Solr Bundle

> Solr Search Engine Bundle is bundled by default in eZ Platform and these instructions are for configuring it, for eZ Publish Platform 5.4 instructions see Solr Search Engine Bundle.

> Version 1.0.x of Solr Bundle primarily aims to be used as replacement for Legacy (SQL-based) search engine for better scalability and performance *(especially with field criteria and sort clauses).* And while it also provides better full text search thanks to Solr, more advance search features like Faceting will come in later releases.

What is Solr Search Engine Bundle?

ezplatform-solr-search-engine as the package is called, aims to be a transparent drop in replacement for the SQL based "Legacy" search engine powering Search API by default. By enabling Solr and re-indexing your content, all your exising Search queries using SearchService, will be powered by Solr automatically. This allows you to scale up your eZ Platform installation and be able to continue development locally against SQL engine, and have test infrastructure, Staging and Prod powered by Solr. Thus remove considerable load from your database so it can focus on more important things, like publishing 😊.

Se Architecture page for further information on the architecture of eZ Platform.

How to set up Solr Search engine

Step 1: Enabling the Bundle

> **Not needed with eZ Platform**
> This step is not needed for eZ Platform as of 15.09, however it is kept here for reference in case you have previously disabled the bundle.

1. Add/Update composer dependencies:

   command line

   ```
   composer require --no-update ezsystems/ezplatform-solr-search-engine:~1.0
   composer update
   ```

2. Activate EzPublishSolrSearchEngineBundle by adding the following lines to your `app/AppKernel.php` file: new `EzSystems\EzPlatformSolrSearchEngineBundle\EzSystemsEzPlatformSolrSearchEngineBundle()`

Step 2: Configuring & Starting Solr

> Example here is for single core, look to Solr documentation for configuring Solr in other ways, also see the provided configuration for some examples.

First, download and extract Solr, **we currently support Solr 4.10.4**:

- solr-4.10.4.tgz or solr-4.10.4.zip

Secondly, copy configuration files needed for eZ Solr Search Engine bundle, *here from the root of your project to the place you extracted Solr*:

Command line example

```
# Make sure to change the /opt/solr/ path with where you have placed Solr
cp -R vendor/ezsystems/ezplatform-solr-search-engine/lib/Resources/config/solr/*
/opt/solr/example/solr/collection1/conf/

/opt/solr/bin/solr start -f
```

Thirdly, Solr Bundle does not commit solr index changes directly on repository updates, leaving it up to you to tune this using `solrconfig.xml` a s best practice suggests, example config:

solrconfig.xml

```
<autoCommit>
  <!-- autoCommit is here left as-is like it is out of the box in Solr 4.10.4, this
controls hard commits for durability/replication -->
  <maxTime>${solr.autoCommit.maxTime:15000}</maxTime>
  <openSearcher>false</openSearcher>
</autoCommit>

<autoSoftCommit>
  <!-- Soft commits controls mainly when changes becomes visible, by default we change
value from -1 (disabled) to 100ms, to try to strike a balance between Solr performance
and staleness of HttpCache generated by Solr queries -->
  <maxTime>${solr.autoSoftCommit.maxTime:100}</maxTime>
</autoSoftCommit>
```

Step 3: Configuring bundle

The Solr search engine bundle can be configured many ways, here are some examples:

Single Core example (default)

Out of the box in eZ Platform the following is enabled for simple setup:

config.yml

```
ez_search_engine_solr:
    endpoints:
        endpoint0:
            dsn: %solr_dsn%
            core: collection1
    connections:
        default:
            entry_endpoints:
                - endpoint0
            mapping:
                default: endpoint0
```

Shared Core example

In the following example we have decided to separate one language as the installation contains several similar languages, and one very different language that should be receive proper language analysis for proper stemming and sorting behavior by Solr:

config.yml

```
ez_search_engine_solr:
    endpoints:
        endpoint0:
            dsn: %solr_dsn%
            core: core0
       endpoint1:
            dsn: %solr_dsn%
            core: core1
    connections:
        default:
            entry_endpoints:
                - endpoint0
                - endpoint1
            mapping:
                translations:
                    - jpn-JP: endpoint1
                # Other languages, for instance eng-US and other western languages are sharing
core
                default: endpoint0
```

Multi Core example

If full language analysis features are preferred, then each language can be configured to separate cores.

Note: Please make sure to test this setup against single core as it might perform worse then single core if your project uses a lot for language fallbacks per siteaccess as queries will then be performed across several cores at once.

config.yml

```
ez_search_engine_solr:
    endpoints:
        endpoint0:
            dsn: %solr_dsn%
            core: core0
       endpoint1:
            dsn: %solr_dsn%
            core: core1
       endpoint2:
            dsn: %solr_dsn%
            core: core2
       endpoint3:
            dsn: %solr_dsn%
            core: core3
       endpoint4:
            dsn: %solr_dsn%
            core: core4
       endpoint5:
            dsn: %solr_dsn%
            core: core5
       endpoint6:
            dsn: %solr_dsn%
            core: core6
    connections:
        default:
            entry_endpoints:
                - endpoint0
                - endpoint1
                - endpoint2
                - endpoint3
                - endpoint4
                - endpoint5
                - endpoint6
            mapping:
                translations:
                    - jpn-JP: endpoint1
                    - eng-US: endpoint2
                    - fre-FR: endpoint3
                    - ger-DE: endpoint4
                    - esp-ES: endpoint5
                # Not really used, but specified here for fallback if more languages are
suddenly added by content admins
                default: endpoint0
                # Also use separate core for main languages (differs from content object to
content object)
                # This is useful to reduce number of cores queried for always available
language fallbacks
                main_translations: endpoint6
```

Step 4: Configuring repository with the specific search engine

The following is an example of configuring Solr Search Engine, where `connection` name is same as in example above, and engine is set to `solr`:

ezplatform.yml

```
ezpublish:
    repositories:
        main:
            storage:
                engine: legacy
                connection: default # This should be the connection you had from before for
repository
            search:
                engine: solr # One of legacy (default) or solr
                connection: default # If legacy same as storage applies, if solr use same as
you defined in step 3
```

Step 5: Run CLI indexing command

> **Make sure to configure your setup for indexing**
> Some exceptions might happen on indexing if you have not configured your setup correctly, here are the most common issues you may encounter:
>
> - Exception if Binary files in database have an invalid path prefix
>   - Make sure `ezplatform.yml` configuration `var_dir` is configured properly.
>   - If your database is inconsistent in regards to file paths, try to update entries to be correct *(but make sure to make a backup first)*.
> - Exception on unsupported Field Types
>   - Make sure to implement all Field Types in your installation, or to configure missing ones as NullType if implementation is not needed.
> - Content not immediately available
>   - Solr Bundle is on purpose not committing changes directly on Repository updates *(on indexing)*, but letting you control this using Solr configuration. Adjust Solr **autoSoftCommit** *visibility of change to search index)* and/or **autoCommit** *(hard commit, for durability and replication)* to balance performance and load on your Solr instance against needs you have for "NRT".
> - Running out of memory during indexing
>   - In general make sure to run indexing using prod environment to avoid debuggers and loggers from filing up memory.
>   - Stash: Disable in_memory cache as recommended on Persistence cache for long running scripts.
>   - Flysystem: An open issue exists where you can find further info https://jira.ez.no/browse/EZP-25325

Last step is to execute initial indexation of data:

```
php app/console --env=prod --siteaccess=<name> ezplatform:solr_create_index
```

Providing feedback

After completing the installation you are now free to use your site as usual. If you get any exceptions for missing features, have feedback on performance, or want to discuss, join our community slack channel at https://ezcommunity.slack.com/messages/ezplatform-use/