

# MEAN Stack Template List of Components

acraig, dyang, jliu, sbae, sratna

08-15-16

## 1 MEAN

Our web application will be implemented using the MEAN stack, with special emphasis on Angular 2 which has typescript. This MEAN stack implementation must work on Microsoft Visual Studio 2015 for business reasons. Since we are not a for profit company the code must be shown in the environment so it is easily accessible by others. In addition, our goal is that others can use this nPDS template to build a nPDS server with their specifications.

- MongoDB: <https://www.mongodb.com/> (v3.2)
- Express: <https://expressjs.com/> (v4.14.0)
- Angular JS: <https://angularjs.org/> (v2.0: Comes with Typescript ES6)
- Node JS: <https://nodejs.org/en/> (v4.4.7 LTS: Long Term Support, Node.js highly maintained)

## 2 View Templates

The view templates of the web application will have URL for every webpage to have a similar organization as webpages in the Semantic Web. We will utilize the view templates by AngularJS. Any component that is used will be backed by a successful, well-established, for-profit organization, that is not a one-off.

## 3 User Authentication

The user authentication system will be readily maintained and supported. the input credentials will be on the client front end and checks will be done using Node and Express. Finally, MongoDB will store all authentication attempts in a database. This system will use PassportJS (<http://passportjs.org/> (v1.0.0) \*can be downloaded through node package manager) as its main organization and validation method. We will implement a Nexus Portal Doors user management system. This will be built within NPDS that allows for any user authentication system to be connected to the system.

## 4 Package Management

Our web application will have a simple one-click download for the template in order to ensure easy functionality by the user.

## 5 Database Agent

To get in and out of the database we will use Mongoose (v4.5.8) and MongoDB. This will be based on the URL of the webpage. Mongoose will enable the developer to create a schema within the database, as well as insert and retrieve data.

## 6 Priority of Environments

- JetBrains WebStorm: <https://www.jetbrains.com/webstorm/> : Lightweight yet powerful IDE, perfectly equipped for complex client-side development and server-side development with Node.js
- Microsoft Visual Studio Community 2015: <https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx> : A free, fully-featured, and extensible IDE for creating modern applications for Windows, Android, and iOS, as well as web applications and cloud services
- Microsoft Visual Code: <https://code.visualstudio.com/download> : code editor redefined and optimized for building and debugging modern web and cloud applications, free, open-source, runs anywhere

## 7 Server Types

- NPDS Read/Write Registrar
- NEXUS Read Only Diristry : a PORTAL registry and DOORS directory combined in a single server
- PORTAL Read Only Registry : a registry for storing and distributing descriptions of resources relevant to a given problem domain. These descriptions include both lexical tags and semantic labels as well as locations and cross-references. Back-end use of traditional relational database stores, may be established by an organization or person who maintains any local policies governing registration of resources at that particular primary PORTAL registry
- DOORS Read Only Directory : a directory for efficient lookup of resources registered with a PORTAL registry using the associated tags and labels and has RDF-triple database stores

## 8 Interfaces

- Agent Interactive RESTful Web API Service
- Human User Interactive Website Application