Twitter Crawler

Rashid Goshtasbi and Brendan Cheng

SID: 861056442 and 861055750

Hello there! T-Crawler is an open source project for Linux to download tweets over twitter that has geolocation enabled. These tweets are indexed and can be retrieved by search through a Java servlet that acts as a search engine.

Collaboration Details

Description of contributions of each team member

Rashid

- Researched on how to configure Apache Tomcat with Eclipse
- Researched on how .XML, .JSP, and .Java files interact
- Configured server host for Tomcat
- Configured internal and external .jar libraries
- Ranked indexes based on search query
- Configure Lucene to accept JSON Files
- Convert JSON objects to strings
- Setup string values for index search algorithm to pass from .java to .jsp files to display to webpage
- Setup core .jar libraries
- Files: LuceneExample.java, textOutput.java, search_engine.jsp, json.jar, lucene.jar

Brendan

- Researched on how to configure Apache Tomcat with Eclipse
- Researched on how .XML, .JSP, and .Java files interact
- Configured server host for Tomcat
- Setup XML file to deploy descriptors for our servlet
- Configured .JSP and .HTML files to create a UI interface
- Convert JSON objects to strings
- Setup string values for index search by concatenation

- Files: LuceneExample.java, textOutput.java, search_engine.jsp, web.xml, search.html

Overview of System

- **Architecture:**
    - LuceneExample.java:
        - WebDocument class
            - Organizes string of id, text, and created at for storing into index files
        - Main goes through each JSON file and casts it into an object variable to allow for parsing into an index
        - ReadJSON
            - Takes in the files at directory location and retrieves each type of object file such as created at, id, and twitter text.
            - sends in the string values of created at, id, and twitter text and passes it into the WebDocument class and then index's that information
        - Index Class
            - Indexes the information from WebDocument and creates fields to read for the search
    - web.xml:
        - Deploys descriptor for our Java servlet
    - search_engine.jsp:
        - Accepts ranked indexes and outputs results
    - search.html:
        - Provides the interface for our Java Servlet
        - Text box, search button, headers, titles, background colors
    - textOutput.java
        - Ranks the indexes
        - Takes the variable that the user types into the search engine box and uses those words in a query and searches the index for the top 10 relevant tweets and outputs it on the screen for the user to see
- **Index Structures**
    - We take an input reader that reads the json data files at the specified location
    - From there we input each line of the json file and insert the line into an function of ArrayList of JSON

    Object files
    - There we cast each object we need into a string and pass those into a class provided from our TA that will create a "page"
    - Afterwards we pass that page into our index function
    - There we manipulated this function to create fields such as "text", "id", "username", and "created at" so we can later pull them when we are searching for words from our query.
    - We modify the way Lucene indexes our tweets through the use of .setBoost(). Through the use of this function, we assign each field of our tweets with a weight in order to give priority to text, followed by id.

- **Search Algorithm**

    - We use the built-in functionality of Lucene. Lucene indexes our files by breaking our tweets into terms that are generated using a built-in analyzer. An index file is created to contain these terms. When a query is received, it is processed through the same analyzer to look for matching terms within the index file. A list of tweets that have a match with the query is then created.

- **Libraries:**

    - Java.io
    - Java.lang
    - Java.util
    - JSON Simple
    - Apache Lucene
    - Tomcat
    - Java Server Pages

Limitations

- Biggest limitation was for us to parse JSON array objects which carried more information such as location to use google maps API
- Need to install all relevant libraries
- Returns 10 search results, no additional pages of results
- Results are tweets, not links
- Formatting of the outputs aren't simplified, uses "::" to spread the created dates, user id and tweet text

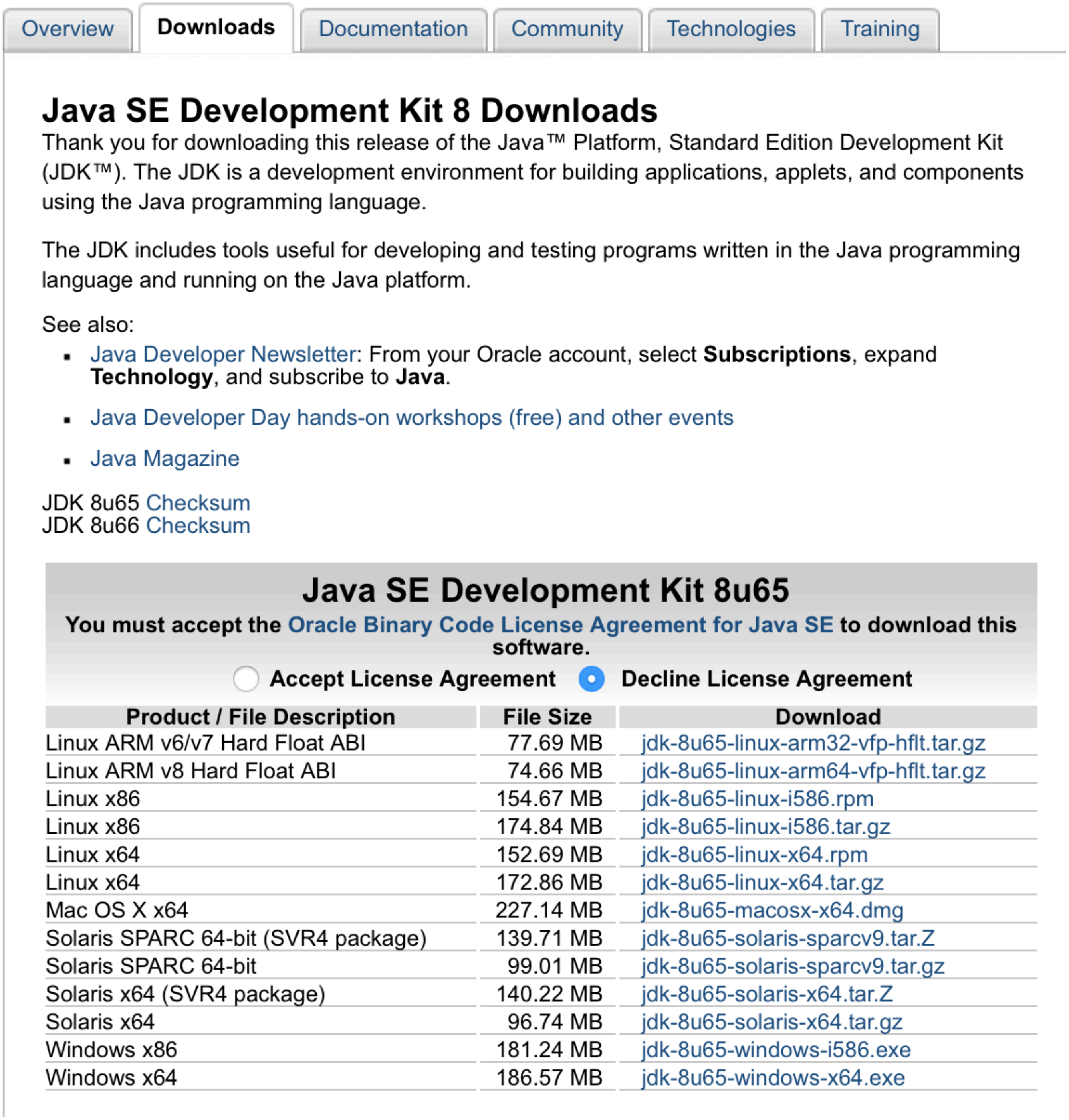
Instructions

Installation and Running of Tcrawler on Eclipse Browser and/or LocalHost

Browser "localhost:8080/ui_search/search.html"

1. Take our tcrawler_project.zip folder and un-zip it to a location that is easily accessible for you.
2. Unzip Apache Tomcat file from our tcrawler_project
    - Our Tomcat file has been modified to include lucene.jar file library to handle exceptions.
    - If you wish to use a different Tomcat file, it will not handle exceptions in our servlet

3. Download Java SE Development Kit from: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
    - Version 8 is preferred with our program Inline-style:



Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u65 Checksum
JDK 8u66 Checksum

Java SE Development Kit 8u65

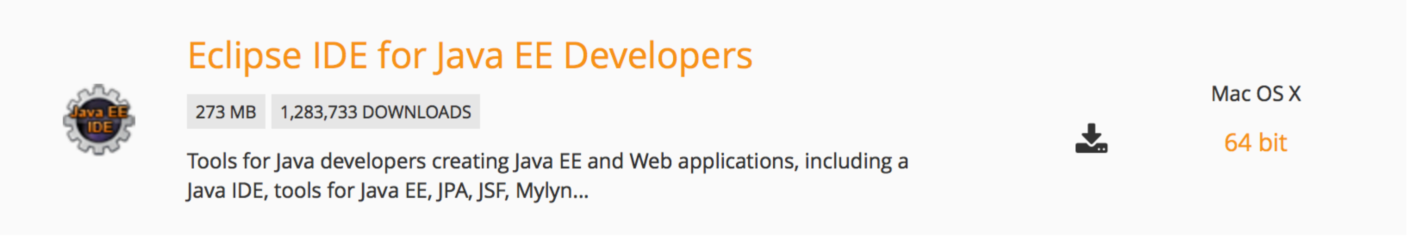
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ **Accept License Agreement** ● **Decline License Agreement**

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u65-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.66 MB	jdk-8u65-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.67 MB	jdk-8u65-linux-i586.rpm
Linux x86	174.84 MB	jdk-8u65-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u65-linux-x64.rpm
Linux x64	172.86 MB	jdk-8u65-linux-x64.tar.gz
Mac OS X x64	227.14 MB	jdk-8u65-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.71 MB	jdk-8u65-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.01 MB	jdk-8u65-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.22 MB	jdk-8u65-solaris-x64.tar.Z
Solaris x64	96.74 MB	jdk-8u65-solaris-x64.tar.gz
Windows x86	181.24 MB	jdk-8u65-windows-i586.exe
Windows x64	186.57 MB	jdk-8u65-windows-x64.exe

-

4. Install Eclipse from: https://eclipse.org/downloads/ Inline-style:



Eclipse IDE for Java EE Developers

273 MB 1,283,733 DOWNLOADS

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

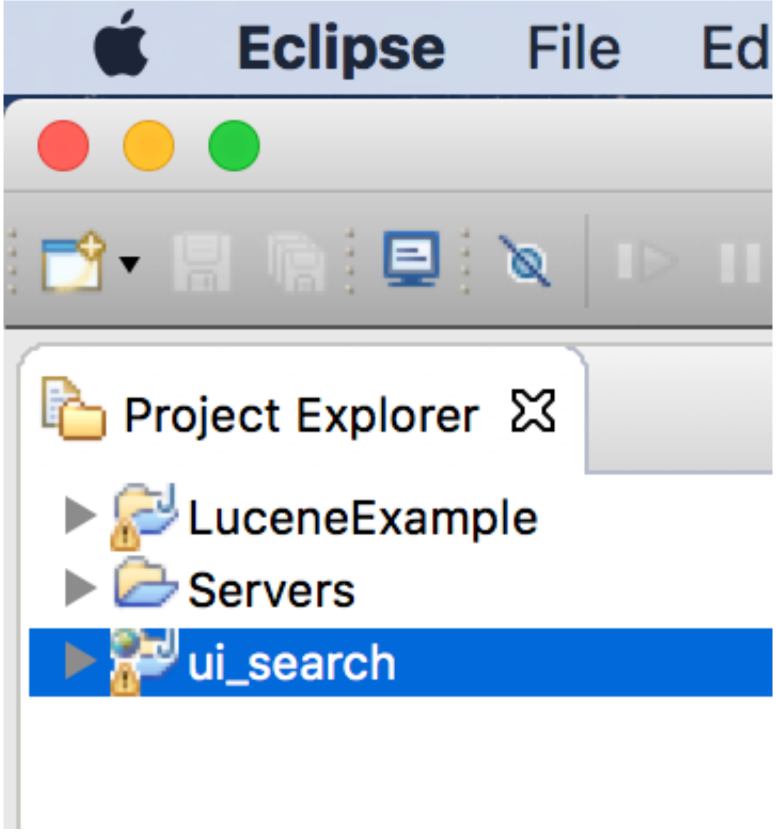
Mac OS X

64 bit

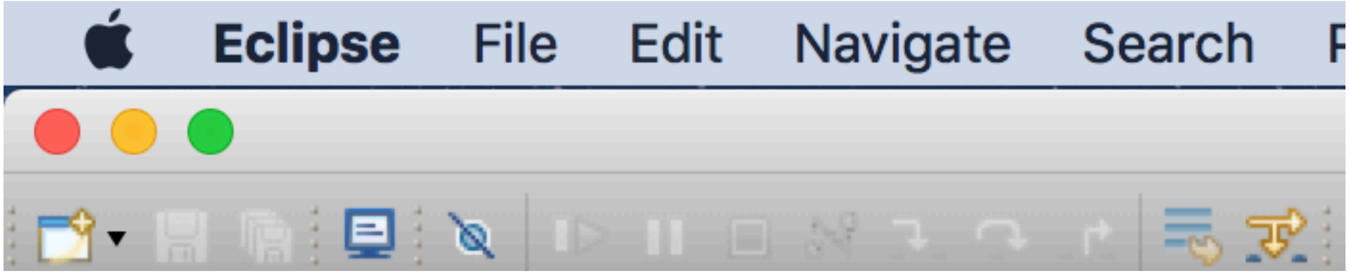
-

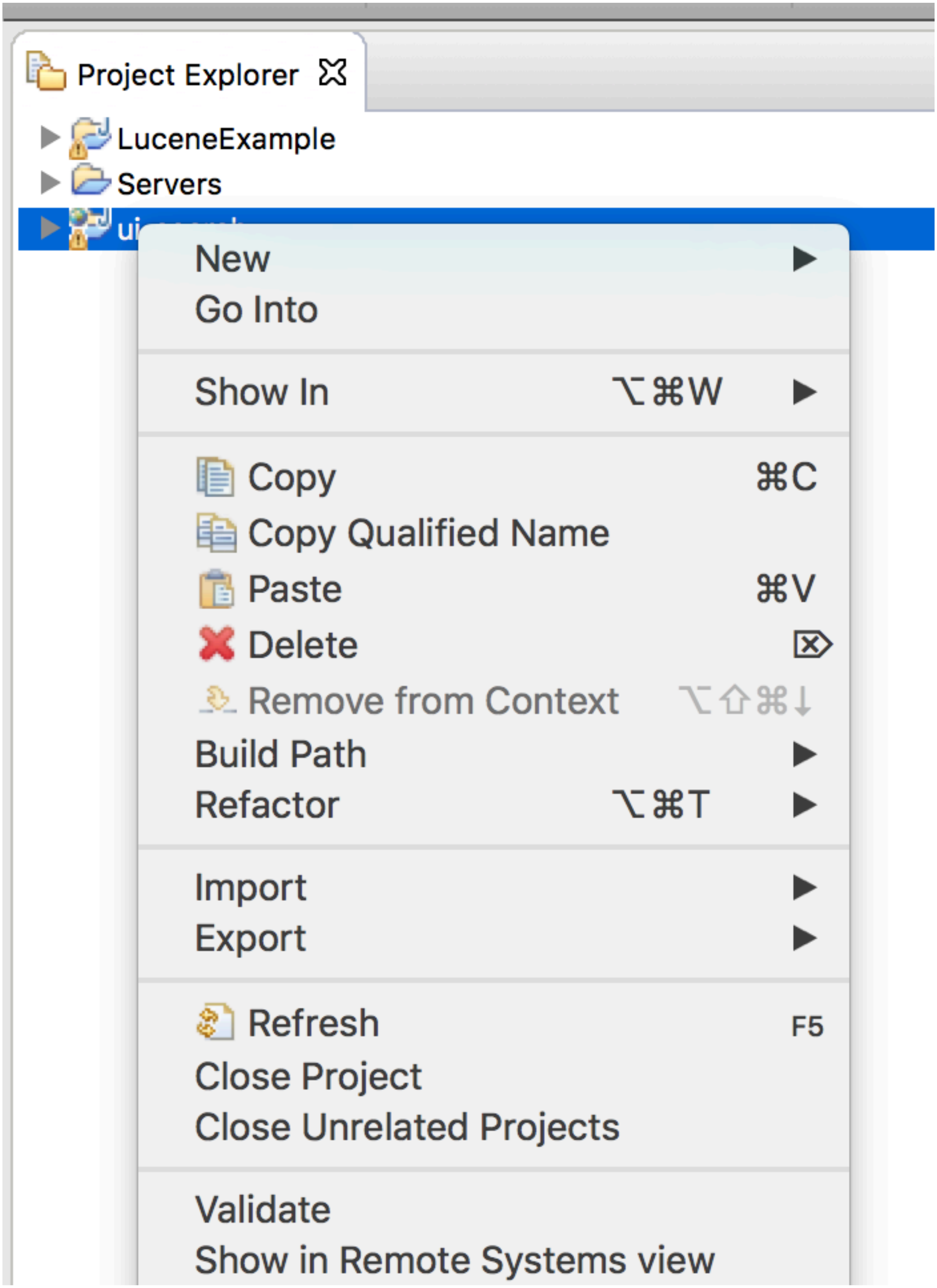
5. Import tcrawler project into Eclipse
   - Open eclipse and select File > Import > General > Archive File and click next and under "From archive file:" select the tcrawler.zip file in the provided tcrawler*project folder you un-zipped earlier. Eclipse will import the .zip as "ui*search" in your Eclipse workspace.
   - On our workspace, select the Servers tab and select the option to create a new server. Select Tomcatv8.0, then select the Tomcat directory we provided for you.
   - After the server is created, you will see it listed under the servers tab. Right click and go to Properties, and click on "change the location." Apply and press ok.
   - Under the server tab, Double click the server listed for Tomcat and under Server Locations, select the option for "Use Tomcat installation(takes control of Tomcat installation). Make sure the port name HTTP.1.1 corresponds to the 8080 Port Number.

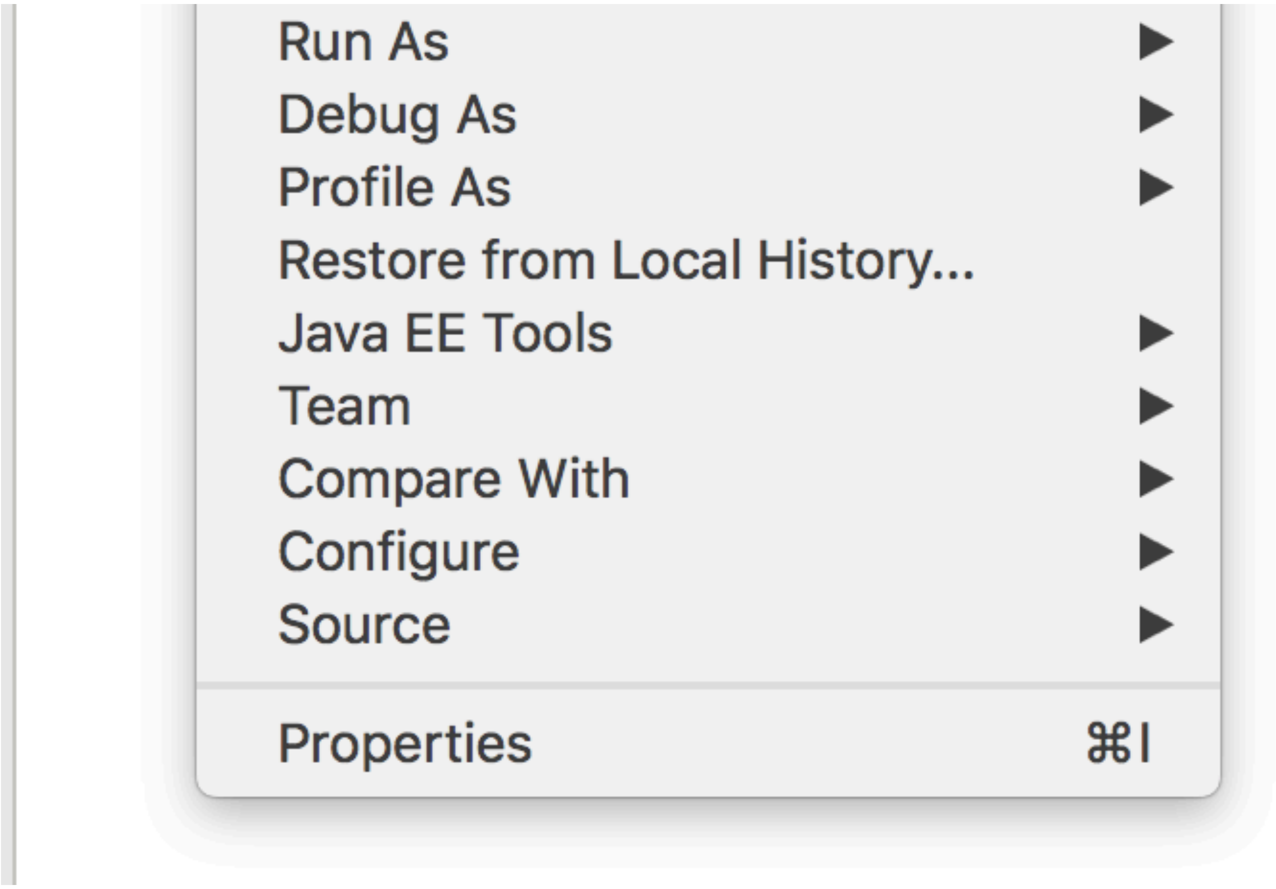
6. Importing .jar file libraries for project files
   - Select the project folder for ui_search in Eclipse, right-click and select properties.
   - Under "Java Build Path," select "Add External JARs..." and select the two .jar file included in the zip folder we provided you.



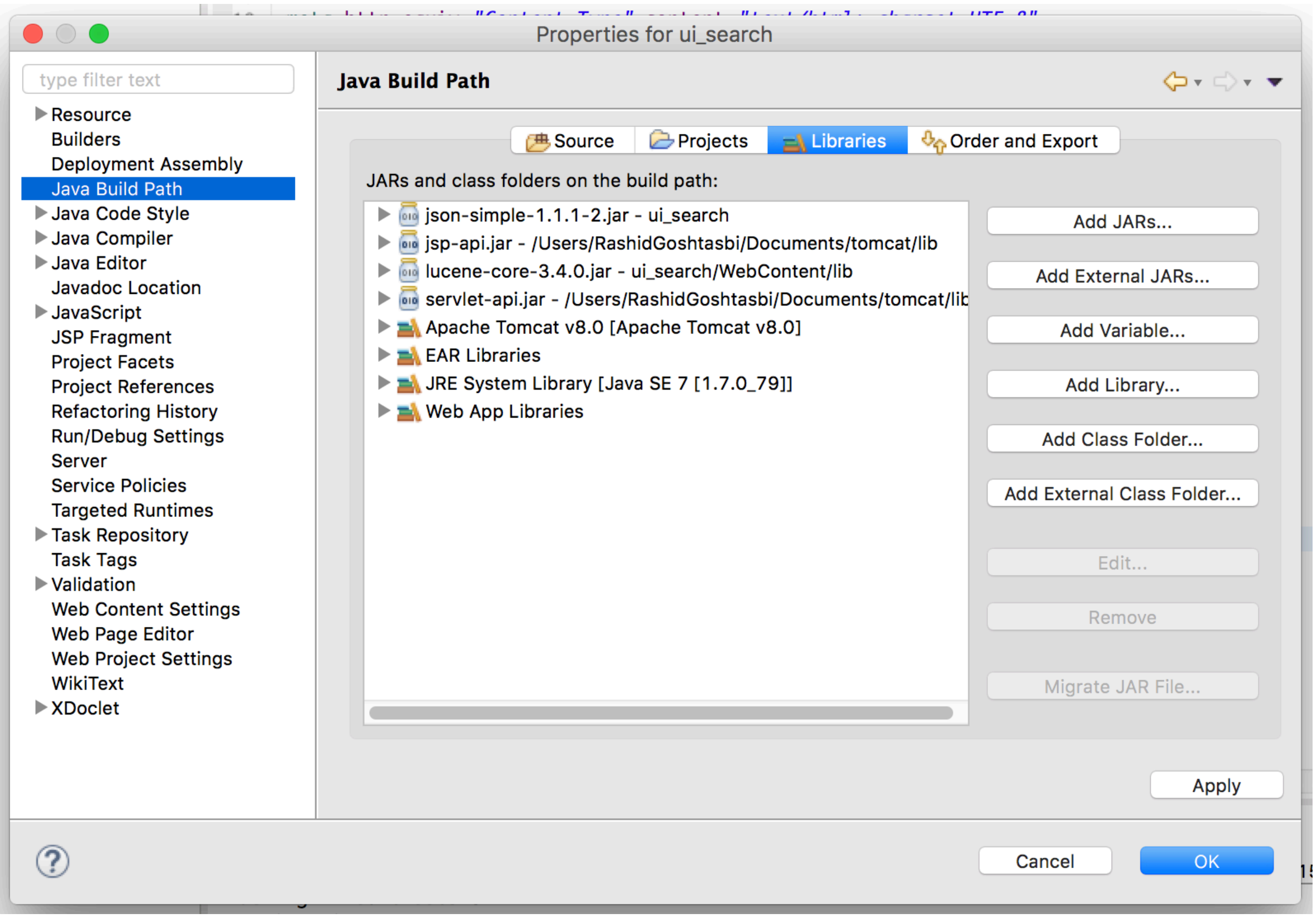
   - Click "Apply", then "OK". -




Project Explorer
LuceneExample
Servers
ui
New
Go Into
Show In ⌥⌘W
Copy ⌘C
Copy Qualified Name
Paste ⌘V
Delete ⌦
Remove from Context ⌥⇧⌘↓
Build Path
Refactor ⌥⌘T
Import
Export
Refresh F5
Close Project
Close Unrelated Projects
Validate
Show in Remote Systems view


Run As ▶
Debug As ▶
Profile As ▶
Restore from Local History...
Java EE Tools ▶
Team ▶
Compare With ▶
Configure ▶
Source ▶
Properties ⌘I

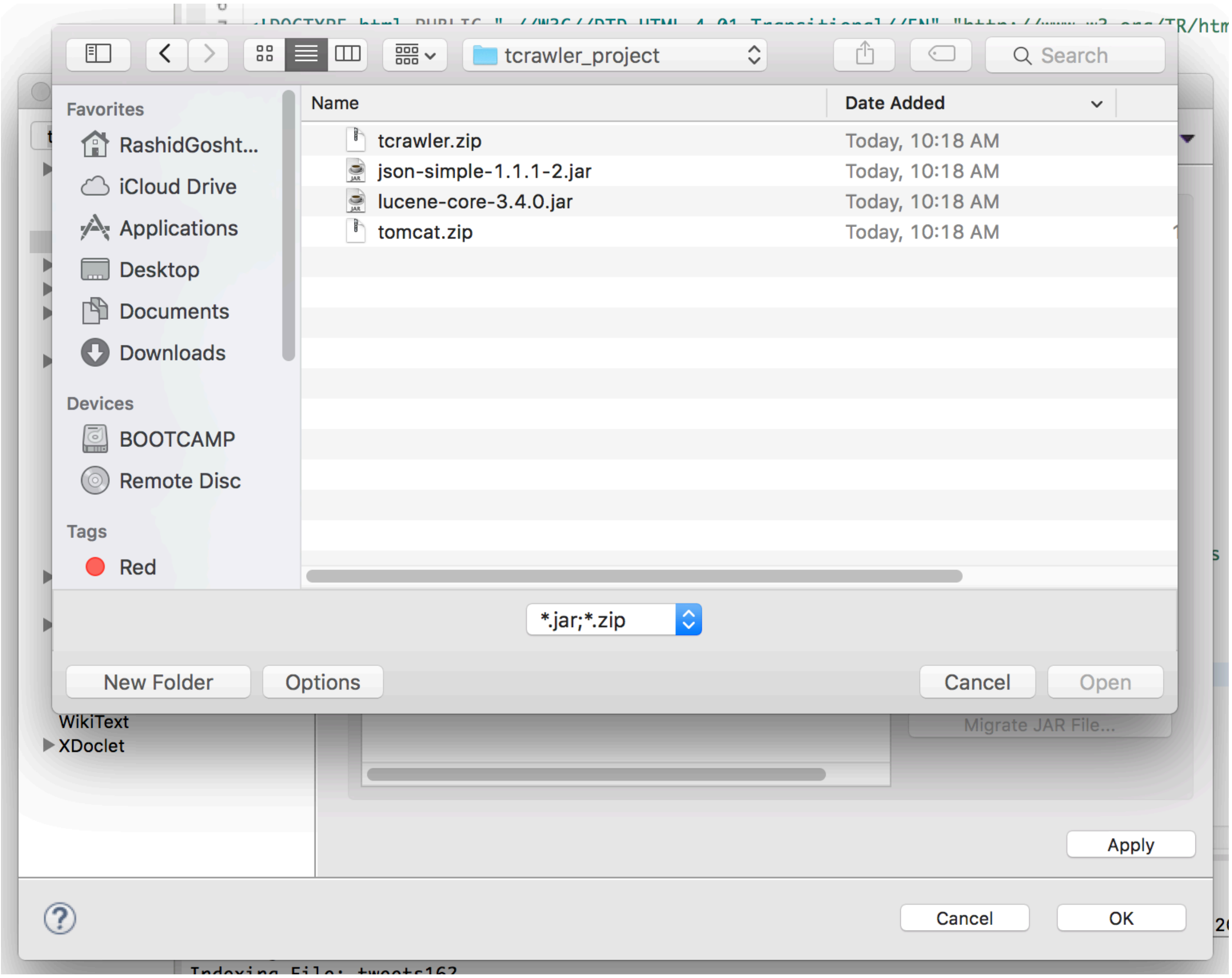
-



JARs and class folders on the build path:

- json-simple-1.1.1-2.jar - ui_search
- jsp-api.jar - /Users/RashidGoshtasbi/Documents/tomcat/lib
- lucene-core-3.4.0.jar - ui_search/WebContent/lib
- servlet-api.jar - /Users/RashidGoshtasbi/Documents/tomcat/lib
- Apache Tomcat v8.0 [Apache Tomcat v8.0]
- EAR Libraries
- JRE System Library [Java SE 7 [1.7.0_79]]
- Web App Libraries

-



Name	Date Added
tcrawler.zip	Today, 10:18 AM
json-simple-1.1.1-2.jar	Today, 10:18 AM
lucene-core-3.4.0.jar	Today, 10:18 AM
tomcat.zip	Today, 10:18 AM

-

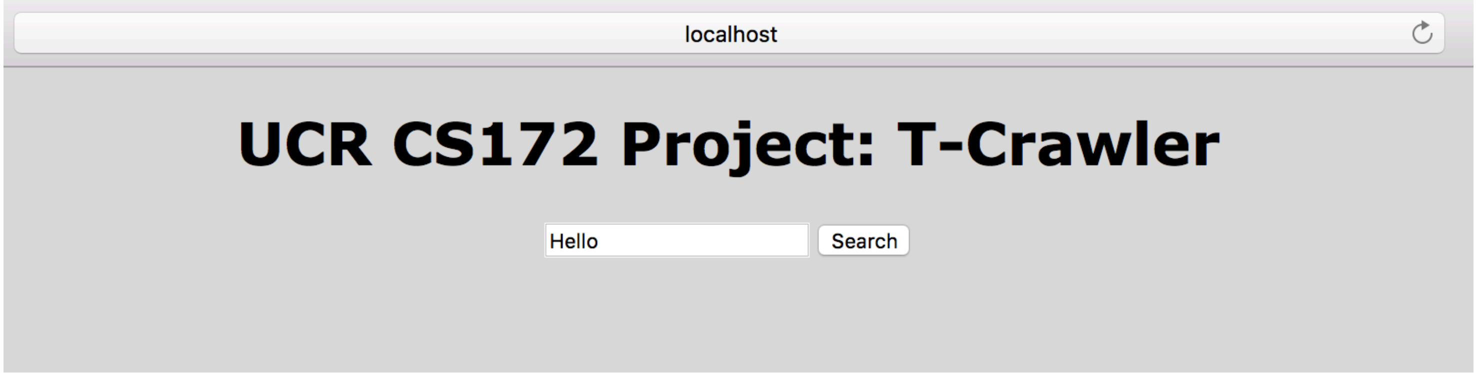
7. Inside textOutput.java and LuceneExample.java please change "INDEX_DIR" to be pointing to your output folder for where you want to store your index and where your JSON data files are currently located at, respectively.
8. Running the project
    - Under the project folder in Eclipse, open ui_search > WebContent > right-click the "search.html" file > "Run As" > "Run on Server"
    - A new window should open up and select "Finish".
    - Eclipse will compile the project and webpages relevant files and will open up a new browser window in Eclipse. In there the search bar and search function will show.

IF YOU WANT TO RUN TCRAWLER FROM TOMCAT'S LOCALHOST TERMINAL:

1. Navigate to directory with Tomcat folder
2. type: sudo chown -R Tomcat
3. sudo chmod +x Tomcat/bin/*.sh

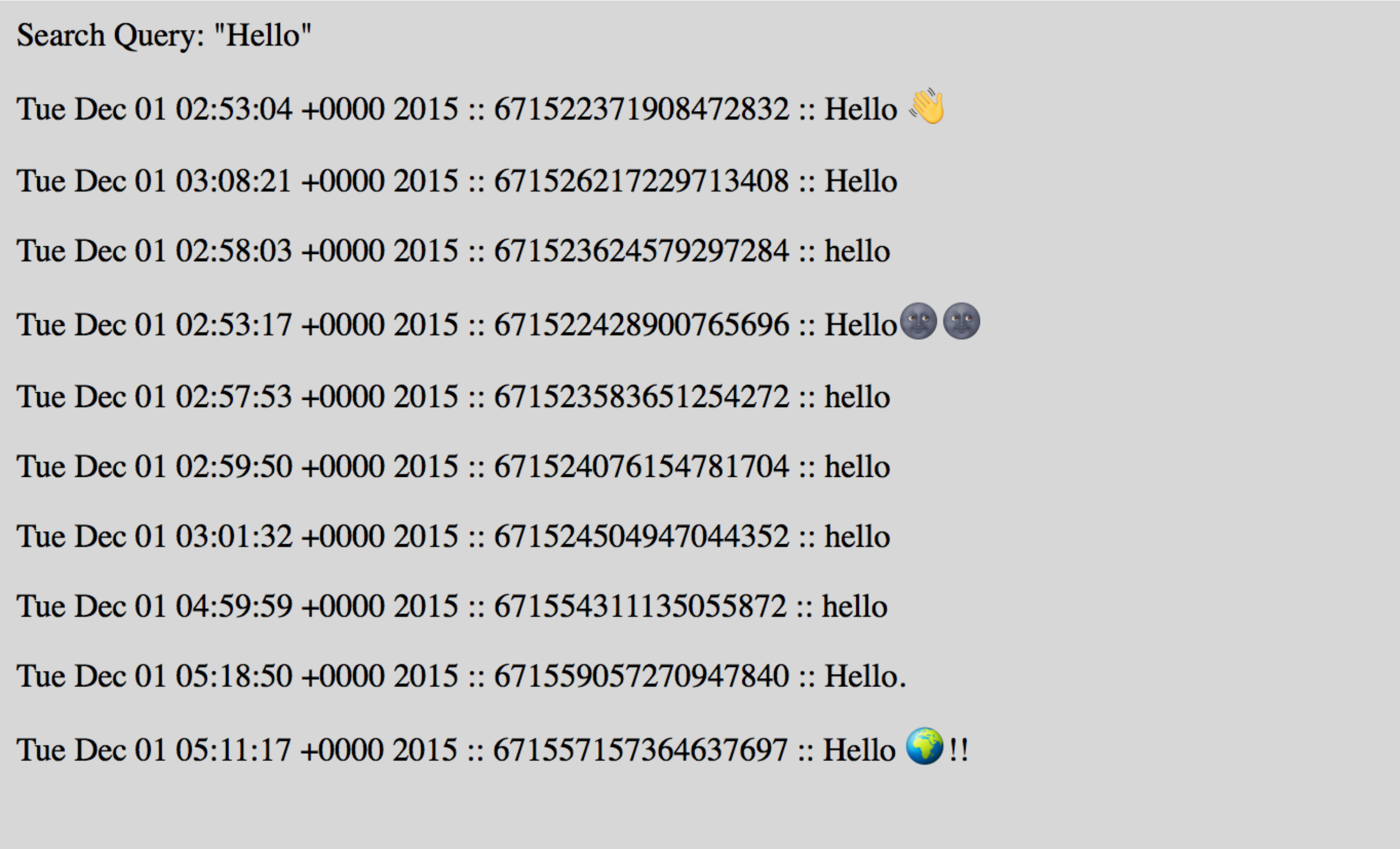
4. To start tomcat, type: Tomcat/bin/startup.sh
5. In your browser, type in "localhost:8080"
6. You will be directed to the Tomcat page.
7. Then in the url bar, type: **localhost:8080/ui_search/search.html**
8. This is the WebApp created in Eclipse
9. To shutdown tomcat, type: Tomcat/bin/shutdown.sh

Images In Action

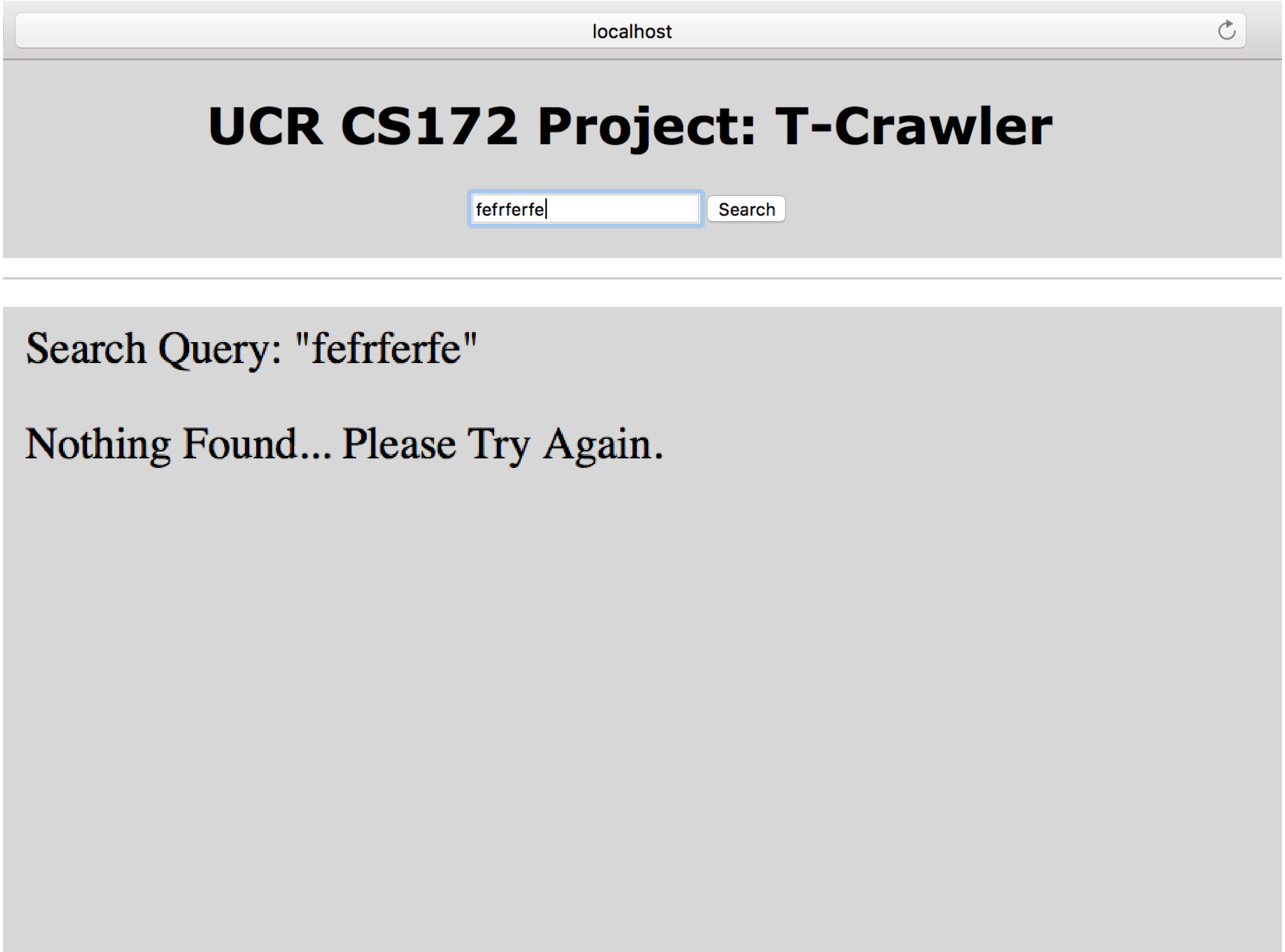


UCR CS172 Project: T-Crawler

Hello



```
Search Query: "Hello"

Tue Dec 01 02:53:04 +0000 2015 :: 671522371908472832 :: Hello 👋

Tue Dec 01 03:08:21 +0000 2015 :: 671526217229713408 :: Hello

Tue Dec 01 02:58:03 +0000 2015 :: 671523624579297284 :: hello

Tue Dec 01 02:53:17 +0000 2015 :: 671522428900765696 :: Hello🌚🌚

Tue Dec 01 02:57:53 +0000 2015 :: 671523583651254272 :: hello

Tue Dec 01 02:59:50 +0000 2015 :: 671524076154781704 :: hello

Tue Dec 01 03:01:32 +0000 2015 :: 671524504947044352 :: hello

Tue Dec 01 04:59:59 +0000 2015 :: 671554311135055872 :: hello

Tue Dec 01 05:18:50 +0000 2015 :: 671559057270947840 :: Hello.

Tue Dec 01 05:11:17 +0000 2015 :: 671557157364637697 :: Hello 🌍!!
```



UCR CS172 Project: T-Crawler

Search Query: "fefrferfe"

Nothing Found... Please Try Again.