

# Week 10

The Last Week (Chapter 25)

# Week 10

- shoulda
- paperclip
- sending emails
- delayed job
- security!

# Shoulda

tests made easier

```
class TestThing < Test::Unit
  context "a user" do
    setup do
      @user = User.find(:first)
    end

    should "return first name" do
      assert_equal "joe", @user.first_name
    end
  end
end
```

# Nested Contexts

```
class TestThing < Test::Unit
  context "a user" do

    ....

    context "with a profile" do
      setup do
        @user.profile = Profile.find(:first)
      end

      should "be awesome" do
        assert @user.awesome?
      end
    end
  end
end
```

# Macros

ahhhhhh so easy

```
class PostTest < Test::Unit::TestCase
  should_belong_to :user
  should_have_many :tags, :through => :taggings

  should_require_unique_attributes :title
  should_require_attributes :body, :message => /wtf/
  should_require_attributes :title
  should_only_allow_numeric_values_for :user_id
end
```

```
context "on GET to :show for first record" do
  setup do
    get :show, :id => 1
  end

  should_assign_to :user
  should_respond_with :success
  should_render_template :show
  should_not_set_the_flash

  should "do something else really cool" do
    assert_equal 1, assigns(:user).id
  end
end
```

# Paperclip

Image Attachments

# Sending Emails

I LOVE EMAILS! THEY ARE THE BEST

```
config.action_mailer.delivery_method = :smtp  
config.action_mailer.delivery_method = :sendmail  
config.action_mailer.delivery_method = :test
```

```
ruby script/generate mailer AwesomeMailer
```

```
class AwesomeMailer < ActionMailer::Base
  def mail_to_client account
    subject 'Video Professor'
    recipients account.email
    from 'video_prof'
    body :greeting => 'try my product'
  end
end
```

# Special Methods

- `AwesomeMailer#create_mail_to_client`
- `AwesomeMailer#deliver_mail_to_client`

# Content Types

- `mail_to_client.text.html.erb`
- `mail_to_client.text.plain.erb`

# Delayed Job

Do It Later

[http://github.com/tobi/delayed\\_job/tree/master](http://github.com/tobi/delayed_job/tree/master)

```
create_table :delayed_jobs, :force => true do |table|
  table.integer :priority, :default => 0
  table.integer :attempts, :default => 0
  table.text :handler
  table.string :last_error
  table.datetime :run_at
  table.datetime :locked_at
  table.datetime :failed_at
  table.string :locked_by
  table.timestamps
end
```

```
class SpamJob < Struct.new(:account_id)
  def perform
    account = Account.find(account_id)
    AwesomeMailer.deliver_mail_to_client(account)
  end
end
```

```
Delayed::Job.enqueue(  
  SpamJob.new(  
    Account.find(:first).id  
  )  
)
```

rake jobs:work

# Security

mass assignment is bad

```
create_table :users do |t|  
  t.name  
  t.email  
  t.account_id  
end
```

```
@user.attributes = { ... }
```

attr\_accessible

```
ActiveRecord::Base.send :attr_accessible, nil
```

```
class User < ActiveRecord::Base
  attr_accessible :name, :email
end
```

Show And Tell

Final Words

# Keep These Handy

- <http://api.rubyonrails.org/>
- <http://google.com/>
- gem server

Questions?