[**https://github.com/Arkos-LoG/ReactContactsApp.git**](https://github.com/Arkos-LoG/ReactContactsApp.git)

[**https://github.com/Arkos-LoG/ReactContactsApp**](https://github.com/Arkos-LoG/ReactContactsApp)

**Setting up the React App**

Starting the React App is fairly straight forward:

Nmp install

Webpack make sure it got installed

npm run dev

go to localhost:8080

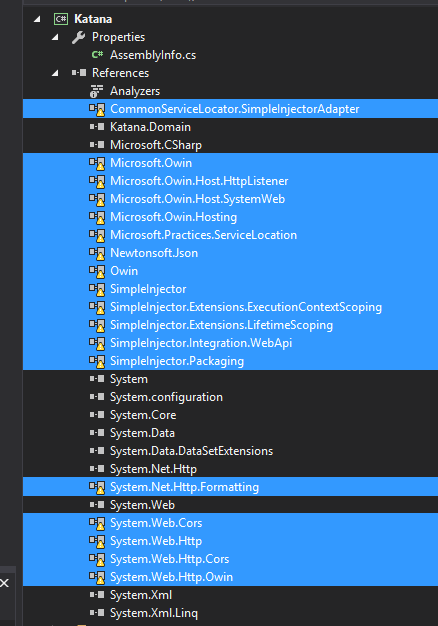
**Note: nothing will work until the Katana exe ( MS implementation of OWIN ) is up and running like in pic I included below.**

I chose to do WebAPI in selfhost, so you wouldn’t have to deal with IIS.

**Setting up the Server**

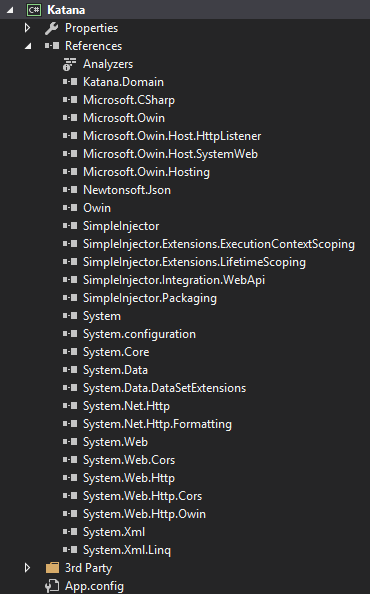
Getting the server working is bit more of a challenge. Bringing it down from github, it will have all sorts of dependency issues.

How I got it working is removing the bad references…

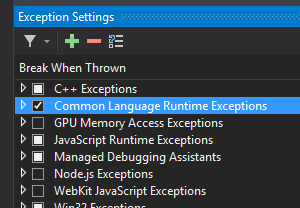


Then I wrestled with package manager and nugget until I had the following:

**Note:** For SimpleInjector, the dlls are in folder named SimpleInjector at the Katana project



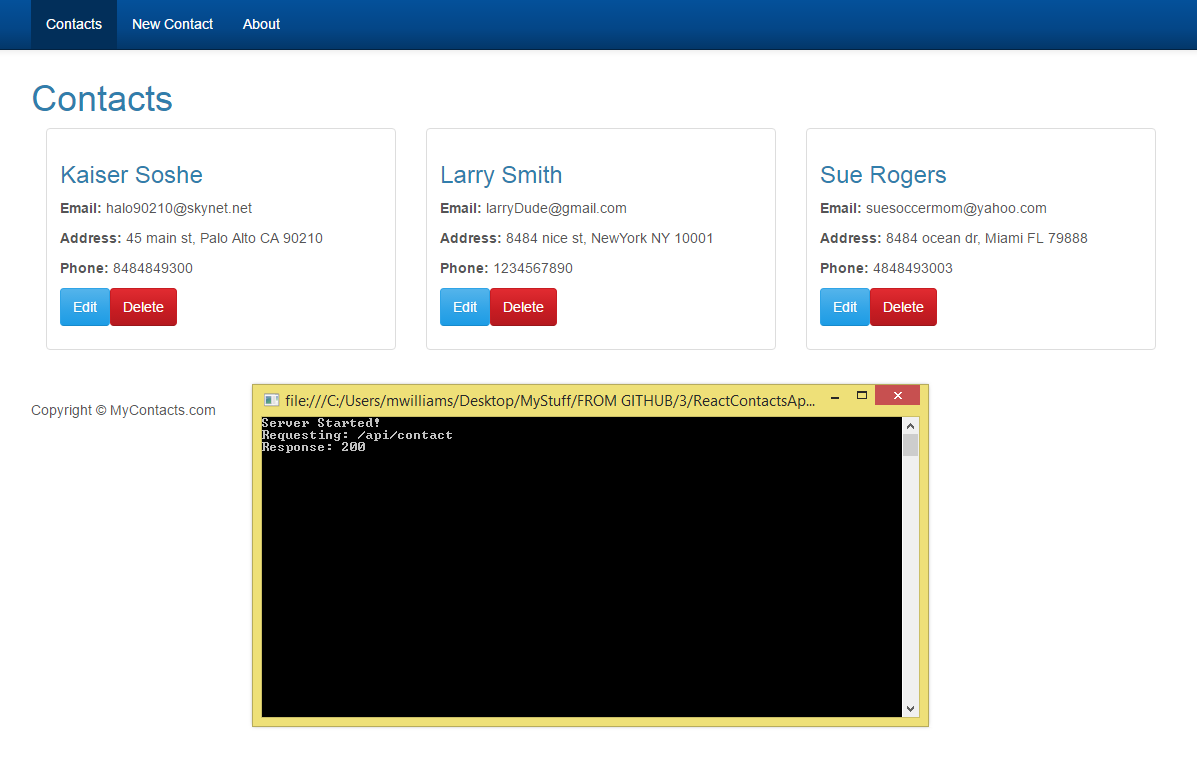
Just to make sure it’s all running correctly I ran it under the debugger with CLR exceptions turned on



I had it at one point where it would run, but something failed to load an assembly during runtime. The exception

Told me what else I was missing. There is probably a much easier way to do this.

When everything is up you should see the following. If you see Server Started! But no contacts, refresh the page.



**Items I’m still working on are the tests, and I need some better validation on for editing a contact.**

**React Contacts App built with the following:**

**Front End – with VS Code** <https://code.visualstudio.com/>

* React, Flux, React Router
* Bootstrap with <https://bootswatch.com/cerulean/>
* Responsive Design thanks to Bootstrap
* Babeljs for ES2015 features <https://babeljs.io/docs/learn-es2015/>
* Webpack
* Webpack-Dev-Server localhost:8080
* axios for promise based HTTP client <https://github.com/mzabriskie/axios>
* Nodejs Event Emitter - used in the Store part of the Flux pattern
* lodash <https://lodash.com/docs>
* react-validation-mixin for validation <https://github.com/jurassix/react-validation-mixin>
* joi for object schema validation <https://www.npmjs.com/package/joi> <https://github.com/hapijs/joi>
* Tests are in progress…

**Back End - with VS 2015**

* Katana: Microsoft's implementation of OWIN (Open Web Interface for .Net) to self-host the Web API framework
  + The server is using localhost:9000; CORS enabled to avoid Access-Control-Allow-Origin errors between the client and server
* SimpleInjector (included in the Katana project in 3rd Party folder) <https://github.com/simpleinjector/SimpleInjector/>
* Tests are in progress

**Trouble shooting tools**

* Fiddler
* Beyond Compare
* Agent Ransack