



Create and Read with
Models

But before...

Rails
console!

\$ rails console

Model: Project

Project
<ul style="list-style-type: none">- name- description

Create

Create

```
> my_project = Project.new  
> my_project.name = "Ironhack"  
> my_project.description = "Description"  
> my_project.save
```

Create

```
> my_project = Project.new(name: 'Ironhack',  
  description: 'Description')  
> my_project.save
```

Create

```
> my_project = Project.create(  
    name: "Ironhack",  
    description: "Description")
```

Exercise

Create 3 projects using the different methods we've seen so far.

Check the value of the attribute id and the timestamps.

Exercise

Check the value of the attribute `id` and the timestamps on a new instance:

```
> p = Project.new  
> p.id  
> p.created_at
```

Read

Retrieve a single object

Retrieve by attribute

::find_by

```
> Project.find_by(name: "Ironhack")
```

When there are no
elements, methods
return `nil`.

Retrieve a collection

Retrieve all

::all

```
> projects = Project.all
```

Retrieve by conditions

#where

As a string

```
> Project.where("name='Ironhack'")
```

#where

As a hash

```
> Project.where(name: "Ironhack")
```

#where

As a string with placeholders

```
> Project.where("name=?", 'Ironhack')
```

(Protects against SQL injection)

https://imgs.xkcd.com/comics/exploits_of_a_mom.png

#where

Chaining

```
> Project.where(name: "Ironhack")  
  .where("created_at < ?",  
        Date.yesterday)
```

#where

Negation

```
> Project.where.not("id < ?", 4)
```

::find_by vs ::where

Use `find_by` when looking for one object.

```
> Project.find_by(id: 1)
```

```
# => #<Project id: 1, name: "Finish my Personal  
Website", created_at: "2016-03-09 18:49:59",  
updated_at: "2016-03-09 18:49:59", description: "Been  
putting this off forever, need to finish my p...">
```

::find_by vs ::where

Use `where` when looking for multiple objects.

```
> Project.where("id > 1")
```

```
=> #<ActiveRecord::Relation [#<Project id: 2, name:
"TurfSurf Project", created_at: "2016-03-09 18:49:
59", updated_at: "2016-03-09 18:49:59", description:
"Need to make the site production ready and get
the...">, ... ]
```

::find_by vs ::where

`Where` returns an `ActiveRecord::Relation`. For all intents and purposes, this acts like an array, but we can use any `ActiveRecord` method on this special array (`where`, `order`, etc).

::find_by vs ::where

It is going to be array-like *even if there is only one record that matches the where clause.*

::find_by vs ::where

`find_by` returns either **one** object, or nil.

Retrieve with order

#order

- > `Project.order("created_at DESC")`
- > `Project.order("name ASC")`

Active support dates

```
d = Date.current
```

```
# => Mon, 14 Mar 2016
```

```
d + 1.year
```

```
# => Tue, 14 Mar 2017
```

```
d - 3.hours
```

```
# => Mon, 14 Mar 2016 03:00:00 UTC +00:00
```



Write The Following Queries

1. Get all of the projects that aren't named Ironhack.
2. Get the projects created in the last month ordered by name alphabetically
3. Get the projects updated yesterday
4. Get **one** project with the ID of 3

Solutions

1

```
Project.where.not(name: "Ironhack")
```

2

```
date = Time.now.midnight - 1.month  
  
projects = Project.where("created_at > ?", date)  
              .order("name ASC")
```


3

```
from = Time.now.midnight - 1.day
```

```
to    = Time.now.midnight
```

```
Project.where(updated_at: from..to)
```

4

```
Project.find_by(id: 3)
```

Same result?

```
Project.order("name ASC")  
  .where("created_at > ?", date)
```

```
Project.where("created_at > ?", date)  
  .order("name ASC")
```

Add `.to_sql` and compare the queries!



Remember to commit
your changes in git