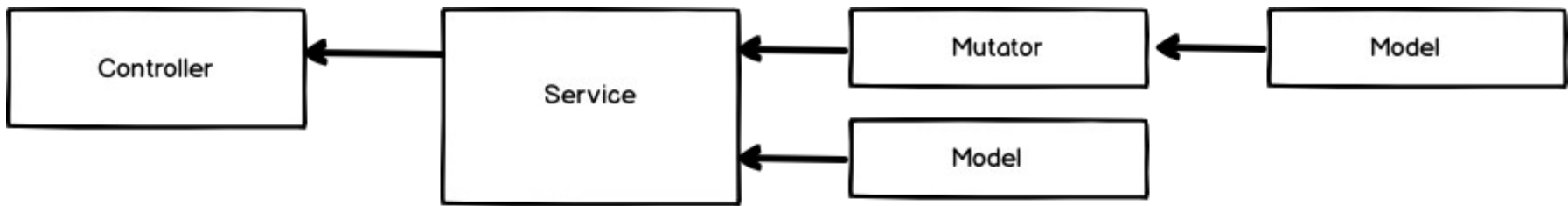
```
before_create do
  last_number = Train::Queue.last_number
  number = last_number + 1
end
```



```
class TrainQueueMutator
  def self.create(train)
    last_number = Train::Queue.last_number
    number = last_number + 1
    Train::Queue.create(number: number, train: train)
  end
end
```

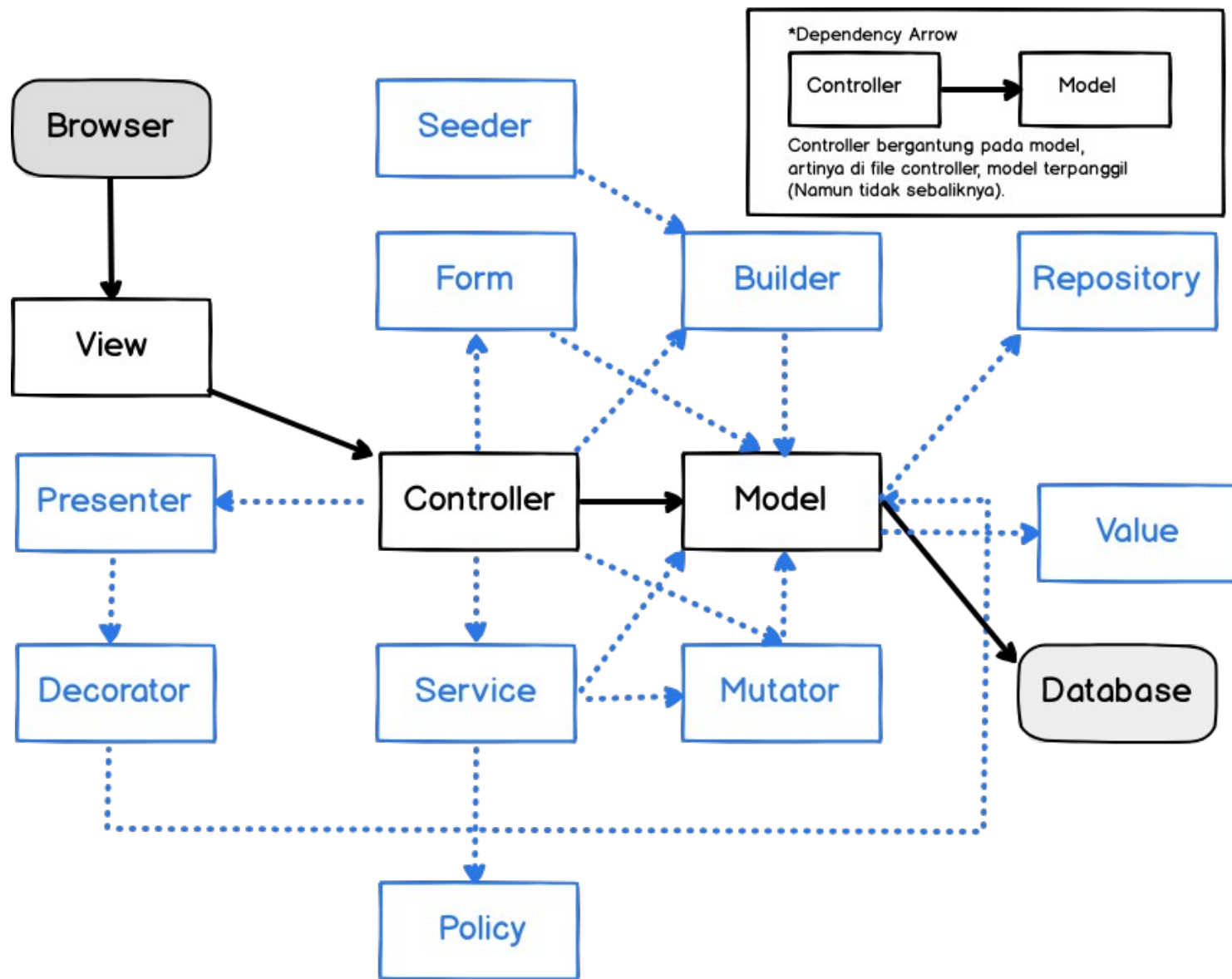


Action	Simple	Complex
Create	User.create	UserMutator.create(user)
Destroy	User.destroy	UserMutator.destroy(user)
Save	User.save	UserMutator.save(user)
Update	User.update	UserMutator.update(user)



Rules

- Service (High Level Application Logic)
 - Ketika logic memuat lebih dari satu model!
- Mutator (Low Level Application Logic)
 - Application Logic yang memuat satu model saja!
 - before_create = UserMutator.create(user)
 - before_destroy = UserMutator.destroy(user)
 - before_save = UserMutator.save(user)
 - ...



```
├─ app
├─ └─ builders
├─ └─ controllers
├─ └─ decorators
├─ └─ forms
├─ └─ models
├─ └─ mutators
├─ └─ policies
├─ └─ presenters
├─ └─ repositories
├─ └─ services
├─ └─ values
├─ └─ views
```

Thanks!

