# Express

Earlier we discussed how to use Node.js with Visual Studio Code but Node.js is a runtime and if you are going to create an application of specific type (Web, IoT, Notification service module and so on), it’s better to utilize an additional framework. In case of Web applications, I would recommend to use Express framework that is developed for Node.js and contains great API, an application generator and support the best patterns for Web applications.

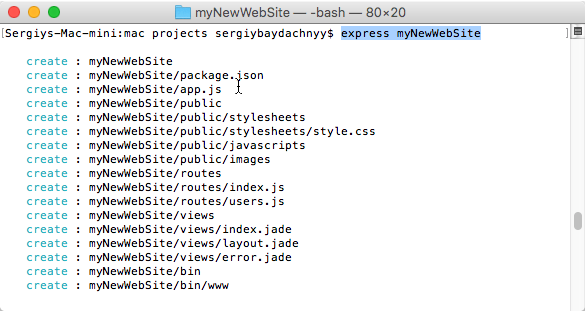
Before starting you need to install express generator that we will use to create a template for our site. You can do it with the help of the next command:

**sudo npm install -g express-generator**

Once the generator is installed you can navigate the Terminal to your working folder where you are going to create a new project. To create the project you can use this command:

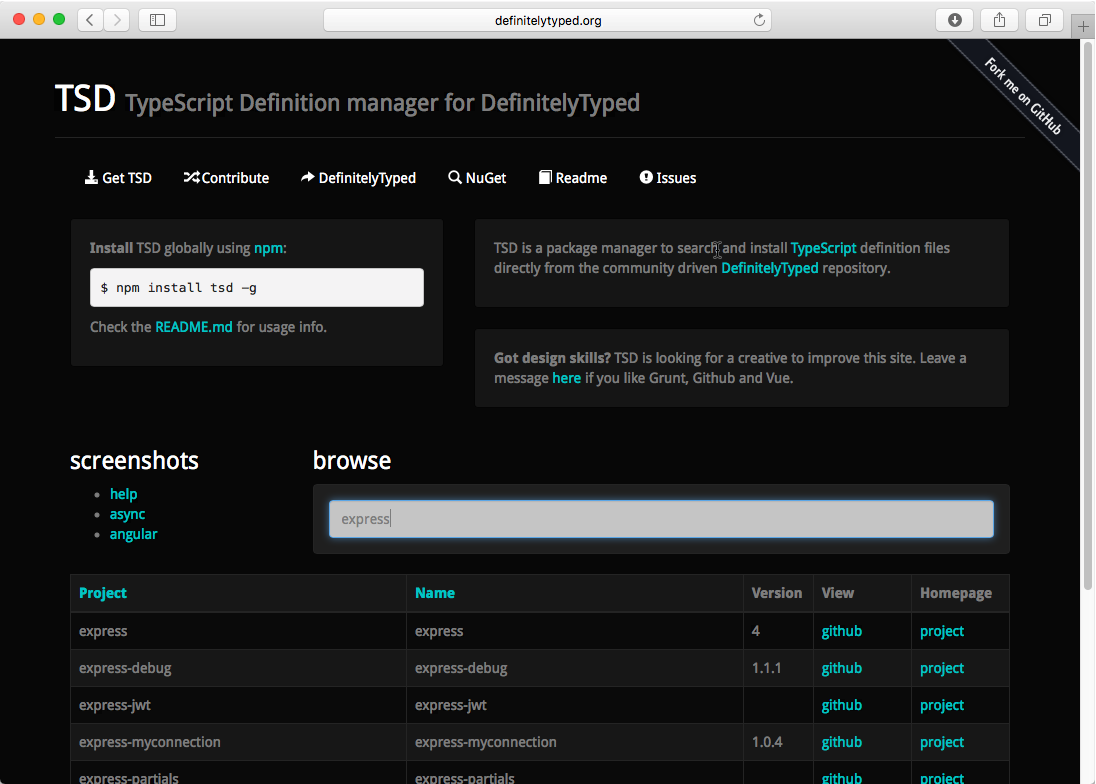
**express myNewWebSite**

Of course, this command will not install express itself and all needed components but it will create a new folder and a ready to use web application inside:



Therefore, to finish configuring the application you need to navigate Terminal windows to a new folder and execute **npm install**. Node.js package manager will add all needed modules and you can type **npm start** to execute the application. To test the result you need to open a browser and navigate to <http://localhost:3000>. The latest version of application’s template contains exactly this port.

Ok. We have created our web applications but what about Visual Studio Code? We can open the application folder right now and if you use Visual Studio Code you expect two important features: IntelliSense and Debugging. Once you open the folder you can see that IntelliSense system doesn’t work properly, because Code knows nothing about node.js and express. But we can provide all needed information using TypeScript Definition files. Visit <http://definitelytyped.org/tsd/> where you can find express and node.js definition files:



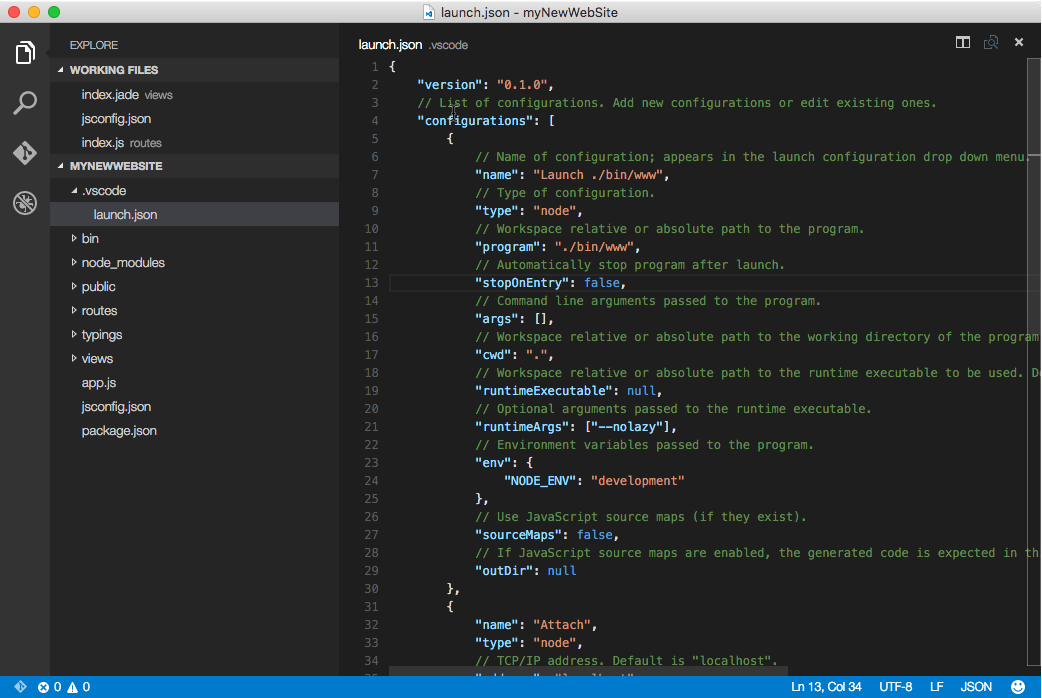
So, using **tsd** tool you need to execute two commands in context of your project folder:

**tsd install express**

**tsd install node**

Right after that you can use VS Code IntelliSense system without any problems.

In order to use Debugger you simply need to create a configuration file (launch.json). To do it, open Debug window and click Start button to force Code to create the configuration file for you. Visual Studio Code is smart enough to check **package.json** and read all parameters from there. So, in general, you should not modify **launch.json** – just check that name and program elements refer to **./bin/www**:



To run the application in Debug mode, you can open routes/index.js file, set some breakpoint there and click Start button in Debug windows. Once the application is started, open your browser and navigate it to <http://localhost:3000>. The debugger works great:

