SEM5640 Group Project

Solr Technical Report

|  |  |
| --- | --- |
| Author: | Morgan Jones |
| Config Ref: | SEM5640.2019.str |
| Date: | 3rd November 2019 |
| Version: | 1.0 |
| Status: | Release |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. Search with solr 3

REFERENCES 3

DOCUMENT HISTORY 4

# Introduction

## Purpose of this Document

The purpose of this document is to provide a technical report for the usage of Apache Solr [1] within the 2019 SEM5640 masters’ year group project.

## Scope

This report aims to describe our research of Apache Solr [1] and evaluate its potential for use within the project as an alternative to a bespoke implementation of the system’s required search functionality.

## Objectives

The objectives of this document are:

* Outline Solr and its suitability for the required search functionality
* Evaluate the adjustments required for using Solr
* Evaluate Solr’s interoperability with other system components/applications.
* Compare Solr use to bespoke search implementation

# Search with solr

Solr is an open-source enterprise-search platform, written in Java, from the Apache Lucene project. Solr is a powerful search tool used by many heavily-trafficked websites and applications. Solr’s search capabilities are far greater than the search required for our project; therefore all search requirements can be met through use of Solr as alternative to bespoke search implementation.

Solr has a Java API called Solrj [2] that makes it easy for Java applications to interface with a Solr server. We are planning to build our system’s message store microservice in Java EE therefore use of Solr for search would be an easy adjustment. Both Solr and Solrj have comprehensive online documentation and tutorials so usage should be fairly straightforward.

Solr would be running on the same machine (Docker container) as the message store microservice but on a different port. The message store microservice would then make use of Solrj to make calls to the Solr collection’s REST API. These calls would involve both indexing and querying on the Solr server. Where indexing would provide Solr with the input documents for it to search through and querying would provide Solr with the details of the search itself. Results would then be returned to the HttpSolrClient in the message store application.

Overall we believe use of Solr to be a good choice for this project. Solr is a professional way to provide search functionality to our web-based system. Integrating Solr with our current choice of technology is easy. Use of Solr will be able to meet our search related functional requirements and save us time in not having to code and test our own search implementation.

REFERENCES

|  |  |
| --- | --- |
| [1] | “Solr Homepage,” 26 10 2019. [Online]. Available: https://lucene.apache.org/solr/. |
| [2] | “Using Solrj,” Apache Software Foundation, 27 10 2017. [Online]. Available: https://lucene.apache.org/solr/guide/7\_1/using-solrj.html. [Accessed 3 11 2019]. |

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 0.1 | N/A | 26/10/2019 | N/A – Initial Creation | MWJ7 |
| 1.0 | N/A | 03/11/2019 | Added main content & references | MWJ7 |
|  |  |  |  |  |
|  |  |  |  |  |