

RAILS ORM

de-mystifying ActiveRecord

BlazingCloud™

NoteToSelf: Know your audience

- Who has used an ORM?
- Daily:
 - Who uses Relational Databases?
 - Who uses Java,Erlang,ObjectiveC?
 - Who uses PHP, Python?
 - Who uses Javascript?
 - Who uses [Perl, Smalltalk, Lisp, Ruby,]

BlazingCloud™

NTS: Address a Need

What I want =>

- How can I do my job smoother, faster, and to a higher standard?

What I need first =>

- Where can I can get a 'run-time' model of rails?

BlazingCloud™

NTS: Fulfill need

- What does Rails do for you?
- What do you need to do for Rails?

Rails is Ruby,

Ruby is Awesome,

Transitive Relation ,

Rails is Awesome.

BlazingCloud™

What you are about to see

- Papa Bears Honey Sharing Co-Operative

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized sky or water with wavy lines.

Papa Bear Honey Share

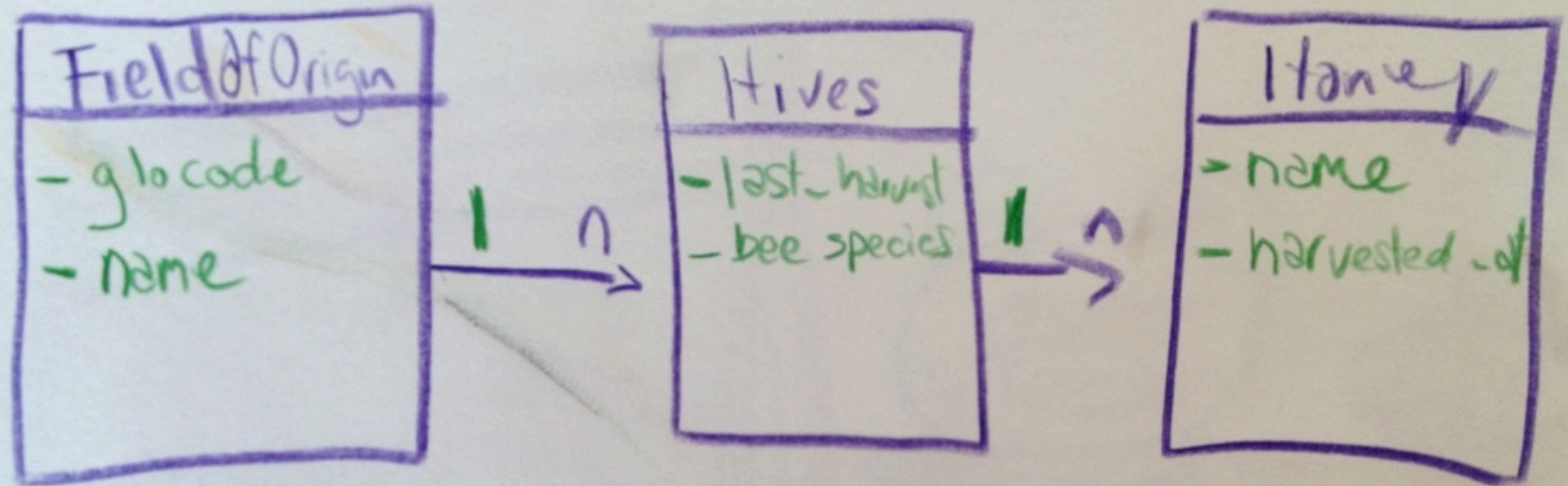


What you are about to see

- (2) one-to-many relationships

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

Internal Data Model



Internal

EXTERNAL

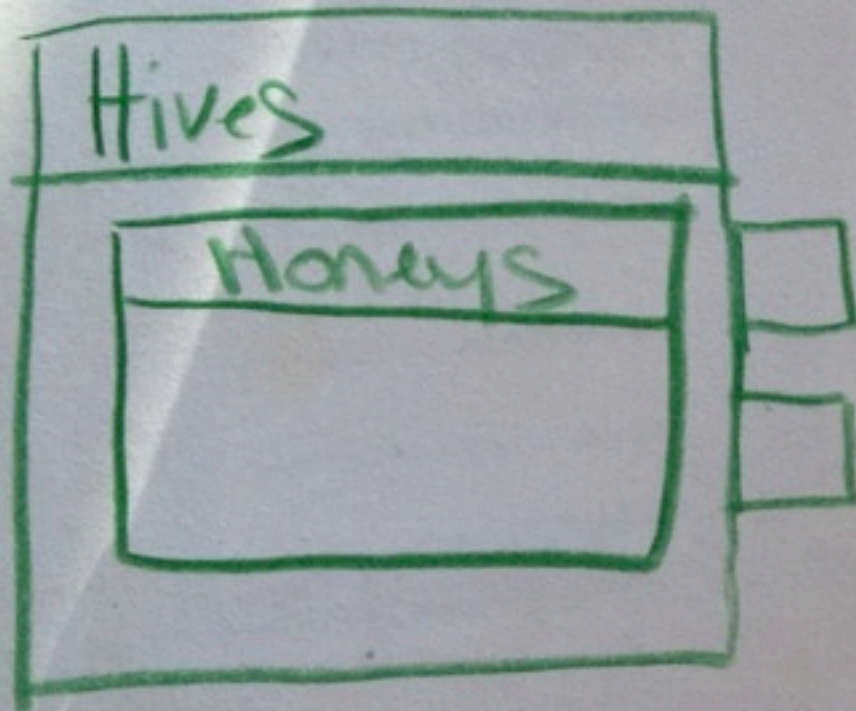
What you are about to see

- External Resource Scheme

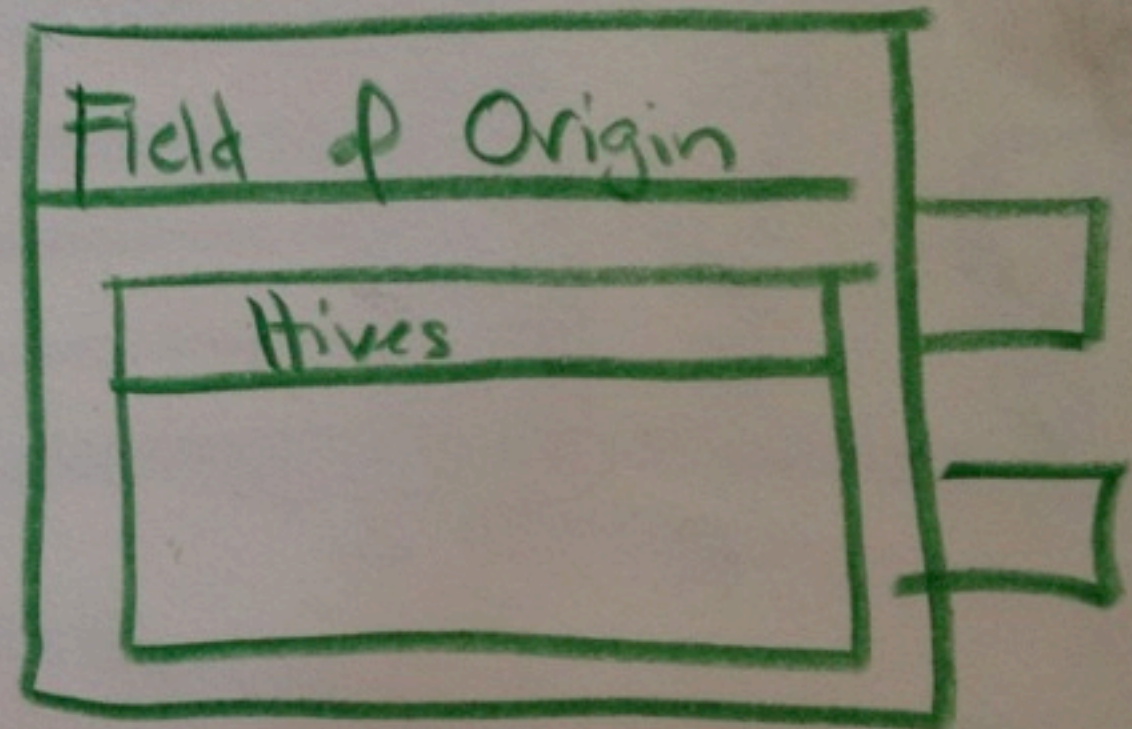
The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

External REST Resource API

EXTERNAL



/hives/:id/honeys
/hives/



/field-of-origins/:id/hives
/field-of-origins/

git@github.com:blazingcloud/papabear.git

BlazingCloud™

What you are about to see

- database model implies methods

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

`rake db:reset`

BlazingCloud™

rails db

BlazingCloud™


```
PRAGMA table_info(hives);
```

BlazingCloud™

```
module ActiveRecord
  module ConnectionAdapters
    class SQLiteAdapter < AbstractAdapter

def table_structure(table_name)
  structure = exec_query("PRAGMA
table_info("#{quote_table_name(table_name)}"),
'SHEMA').to_hash
  raise(ActiveRecord::StatementInvalid, "Could not
find table '#{table_name}'") if structure.empty?
  structure
end
```

https://github.com/rails/rails/blob/master/activerecord/lib/active_record/connection_adapters/sqlite_adapter.rb

BlazingCloud™

for all table attributes
define a method
to change attribute
to read attribute

story!

BlazingCloud™

What you are about to see

- Inserting a bunch of pre declared data via External API
- Hand waving

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave.

rails server &

BlazingCloud™

```
cat curl.txt  
sh curl.txt
```

BlazingCloud™

<http://localhost:3000/>

BlazingCloud™

What you are about to see

- 'has_many' declaring an association to 'honeys'

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

app/models/hive.rb

```
class Hive < ActiveRecord::Base  
  belongs_to :field_of_origin  
  has_many :honeys # product  
end
```

BlazingCloud™

activerecord/lib/active_record/associations.rb

```
module ActiveRecord  
  module Associations
```

```
    def has_many(name, options = {}, &extension)  
      Builder::HasMany.build(self, name, options, &extension)  
    end
```

https://github.com/rails/rails/blob/master/activerecord/lib/active_record/associations.rb

BlazingCloud™

[activerecord/lib/active_record/associations/builder/association.rb](https://github.com/rails/rails/blob/master/activerecord/lib/active_record/associations/builder/association.rb)

```
module ActiveRecord::Associations::Builder
  class Association

    def self.build(model, name, options)
      new(model, name, options).build
    end

    def define_readers
      name = self.name
      mixin.redefine_method(name) do |*params|
        association(name).reader(*params)
      end
    end
  end
end
```

https://github.com/rails/rails/blob/master/activerecord/lib/active_record/associations/builder/association.rb

BlazingCloud™

What you are about to see

- ‘honeys’ reader method
- ‘has_many’ builds methods like ‘create!’ on the reader association

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

app/controllers/honeys_controller.rb

```
class HoneysController < ApplicationController
  def create
    current_hive = Hive.find(params[:hive_id])
    if current_hive
      current_hive.honeys.create!(params[:honey])
      render :json => current_hive.to_json(:include => :honeys)
    else
      head :error
    end
  end
end
```

BlazingCloud™

```
module ActiveRecord
  module Associations
    class CollectionAssociation < Association

      def create!(attributes = {}, options = {}, &block)
        create_record(attributes, options, true, &block)
      end
    end
  end
end
```

https://github.com/rails/rails/blob/master/activerecord/lib/active_record/associations/collection_association.rb

BlazingCloud™

SEE CURL POST

```
curl -d 'honey[name]=Cinnamon%20Manuka%20&honey[volume_in_ml]=2520' http://localhost:3000/hives/1/honeys
```

BlazingCloud™

What does Rails ORM do for me?

- Keeps me focused on describing the domain
- Keeps me from re-expressing my data model in query/strings/logic/controllers/libraries
- Keeps me from having to write generic code - focus on what is specific.

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

What do I need to do for Rails ORM?

- learn Rails conventions
- learn Rails limits
- express my design in it's language

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or cloud.

Where can I get a Realtime / Runtime understanding of what Rails is doing?

- practice
- errors
- reflection
- rinse & repeat

The logo for BlazingCloud, featuring the text "BlazingCloud" in a white, sans-serif font with a trademark symbol, set against a light blue background that resembles a stylized wave or water surface.

quest-ions?

BlazingCloud™