

Autentifikācija un autorizācija

Termini

- Autentifikācija
Pārliedzināties, ka lietotājs ir tas par ko uzdodas
- Autorizācija
Atļaujam lietotājam veikt darbības saskaņā ar definētajiem likumiem

Autentifikācijas metodes

- IP adrese
- Lietotājs / parole
- Uzticama trešā puse
- Dažādi faktori – “ko es zinu”, “kas man ir”, “kas es esmu”, “kur es esmu”

Autentifikācijas risinājumi

Veids	Risinājums
Lietotājs / parole	http_authentication restful_authentication (acts_as_authenticated)
Trešā puse	open_id_authentication simple_ldap_authenticator acts_as_google_account ...

restful_authentication demo

Nestandarta piemēri

Vajadzība	Risinājums
Autentifikācija ar Windows domēna lietotāju	restful_authentication + ruby-ldap
Autentifikācija ar internet-bankas lietotāju	restful_authentication + sava plugina izveide

Autorizācija

- Realizācija atkarībā no vajadzībām
- Jāierobežo pieeja pie darbībām ar jebkuru resursu:
=> kontrolieru `before_filter` metodes
- Jāierobežo pieeja pie konkrētiem datiem:
=> modeļu metodes

Autorizācijas modeļi

- User
- User has and belongs to many Roles
- User has and belongs to many Roles,
Roles has and belongs to many
Permissions

AutORIZĀCIJAS plugini

Vajadzība	Risinājums
Pieeja kontrolieriem atkarībā no lomām	acl_system
Pieejas atkarībā no modeļos definētām metodēm	authorization

acl_system

```
class PostsController < ApplicationController
  before_filter :login_required, :except => [:list, :index]
  access_control [:new, :create, :update, :edit] => '(admin | user | moderator)',
                :delete => 'admin & (!moderator & !blacklist)'

  access_control :DEFAULT => '!guest',
                [:new, :create, :update, :edit] => '(admin | user | moderator)',
                :delete => 'admin & (!moderator & !blacklist)'

  <% restrict_to "(admin | moderator) & !blacklist" do %>
    <%= link_to "Admin & Moderator only link", :action => 'foo' %>
  <% end %>
```

authorization

```
class MeetingController < ApplicationController
  permit "rubyists and wanna_be_rubyists", :except => :public_page

  def public_page
    render :text => "We're all in Chicago"
  end

  def secret_info
    permit "(matz or dhh) and interested in Answers" do
      render :text => "The Answer = 42"
    end
  end

  def find_apprentice
    @founder = User.find_by_name('matz')
    permit "'inner circle' of :founder" do
      if request.post?
        apprentice = User.find_by_skillset(params[:uber_hacker])
        ruby_community = Group.find_by_name('Ruby')
        ruby_community.accepts_role 'yarv_builder', apprentice
      end
    end
  end
end
```

authorization

```
def rails_conf
  @meeting = Meeting.find_by_name('RailsConf')
  permit "attendees of :meeting or swedish_mensa_supermodels" do
    venue = Hotel.find_by_name("Wyndham O'Hare")
    current_user.is_traveller_to venue
    if permit? "traveller to :venue and not speaker"
      Partay.all_night_long
      @misdeeds = current_user.is_participant_in_what
    end
  end
end
end
end
end
```