```
mddir c:\dev\sfprojects
cd dev\sfprojects
mkdir jobeet\lib\vendor
cd jobeet lib\vendor
unzip symfony
rename symfony-1.4.0 to symfony

/* right click computer->Advanced system settings ->Environment Variables -> select Path in System variables
click Edit and add something like this ';C:\wamp\bin\mysql\mysql5.1.53\bin;C:\wamp\bin\php\php5.3.4;C:\wamp\www'
to Variable value text box.
*/

php lib\vendor\symfony\data\bin\symfony -V /* to verify the symfony version */

php lib\vendor\symfony\data\bin\symfony generate:project jobeet
/* c:\> copy lib\vendor\symfony\data\bin\symfony.bat .*/
php symfony generate:app frontend

/* locate and open httpd.conf configuration file and add the following configuration at
the end
NameVirtualHost 127.0.0.1:8080
Listen 127.0.0.1:8080
<VirtualHost 127.0.0.1:8080>
DocumentRoot "C:\wamp\www\dev\sfprojects\jobeet\web"
DirectoryIndex index.php
<Directory "C:\wamp\www\dev\sfprojects\jobeet\web">
AllowOverride All
Allow from All
</Directory>
Alias /sf "C:\wamp\www\dev\sfprojects\jobeet\lib\vendor\symfony\data\web\sf"
<Directory "/home/sfprojects/jobeet/lib/vendor/symfony/data/web/sf">
AllowOverride All
Allow from All
```

```
</Directory>
</VirtualHost>

*/

mysqladmin -uroot -p create jobeet
php symfony configure:database "mysql:host=localhost;dbname=jobeet" root /* or root password if password avai */
/* create schema.yml file*/
# config/doctrine/schema.yml
JobeetCategory:
actAs: { Timestampable: ~ }
columns:
name: { type: string(255), notnull: true, unique: true }


/*
php symfony doctrine:build --model
php symfony doctrine:build --sql
php symfony doctrine:insert-sql
*/
/* create data yml files */
# data/fixtures/categories.yml
JobeetCategory:
design:
name: Design
programming:
name: Programming
manager:
name: Manager
administrator:
name: Administrator

php symfony doctrine:build --all --no-confirmation /* use this for the last 3 line */
```

php symfony doctrine:data-load

php symfony doctrine:build --all --and-load /* replace last 2 line */

php symfony doctrine:generate-module --with-show --non-verbose-templates frontend job JobeetJob
(php symfony generate:module frontend content)

Modify apps/frontend/templates/layout.php

```
Modify apps/frontend/config/view.yml /* normally don’t need */
stylesheets: [main.css]
javascripts: []
has_layout: true
layout: layout
```

Create apps/frontend/modules/job/config/view.yml /*for customize the view of a module normally don’t need*/
Modify apps/frontend/modules/job/templates/indexSuccess.php to make it to look as you requirement
Modify apps/frontend/modules/job/templates/showSuccess.php to make it to meet your requirement

```
// apps/frontend/templates/layout.php   /* adding slot */
<title> <?php include_slot('title', 'Jobeet - Your best job board') ?> </title>
// apps/frontend/modules/job/templates/showSuccess.php
<?php slot('title',sprintf('%s is looking for a %s', $job->getCompany(),$job->getPosition()))?>
```

```
# apps/frontend/config/routing.yml  /* setting up the route */
job:
  class: sfDoctrineRouteCollection /* automatic create seven sfDoctrineRoute routes */
  options: { model: JobeetJob }

job_show_user:
  url: /job/:company/:location/:id/:position
  class: sfRequestRoute
  options: { model: JobeetJob, type: object }
/* to list the routes information use following 2 line */
$ php symfony app:routes frontend
$ php symfony app:routes frontend job_edit
```

```
/* sfDoctrineRouteCollection creats all routes so you could remove some from the routing.yml */
```

```
url_for('job_show_user', $job) /*  change in indexSuccess.php */
```

```
// lib/Jobeet.class.php  create this class for slugify the text to make the URL more readable
class Jobeet
{
static public function slugify($text)
{
// replace all non letters or digits by -
$text = preg_replace('/\W+/', '-', $text);
// trim and lowercase
$text = strtolower(trim($text, '-'));
return $text;
}
}

// lib/model/doctrine/JobeetJob.class.php  added for calling the slugify function just created above
public function getCompanySlug()
{
return Jobeet::slugify($this->getCompany());
}
public function getPositionSlug()
{
return Jobeet::slugify($this->getPosition());
}
public function getLocationSlug()
{
return Jobeet::slugify($this->getLocation());
}
```

```
$this->job = $this->getRoute()->getObject(); /* add to executeShow in actions.class.php getRoute() method
automatic thrown error if any.
It Same like this:
$this->job = Doctrine_Core::getTable('jobeetjob')->find(array($request->getParameter('id')));
```

/* use frontend_dev.log to view the log file */
/* to override method go to lib/model/doctrine/.. */

```
/* for application specific settings change apps/frontend/config/app.yml */
/* use sfConfig::get('app_active_days') to get from yml file */
```

```
/* Refactoring (move Doctrine_Query code from action to model like lib/model/doctrine/JobeetJobTable.class.php */
class JobeetJobTable extends Doctrine_Table
{
public function getActiveJobs()
{
$q = $this->createQuery('j')
->where('j.expires_at > ?', date('Y-m-d H:i:s', time()));
return $q->execute();
}
}
```

```
# config/doctrine/schema.yml
JobeetCategory:
  actAs:
    Timestampable: ~
    Sluggable:
      fields: [name]
…..
```

Create the partial file `_list.php` file:

```
<?php include_partial('job/list', array('jobs' => $jobs)) ?> /* include the partial */
```

```
/* to paginate add this to showSuccess
$this->pager = new sfDoctrinePager('JobeetJob',sfConfig::get('app_max_jobs_on_category'));
$this->pager->setQuery($this->category->getActiveJobsQuery());
$this->pager->setPage($request->getParameter('page', 1));
$this->pager->init();
```

```
php symfony test:unit Jobeet
php test/unit/JobeetTest.php /* to run JobeetTest.php for testing */
php symfony test:unit
php symfony test:functional frontend
php symfony test:all
```

```
/* unit testing */
```

```
php symfony configure:database --name=doctrine --class=sfDoctrineDatabase --env=test
"mysql:host=localhost;dbname=jobeet_test" root mYsEcret
mysqladmin -uroot -pmYsEcret create jobeet_test
php symfony doctrine:insert-sql --env=test
copy data/fixtures to test/fixtures
$ php test/functional/frontend/categoryActionsTest.php
```

```
php symfony jobeet:cleanup --days=10 --env=dev /* if cleanup function created by you can run it */

$this->mergeForm(new AnotherForm()); /* merge form into another */
$this->embedForm('name', new AnotherForm()); /* embed form */

lib/form/doctrine/JobeetJobForm.class.php
unset(
$this['created_at'], $this['updated_at'],
$this['expires_at'], $this['is_activated']
);
$this->useFields(array('category_id', 'type', 'company', 'logo',
'url', 'position', 'location', 'description', 'how_to_apply',
'token', 'is_public', 'email'));
$this->validatorSchema['email'] = new sfValidatorAnd(array($this->validatorSchema['email'],
new sfValidatorEmail(),));

$this->widgetSchema['type'] = new sfWidgetFormChoice(array('choices' =>
Doctrine_Core::getTable('JobeetJob')->getTypes(),'expanded' => true,));

$this->widgetSchema['logo'] = new sfWidgetFormInputFile(array('label' => 'Company logo',));
$this->validatorSchema['logo'] = new sfValidatorFile(array('required' => false,'path' =>
sfConfig::get('sf_upload_dir').'/jobs','mime_types' => 'web_images',));
$this->widgetSchema->setLabels(array('category_id' => 'Category','is_public' => 'Public?', ));

php symfony generate:app backend
move apps/frontend/config/app.yml file to config/app.yml
php symfony doctrine:data-load --application=frontend
php symfony doctrine:generate-admin backend JobeetJob --module=job
php symfony doctrine:generate-admin backend JobeetCategory --module=category
php symfony plugin:publish-assets
apps/backend/modules/category/config/generator.yml
apps/backend/modules/job/config/generator.yml
```

```
list:
  batch_actions:
    _delete: ~
    extend: ~
```

create an `executeBatchExtend()` method in actions.class.php
Update the `extend()` method in JobeetJob.class.php
create an `executeListExtend ()` method in actions.class.php

```
list:
  actions:
    deleteNeverActivated: { label: Delete never activated jobs }
```

Create the `listDeleteNeverActivated` method in actions.class.php
reduce the number of queries `use table_method: retrieveBackendJobList`

create a `BackendJobeetJobForm to overridden display`

```
form:
  class: BackendJobeetJobForm
```

`use sfWidgetFormInputFileEditable` widget to add editing capabilities to a simple input file

```
filter:
  display: [category_id, company, position, description, is_activated,is_public, email, expires_at]
```

```
$this->getUser()->setFlash('notice', sprintf('Your job validity has been
extended until %s.', $job->getDateTimeObject('expires_at')->format('m/d/
Y'))); /* setting the flash */
```

```
/* include the flash message in template or layout*/
<?php if ($sf_user->hasFlash('notice')): ?>
<div class="flash_notice"><?php echo $sf_user->getFlash('notice')
?></div>
<?php endif ?>
```

```
$jobs = $this->getUser()->getAttribute('job_history', array());
array_unshift($jobs, $this->job->getId());
$this->getUser()->setAttribute('job_history', $jobs);
```

```
php symfony plugin:install sfDoctrineGuardPlugin
php symfony guard:create-user fabien@example.com fabien SecretPass
```

```
$ php symfony list guard
php symfony guard:promote Fabien

config/ProjectConfiguration.class.php

symfony doctrine:build --all --and-load --no-confirmation

// apps/backend/lib/myUser.class.php
class myUser extends sfGuardSecurityUser

apps/backend/config/settings.yml
```