Week 8

CSPP 52553: Web Development
Winter 2013

- Gems (Pagination Example)
- Deployment
- Search Functionality
- Model Callbacks
- Sending Email

RubyGems

A gem is a versioned package of Ruby files.

The gem command lets you add and remove gems from your computer.

RubyGems

gem list
gem install
gem uninstall

You might need ***sudo*** *in front of each.*

RubyGems

For Rails projects, use the Gemfile to manage the gems required by your application

instead of installing them manually.

Deployment Demo: Heroku

Heroku Account
SSH keys
Local git repository

```
bundle install --without production
heroku run rake db:migrate
```

Search Functionality

Three primary ways to provide query conditions:

1. where column: value

2. where "SQL Fragment"

3. where "SQL with ? Placeholders", args...

Search Functionality

Three primary ways to provide query conditions:

1. `where(column: value)`

2. `where("SQL Fragment")`

3. `where("SQL with ? Placeholders", args...)`

Search Functionality

Use LIKE and % signs as wildcards inside of SQL fragments:

```
where("name LIKE 'M%'")
```

Security: use ? placeholders

```
where("name LIKE 'M%'")

where("name LIKE ?", "M%")
```

Model Callbacks

Model Lifecycle

Models can be *validated, saved, loaded,* and *deleted*.

Model Lifecycle

Models can be *validated, saved, loaded,* and *deleted*.

Note: there are actually two "saved" scenarios:

1. *Create*. An object is brand new, and will be inserted into the database for the first time.

2. *Update*. An object is already associated to a row in the database, and the object data will update the row data.

Model Callbacks

Callbacks are methods that execute at certain moments of an object's life cycle.

With callbacks it is possible to write code that will run whenever an Active Record object is created, saved, updated, deleted, validated, or loaded from the database.

- Rails Guides

Model Callbacks

Callbacks are methods that execute

whenever an Active Record object is created, saved, updated, deleted, validated, or loaded from the database.

Model Callbacks

create	***update***	***delete***
before_validation		
before_save		
before_create		
INSERT ...		
after_create		
after_save		

Model Callbacks

create	***update***	***delete***
before_validation	before_validation	
before_save	before_save	
before_create	before_update	
INSERT ...	*UPDATE ...*	
after_create	after_save	
after_save		

Model Callbacks

create	***update***	***delete***
before_validation	before_validation	
before_save	before_save	
before_create	before_update	
INSERT …	*UPDATE …*	
after_create	after_save	
after_save		

Model Callbacks

create

before_validation

before_save

before_create

INSERT ...

after_create

after_save

update

before_validation

before_save

before_update

UPDATE ...

after_save

delete

before_destroy

DELETE ...

after_destroy

Sending Email

Sending Email

Mailers are how we send email from a Rails app.

- Plain text
- HTML
- Multipart Emails
- Attachments

Sending Email

Mailers are like controllers.

- Each method sends a specific message.
- They create instance variables for their views.

Sending Email

```
class MyMailer < ActionMailer::Base

  def welcome(recipient)
    @account = recipient

    mail(:to => recipient.email_address,
         :subject => "Thanks for Signing Up")
  end

end
```

Sending Email

```
rails generate mailer
     ReservationMailer confirm canceled
```