test "can set the genre" do

genre = FactoryGirl.build (:genre)

book = FactoryGirl.build(:book, :genre => genre)

assert\_equal genre, book.genre

end

test "setting a genre that doesn't exist makes the factory create it" do

book = FactoryGirl.build(:book, :genre\_name => "History")

assert\_equal "History", book.genre.name

assert Genre.exists?(:name => "History")

end

FactoryGirl.define do

factory :book do

title "MyString"

transient do

genre\_name "No Genre"

end

genre do

Genre.find\_by(name: genre\_name) || FactoryGirl.create(:genre, name: genre\_name)

end

end

end

test "create checked out" do

book = FactoryGirl.build(:book, :out, :title=> "my book")

assert\_equal "my book", book.title

assert book.checked\_out

end

# This factory lets you create a book. It uses traits so that you can specify if it is

# current checked out or not

#

# Note, that this factory defaults the title to "MyString" but you can override that by adding

# title: "Your title" to your creation statement

#

# Example:

# FactoryGirl.create :book, :out, title: "Your title"

#

# FactoryGirl.create :book, :in, :title => "Your title"

#

FactoryGirl.define do

factory :book do

title "MyString"

# this trait marks the book as being in the library

trait :out do

checked\_out true

end

#this trait marks the book as out of the library with a patron

trait :in do

checked\_out false

end

end

end

test "gets many floors" do

genre = FactoryGirl.create :genre, :with\_floors, :number\_of\_floors => 3

assert\_equal 3, genre.floors.size

end

factory :genre do

name "one genre'"

end

trait :with\_floors do

transient do

number\_of\_floors 5

end

# after we create the genre, we get it and an evaluator that knows about

# our transient variables

after(:create) do |genre, evaluator|

# create a list using the :floor factory, our evaluator can tell us how big the list

# should be and we want the genres on that floor to be a list containing only this genre

# That will create the entries in the many-to-many table so that we see this genre on

# those floors

create\_list(:floor, evaluator.number\_of\_floors, :genres => [genre])

end

end