Object Oriented WEB Programming Using Ruby

Day 14: 19/July/2012

Anti Virus Policy and
Object Oriented Web Application

Message from the Next Week

The Final lecture. The plan is 'the summary of the semester.'

What I think now is to let all students think about the 'System Design' keys of Object Oriented WEB and DB system. In other words, what is(/are) the best approach(es) to make the most of the Object Oriented characteristics of the ruby language environment.

Very vague? Exactly!

Anti Virus Policy

The other thing that the original lecture plan included was the security issue.

One of the point which had been introduced from version 2, was anti CSRF policy. Now our rails version 3.2, we see the line below every time we generate the project;

```
<%= csrf_meta_tags %>
```

in layouts/application.html.erb

Let us have a glance at CSRF now.

What is CSRF

Cross-site request forgery, also known as a one-click attack or session riding and abbreviated as CSRF (sometimes pronounced sea-surf) or XSRF, is a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts.

Well, let's see Wiki, and some other useful pages.

http://en.wikipedia.org/wiki/Cross-site_request_forgery
http://wiki.developerforce.com/page/Secure_Coding_Cross_Site_Request_Forgery

Rails and CSRF

Ruby on Rails 2.0 provides a protect_from_forgery feature. This implementation does not meet salesforce.com's requirements for CSRF protection if used with the :secret option because the token value will be the same for all users. See General Guidance, above, for anti-CSRF token requirements. Use of the protect_from_forgery feature without the :secret option with Ruby on Rails 3.3 and above creates a random token that meets Salesforce.com security requirements. See the documentation for ActionController::RequestForgeryProtection for more information.

http://wiki.developerforce.com/page/Secure_Coding_Cross_Site_Request_Forgery

General Guidance against CSRF (1/2)

All requests that create, update or delete data or have side-effects require protection against CSRF.

The most reliable method is to include an anti-CSRF token as a hidden input with every application action. This token should be included in all forms built by the genuine application and validated to be present and correct before form data is accepted and acted upon.

http://wiki.developerforce.com/page/Secure_Coding_Cross_Site_Request_Forgery#General_Guidance

General Guidance against CSRF (2/2)

Use the POST method for requests requiring protection to avoid disclosing the token value in Referer headers.

Token values must be unique per user session and unpredictable.

http://wiki.developerforce.com/page/Secure_Coding_Cross_Site_Request_Forgery#General_Guidance

We are WEB developer!

We are now learning how to write WEB program. And, the WEB pages we write are to be attached from malicious pieces of codes, such as CSRF.

Also there will be a chance that new security threat may appear, and it is clear that the WEB system developer always have to handle those security problems.

Rails had provided the anti-CSRF embedded mechanism, but sometimes, we ourselves have to write the code against such threats.

OWASP

Oh! WASP? Probably not.

The Open Web Application Security Project.

We often have to write the defending algorithms in the application we develop. For those cases, we have to search for the knowledge on characteristics of the threat, and orthodox approaches against the threat. For those cases, some of the WEB sites are very useful.

https://www.owasp.org/index.php/Category:OWASP_Project

Object Oriented WEB system

Once again, the quite vague title is the LAST report theme to you, students!

What does ‘Perfect Object Oriented Installation’ mean?
What is the merit of that?

Class and the instances

Rails environment allows us a rapid development frameworks.

Also, we have built-in TDD (Test Driven Development) framework in rails.

Test cases can be an instances of the Test Class. Is there any merit that they are installed (automatically generated) as Class?

See the samples in the next page.

Causes_controller_test.rb

```ruby
require 'test_helper'

class CausesControllerTest < ActionController::TestCase
  setup do
    @cause = causes(:one)
  end

  test "should get index" do
    get :index
    assert_response :success
    assert_not_nil assigns(:causes)
  end

  test "should get new" do
    get :new
    assert_response :success
  end

```

app/test/functional/causes_controller_test.rb

Our Last Report

Please report the merit and demerit of the installation based on the Ruby on Rails, which employs the Object Oriented Technology, and the concepts.

300 to 1,000 words report is accepted.