Opzet gebruik SOLR

Input

1. Converteer MARC XML naar SOLR XML
2. Verzamel eventueel extra content
3. Maak SOLR XML
4. Upload

Ad 1. Globale structuur SOLR XML

"echte velden", indexed=true, meestal woord-voor-woord, soms in z'n geheel

title:

author: in JSON een array, in XML repeated field

year:

…

"berekende velden", indexed=true,

tekst: alle echte velden bij elkaar

…

"berekende velden", speciaal voor facets, indexed=true,

yearForFacet: 2010

…

"berekende velden", indexed=false, stored=true

listEntry: een slimme representatie van de data nodig voor de list, eventueel al in HTML, of in JSON, zó dat er snel HTML van kan worden gemaakt

indexed=false, stored=true

detailEntry: idem voor de detail representatie

Zoeken

Gebruik PHP's SolrClient (https://www.php.net/manual/en/class.solrclient.php)

Het zoekscherm

Het zoekscherm bestaat uit de volgende onderdelen:

1. Query input met de query zoals ingevoerd door de gebruiker
2. Header, met aantal gevonden, volgende/vorige pagina, etc.
3. Facets
4. Details

Bij elke zoekactie genereert SOLR een zoekresultatenlijst met alle gegevens. Dus ook **listEntry's** en **detailEntry's**.

De listEntry's zijn handig voor het snel opmaken van de lijst.

Wanneer op een element in de lijst wordt geklikt kan het detailscherm worden opgemaakt door de bjbehorende detailEntry te vertonen.

Het lijkt mij handig om een class te maken voor het zoekscherm:

```
<?php

class SearchPage {
  public $userQuery = ''; //of "*.*"
  private $parsedQuery = '';
  public $start = 0;
  public $numRows = 10;

  private $solrOptions = array (
  'hostname' => SOLR_SERVER_HOSTNAME,
  'login'    => SOLR_SERVER_USERNAME,
  'password' => SOLR_SERVER_PASSWORD,
  'port'     => SOLR_SERVER_PORT,
  );
  private $facetFields = array(

  );
  private $solrClient = nulll //of type SolrClient
  private $solrQuery = null; //of type SolrQuery

  public $searchResults = array();


  public function __construct() {
    $client = new SolrClient($this->solrOptions);
    $query = new SolrQuery();
  }

  public function prepareSearch($userInput) {
    $this->userQuery = $userInput;
    $this->parsedQuery = $this->parse($userInput);

    $this->query->setQuery($this->parsedQuery);
    $this->query->setStart($this->start);
    $this->query->setRows($this->numRows);
    //etc

    $this->search();
  }

  private function search() {
    $query_response = $this->client->query($this->query);
    $query_response->setParseMode(SolrQueryResponse::PARSE_SOLR_DOC);  //??
    $this->searchResults = $query_response->getResponse();
    $this->
  }

  public function nextPage() {
    $this->start = $this->start + $this->numRows;
```

```
    if ($this->start < $this->searchResults['response']['numFound']) $this-
>search();
  }

  public function prevPage() {
    $this->start = $this->start - $this->numRows;
    if ($this->start > 0) $this->search();
  }


  //etc

}
```