

Columbus Ruby Brigade

Aaron J. Bedra
April 16th 2007

Installing Net::LDAP

```
gem install ruby-net-ldap
```

```
require 'rubygems'  
require 'net/ldap'
```

```
ldap = Net::LDAP.new :host => '192.168.1.6',  
  :port => 389,  
  :auth => {  
    :method => :simple,  
    :username => "cn=admin,dc=example,dc=com",  
    :password => "secret"  
  }
```

```
if ldap.bind  
  puts ldap.inspect  
else  
  puts ldap.get_operation_result.message  
end
```

Establish a connection and then...

```
if ldap.bind
  filter = Net::LDAP::Filter.eq( "cn", "*" )
  treebase = "dc=example,dc=com"

  ldap.search( :base => treebase, :filter => filter ) do |entry|
    puts "DN: #{entry.dn}"
    entry.each do |attribute, values|
      puts "  #{attribute}:"
      values.each do |value|
        puts "    --->#{value}"
      end
    end
  end
end
else
  puts ldap.get_operation_result.message
end
```

```
if ldap.bind
  dn = "cn=Example User,dc=example,dc=com"
  attr = {
    :cn => "Example User",
    :objectclass => ["top", "inetorgperson"],
    :sn => "User",
  }

  if ldap.add( :dn => dn, :attributes => attr )
    puts "User successfully added"
  else
    puts "User could not be added [#{ldap.get_operation_result.message}]"
  end
else
  puts "Directory bind operation failed [#{ldap.get_operation_result.message}]"
end
```

```
if ldap.bind
  dn = "cn=Example User,dc=example,dc=com"
  if ldap.delete :dn => dn
    puts "User successfully deleted"
  else
    puts "User could not be deleted [#{ldap.get_operation_result.message}]"
  end
else
  puts "Directory bind operation failed #{ldap.get_operation_result.message}"
end
```

```
if ldap.bind
  dn = "cn=Example User,dc=example,dc=com"
  ops = [
    [:add, :mail, "aliasaddress@example.com"],
    [:replace, :mail, ["newaddress@example.com", "newalias@example.com"]],
  ]
  if ldap.modify :dn => dn, :operations => ops
    puts "User successfully modified"
  else
    puts "User could not be modified [#{ldap.get_operation_result.message}]"
  end
else
  puts "Directory bind operation failed [#{ldap.get_operation_result.message}]"
end
```

LDAP on Rails

First create a module and put it in the lib folder

```
require 'net/ldap'

module LDAP

  class << self

    def authenticate(user,password)
      return false if user.blank? || password.blank?
      connection = Net::LDAP.new({:host => '192.168.1.5',
        :port => 389,:auth=>{:method=>:simple,:username => "cn=#{user},dc=example,dc=com",
        :password => password } } )
      if connection.bind
        LDAP_CONNECTION_LOG.info "[#{Time.now}] -- User: #{user} successfully logged authenticated"
        return true
      else
        LDAP_CONNECTION_LOG.info "[#{Time.now}] -- User: #{user} failed authentication"
        return false
      end
    end

  end

end

end
```

Controller Actions

```
class LoginController < ApplicationController
  skip_before_filter :authenticate

  def index
  end

  def authenticate
    if LDAP.authenticate(params[:login][:name], params[:login][:password])
      flash[:success] = 'Login Successful'
      redirect_to :action => 'index'
    else
      flash[:error] = 'Login failed!'
      redirect_to :action => 'index'
    end
  end

  def logout
    reset_session
    flash[:notice] = 'Logged out'
    redirect_to :action => 'index'
  end
end
```

```
class ApplicationController < ActionController::Base
  session :session_key => '_ldap_session_id'

  before_filter :authenticate
  filter_parameter_logging 'password'

  protected

  def authenticate
    unless session[:person]
      session[:return_to] = @request.request_uri
      redirect_to :controller => 'login'
      return false
    end
  end
end

end
```