**Prerequisite :-**

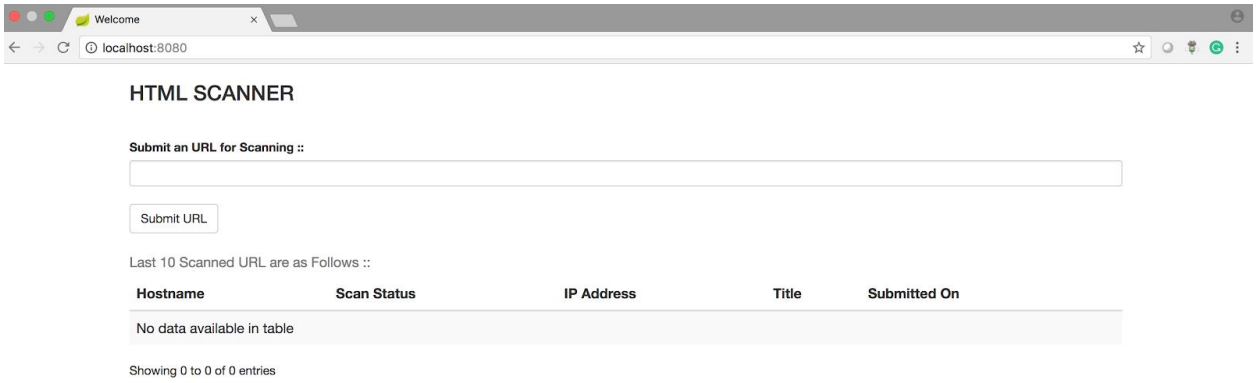
1) Java 1.8
2) Maven (Any version)
3) MySql installed

**Steps to run the server :-**

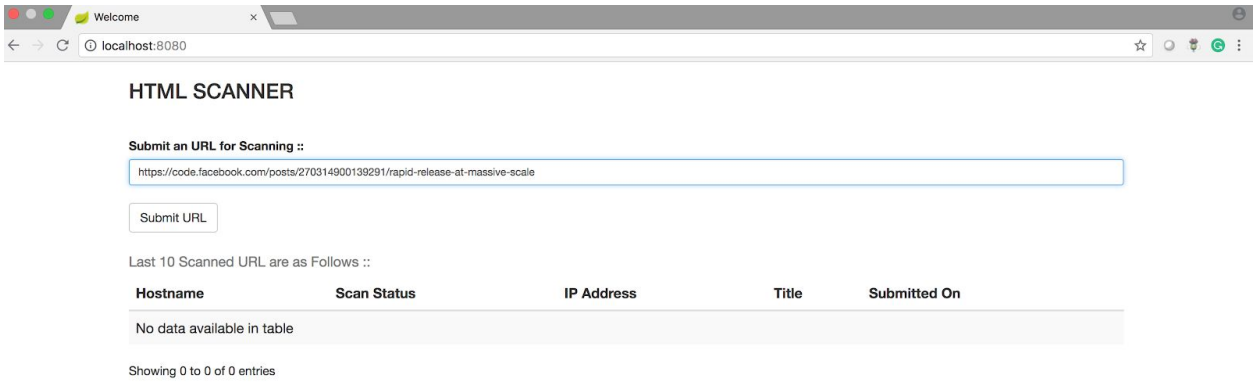
1) Unzip the zip file
2) Go to the folder htmlparser
3) "**mvn package**"
4) This would create a JAR in "**<$PROJECT_PATH>/htmlparser/htmlparser-0.0.1-SNAPSHOT.war**
5) Create a database in mysql named **htmlparser**
6) Run the jar "**java -jar htmlparser-0.0.1-SNAPSHOT.war**"
7) It shall run the DB migration with all the tables created.
8) This should run the server on **http://localhost:8080**

**Steps to follow the Scanning and Retrieving the data :-**

1) After the server is up go to **http://localhost:8080** **:-**
   Result is currently empty however the top 10 results should be displayed on loading

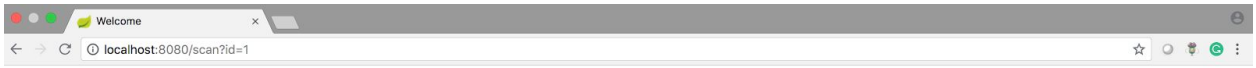



2) Enter the url in the submit text box




3) The top 10 result should be displayed sorted on submitted time


4) Click on the Hostname section to view the Scanned URL details




Scan Details

| URL | Redirection URL | Submitted On | Website Title | Website Body Content | Image Count | Links Count |
|---|---|---|---|---|---|---|
| code.facebook.com | | 2018-06-19 22:47:10.0 | Rapid release at massive scale Engineering Blog Facebook Code Facebook | Facebook Code Search Open Source Platforms Android iOS Web Infrastructure Systems Core Data Data Infrastructure Developer Tools Production Engineering Security Hardware Infrastructure Connectivity Data Center Networking Traffic Video VR Video Engineering Virtual Reality Artificial Intelligence Applied Machine Learning Research Infra Web Scale Backend Development Tools Tooling Rapid release at massive scale Chuck Rossi Over time the software industry has come up with several ways to deliver code faster safer and with better quality Many of these efforts center on ideas such as continuous integration continuous delivery agile development DevOps and test driven development All these methodologies have one common goal to enable developers to get their code out quickly and correctly to the people who use it in safe small incremental steps The development and deployment processes at Facebook have grown organically to encompass many parts of these rapid iteration techniques without rigidly adhering to any one in particular This flexible pragmatic approach has allowed us to release our web and mobile products successfully on rapid schedules rate of code delivery scaled with the size of the team But it took a certain amount of human effort in the form of release engineers in addition to the tools and automated systems in place to drive the daily and weekly pushes out the door We understood that batching up larger and larger chunks of code for delivery would not continue to scale as the team kept growing We decided to move facebook com Continuous delivery at scale While a true continuous push system would deliver every individual change to production soon after it landed the code velocity at Facebook required us to develop a system that pushes tens to hundreds of diffs every few hours The changes that get made in this quasi continuous delivery mode are generally small and incremental and very few will have a visible effect on the actual user experience Each release is rolled out to 100 percent of production in a tiered fashion over a few hours so we can stop the push if we find any problems Many of the changes are initially kept behind our Gatekeeper system which allows us to roll out mobile and web code releases independently from new features helping to lower the risk of any particular update causing a problem If we | 6 | 75 |