# HOW IT WORKS

Directory contains 2 folders: Released and SourceCodes

**Released Folder:**

1. Deploy “FileOperation.war” file to the server and copy FileAnalyzer.jar and then paste it to C:/ directory.
   1. The user will be able to change configuration on: Tomcat8.5\webapps\FileOperation\WEB-INF\classes\config.properties
2. Server will run for each request FileAnalyzer.jar.
   1. The program will work on the given path and create a file with .chc extension for each text file given target directory and its sub directories.
   2. .chc file keeps the last modified date of text file and analysed data results for the performance which also means if the modified date of the text file is the same with the date in the .chc file, program will not reanalyse the text file again and again.The program attached the results to the output query.
3. SimpleClientWordProcessor.exe is a simple client for the service.

**SourceCodes Folder:**

All the source codes are provided in this folder.

**Tested on:**

Apache TomCat 8.5

Java version "1.7.0\_75" , Java(TM) SE Runtime Environment (build 1.7.0\_75-b13)

.Net 3.5

Further information / questions:

* The IT department is known to change their mind with regards to the information they need from time to time, keep that in mind while designing the service
* How can you make sure that the service works as expected before handing it out to IT?
* Feel free (and encouraged) to use any existing frameworks to fulfil the task as efficiently as possible

Answers:

1. I have divided service and server side. I have created an interface in the server side connection class for the future extensions. This provides flexibility to IT department feel to change the service anytime. The file path and default search path are configurable in the server settings.
2. To make sure that we deliver the service with expected results, we can write unit tests. Unit test reduces concerns about service and shows how stable it works. If the unit testing is successful, there can be possible problems related to configuration. It can be avoided by providing a setup wizard programs.
3. In this project, I have used following technologies and framework:

**Server side (FileOperation.war):**

Java based, Maven,Spring

**Service side (FileAnalyzer.jar):**

Java based, Gson

**Client side (SimpleClientWordProcessor.exe):**

.Net based, Newton.Json