Peernet

Deployment Manual

Prepared by Christina Constas & Alexandre Meyer

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to help administrators deploy the PeerSet web application on a Linux operating system.

1.2. Summary

Section 1 of this document defines the purpose of the document, lists commonly used words, acronyms and abbreviations with their definitions, and lists the references used for preparing this document. Section 2 outlines the software requirements for installing and running PeerSet. Section 3 lists the steps for installation and includes a subsection listing the necessary steps for creating a Google Client ID.

1.3. Definitions, Acronyms & Abbreviations

- **Backend:** The engine, database and networking code used to run PeerSet. Users will not be viewing and interacting with the backend of the application.
- **cid:** Client ID. You will be required to create a Google cid in order for users to log in to PeerSet with their Google accounts.
- **Frontend:** The user interface of the application. This is the part of PeerSet that users will be viewing and interacting with.
- **JDK 11:** JDK stands for Java Development Kit. JDK 11 is required to run PeerSet.
- **Maven:** A project management tool by Apache, used primarily for Java applications.
- **MySQL:** The database used to store and manage information for PeerSet.
- **Node.js:** A JavaScript runtime used to build scalable network applications.
- **npm:** The software registry used in the development of PeerSet.

1.4. References

- Project link
- MySQL Installation Guide
- MySQL Tutorial
- Maven Installation Guide

- JDK 11 Installation Guide
- NPM & NodeJS Installation Guide
- Client ID in Google Analytics: What it is and how to set it up (Zhovtonizko & Komarovskiy, 2019)
- Guided steps for creating a Google Client ID
- About Node.js

2. Preparation

This section of the deployment manual describes the preparations that must be undertaken before installing and running PeerSet. Please note that, at this time, PeerSet is only able to run on Linux operating systems.

2.1. Backend

The following subsections describe the additional software requirements for running the backend of PeerSet.

2.1.1. MySQL

For server maintenance, MySQL must be installed before downloading and running PeerSet.

Here is the MySQL installation guide, with instructions for downloading and installing MySQL on Windows, Mac or Linux operating systems.

Please see this tutorial for more information about using MySQL.

2.1.2. Maven

Maven is a project management tool that must be installed before downloading and running PeerSet. To install Maven on your computer, please see this installation guide.

2.1.3. JDK 11

JDK 11 must be installed in order to run Java applications. To install JDK 11 on your computer, please see this installation guide.

2.2. Frontend

Node.js and npm must be installed in order to run the frontend of PeerSet. To install Node.js and npm onto your computer, please see this installation guide.

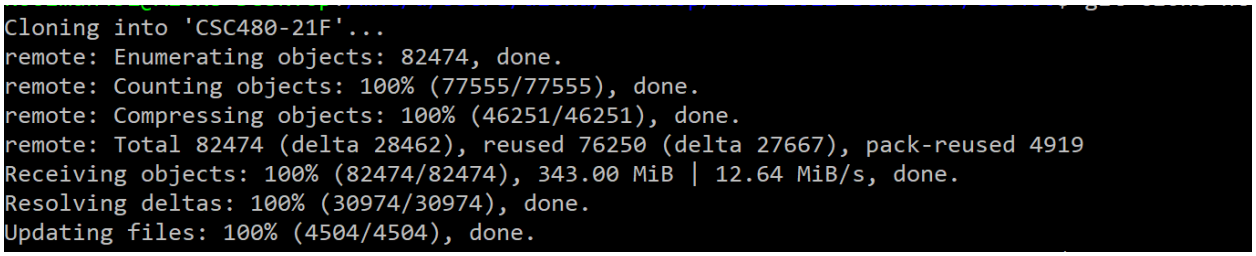
3. Installation

3.1. Backend

In order to run PeerSet, you must first install the backend. This must be done by running a script file named **launcher.sh,** located in the main directory of PeerSet's GitHub repository.

To start, download the zip file of PeerSet's Github repository into your preferred directory. You can also clone the repository by opening the Terminal and entering the following command:
Git clone https://github.com/vanessamaike/CSC480-21F.git

The following should happen:



Next, use the command line to the directory where the repository was downloaded.

To add professor accounts to the configuration file, run the command **./launcher.sh allow**. If you have a document prepared with the professor email accounts you wish to add to the program, follow the **./launcher.sh allow** command with the directory where that document is stored. If the directory is invalid, or if you don't have a document, you will be prompted to enter the email addresses manually.

To start the server, run the following command: **./launcher.sh run**

To stop the server, run the following command: **./launcher.sh end**

3.2. Frontend

To run the frontend of PeerSet, use the command line to the directory where the repository was downloaded. Next, enter **cd CSC480-21F,** then enter **cd frontend.** This will direct you to the frontend folder of the repository.

Once you've navigated to the frontend folder, enter the command **npm install.** Then, enter the command **npm start** to start the frontend.

3.3. Creating Google Client IDs

The Client ID (cid) is a unique identifier that tells Google Analytics what actions to link to a user on PeerSet. This ID will need to be manually created in order to allow users of the web application to log in to PeerSet with their Google Accounts.

The guided steps on how to create a Google Client ID can be found here: https://developers.google.com/identity/gsi/web/guides/get-google-api-clientid

For the sake of deployment, you can follow the manual instructions listed below.

1. Go to the site: https://console.cloud.google.com/getting-started

2. If not already signed in, click on Sign In in the top-right corner and sign into your Google Account where you would like to extract the google client ID.

3. Click on "Select a Project."

4. Click on "New Project."

5. Enter the necessary information and select "Create." Please keep in mind some workspaces may have disabled Google Cloud Platform service so contact your administrator to turn the service on in the Google Workspace Admin console.

6. Go to https://console.cloud.google.com/apis/dashboard

7. In the drop-down menu in the top left, make sure you are in the current directory of the project you just named.

8. Click on "Credentials."

9. Click "Create Credentials."

10. Click on "OAuth client ID."

11. If not already configured, click on "Configure Consent Screen."

12. CLick the necessary input based on the User Type.

13. Click "Create."

14. Follow the step-by-step guide and type in the necessary input.

15. Click on "Credentials."

16. Click "Create Credentials."

17. Click on "OAuth client ID."

18. Click on "Application type" and select "Web application."

19. Under "Authorized JavaScript origins", write the necessary JavaScript Origins URL where the web application will be hosted. For example, "http://pi.cs.oswego.edu:5685/"

20. Click "Create."

21. Copy the necessary "Client ID" and paste it inside the microprofile-config.properties within the ``backend`` folder of the project.

3.4. Acknowledgements

The Requirements team would like to thank John Owens for providing the instructions for creating Google Client IDs.