Gisele Monarrez

Web App/Pilgrim

Blog Post- Active Records

4.25.16

Alright, first let’s talk about what ActiveRecords Associations really is. Is it a real association? Although the name makes it sound like a real association, it is not. Instead what it is, is it is a class method used in ruby. The purpose of ActiveRecords Associations is to tie objects together by using foreign keys, making associations. The Rails Association supports six types of associations:

* belongs\_to
* has\_one
* has\_many
* has\_many :through
* has\_one :through
* has\_and\_belongs\_to\_many

The way that the associations are expressed are through association symbols or options hash. Here is an example of one if you are curious as to what it looks like:

**class Project < ActiveRecord::Base**

belongs\_to :portfolio

has\_one :project\_manager

has\_many :milestones

has\_and\_belongs\_to\_many :categories

**end**

The class is “Project” and the lines that follow are expressing all the different associations the project has to other things. It is stating that project belongs to the portfolio, it has one project manager, has many milestones, and belongs to many categories. The beautiful thing about ActiveRecords Associations is that the code is not only easy, but the user can create a vast number of orders. If the user decides they want to delete orders but don’t want to go through the hassle of deleting each one one by one. They can make a relationship saying that all those orders belong to *blank* user and then deleting them all from that one user. This is just like what we were doing in class when we were drawing the different relationships in school. ActiveRecords Associations allows us to show what we did in class through code; instead of drawing lines to connect the boxes that were a representation of the different relationships, we instead now have the Rails Associations to do that for us. ActiveRecords Associations is very helpful and can help make life that much easier for coders globally.

Word Count: 315

<http://apidock.com/rails/ActiveRecord/Associations/ClassMethods/belongs_to>

<http://api.rubyonrails.org/classes/ActiveRecord/Associations/ClassMethods.html>

<http://guides.rubyonrails.org/association_basics.html>