

CIS 4911

SENIOR PROJECT: ENVOSCHOLAR v3.0

---

# Installation Guide

---

*Team Members:*

Dairon HERNANDEZ

Jorge EUCEDA

*Product Owner:*

Mark FINLAYSON

*Instructor:*

Masoud Sadjadi

*Mentors:*

Maria A. PRESA REYES,

Deya BANISAKHER



## Contents

|          |                                                             |          |
|----------|-------------------------------------------------------------|----------|
| <b>1</b> | <b>Introduction and Needed Information</b>                  | <b>2</b> |
| 1.1      | Angular Frontend . . . . .                                  | 2        |
| 1.2      | Backend Database Server and Database API . . . . .          | 2        |
| 1.3      | Custom Ontology Graph View using WebVOWL . . . . .          | 3        |
| 1.4      | Installation Prerequisites . . . . .                        | 3        |
| <b>2</b> | <b>Running EnvoScholar v3.0</b>                             | <b>4</b> |
| 2.1      | Running EnvoScholar Locally . . . . .                       | 4        |
| 2.1.1    | Running MongoDB Server and API on MacOS . . . . .           | 4        |
| 2.1.2    | Running MongoDB Server and API on Windows 10 . . . . .      | 4        |
| 2.1.3    | Running MongoDB Server on and API Linux . . . . .           | 5        |
| 2.2      | Running EnvoScholar Remotely . . . . .                      | 5        |
| 2.2.1    | Running the Angular Frontend in Production Server . . . . . | 5        |
| 2.2.2    | Running the MongoDB Server in Production Server . . . . .   | 6        |
| 2.2.3    | Running the MongoDB API in Production Server . . . . .      | 6        |

# 1 Introduction and Needed Information

This document provides a step-by-step instruction on how to set up development environment and deployment environment. The YouTube video ([here](#)) also goes through the steps to install and run EnvoScholar. Prior to using EnvoScholar v3.0, make sure you follow all the steps described below to ensure functionality of all features.

Navigate to the `nvos-scholar` folder. Inside you will find three folders, `nvos-scholar`, `user-database-server` and `webvowl-filtered`. These folders contain all the source code you will need. The `nvos-scholar` contains all the frontend components of the website. The `user-database-server` contains all the code for the database API and the database collection specifications.

## 1.1 Angular Frontend

The `nvos-scholar` folder contains all the source code for the Angular frontend. Inside you will see a folder named `src`. Navigate to that folder. Inside the `src` folder, navigate to the `app` folder. This app folder contains the main app component (`app.component.ts`) which provides an entry point for the router. In this same folder you'll find `routes.ts`, which handles our frontend URL routing API.

You'll also find folders that divide the components into major categories. Inside each of the component folders you'll find their respective `Component.ts`, `HTML`, and `CSS`, with the exception of the `webvowl-graphview` component folder. This component contains the compiled `webvowl-filtered` folder code, which will be explained in its own section. Also note that the ontology page was not created using Angular. It was created using `HTML` and `JQuery`, and all of its details can be found in the ontology folder, still under the `app` folder.

## 1.2 Backend Database Server and Database API

Inside the `user-database-server` folder, the three main folders you will be working with if you need to add/change anything in the database are the controllers, models, and routes folders. The main features added when compared to EnvoScholar 2.0 are mainly the ability to store and retrieve Developer Statistics such as Query History and Query Information using newly-added models, routes, and controllers. `config` folder will probably stay the same, and `server.js` file will also probably not need anything changes/additions. The `server.js` file is the main file in the `user-database-server` folder.

### 1.3 Custom Ontology Graph View using WebVOWL

Inside the `webvowl-filtered` folder, you'll find a heavily customized version of the open-source project named **WebVOWL**. This folder contains an `src` folder. This project is heavily-layered and complex, so we will cover it in a separate video found (here) explaining the underlying code and how it was customized. The main concept you should be aware of right now is that you can build this program using node js and npm, which will generate a `deploy` folder that we can copy and paste into the `webvowl-graphview` folder component found in `nvos-scholar` folder containing the frontend that was talked about in the "Angular Frontend" section of this guide.

### 1.4 Installation Prerequisites

1. NodeJS: Can be downloaded at the following link: <https://nodejs.org/en/>
2. Angular: Go into the project folder named `nvos-scholar` and type:

```
$ npm install -g @angular/cli
```

*Note:* This will allow you to run the command `npm run start:dev`, which runs the angular application in a development environment locally. You should not have to install any additional libraries because we've included the `node_modules` folder for convenience. But just in case that any libraries are missing, you can go to either the `nvos-scholar` folder or the `user-database-server` folder and run the command `npm install`. This command will reinstall all of the necessary libraries for our project. If you're on a Linux or Mac OS, you might need to run this command as an elevated command using `sudo npm install` if you receive any permission conflicts, and you'll need to type your computer's laptop when prompted.

3. MongoDB: can be installed at the following link (for Windows, Linux, MacOS): <https://www.mongodb.com/download-center/community>.

- **Windows**

We need to create the database directory. For Windows 10 Specifically, you need to open the command prompt navigate to the root folder of your main drive, then create the directory where the database will be served. For example, you can do `$ cd C:\`, then run `$ md "\data\db"`. This will create the directory in which the database will live.

- **MacOS and Linux**

On Linux and MacOS, you can simply open a terminal and run the command `$ sudo mkdir -p /data/db`. Note that this requires elevated permissions as we are creating this folder in the root of the filesystem.

## 2 Running EnvoScholar v3.0

EnvoScholar v3.0 configures the application to be run both locally in a development mode, and running it in production mode using a web server. The two main sections below will cover both instances.

### 2.1 Running EnvoScholar Locally

Open three terminals in any of your operating systems. With the first terminal, navigate to the `nvos-scholar` folder and run the command `npm run start:dev`. **Note:** if you are on MacOS or Linux, you need to run `sudo npm run start:dev` instead due to needing elevated permissions. Once this finishes compiling, open the second terminal.

The second and third terminals will be used to run the MongoDB server and the MongoDB API which sends and receives requests from the Angular Frontend to the MongoDB Server. Run the following commands:

#### 2.1.1 Running MongoDB Server and API on MacOS

If you installed MongoDB through HomeBrew (recommended) then you can simply start and stop MongoDB with the following commands in your second terminal, respectively:

- `$ brew services start mongodb`
- `$ brew services stop mongodb`

#### 2.1.2 Running MongoDB Server and API on Windows 10

Type the below command int your second terminal:

```
$ "C:\Program_Files\MongoDB\Server\4.0\bin\mongod.exe" --dbpath="c:\data\db"
```

**Note:** You might need to change the 4.0 after `Server\` if a newer version of MongoDB community is installed in your machine.

### 2.1.3 Running MongoDB Server on and API Linux

Assuming you've already created the `\data\db` directory, all you need to do on Linux is execute the command `$ sudo mongod` in your second terminal.

## 2.2 Running EnvoScholar Remotely

To run EnvoScholar in Linux server provided by the product owner, you need to learn about how to run terminal processes in the background using linux. In our case, we decided to use a linux program called `tmux`, which is a terminal multiplexer that allows you to do the aforementioned. We will still need to have three terminals opened, but now we will "attach" and "detach" from these three terminals as needed so the processes remain running in the background. You can find a video ([here](#)) showing this deployment process.

In a general way, you can create a session by running `$ tmux new -s <session-name>` in the command line. You can detach from this session by pressing the `CTRL+D` combination in your keyboard. You can also reattach to a session by running the command `$ tmux attach -t <session-name>`.

**Note:** Before proceeding, please make sure to understand how to create, detach, and reattach from terminal sessions using `tmux`, as the below steps heavily depend on it.

### 2.2.1 Running the Angular Frontend in Production Server

The first thing you must do is to create a new session for our first terminal using the following command:

```
$ tmux new -s nvos-frontend
```

This will automatically attach you to this session. Once you run this command, execute the angular frontend by running the following command:

```
$ sudo npm run start:prod
```

**Note:** this command will prompt you to enter your password, this is the same password that the product owner provided you with to ssh into this machine.

### 2.2.2 Running the MongoDB Server in Production Server

The same command that was mentioned to run the MongoDB Server on linux applies here, with the difference being that you need to create a terminal session using tmux with the following command:

```
$ tmux new -s mongo-server
```

Once attached to this session, execute the mongodb server by running the following command:

```
$ sudo mongod
```

**Note:** this command will prompt you to enter your password, this is the same password that the product owner provided you with to ssh into this machine.

### 2.2.3 Running the MongoDB API in Production Server

The same command that was mentioned to run the MongoDB API on Linux applies here, with the difference being that you need to create a terminal session using tmux with the following command:

```
$ tmux new -s mongo-api
```

Once attached to this session, execute the mongodb server by running the following command in the `user-database-server` folder:

```
$ sudo node server.js
```

**Note:** this command will prompt you to enter your password, this is the same password that the product owner provided you with to ssh into this machine.

Now that you've followed the above steps, you are able to access the website in the domain provided by the product owner. Currently, this domain is located in the following URL: <http://envoscholar.cs.fiu.edu>.