

SEE Installation Guide

5/30/14

Prepared by Lantana Consulting Group

Lantana Consulting Group   
[liora.alschuler@lantanagroup.com](mailto:liora.alschuler@lantanagroup.com)   
<http://www.lantanagroup.com>

© 2013 Lantana Consulting Group  
All rights reserved.

**Lantana Consulting Group**PO Box 177  
East Thetford, VT 05043  
[www.lantanagroup.com](http://www.lantanagroup.com)

Angelo Kastroulis  
Sr Strategist  
[angelo.kastroulis@lantanagroup.com](mailto:angelo.kastroulis@lantanagroup.com)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 5/30/14 | 1.0 | Original Guide | Angelo Kastroulis |
| 6/12/14 | 1.1 | Client Loader/System Loader | Angelo Kastroulis |

Table of Contents

Executive Summary 5

1 Prerequisites 6

1.1 Nodejs 0.10.25 6

1.2 MongoDB 6

2 Install SEE 7

2.1 Copy the files 7

2.2 Install SEE 7

2.3 Configure SEE database and application server settings 7

2.4 Load SEE data 8

2.5 Configure inbound/outbound Orion interfaces 8

2.6 Start SEE 8

Executive Summary

This installation guide provides guidance for installing and configuring the SEE tool. While it includes examples of procedures for installing prerequisites, they are included only for reference. Each distribution of operating systems may have its own specific steps that differ and are outside the scope of this document.

# Prerequisites

## Nodejs 0.10.25

Install Nodsjs in your environment. Using either an installer (Windows, Mac OSX), a repository (Linux’s epel, yum, rpm, etc), or a make strategy (compiling code).

For example, some Linux distributions work this way using the epel repository:

**Install epel repository:**

wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

rpm -ivh epel-release-6-8.noarch.rpm

**Install nodejs**

sudo yum install nodejs npm --enablerepo=epel

## MongoDB 2.6.1

Install MongoDB and ensure that it runs as a service (daemon).

For example, some Linux distributions may work like this:

**Configure yum**

Create a /etc/yum.repos.d/mongodb.repo file to hold the following configuration information for the MongoDB repository (for a 64 bit system):

[mongodb]

name=MongoDB

Repository baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86\_64

gpgcheck=0

enabled=1

Use yum to install MongoDB packages and tools

sudo yum install mongodb-org

Start the [mongod](http://docs.mongodb.org/manual/reference/program/mongod/#bin.mongod) process

sudo service mongod start

# Install SEE

## Copy the files

Create a SEE folder (e.g. home/user/SEE)

Copy the SEE files into that folder. Do not include the node\_modules folder. If that folder is carried over, then it should be deleted before the next step.

## Install SEE

CD into the SEE folder and install the dependant packages using npm (Node Package Manager).

sudo npm install

## Configure SEE database and application server settings

The settings are configured in a javascript file located in SEE/config.

module.exports = {

Settings: {

dbHost: "localhost",

dbPort: 27017,

dbUser: "see\_user",

dbPassword: <password here>,

appHost: "localhost",

appPort: 8000,

fileDropLocation: "./drop/send/",

fileLoadLocation: "./drop/load/",

fileProcessedDropLocation: "./drop/archive/send/",

fileProcessedLoadLocation: "./drop/archive/load/",

fileProcessedErrorLocation: "./drop/archive/error/",

sampleHeaderLocation: "./drop/",

logLevel: "verbose", //can be verbose, info, http, warn, error (progressively restrictive)

logPath: "./see.log",

inboundLogLevel: "verbose",

inboundLogPath: "./see-inbound.log"

}

}

* dbHost and dbPort are the host and port of the Mongodb instance
* dbUser is the username the application will run as in the MongoDB instance and dbPassword is that user’s password (not what you enter here will be used in subsequent steps).
* appHost and appPort are the host and port that this services should be started on in Nodejs
* fileDropLocation, fileLoadLocation, fileProcessedXXX, and sampleHeaderLocation are related to internal import features for SEE. Do not change these.
* logLevel sets the level of detail logged in messages realted to SEE’s APIs, with verbose being the most detail.
* logPath is the location to create the log file. If one does not exist, it will be created.
* inboundLogLevel is the level of detail for messages sent or received in the “inbound” service (Orion interface)
* inboundLogPath is the location of the log file. If one does not exist, it will be created.

## Load SEE System users

The system users and tables (see\_user above) are loaded into the database with the a javscript file called systemDataLoader.js alocated in the SEE/Data folder and needs to be run with node while Mongodb is running. Run this command from the SEE folder.

node ./Data/systemDataLoader.js

## Load SEE System data

The system data is bulk loaded using MongoDB’s MongoImport tool which needs to be run with node while Mongodb is running. It loads system-wide medications, medication routes, and Snomed problems. Run this command from the location of the Mongo install.

For example, for Medications:

./bin/mongoimport --db medication --collection medication --host <HOST> --port <PORT> --file <SEE LOCATION>/Data/medications.js --jsonArray --stopOnError --drop --username see\_user --password <PASSWORD>

For example, for Medication Routes:

./bin/mongoimport --db route --collection route --host <HOST> --port <PORT> --file <SEE LOCATION>/Data/routes.js --jsonArray --stopOnError --drop --username see\_user --password <PASSWORD>

For example, for Snomed Problems:

./bin/mongoimport --db snomed\_problem --collection problem --host <HOST> --port <PORT> --file <SEE LOCATION>/Data/problems.js --jsonArray --stopOnError --drop --username see\_user --password <PASSWORD>

Notes:

**--port**

Specifies the port the MongoDB instance is running on. Default is 27017. Use the value specified in the config above.

**--host**

Specifies the host the MongoDB instance is running on. Default is localhost. Use the value specified in the config above.

**--drop**

An optional parameter. Include if you want to drop the collection before the inserts are done, otherwise data will be duplicated.

**--username**

Optional. Use the value specified in the configuration file above.

**--password**

Optional. Use the value specified in the configuration file above.

**--file**

enter the full path to the file in the SEE/Data folder.

## Load SEE Client data

Client data may contain users, organizations, and assigning authority lists. The client data is loaded into the database with the a javscript file in the same way that the system data is loaded. The template file is called clientDataLoader.js. However, depending on the environment (Dev, Test, Production) various versions of the file will need to be run, since the data is environment specific. For example, to load the development environment:

node ./Data/clientDataLoader-dev.js

For the test environment:

node ./Data/clientDataLoader-test.js

For the production environment:

node ./Data/clientDataLoader-production.js

NOTE: BEFORE EXECUTING ANY CLIENT SCRIPT, BE SURE THAT THE DATA IN THE SCRIPT IS CORRECT AND COMPLETE (the data is found in the script’s onDatabaseOpen function, and is represented as JSON). For example, Assigning Authorities lists are wiped clean and completely rebuilt… thus, any changes made with other scripts will be lost.

## Configure inbound/outbound Orion interfaces

Locate the orionconfig.json configuration file found in the SEE/utils folder. It contains entries for a hostname a port of the destination Orion server. These are the location of the services to create messages and attachments. The configuration file looks like this:

{

"hostname": "orion-mao-testhttp1v",

"port": 4443

}

## Start SEE

The see service file is documentService.js. That is the file that needs to be run by node. You can manually start the application with:

node documentService.js

or

npm start

or

Daemonize the application using your favorite method.

For example, you could do it this way (for illustrative purposes only): using forever with start & stop scripts in the SEE folder.

sudo npm install forever

Create a file called start in the SEE folder with these contents:

#!/bin/bash

# Invoke the Forever module (to START our Node.js server).

./node\_modules/forever/bin/forever \

start \

-al forever.log \

-ao out.log \

-ae err.log \

documentService.js

Create a file called stop in the SEE folder with this in it:

#!/bin/bash

# Invoke the Forever module (to STOP our Node.js server).

./node\_modules/forever/bin/forever stop documentService.js

Make sure they have the appropriate execute permissions:

chmod +x ./stop

chmod +x ./start

Start the server

./start