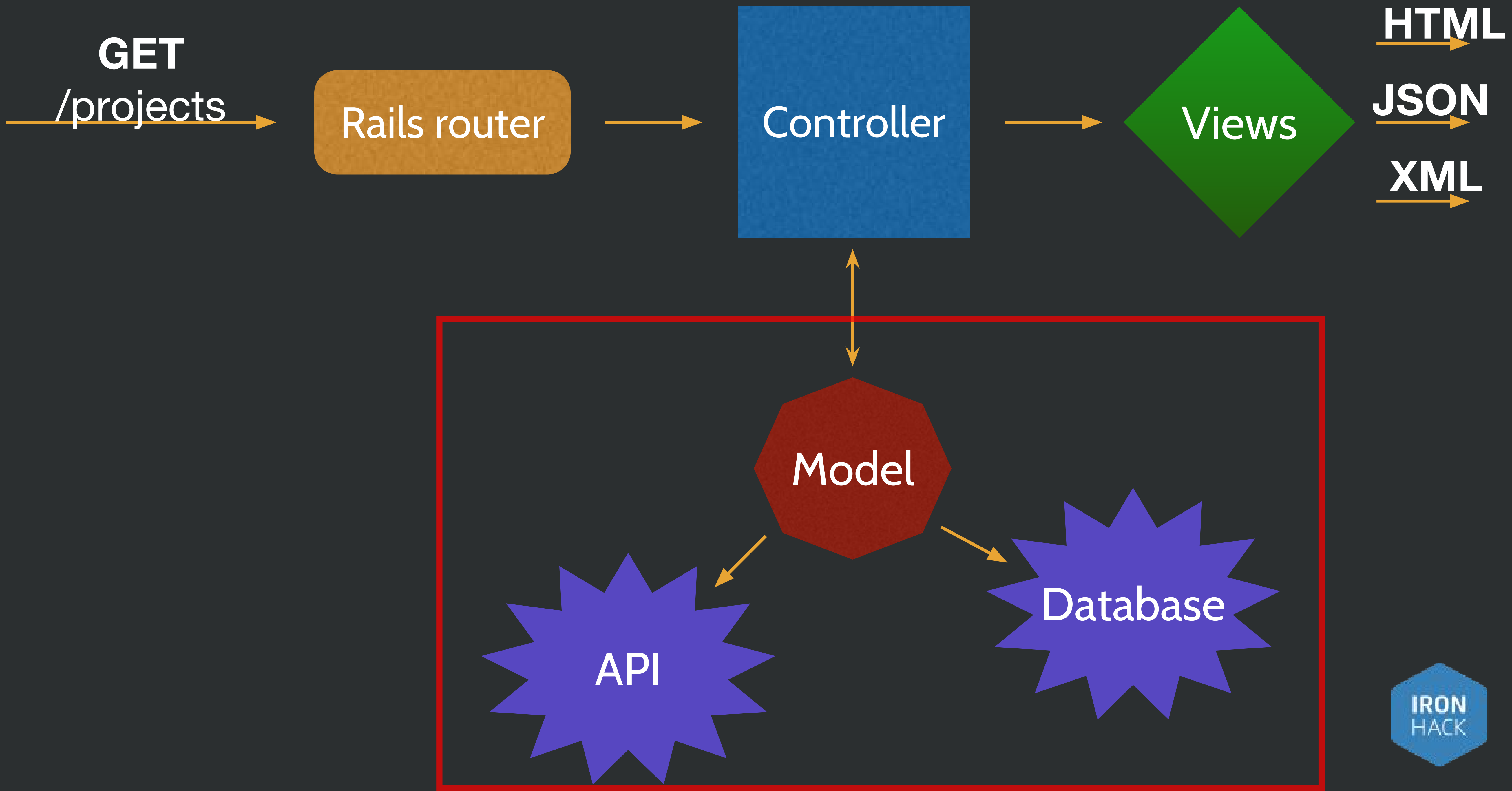
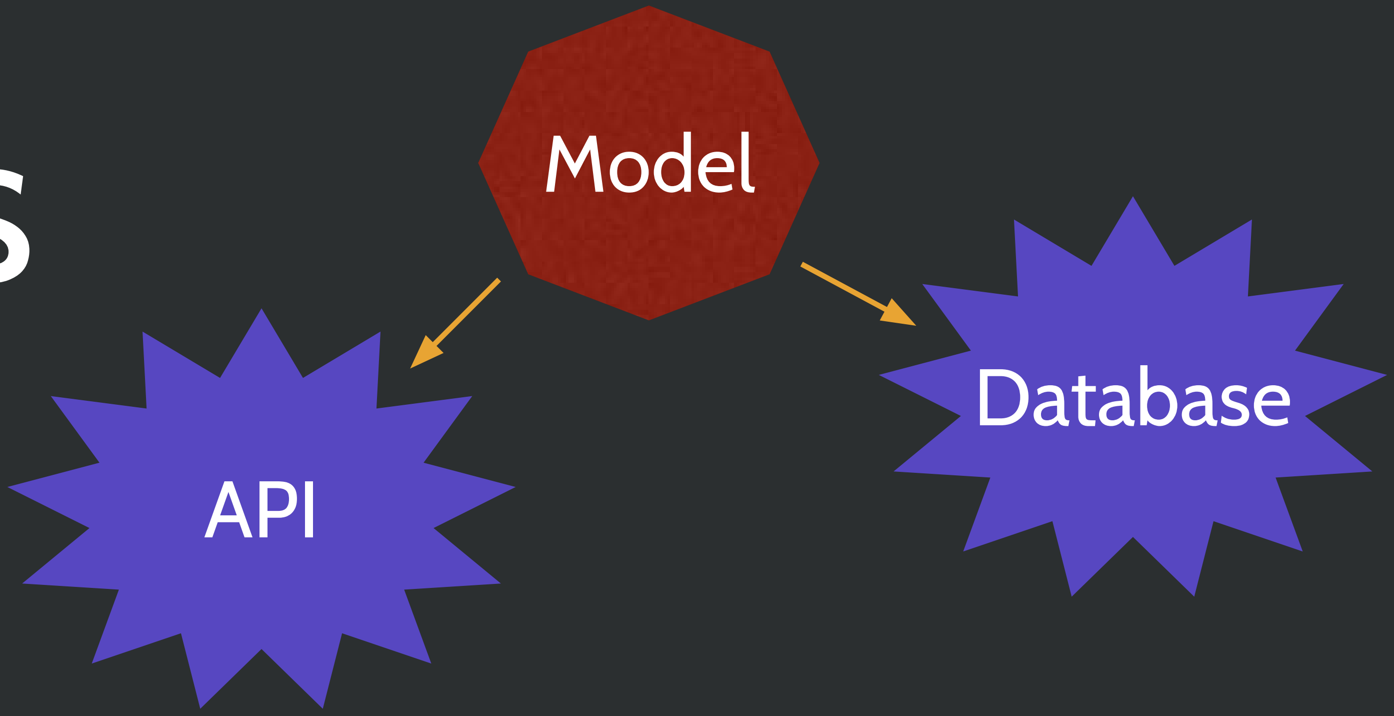
Introducing the models

MVC



IRON HACK

MVC: Models



1. Deal with data
2. Implement business logic rules

Deal with data

- CRUD data from a data source
    - Database
    - API
    - Filesystem
- Ensure data format is valid
    - Implement validations
        - *length of a name is shorter than 50 chars*

Implement business logic rules

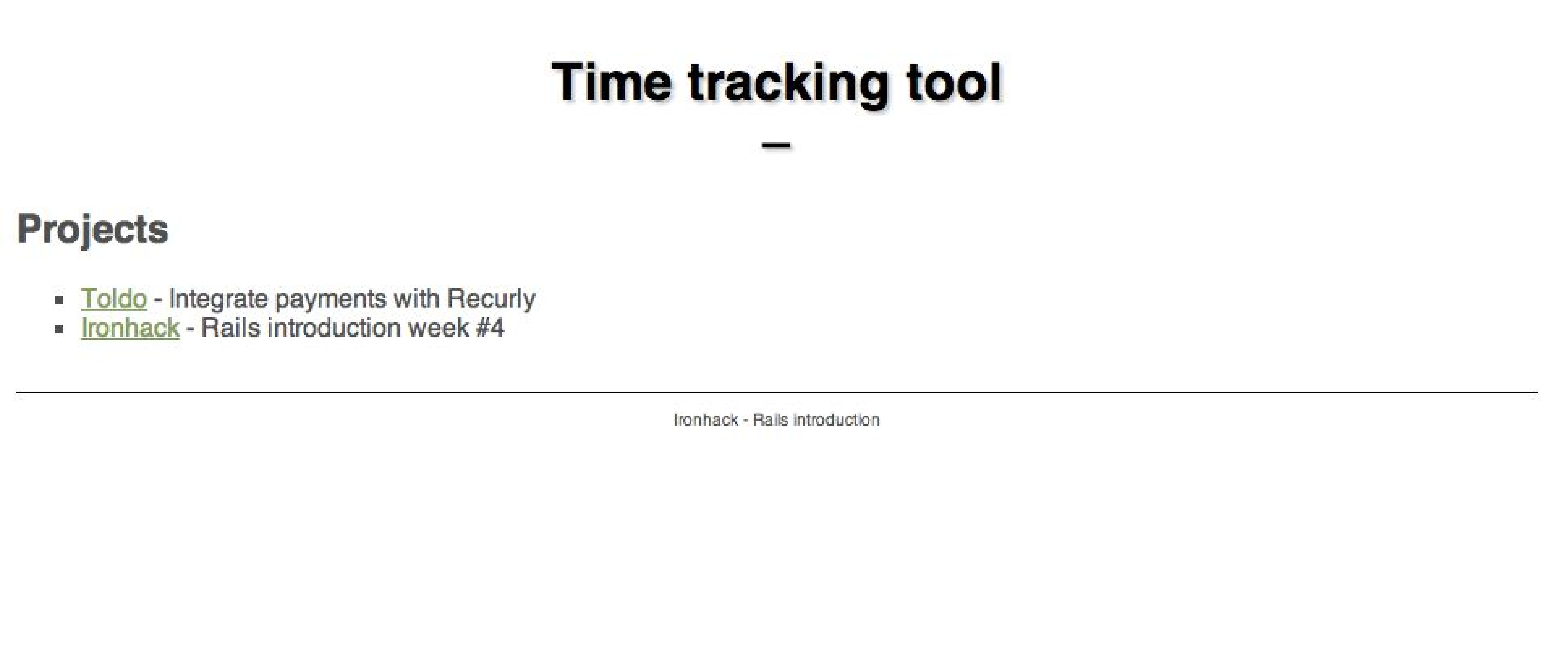
Calculate the letter of the Spanish ID (NIF)

Calculate the VAT of a shopping cart

Generate reports

Single responsibility principle: an object should have only a single responsibility

Back to business: a list with the latest projects



Time tracking tool

–

Projects

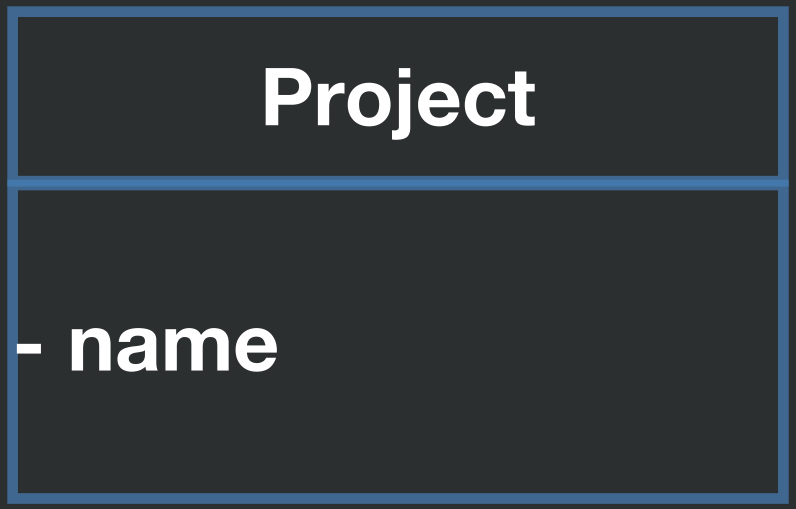
- Toldo - Integrate payments with Recurly
- Ironhack - Rails introduction week #4

Ironhack - Rails introduction

Model: Project

Stored in the database

Only has one attribute: a name (at least for now)



Project

- name

Model: Project

```
$ rails g model Project name:string
```

```
invoke  active_record
create    db/migrate/20140126095033_create_projects.rb
create    app/models/project.rb
```

Model: Project

app/models/project.rb

```
class Project < ActiveRecord::Base
end
```

ActiveRecord

ORM included in Ruby on Rails

Maps Ruby objects to the database

- Classes are mapped to tables
- Objects are mapped to rows

In our example:

- Class Project is mapped to table projects
- Project objects are mapped to rows in table projects

Remember:
convention over
configuration

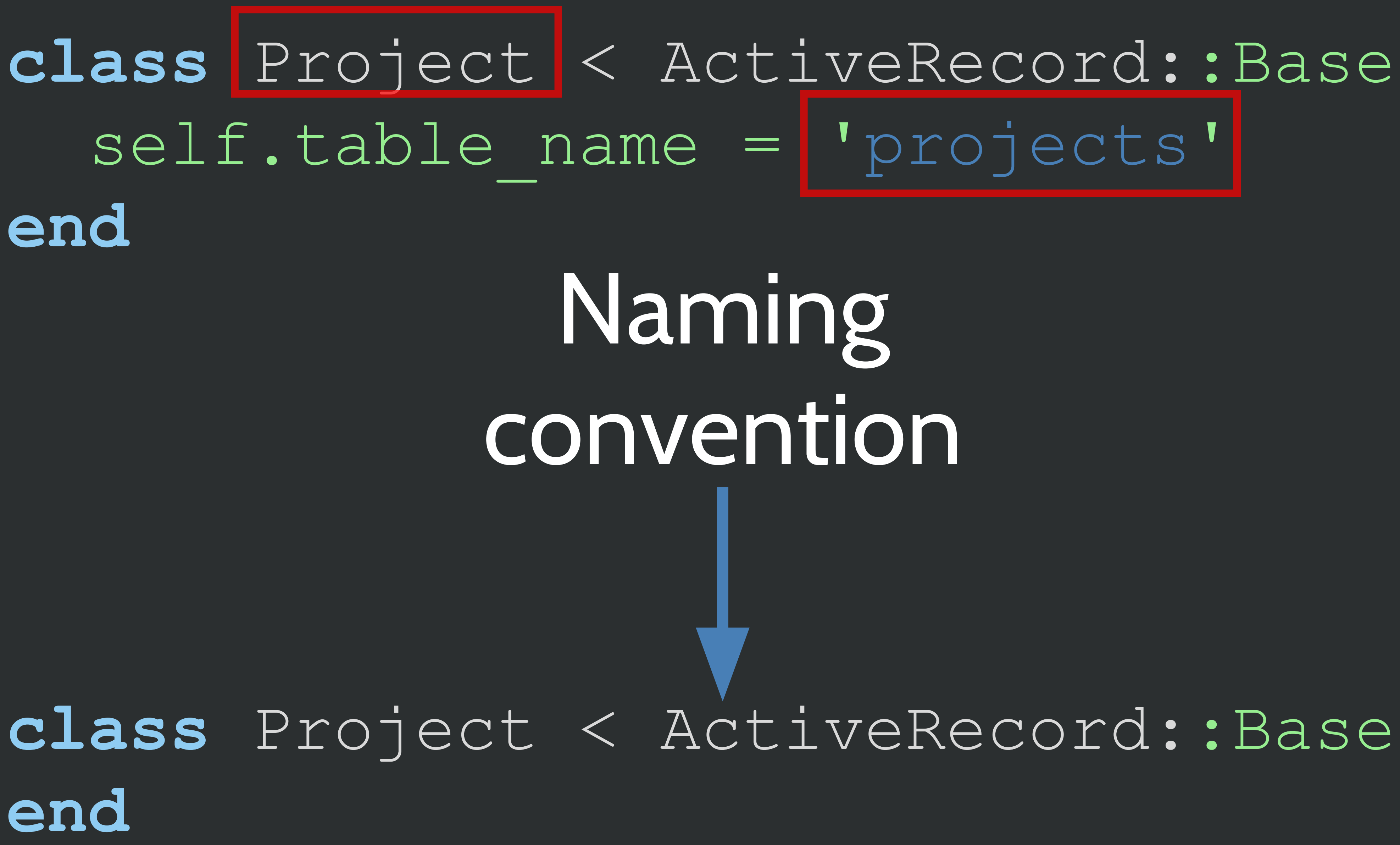
Model: Project

app/models/project.rb

```ruby
class Project < ActiveRecord::Base
end
```

```ruby
class Project < ActiveRecord::Base
  self.table_name = 'projects'
end
```

Model: Project



```
class Project < ActiveRecord::Base
  self.table_name = 'projects'
end
```

Naming convention

```
class Project < ActiveRecord::Base
end
```

Model: Project

db/migrate/xxxxxxx_create_projects.rb

```ruby
class CreateProjects < ActiveRecord::Migration
  def change
    create_table :projects do |t|
      t.string :name
      t.timestamps
    end
  end
end
```

Model: Project

```
$ rake db:migrate
```

```
==  CreateProjects: migrating ==========
-- create_table(:projects)
   -> 0.0302s
==  CreateProjects: migrated (0.0303s)
========================================
```

In schema_migrations rails keeps the last migration run so it can keep track of pending migrations

ActiveRecord Conventions

Tables primary key is named id

Timestamp attributes:

- created_at
- updated_at

Values filled automatically by ActiveRe

Isn’t supposed to be updated by your c

Opinionated!

Migrations

Migrations are a feature of Active Record that allows you to evolve your database schema over time.

ActiveRecord Migrations

Create table: create_table

```
create_table :table_name
```

Create column: add_column

```
add_column :table_name, :column_name, :column_type
```

Remove column: remove_column

```
remove_column :table_name, :column_name
```

Creating a migration

Let's create a new column called ***description***, of type ***text***

Creating a migration

The easiest way to do it:

```
$ rails g migration
    add_description_to_projects
    description:text
```

Check the created migration file, and then run rake db:migrate

Remember to commit the changes in git