

This is an example code by using Zend Framework.

In my application I have a database which includes some technical data of different cars. This database is indexed by search engine Lucene and on a page on my application the user can search car types by a brand and between a certain build period.

Example:

Makes

ALFA ROMEO

Bjvanaf

12/12/1980

BJtot

12/12/1985

Search

4 Results found.

140	152	ALFA ROMEO	33 (905)	01/06/1983	01/12/1985
141	153	ALFA ROMEO	33 (905)	01/06/1983	01/12/1985
142	154	ALFA ROMEO	33 (905)	01/06/1983	01/12/1985
143	155	ALFA ROMEO	33 (905)	01/06/1983	01/12/1985

In my example I have two files:

- 1) The controller page, which creates the form and handles the search function.
- 2) The model class "TecdocIndex " which includes the business logic to make an index with search engine Lucene, makes the search form and the functionality to make a search after the request from the form.

Controller

```
<?
class SearchController extends Application_Model_BaseController
{
    /*
     * Controller to render search form , get search results and render
     * them.
     */
    public function indexAction()
    {
        $form= $this->getForm();

        if ($this->getRequest()->isPost()) {
            $post= $this->_request->getPost();
            if($form->isValid( $post))
            {
                $searchResults =
                    Application_Model_TecdocIndex::searchByQyqueryString($post);
                $this->view->searchResults = $searchResults ;
            }
            $this->view->form = $form;
        }
    }
}
```

Model and business logics for the indexer

<?php

```
class Application_Model_TecdocIndex
{
    /*
     * This function searches the cars from the Index.
     * It takes an input parameter criteriArr, which includes
     * the criteria selected by the user (Brand and optional fields,
     * build year start and build year end)
     *
     * @param $criteriaArr array
     * @return $results array
     */
    static public function searchByQyqueryString($criteriaArr)
    {
        $config=Zend_Registry::get('config');
        $indexPath=$config->indexerPath;

        $index    = Zend_Search_Lucene::open( $indexPath);

        $queriesArr=array();           // array of all queries

        // Add Brand query
        $queriesArr[]=self::makeQueryStringEqual('Makes', $criteriaArr['Makes']);

        // Add build year start query, if it is given
        if ($criteriaArr['BJvanaf'])
        {
            $queriesArr[]=self::getQueryConstructionAPIRange('BJvanaf',
                                                                $criteriaArr['BJvanaf'],'99991212')
;
        }

        // Add build year end query, if it is given
        if ($criteriaArr['BJtot'] )
        {
            $queriesArr[]=self::getQueryConstructionAPIRange('BJtot','11111111',
```

```

$criteriaArr['BJtot']);
    }

    // Combine queries
    $query = new Zend_Search_Lucene_Search_Query_Boolean();
    foreach ($queriesArr as $subQuery)
    {
        $query->addSubquery( $subQuery, true );
    }
    $results = $index->find($query);
    return $results;
}

/*
 * Build a string query for given field with condition equal
 * @param $fieldName      string
 * @param $value          string
 * @return $queryString   string
 */
static private function makeQueryStringEqual($fieldName,$value)
{
    $queryString=$fieldName.'::'.$value;
    try {
        $query =
            Zend_Search_Lucene_Search_QueryParser::parse( $queryString);
    }
    catch (Zend_Search_Lucene_Search_QueryParserException $e) {
        echo "Query syntax error: " . $e->getMessage() . "\n";
    }
    return $query;
}

/*
 * @param $fieldName      string
 * @param $value          string
 * @return $query          Zend_Search_Lucene_Search_Query_Range
 */
static private function getQueryConstructionAPIRange( $fieldName,$valueFrom,
                                                    $valueTo )
{
    $from = new Zend_Search_Lucene_Index_Term($valueFrom, $fieldName);
    $to= new Zend_Search_Lucene_Index_Term($valueTo,$fieldName);
    $query = new Zend_Search_Lucene_Search_Query_Range($from, $to, true);
    return $query;
}

/*
 * Create a search form to search cars
 * @return $form Zend_Form
 */
static public function getForm()
{
    $form = new Zend_Form;
    $Bandeninfogg=new Application_Model_Business_Bandeninfogg();

    $url=$this->getRequest()->getRequestUri();
    $form->setAction( $url)->setMethod('post');
    $form->setAttrib('id', 'login');

    $makes = new Zend_Form_Element_Select( 'Makes');
    $options=$Bandeninfogg->getDistinctMakes('Makes');

```

```

$makes->setMultiOptions($options);
$makes->setLabel('Makes');
$form->addElement($makes);

$dateValidator = new Zend_Validate_Date(array('format' => 'dd/mm/yyyy'));

$BJvanaf=new Zend_Form_Element_Text('BJvanaf');
$BJvanaf->setLabel('BJvanaf');
$BJvanaf->addValidator( $dateValidator);
$form->addElement($BJvanaf);

$BJtot=new Zend_Form_Element_Text('BJtot');
$BJtot->setLabel('BJtot');
$BJtot->addValidator( $dateValidator);
$form->addElement($BJtot);

$form->addElement('submit', 'login', array('label' => 'Search'));

return $form;
}

/*
 * This function creates an index from the database "Bandeninfo"
 */
static public function createIndex()
{
    $config=Zend_Registry::get('config');
    $indexPath=$config->indexerPath;

    $index = Zend_Search_Lucene::create( $indexPath);

    $bb= new Application_Model_Dal_Bandeninfogg();
    $results=$bb->fetchAll(array());

    foreach ($results as $row)
    {
        $BJvanaf= $row->BJvanaf.'01';
        $BJtot=$row->BJtot.'01' ;

        $doc = new Zend_Search_Lucene_Document();
        $doc->addField(Zend_Search_Lucene_Field::UnIndexed('dbId',$row->id));
        $doc->addField(Zend_Search_Lucene_Field::Text ('Makes', $row->Makes ));
        $doc->addField(Zend_Search_Lucene_Field::unIndexed('Models', $row->Models ));
        $doc->addField(Zend_Search_Lucene_Field::unIndexed('Types', $row->Types ));
        $doc->addField(Zend_Search_Lucene_Field::Keyword('BJvanaf', $BJvanaf ));
        $doc->addField(Zend_Search_Lucene_Field::Keyword('BJtot', $BJtot ));
        $index->addDocument($doc);
    }
}

$index->commit();
}

```