

Making Your Content Searchable with Indexing



Xavier Morera

PASSIONATE ABOUT TEACHING

@xmorera www.xaviermorera.com



Indexing

Adding content to a Solr index

If necessary, modifying it

Thus making it searchable





Indexing is Needed For Search

Indexing Content

Multiple content types and from various content sources

- XML, CSV, Word, Excel, Power Point, ...
- Databases

It is possible to index via

- HTTP REST API
- Solr's Java Client API
- No .NET API!

That's why we have SolrNet

- Internally it calls REST API → calls a Handler, i.e. [/update](#)



If Necessary, Modifying It

- Search engines are optimized for reading
- Document is written once
 - Although it can be updated too
- And read many times
 - At query time
- Critical to move transformations as much as possible to index time
 - Instead of every time a document is retrieved
- And now we are ready for Thus making it searchable

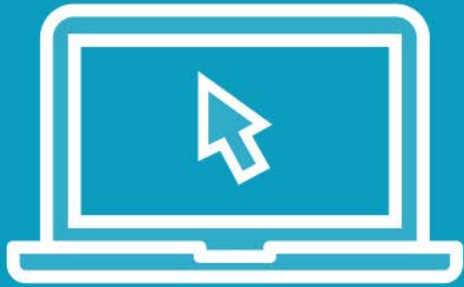


We are going to learn how to index content in C# using SolrNet

Indexing Content



Demo



Indexing $[0, n]$ Documents

M5 D1



Demo Summary



Indexing is straightforward

Mapping

- Attributes
- Dynamic

.Add()

- Object
- IEnumerable (atomic)

Remember to commit!

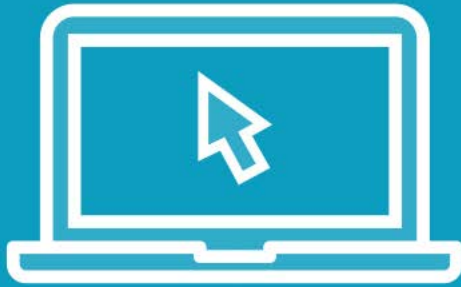
- Rollback

Indexing All Courses

- Manual work only takes you so far
- Let's build a full index
- And use a JSON document as datasource
 - Full list of courses
 - Json.Net to parse it
- You can do multiple content sources
 - Can be a database
 - Push content using C#
 - Configure DataImportHandler
 - Or any other content source (connector)



Demo

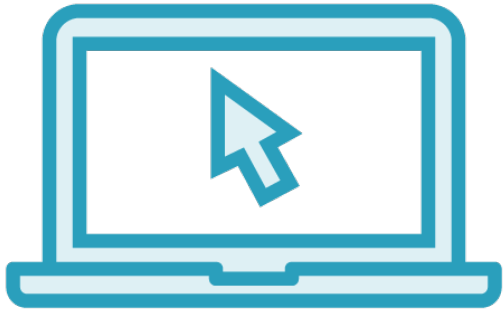


Indexing All Courses

M5 D2



Demo Summary



Full index

Parsed JSON file with Json.Net (easy!)

Extract and push content from any content source

- Connector or access

Solr can pull content with the DataImportHandler

Feed multiple documents (batches)

- Atomic

Feed one at a time

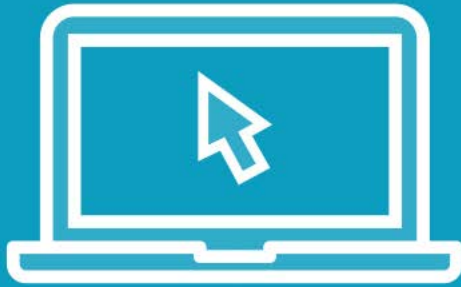


Indexing Binary Files

- Solr is capable of indexing binary files
- Extract request handler
 - Automatically determine input type
 - Or specified
 - Parse and extract content
 - Mapping and boosting also supported
- Apache Tika
 - Different file format parsers
 - Apache PDFBox - A Java PDF Library
 - Apache POI - the Java API for Microsoft Documents



Demo

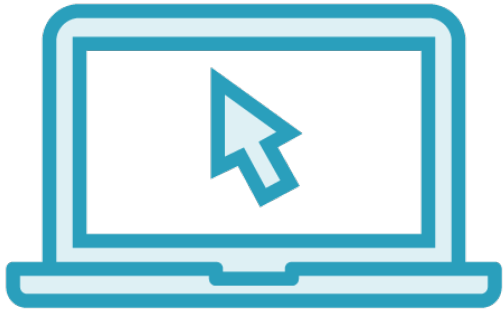


Indexing Binary Files

M5 D3



Demo Summary



Previously we indexed mapped classes

-POCO

Also possible to index text

- JSON, XML, CSV, ...

Solr can index binary files

- Extract request handler
- Word, Excel, PowerPoint, PDF

Uses Apache Tika

- Apache POI, Apache PDFBox, other parsers

Extract() in SolrNet

Takeaway



To search, you need content

To get content, you need to index

SolrNet makes it easy

- Add() & Extract()
- Calls internally /update & /extract

Documents

Binary files

- Parsers