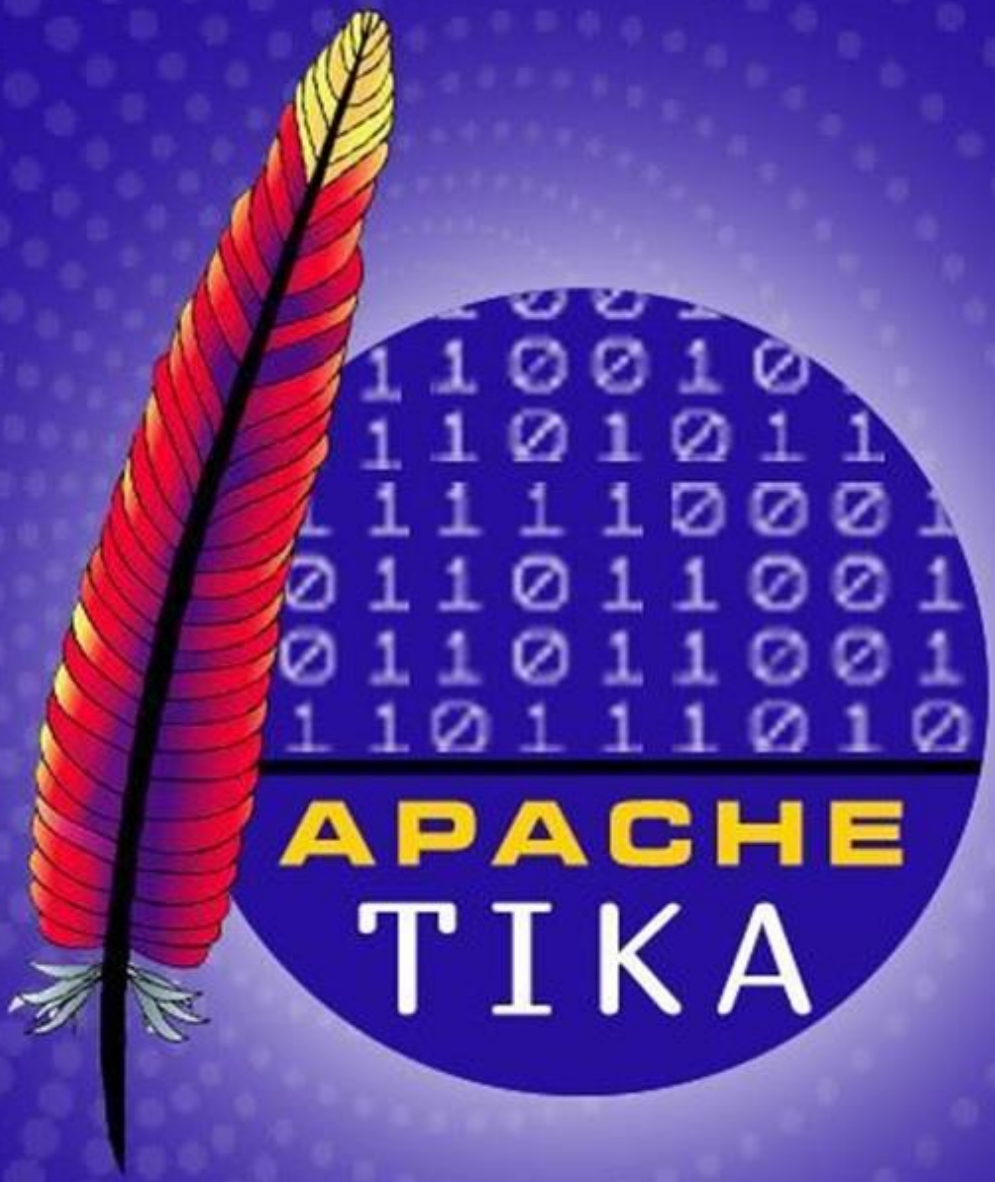


LEARN APACHE TIKA

parser library

tutorialspoint
SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

This tutorial provides a basic understanding of Apache Tika library, the file formats it supports, as well as content and metadata extraction using Apache Tika.

Audience

This tutorial is designed for all Java enthusiasts who want to learn document type detection and content extraction using Apache Tika.

Prerequisites

To make the most of this tutorial, the readers should have prior exposure to Java programming with JDK 1.6 and IO concepts in Java.

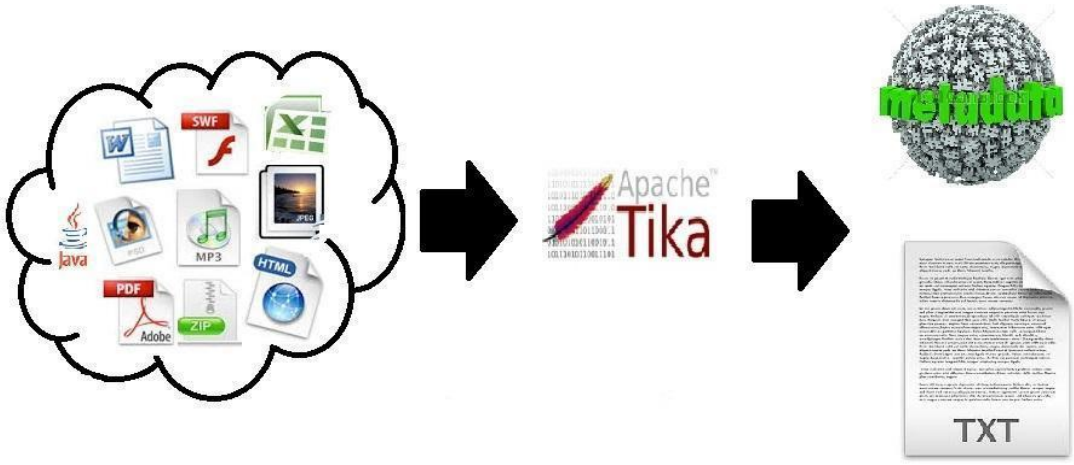
Copyright & Disclaimer

Table of Contents

1. TIKA – OVERVIEW

What is Apache Tika?

- Apache Tika is a library that is used for document type detection and content extraction from various file formats.
- Internally, Tika uses existing various document parsers and document type detection techniques to detect and extract data.
- Using Tika, one can develop a universal type detector and content extractor to extract both structured text as well as metadata from different types of documents such as spreadsheets, text documents, images, PDFs and even multimedia input formats to a certain extent.
- Tika provides a single generic API for parsing different file formats. It uses existing specialized parser libraries for each document type.
- All these parser libraries are encapsulated under a single interface called the **Parser interface**.



Why Tika?

According to filext.com, there are about 15k to 51k content types, and this number is growing day by day. Data is being stored in various formats such as text documents, excel spreadsheet, PDFs, images, and multimedia files, to name a few. Therefore, applications such as search engines and content management systems need additional support for easy

extraction of data from these document types. Apache Tika serves this purpose by providing a generic API to locate and extract data from multiple file formats.

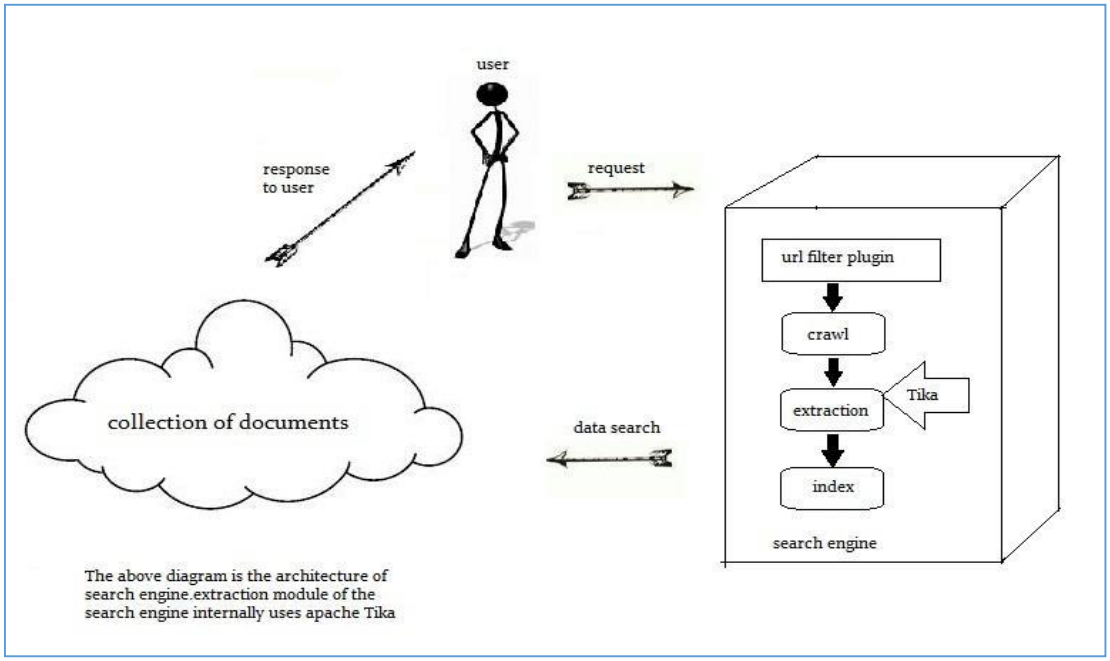
Apache Tika Applications

There are various applications that make use of Apache Tika. Here we will discuss a few prominent applications that depend heavily on Apache Tika.

Search Engines

Tika is widely used while developing search engines to index the text contents of digital documents.

- Search engines are information processing systems designed to search information and indexed documents from the Web.
- Crawler is an important component of a search engine that crawls through the Web to fetch the documents that are to be indexed using some indexing technique. Thereafter, the crawler transfers these indexed documents to an extraction component.
- The duty of extraction component is to extract the text and metadata from the document. Such extracted content and metadata are very useful for a search engine. This extraction component contains Tika.
- The extracted content is then passed to the indexer of the search engine that uses it to build a search index. Apart from this, the search engine uses the extracted content in many other ways as well.



Document Analysis

- In the field of artificial intelligence, there are certain tools to analyze documents automatically at semantic level and extract all kinds of data from them.

- In such applications, the documents are classified based on the prominent terms in the extracted content of the document.

- These tools make use of Tika for content extraction to analyze documents varying from plain text to digital documents.

Digital Asset Management

- Some organizations manage their digital assets such as photographs, e-books, drawings, music and video using a special application known as digital asset management (DAM).

- Such applications take the help of document type detectors and metadata extractor to classify the various documents.

Content Analysis

- Websites like Amazon recommend newly released contents of their website to individual users according to their interests. To do so, these websites follow **machine learning techniques**, or take the help of social media websites like Facebook to extract required information such as likes and interests of the users. This gathered information will be in the form of html tags or other formats that require further content type detection and extraction.

- For content analysis of a document, we have technologies that implement machine learning techniques such as **UIMA** and **Mahout**. These technologies are useful in clustering and analyzing the data in the documents.

- **Apache Mahout** is a framework which provides ML algorithms on Apache Hadoop – a cloud computing platform. Mahout provides an architecture by following certain clustering and filtering techniques. By following this architecture, programmers can write their own ML algorithms to produce recommendations by taking various text and metadata combinations. To provide inputs to these algorithms, recent versions of Mahout use Tika to extract text and metadata from binary content.

- **Apache UIMA** analyzes and processes various programming languages and produces UIMA annotations. Internally it uses Tika Annotator to extract document text and metadata.

History

Year	Development
2006	The idea of Tika was projected before the Lucene Project Management Committee.
2006	The concept of Tika and its usefulness in the Jackrabbit project was discussed.

2007	Tika entered into Apache incubator.
2008	Versions 0.1 and 0.2 were released and Tika graduated from the incubator to the Lucene sub-project.
2009	Versions 0.3, 0.4, and 0.5 were released.
2010	Version 0.6 and 0.7 were released and Tika graduated into the top-level Apache project.
2011	Tika 1.0 was released and the book on Tika "Tika in Action" was also released in the same year.

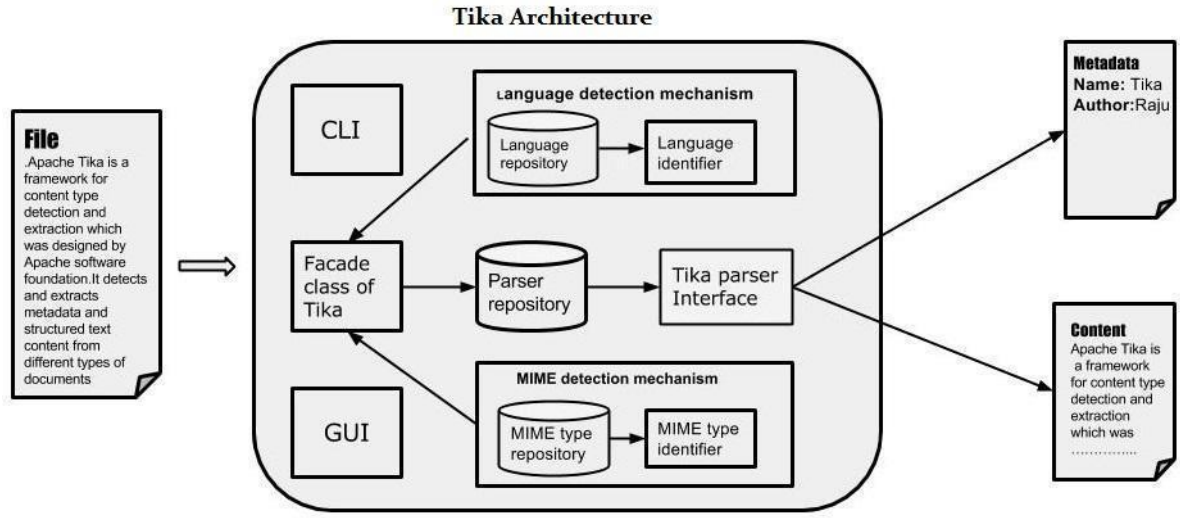
2. TIKA – ARCHITECTURE

Application-Level Architecture of Tika

Application programmers can easily integrate Tika in their applications. Tika provides a Command Line Interface and a GUI to make it user friendly.

In this chapter, we will discuss the four important modules that constitute the Tika architecture. The following illustration shows the architecture of Tika along with its four modules:

- Language detection mechanism
- MIME detection mechanism
- Parser interface
- Tika Facade class



Language Detection Mechanism

Whenever a text document is passed to Tika, it will detect the language in which it was written. It accepts documents without language annotation and adds that information in the metadata of the document by detecting the language.

To support language identification, Tika has a class called **Language Identifier** in the package **org.apache.tika.language**, and a language identification repository inside which contains algorithms for language detection from a given text. Tika internally uses N-gram algorithm for language detection.

MIME Detection Mechanism

Tika can detect the document type according to the MIME standards. Default MIME type detection in Tika is done using **org.apache.tika.mime.mimeTypes**. It uses the **org.apache.tika.detect.Detector** interface for most of the content type detection.

Internally Tika uses several techniques like file globs, content-type hints, magic bytes, character encodings, and several other techniques.

Parser Interface

The parser interface of org.apache.tika.parser is the key interface for parsing documents in Tika. This Interface extracts the text and the metadata from a document and summarizes it for external users who are willing to write parser plugins.

Using different concrete parser classes, specific for individual document types, Tika supports a lot of document formats. These format specific classes provide support for different document formats, either by directly implementing the parser logic or by using external parser libraries.

Tika Facade Class

Using Tika facade class is the simplest and direct way of calling Tika from Java, and it follows the facade design pattern. You can find the Tika facade class in the org.apache.tika package of Tika API.

By implementing basic use cases, Tika acts as a broker of landscape. It abstracts the underlying complexity of the Tika library such as MIME detection mechanism, parser interface, and language detection mechanism, and provides the users a simple interface to use.

Features of Tika

- **Unified parser Interface:** Tika encapsulates all the third party parser libraries within a single parser interface. Due to this feature, the user escapes from the burden of selecting the suitable parser library and use it according to the file type encountered.

- **Low memory usage:** Tika consumes less memory resources therefore it is easily embeddable with Java applications. We can also use Tika within the application which run on platforms with less resources like mobile PDA.

- **Fast processing:** Quick content detection and extraction from applications can be expected.

- **Flexible metadata:** Tika understands all the metadata models which are used to describe files.

- **Parser integration:** Tika can use various parser libraries available for each document type in a single application.

- **MIME type detection:** Tika can detect and extract content from all the media types included in the MIME standards.

- **Language detection:** Tika includes language identification feature, therefore can be used in documents based on language type in a multi lingual websites.

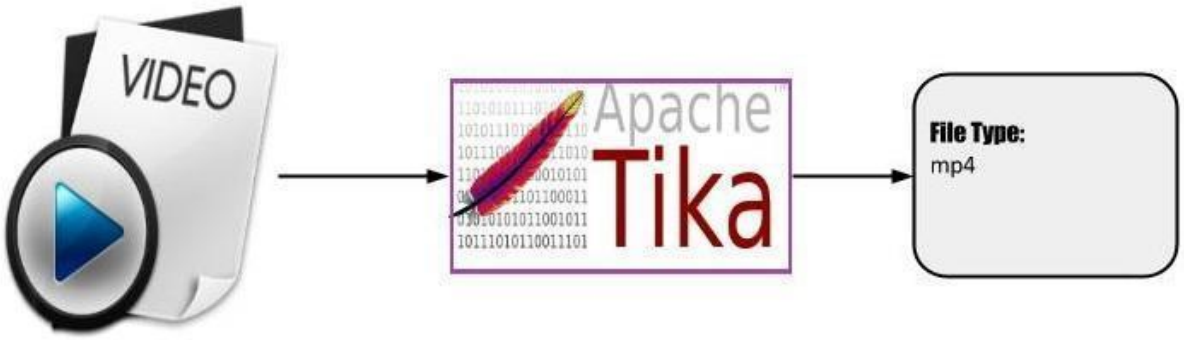
Functionalities of Tika

Tika supports various functionalities:

- Document type detection
- Content extraction
- Metadata extraction
- Language detection

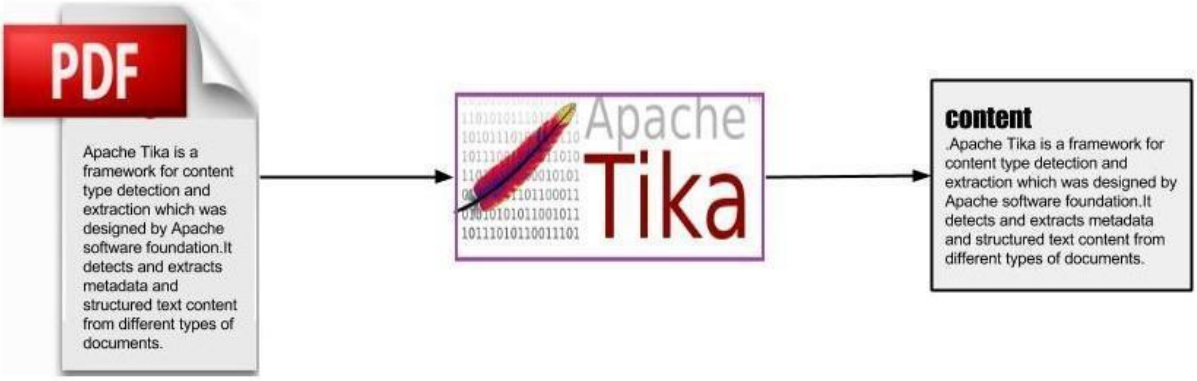
Document Type Detection

Tika uses various detection techniques and detects the type of the document given to it.



Content Extraction

Tika has a parser library that can parse the content of various document formats and extract them. After detecting the type of the document, it selects the appropriate parser from the parser repository and passes the document. Different classes of Tika have methods to parse different document formats.

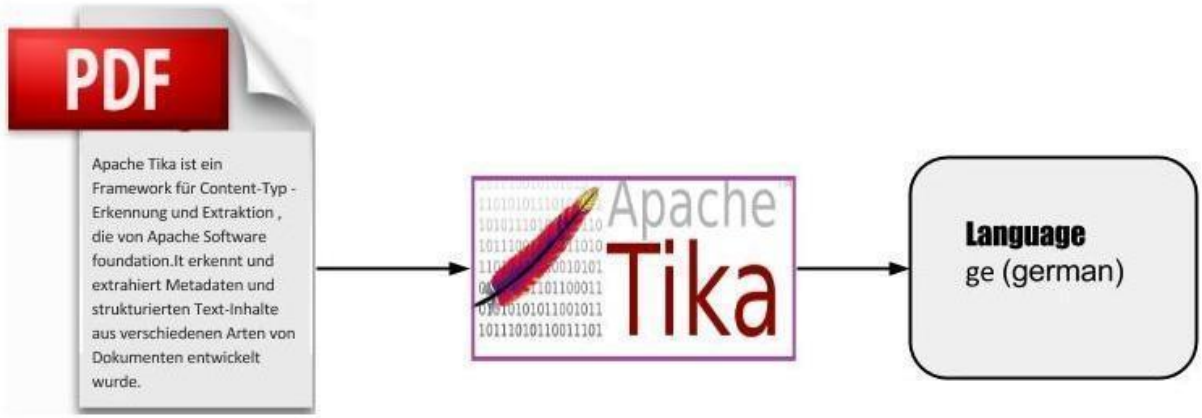


Metadata Extraction

Along with the content, Tika extracts the metadata of the document with the same procedure as in content extraction. For some document types, Tika have classes to extract metadata.

Language Detection

Internally, Tika follows algorithms like **n-gram** to detect the language of the content in a given document. Tika depends on classes like **Languageidentifier** and **Profiler** for language identification.


PDF
Apache Tika ist ein Framework für Content-Typ - Erkennung und Extraktion , die von Apache Software foundation.It erkennt und extrahiert Metadaten und strukturierten Text-Inhalte aus verschiedenen Arten von Dokumenten entwickelt wurde.
Apache
Tika
Language
ge (german)

3. TIKA – ENVIRONMENT

This chapter takes you through the process of setting up Apache Tika on Windows and Linux. User administration is needed while installing the Apache Tika.

System Requirements

JDK	Java SE 2 JDK 1.6 or above
Memory	1 GB RAM (recommeneded)
Disk Space	No minimum requirement
Operating System Version	Windows XP or above, Linux

Step 1: Verifying Java Installation

To verify Java installation, open the console and execute the following **java** command.

OS	Task	Command
Windows	Open command console	\>java –version
Linux	Open command terminal	$java –version

If Java has been installed properly on your system, then you should get one of the following outputs, depending on the platform you are working on.

OS	Output
Windows	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java Hotspot (TM) 64-bit Server VM (build 24.60-b09, mixed mode)
Linux	java version "1.7.0_25" Open JDK Runtime Environment (rhel-2.3.10.4.el6_4-x86_64) Open JDK 64-Bit Server VM (build 23.7-b01, mixed mode)

- We assume the readers of this tutorial have Java 1.7.0_60 installed on their system before proceeding for this tutorial.

- In case you do not have Java SDK, download its current version from http://www.oracle.com/technetwork/java/javase/downloads/index.html and have it installed.

Step 2: Setting Java Environment

Set the JAVA_HOME environment variable to point to the base directory location where Java is installed on your machine. For example,

OS	Output
Windows	Set Environmental variable JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	export JAVA_HOME=/usr/local/java-current

Append the full path of the Java compiler location to the System Path.

OS	Output
Windows	Append the String; C:\Program Files\Java\jdk1.7.0_60\bin to the end of the system variable PATH.
Linux	export PATH=$PATH:$JAVA_HOME/bin/

Verify the command java-version from command prompt as explained above.

Step 3: Setting up Apache Tika Environment

Programmers can integrate Apache Tika in their environment by using

- Command line,
- Tika API,
- Command line interface (CLI) of Tika,
- Graphical User interface (GUI) of Tika, or
- the source code.

For any of these approaches, first of all, you have to download the source code of Tika.

You will find the source code of Tika at http://Tika.apache.org/download.html, where you will find two links:

- apache-tika-1.6-src.zip: It contains the source code of Tika, and
- Tika-app-1.6.jar: It is a jar file that contains the Tika application.

Download these two files. A snapshot of the official website of Tika is shown below.

After downloading the files, set the classpath for the jar file **tika-app-1.6.jar**. Add the complete path of the jar file as shown in the table below.

OS	Output
Windows	Append the String "C:\jars\Tika-app-1.6.jar" to the user environment variable CLASSPATH
Linux	Export CLASSPATH=$CLASSPATH: /usr/share/jars/Tika-app-1.6.tar:

Apache provides Tika application, a Graphical User Interface (GUI) application using Eclipse.

Tika-Maven Build using Eclipse

- Open eclipse and create a new project.
- If you do not having Maven in your Eclipse, set it up by following the given steps.