COMP4321 Final Documentation

Team Members: Li Zhihang
Zhang Xiang
Wang Yiquan

Overall Design

The project is a web based search engine consisting of 3 parts: user interface, search engine and database.

The database contains spider, indexer where the spider first crawl web pages and then pass the pages to indexer. The indexer will first do pre-process on each page's text content including stopword removal and stemming. Then each term will be stored in an inverted file which consists of (term ,posting-list) pairs.

Note that a posting list is a list documents which contains this term and the positions of this term is stored to support the phrase search function. Other information of the page like URL, last modification date will be stored in a forward file. After finish indexing all the pages, each page's PageRank score is calculated and will be used in search engine part.

The search engine receive query from user interface and process the query including stopword removal, stemming and phrase detection. Then it will use vector space model to find the most relevant pages in database. Other rank score like PageRank score, title matching is also considered when decide which pages should be retrieve.

The user interface is responsible for interact with users. User can input query through browser and submit the query to server and the interface will display all the relevant pages in the browser.

File Structure in Database

Inverted File: This is a class where contains Page Content Map and Page Title Map. In this class, it provides functions for other component to have access to these two Maps for functioning such as adding values in (addEntry) and Deletions (deleteEntry). Constructing this class is for over controlling two maps so that it simplifies control flow.

Forward File: This class is designed to store page specific information like title, URL, size and last modification date. It also stores some global statistic information like the tf max of a specific document. It as well contains all the terms a document has to support fast implementation of delete document operation. Relationships between pages are stored in this file since each forward file contains its child page's URLs so we can use the information to access its child pages.

URL and ID Map: The class called UrlToIdMap designed a map that stores URLs of a page with ID matched to each single of them uniquely. This enables us to have a faster access to ID of the page once we have an URL. In this class, method of addEntry is provided for other component of the project to have access to the map, allowing them to add new pairs in. It also enables function that checks whether an URL exists.

Page Content Map: This map is realized in the Inverted File class. It is designed to make a direct connection between a word on a particular webpage and the word's information such as how many times it appears on this page and its position on the page. The information contained related to this word is stored using a linkedlist. We can obtain the detailed information as soon as a word is provided.

Page Title Map: This map has similar functionality with Page Content Map except it only processes page titles.

Algorithms Used

cosine similarity based vector space.

Page rank.

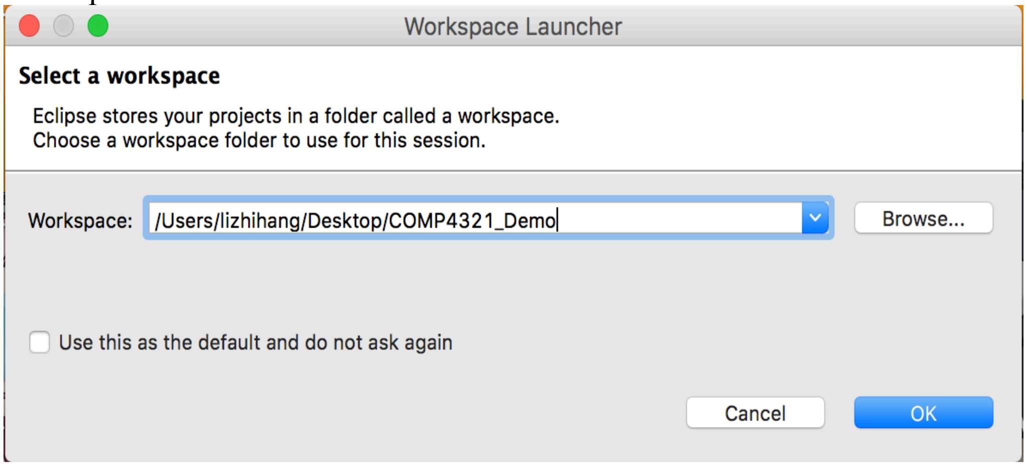
Stopword removal

Porter's algorithm

Details: Given a query, we preprocess it including stop word removal and stemming. For every word(phrase) in the query, we compute its weight as shown in the lecture slides and calculate cosine similarity. If there's match in the title, we increase the score dramatically by 0.3. Finally, we combine the cosine score with its page rank(summation) and return top 50 records or less if not so many documents are related to query).

Installation procedure

1. Download zip file and unzip it.
2. Open eclipse and set the the folder where the unzipped file is located as Workspace



Workspace Launcher

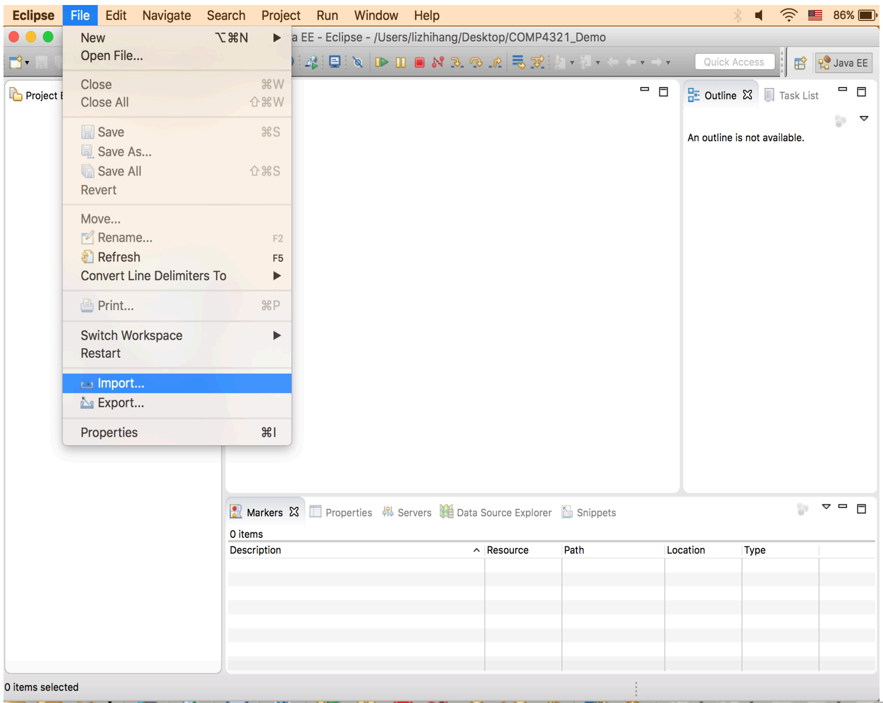
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/lizhihang/Desktop/COMP4321_Demo

Use this as the default and do not ask again

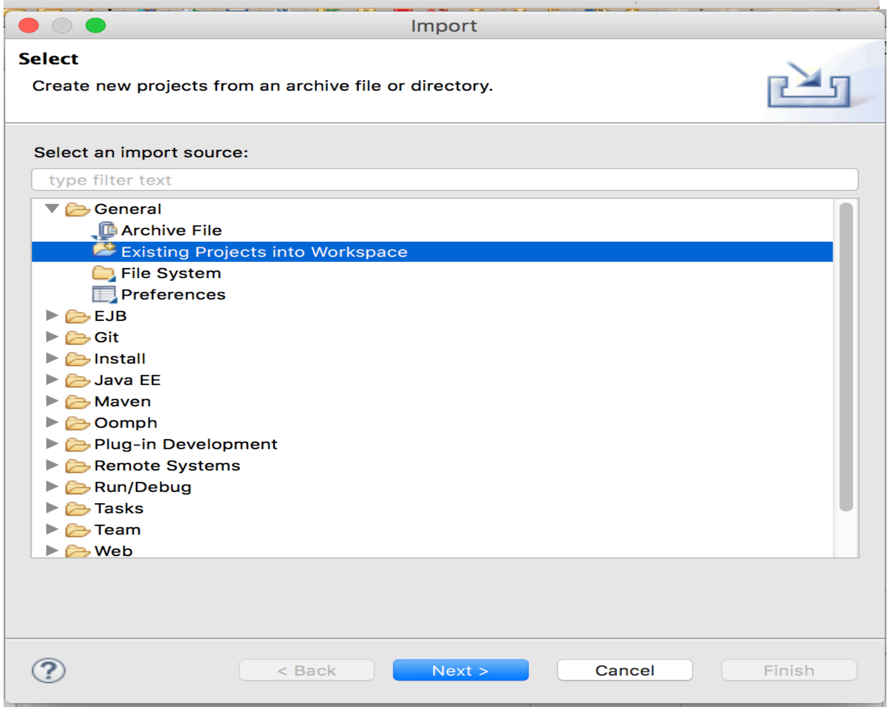
3. Click File -> Import…, then choosing "existing projects into workspace" under General, then click Next. After this, browse the unzipped file in this folder, then Finish.



An outline is not available.

0 items

Description	Resource	Path	Location	Type



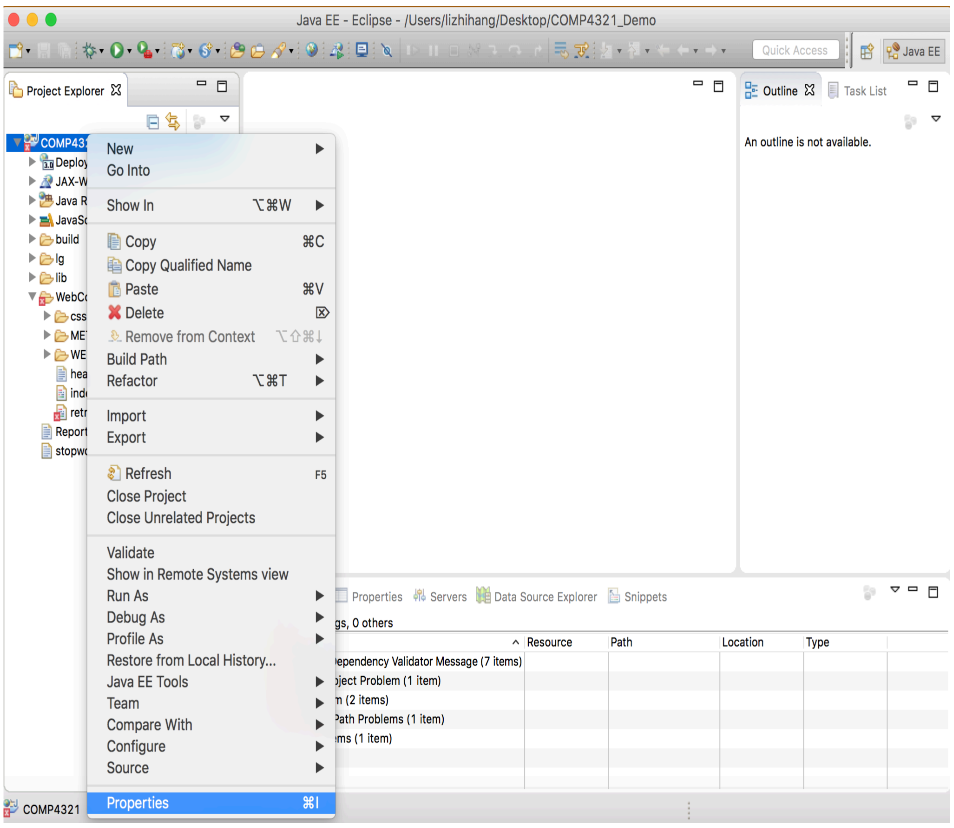
Import

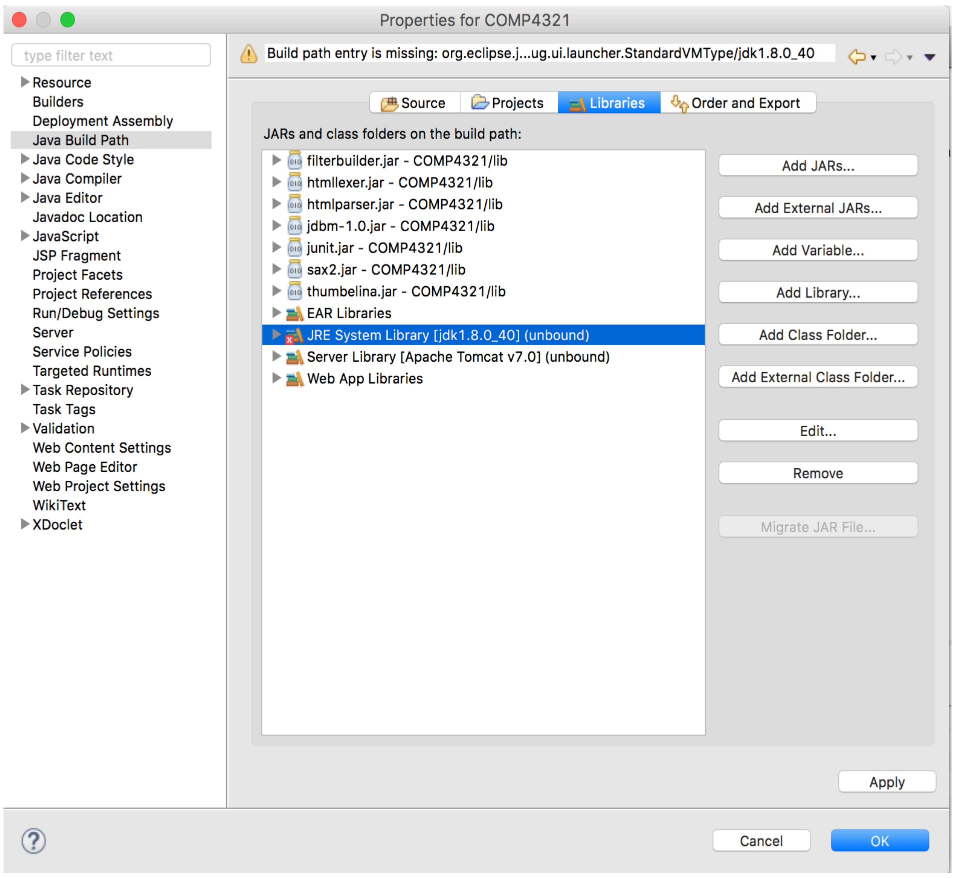
Select

Create new projects from an archive file or directory.

Select an import source:

4. There is potential problem of existing error due to the fact that libraries version and path may be different in different computer. So a user may need to reset them on his or her own computer. You may do the following: right click on the project and select “Properties”, then go to “Java Build Path” and choose “Libraries”. Remove the one that contains error (most likely JRE system library). And then click on “Add library…”, select JRE System Library if that is the case, then click on Finish.

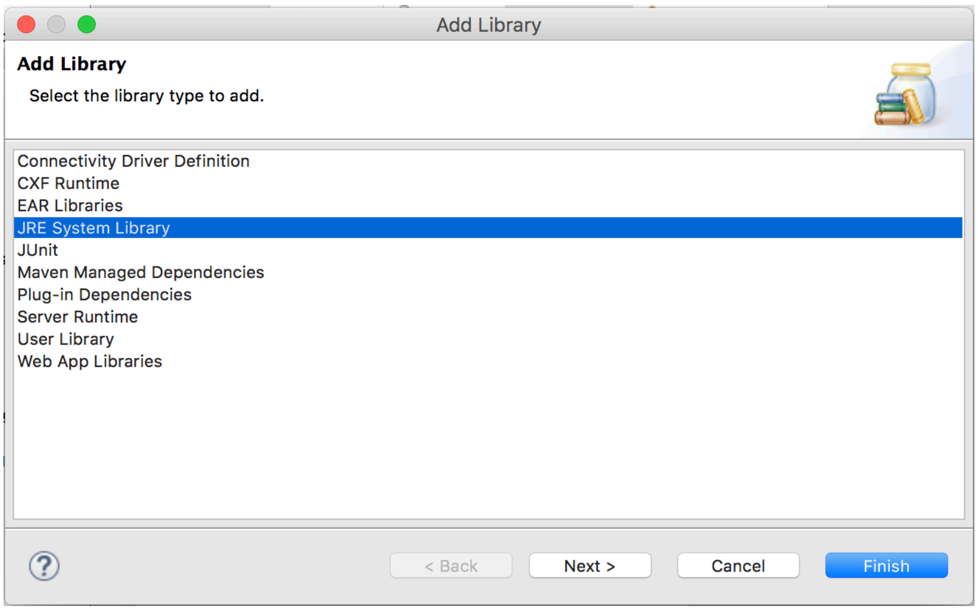




Build path entry is missing: org.eclipse.j...ug.ui.launcher.StandardVMType/jdk1.8.0_40

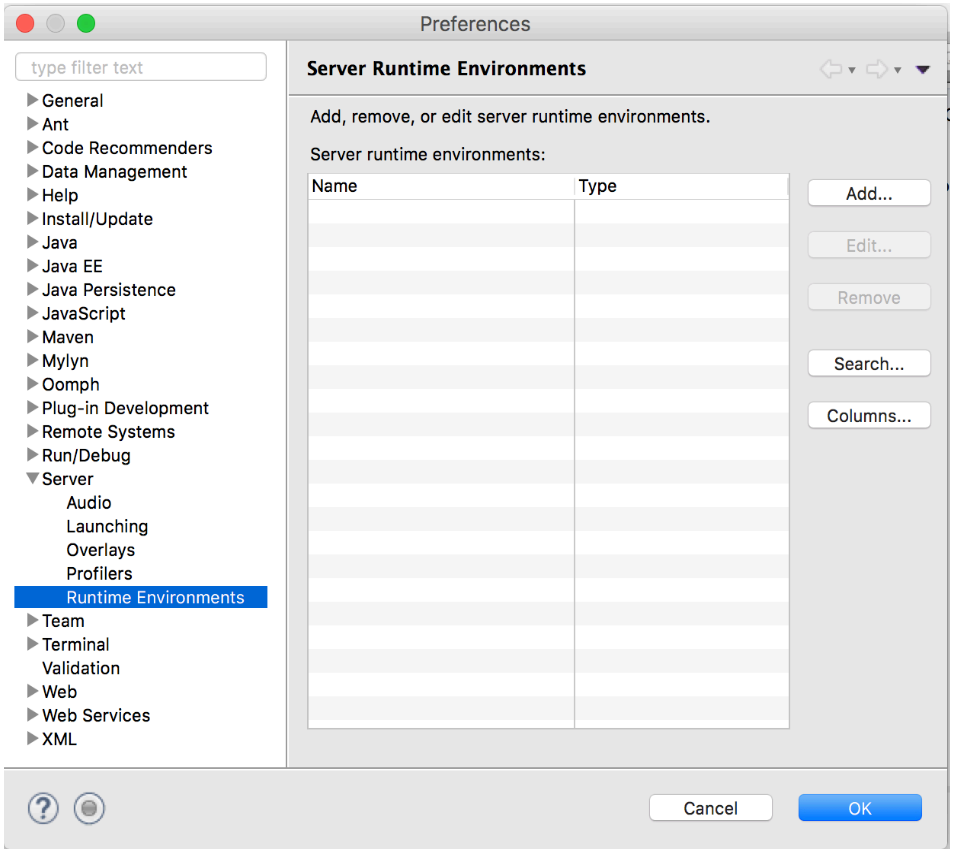
JARs and class folders on the build path:

- filterbuilder.jar - COMP4321/lib
- htmllexer.jar - COMP4321/lib
- htmlparser.jar - COMP4321/lib
- jdbm-1.0.jar - COMP4321/lib
- junit.jar - COMP4321/lib
- sax2.jar - COMP4321/lib
- thumbelina.jar - COMP4321/lib
- EAR Libraries
- JRE System Library [jdk1.8.0_40] (unbound)
- Server Library [Apache Tomcat v7.0] (unbound)
- Web App Libraries



Connectivity Driver Definition
CXF Runtime
EAR Libraries
JRE System Library
JUnit
Maven Managed Dependencies
Plug-in Dependencies
Server Runtime
User Library
Web App Libraries

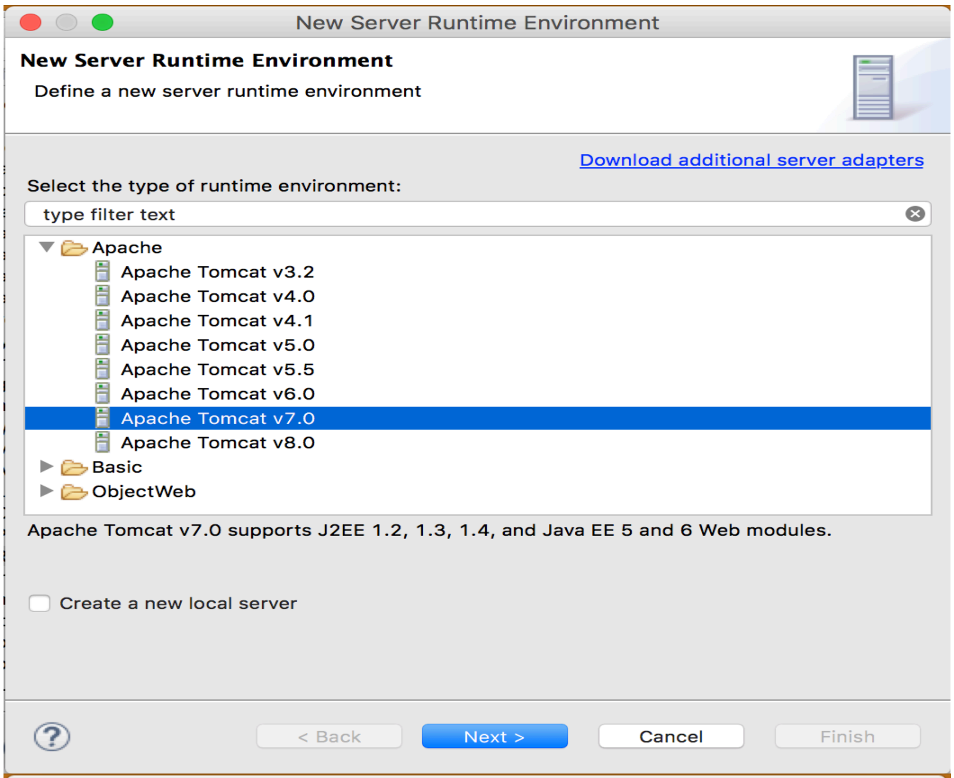
5. The next step is to add Apache Tomcat server, here we use version 7.0. Click preference of eclipse, choose "Runtime Environments" under "Server". Then click on "Add", select Apache Tomcat 7.0 and then click on Next. Browse the folder "apache-tomcat-7.0.69" located in the the project folder then Finish. The error now should be resolved and project is ready to run.



Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type


New Server Runtime Environment
Download additional server adapters
< Back
Next >
Cancel
Finish

New Server Runtime Environment

Define a new server runtime environment

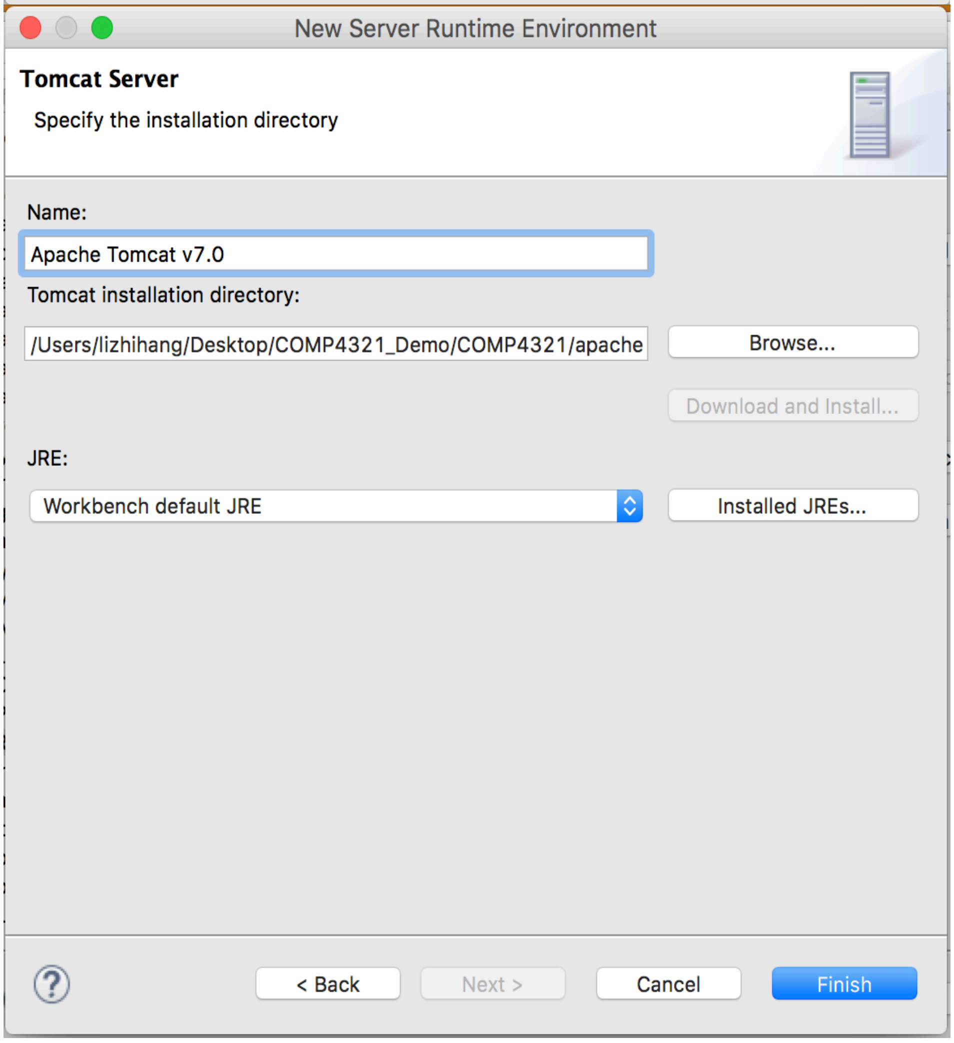
Select the type of runtime environment:

type filter text

- Apache
  - Apache Tomcat v3.2
  - Apache Tomcat v4.0
  - Apache Tomcat v4.1
  - Apache Tomcat v5.0
  - Apache Tomcat v5.5
  - Apache Tomcat v6.0
  - Apache Tomcat v7.0
  - Apache Tomcat v8.0
- Basic
- ObjectWeb

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

☐ Create a new local server



Tomcat Server

Specify the installation directory

Name:

Apache Tomcat v7.0

Tomcat installation directory:

/Users/lizhihang/Desktop/COMP4321_Demo/COMP4321/apache

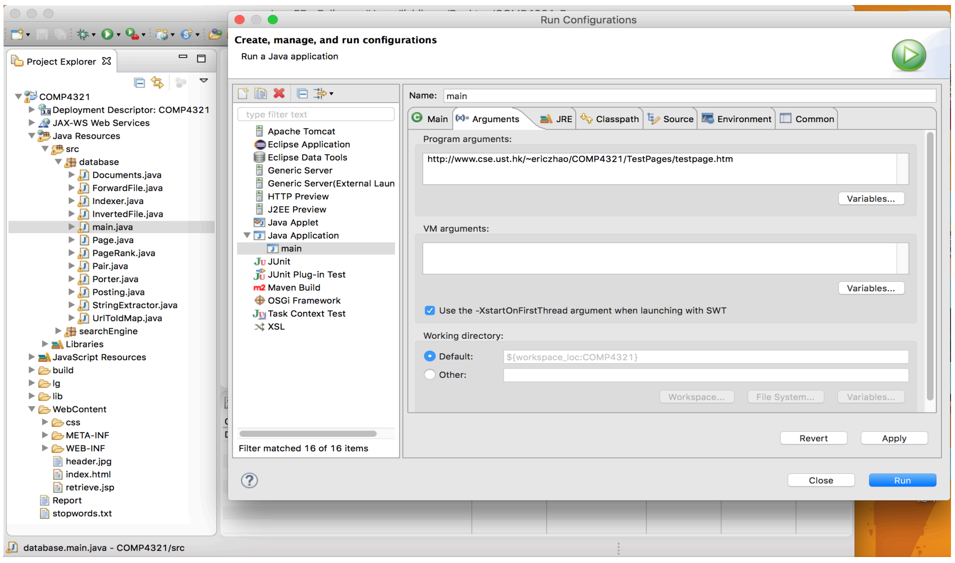
JRE:

Workbench default JRE

6. Note that Running the project should be under Java EE environment. Also, the user needs to change path that where files will be generating by this project in 4 Java files: Indexer.java, InvertedFile.java, ForwardFile.java and UrlToIdMap.java. The way to change it is to change the variable "filepath" into the location of the project where it is located. This variable is located at the very beginning of these 4 files.

```
public final String filepath = "/Users/lizhihang/Desktop/COMP4321_Demo/COMP4321";
```

7. Next step is to use spider to get web pages from online. Right click on main.java under Java Resources -> database, select Run as… and then click on Run Configurations. Select main under Java Application. Paste the beginning site address into Program Arguments under Arguments, for example, the given test site "http://www.cse.ust.hk/~ericzhao/COMP4321/TestPages/testpage.htm". Then click on Run.



Run Configurations

Create, manage, and run configurations

Run a Java application

Name: main

Program arguments:

```
http://www.cse.ust.hk/~ericzhao/COMP4321/TestPages/testpage.htm
```

VM arguments:

[✓] Use the -XstartOnFirstThread argument when launching with SWT

Working directory:

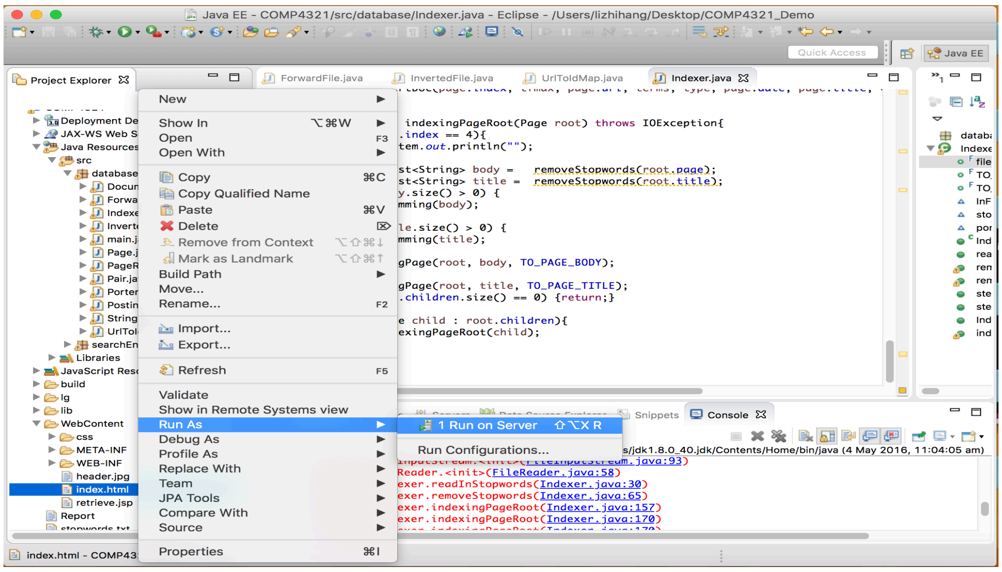
(●) Default: ${workspace_loc:COMP4321}

( ) Other:

Filter matched 16 of 16 items

database.main.java - COMP4321/src

8. After finish running main.java, the search engine is ready to run. Right click on index.html under WEB-INF in WebContent, then click Run as… and run on server. Manually select Tomcat 7.0 Server then Finish.

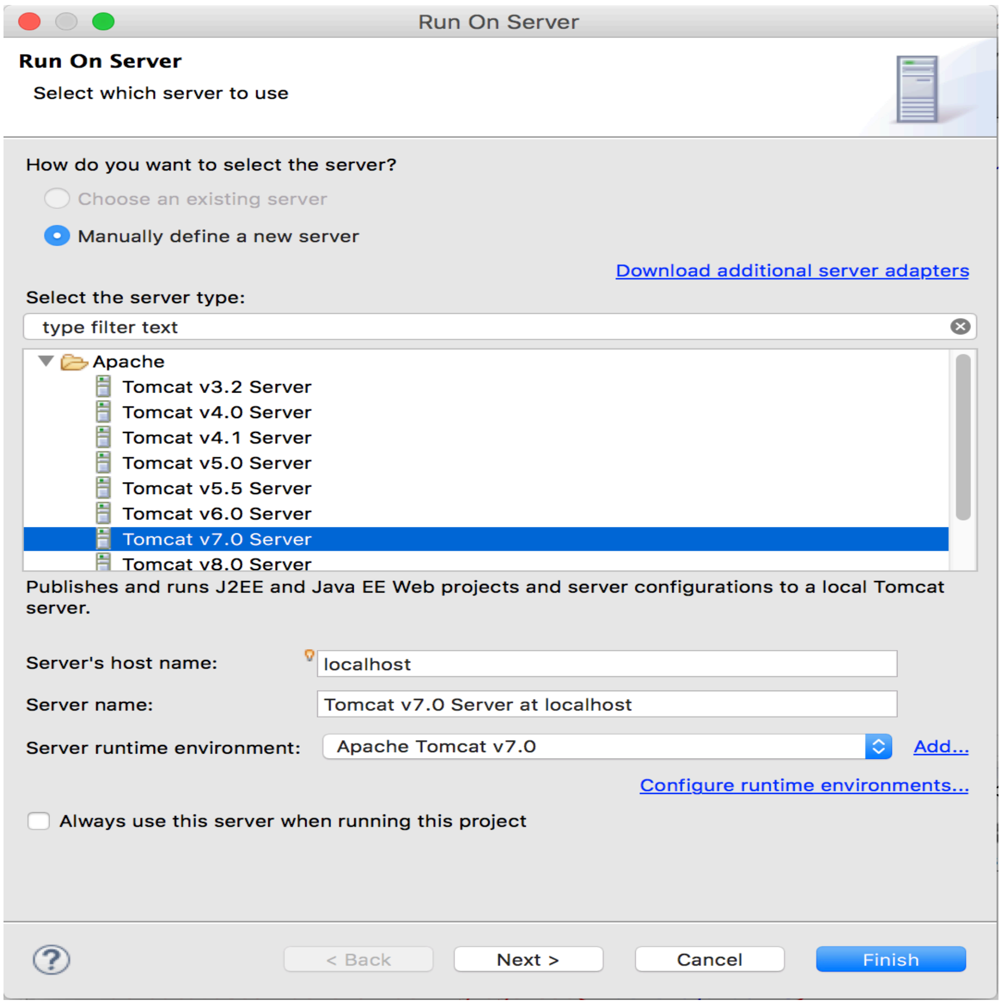


Java EE - COMP4321/src/database/Indexer.java - Eclipse - /Users/lizhihang/Desktop/COMP4321_Demo

```
 indexingPageRoot(Page root) throws IOException{
.index == 4){
tem.out.println("");

st<String> body =  removeStopwords(root.page);
st<String> title =  removeStopwords(root.title);
y.size() > 0) {
mming(body);

le.size() > 0) {
mming(title);

gPage(root, body, TO_PAGE_BODY);

gPage(root, title, TO_PAGE_TITLE);
.children.size() == 0) {return;}

e child : root.children){
exingPageRoot(child);
```

```
s/jdk1.8.0_40.jdk/Contents/Home/bin/java (4 May 2016, 11:04:05 am)
Reader.<init>(FileReader.java:58)
exer.readInStopwords(Indexer.java:30)
exer.removeStopwords(Indexer.java:65)
exer.indexingPageRoot(Indexer.java:157)
exer.indexingPageRoot(Indexer.java:170)
```

index.html - COMP43



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

- ( ) Choose an existing server
- [✓ Manually define a new server]

Download additional server adapters

Select the server type:

type filter text

- Apache
  - Tomcat v3.2 Server
  - Tomcat v4.0 Server
  - Tomcat v4.1 Server
  - Tomcat v5.0 Server
  - Tomcat v5.5 Server
  - Tomcat v6.0 Server
  - [✓ Tomcat v7.0 Server]
  - Tomcat v8.0 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

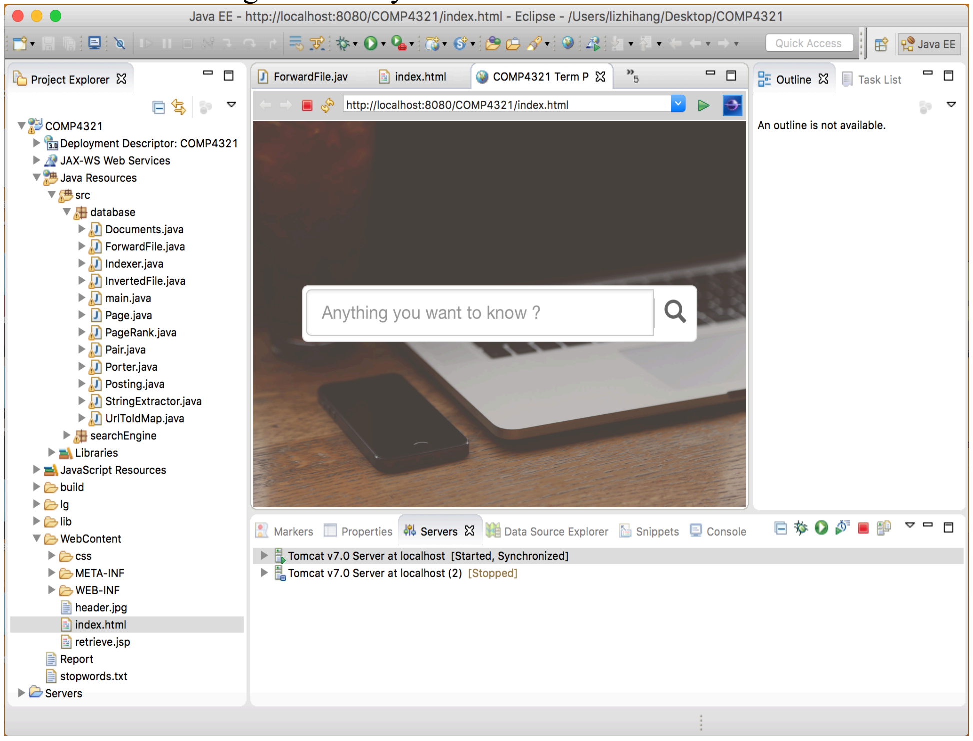
Server's host name:	localhost
Server name:	Tomcat v7.0 Server at localhost
Server runtime environment:	Apache Tomcat v7.0

Add...

Configure runtime environments...

[ ] Always use this server when running this project

9. The web search engine is ready to run.



Java EE - http://localhost:8080/COMP4321/index.html - Eclipse - /Users/lizhihang/Desktop/COMP4321

http://localhost:8080/COMP4321/index.html

An outline is not available.

```
COMP4321
  Deployment Descriptor: COMP4321
  JAX-WS Web Services
  Java Resources
    src
      database
        Documents.java
        ForwardFile.java
        Indexer.java
        InvertedFile.java
        main.java
        Page.java
        PageRank.java
        Pair.java
        Porter.java
        Posting.java
        StringExtractor.java
        UrlToIdMap.java
      searchEngine
    Libraries
  JavaScript Resources
  build
  lg
  lib
  WebContent
    css
    META-INF
    WEB-INF
    header.jpg
    index.html
    retrieve.jsp
  Report
  stopwords.txt
Servers
```

```
Tomcat v7.0 Server at localhost  [Started, Synchronized]
Tomcat v7.0 Server at localhost (2)  [Stopped]
```

Highlight of features

We implement a user-friendly interface using bootstrap and it can be visited both on PC and mobile phone.

We implement the PageRank algorithm and important pages (contains many useful links) will be favored in the retrieve process.

Our spider works at an exceedingly good speed.

Testing of the functions implemented.

We include a lg folder which record the test of database, search engine.

The FFtest is for testing forward file. You can view what pages we spider from website.

The IVFtest is for testing inverted file. You can view all the terms we indexed in our database

The PageRankTest is for testing pagerank implementation. You can view all the pages' PageRank score.

The image UI, query and search result are for testing the UI and search engine besed on web

Conclusion

Our search engine is quite robust because we handle user query carefully and deal with many unexpected situations.

Our search engine is exceedingly fast because we handle frontend – backend interaction carefully and our retrieve process are also quite fast because the page rank is precomputed and store in system.

We would like to add features to do personalization search like using search history, constructing user profile and using collaborative filtering system to do recommendation to user.