

Rails and delegate

For example I have a QueueItem and a Video model.

```
QueueItem < ActiveRecord::Base
  belongs_to :video
end
Video < ActiveRecord::Base
  has_many :queue_items
  belongs_to :category
end
```

If I want to get the category object of the video in first queue_item, I may write

```
queue_item = QueueItem.first
queue_item.video.category.name
#=> "Action"
```

Instead of getting the object via model association, we can use delegate to help us.

```
class QueueItem < ActiveRecord::Base
  belongs_to :video

  delegate :category, to: :video
end
```

Then we can get the category by

```
queue_item = QueueItem.first
queue_item.category.name
#=> "Action"
```

Or even you can set a category_name method in the QueueItem model.

```
class QueueItem < ActiveRecord::Base
  belongs_to :video

  delegate :category, to: :video
  def category_name
    category.name
  end
end
```

```
queue_item = QueueItem.first
queue_item.category_name
#=> "Action"
```

You can set one or more method names (specified as symbols or strings) if you want. And the name of the target object via the `:to` option(also a symbol or string).

There are some options you can use in delegation `:to` — Specifies the target object `:prefix` — Prefixes the new method with the target name or a custom prefix `:allow_nil` — if set to true, prevents a `NoMethodError` to be raised

You can use the `prefix` option to make the method more readable.

```
class QueueItem < ActiveRecord::Base
  belongs_to :video
  delegate :category, to: :video
  delegate :title, to: :video, prefix: :video
end

queue_item = QueueItem.first
queue_item.video_title == queue_item.video.title
# => true
```