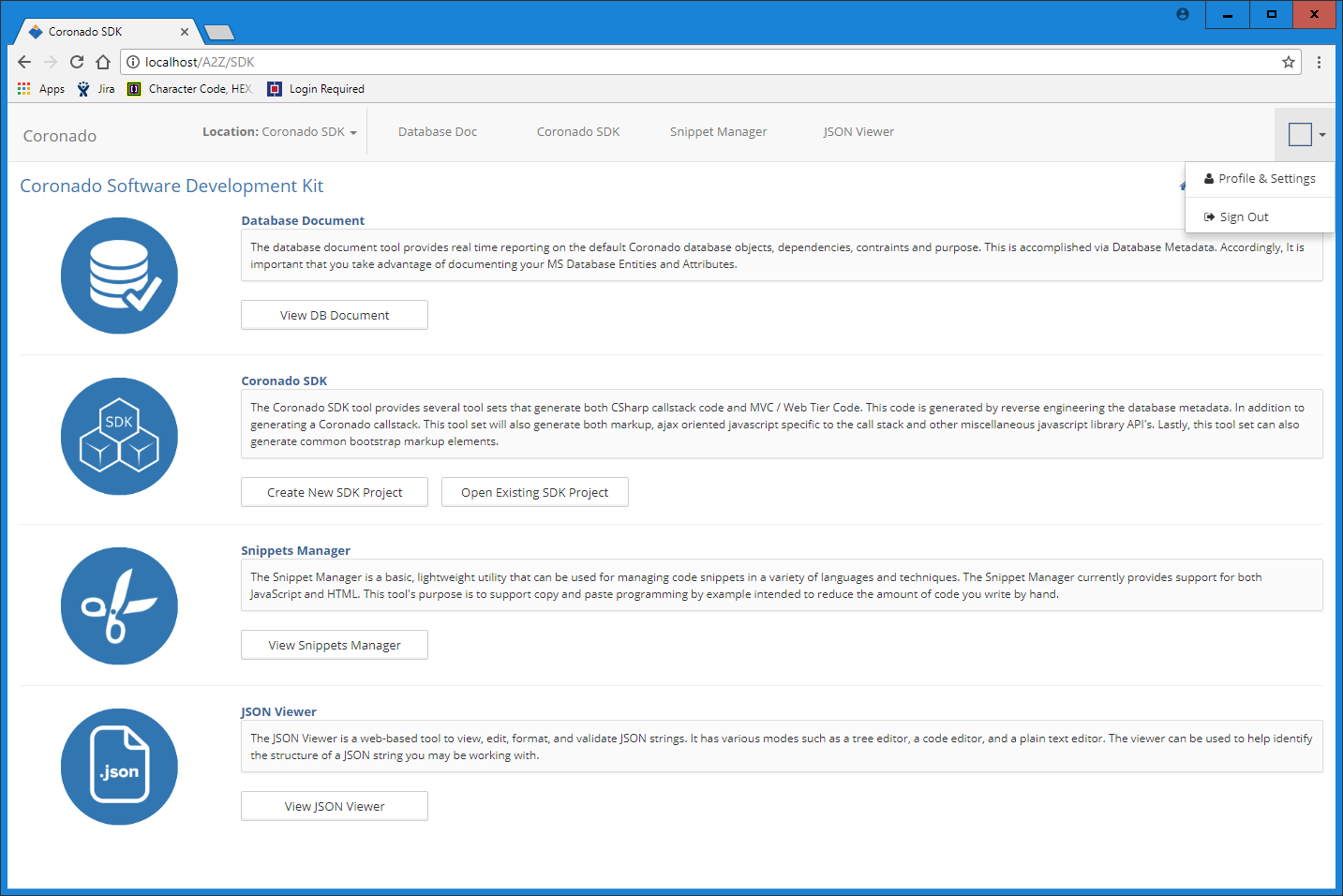
Open SDK Integration

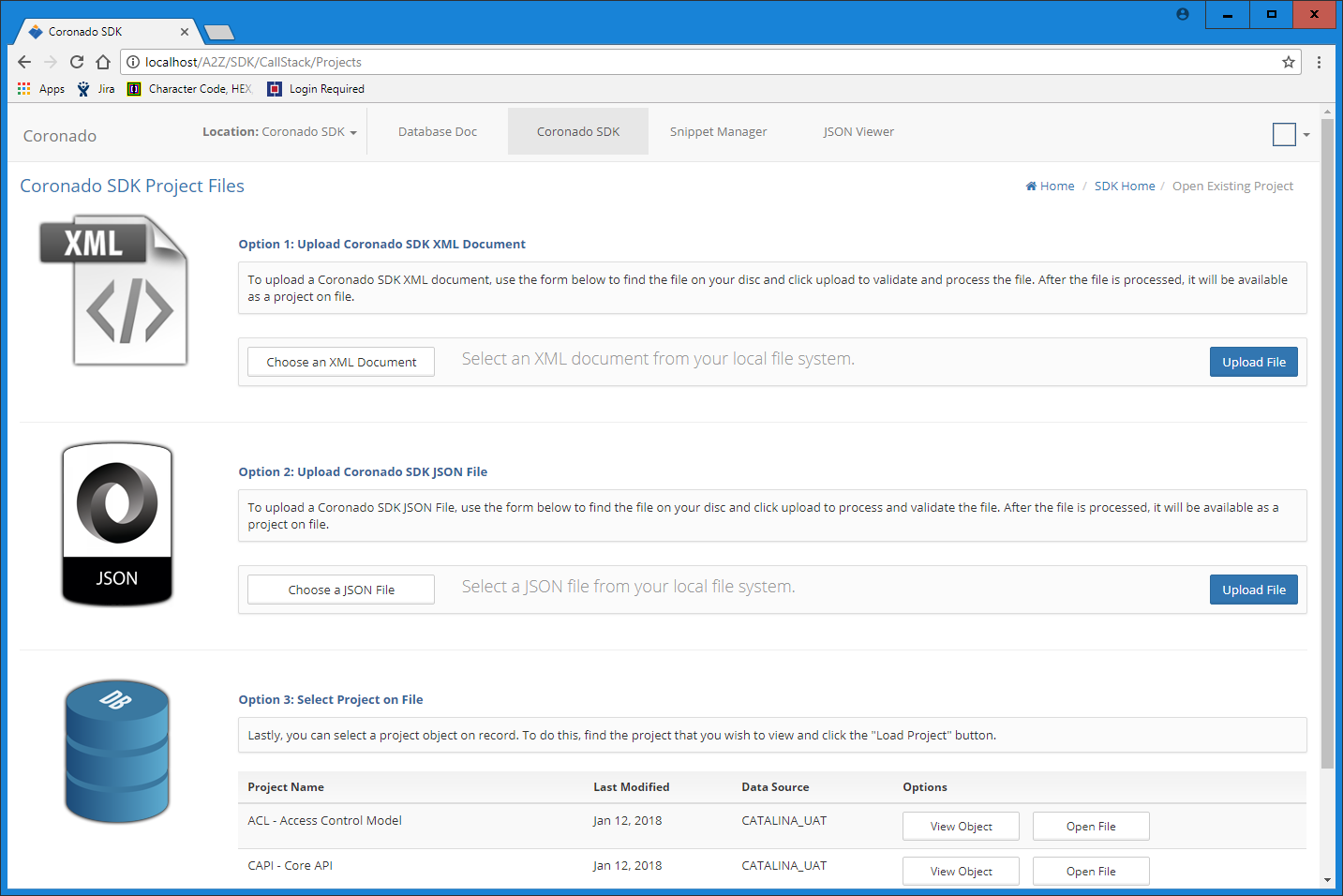
The following screen capture demonstrates a stable working copy of the SDK that is being rewritten to support other languages besides JavaScript, ASP.NET and C#. The purpose of this SDK is to provide a tool that allows you to plan, design and document your project where you can then generate coded functionality as needed. This code is then incorproated with your IDE of choice.

The following screen shots illustrate real working software written by Sean McNeill

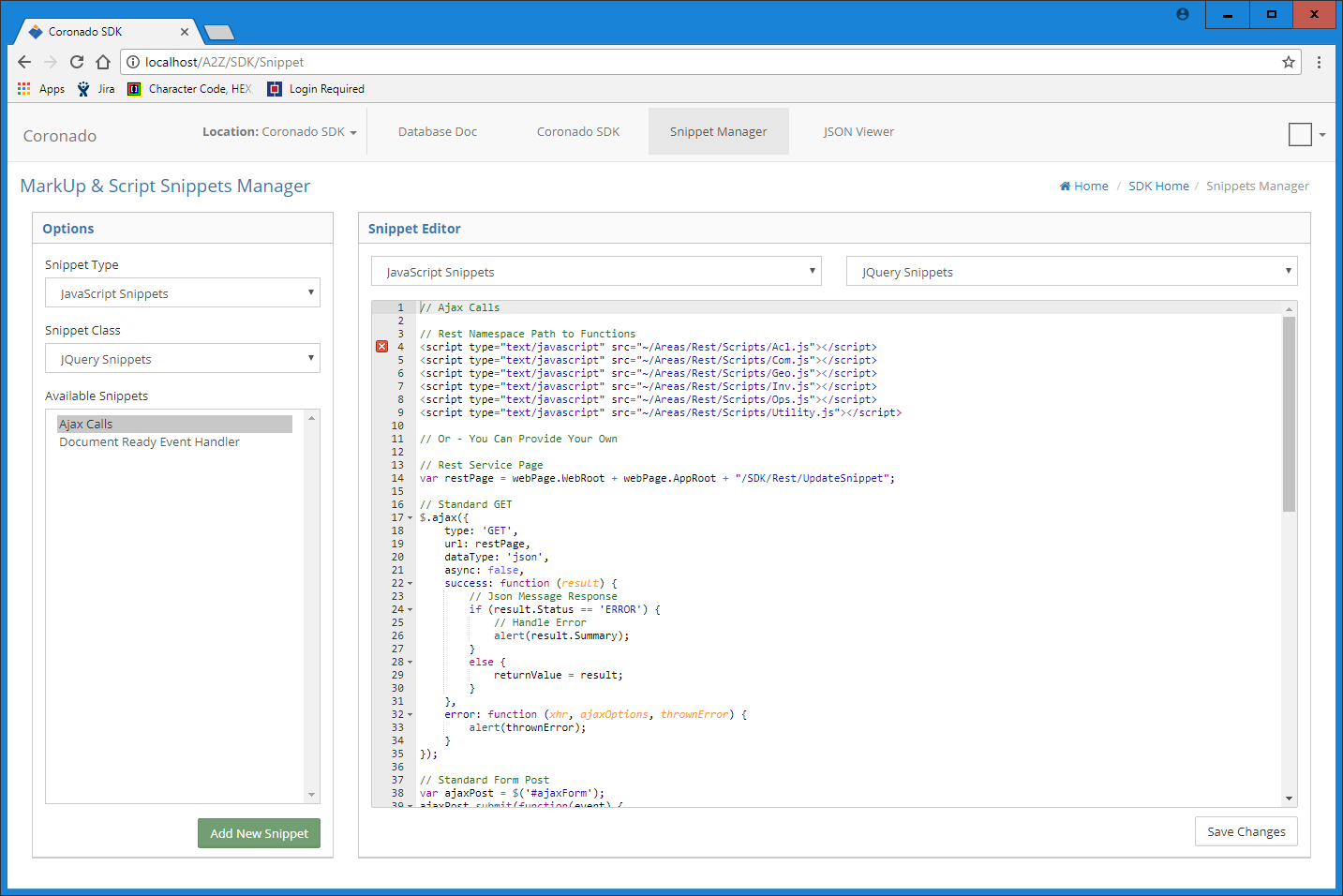
FOR YOUR EYES ONLY



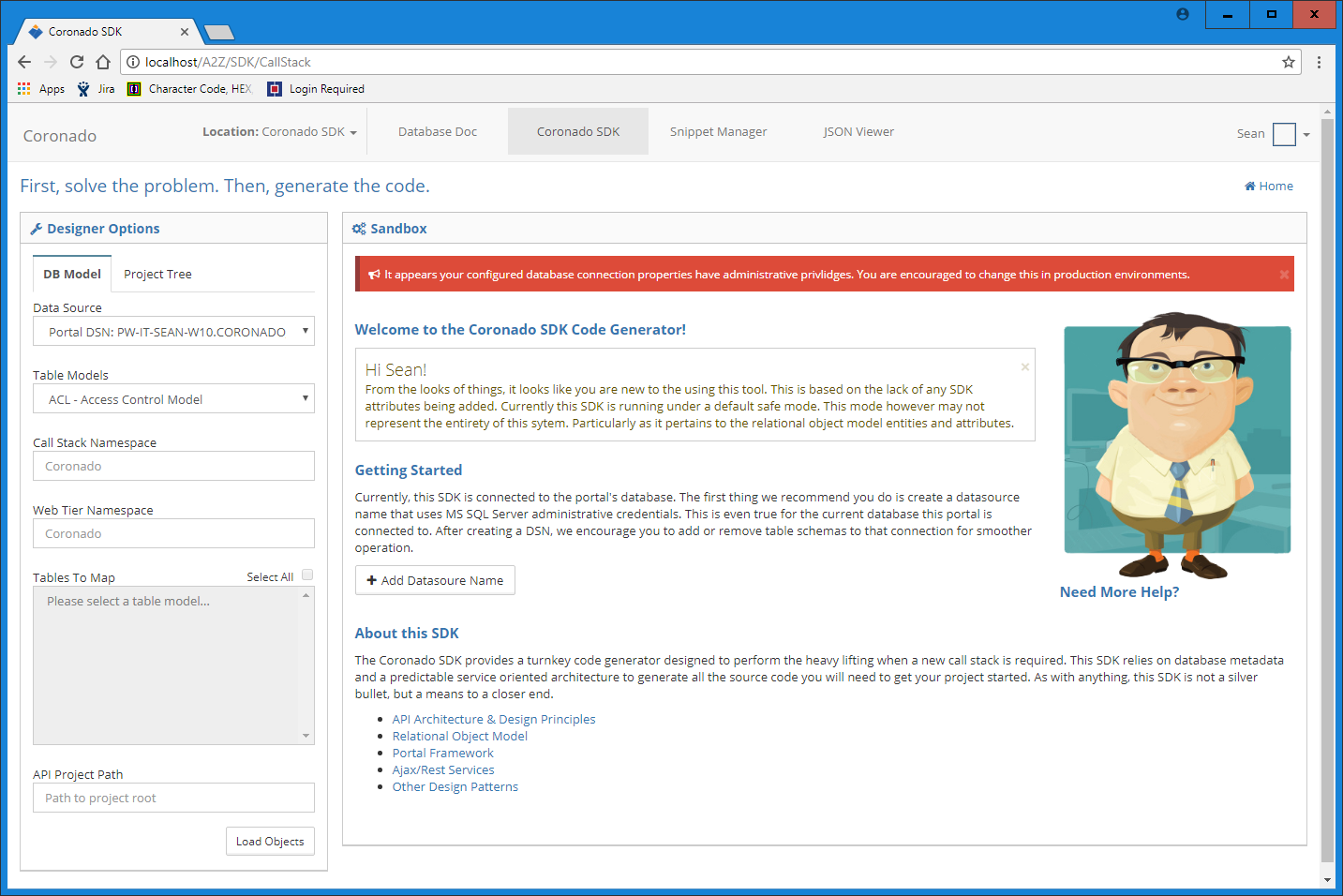
Current dashboard, next version will contain key performance analytics and charts



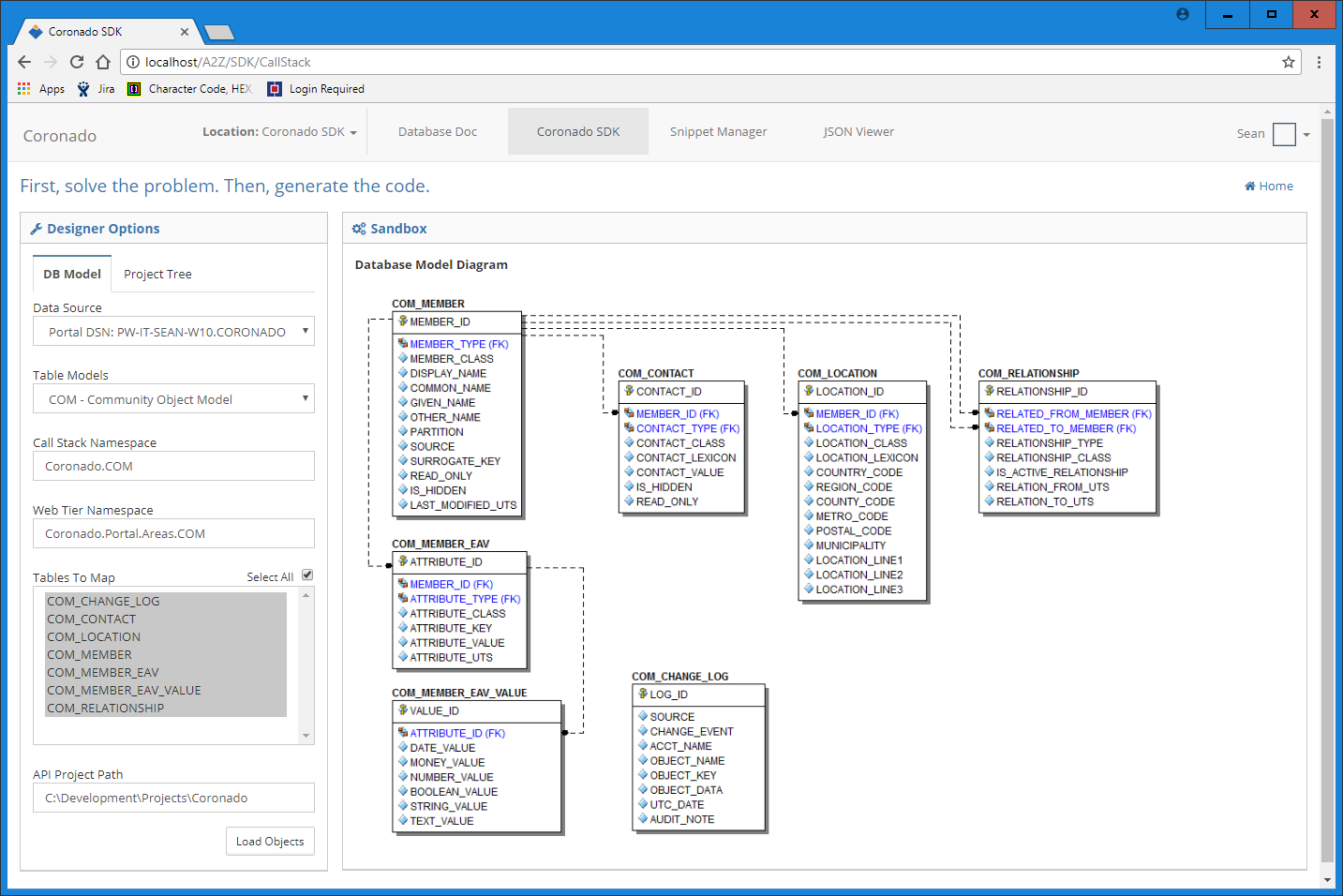
Code generation is only part of a larger project. Managing projects can start with a simple file and be cast into object models and saved in a data source of your choice



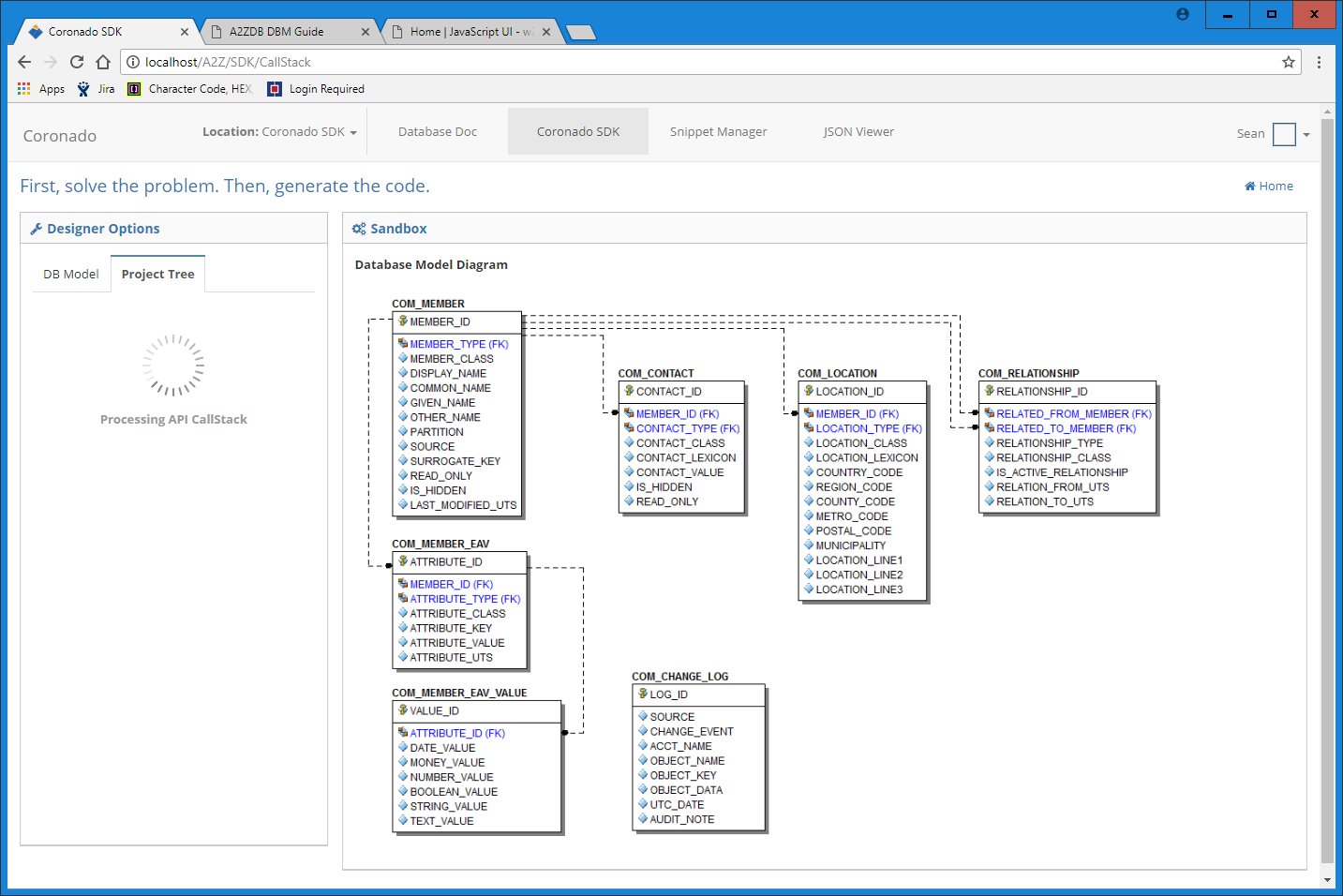
Example of generic Snippets Manager

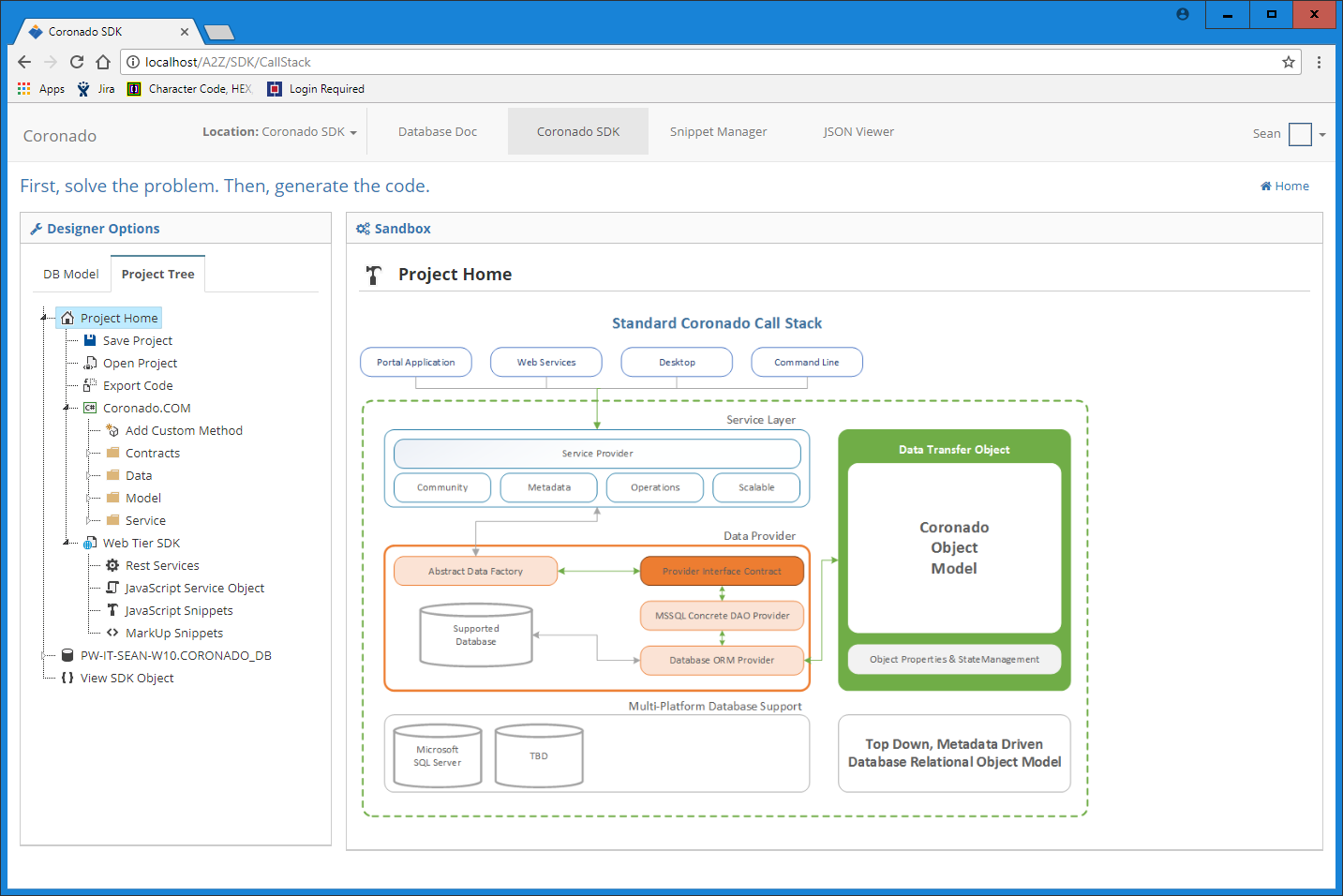


Example of SDK Design Screen. The SDK creates SOA API’s with a service layer. This call stack can be generated via reverse engineering the database layer.

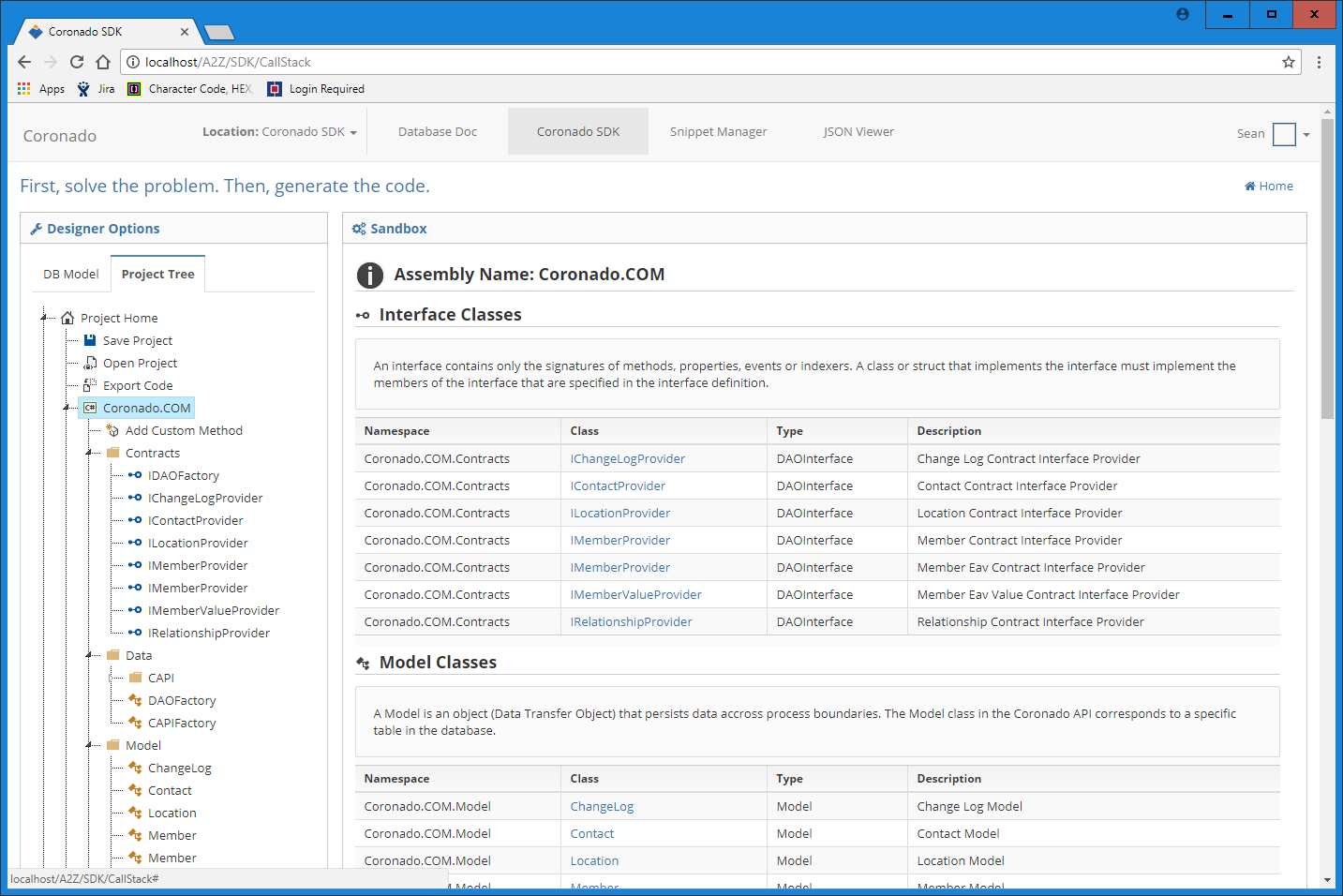


After a taxonomy is selected, you click the load objects button, a progress indicator displays while the call stack is generated (2 to 5 seconds)

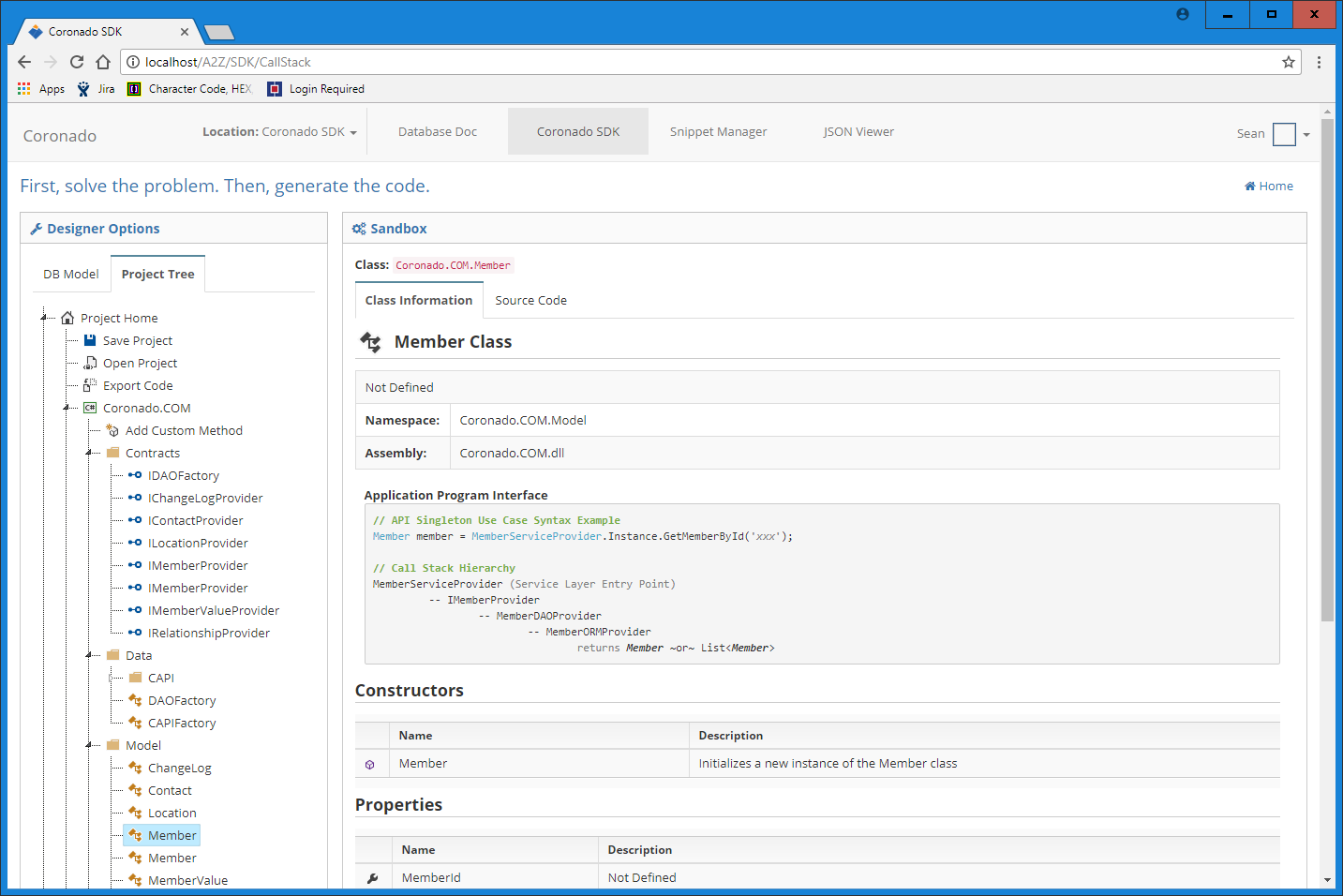




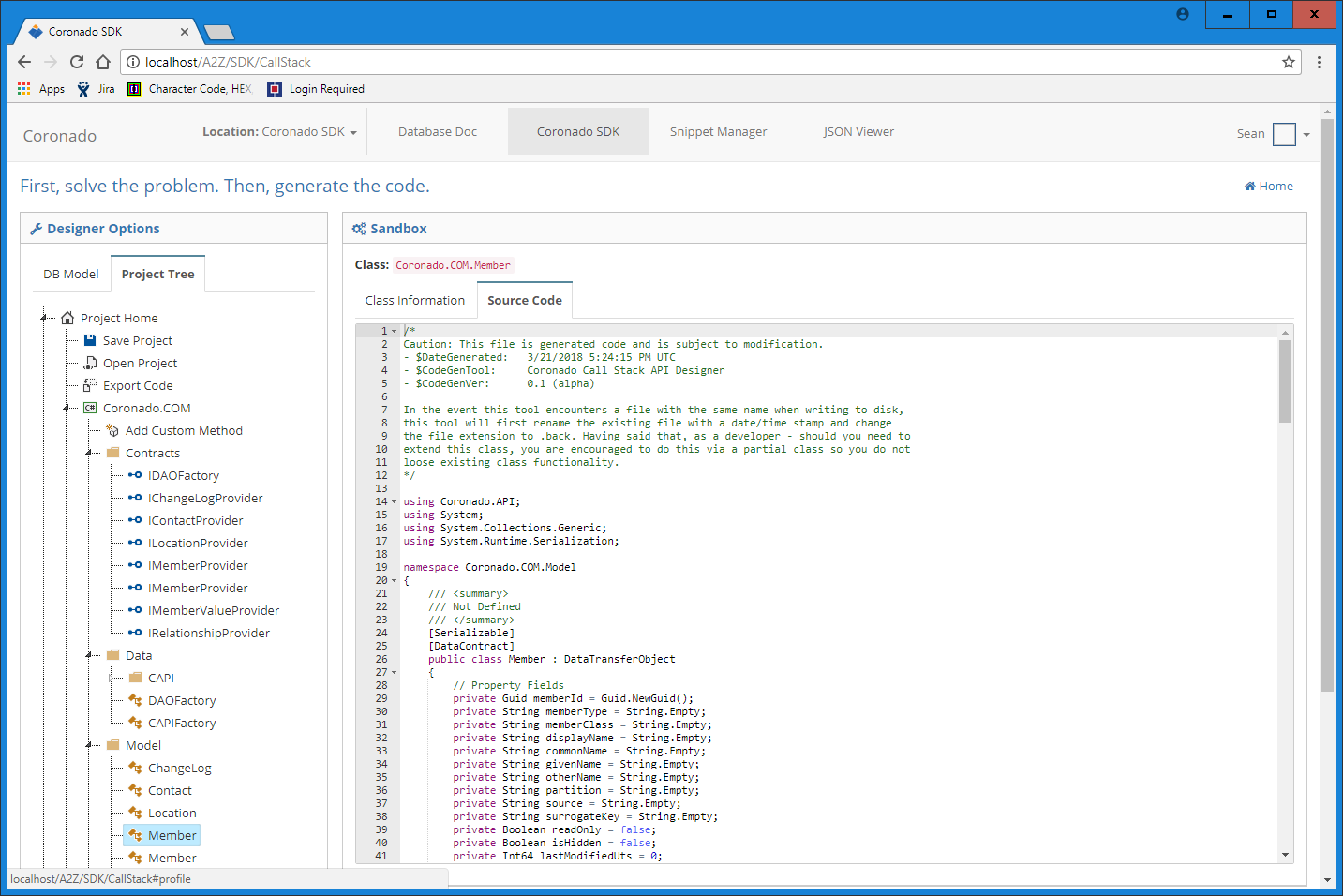
After all the project objects are marshalled and serialized, a project tree is loaded allowing you to see the source code manifest



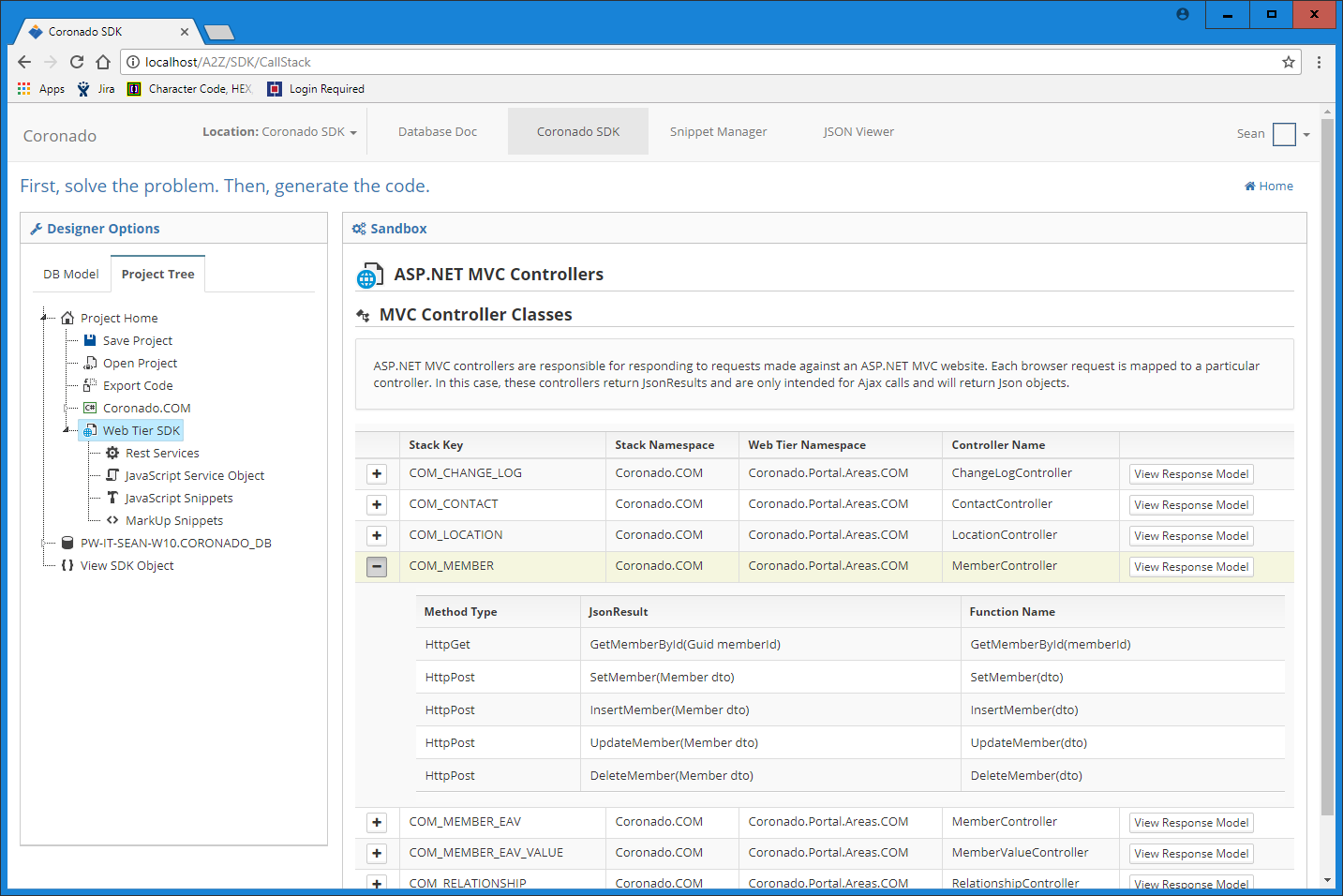
Source code is presented as a documented project where you can add additional project information.



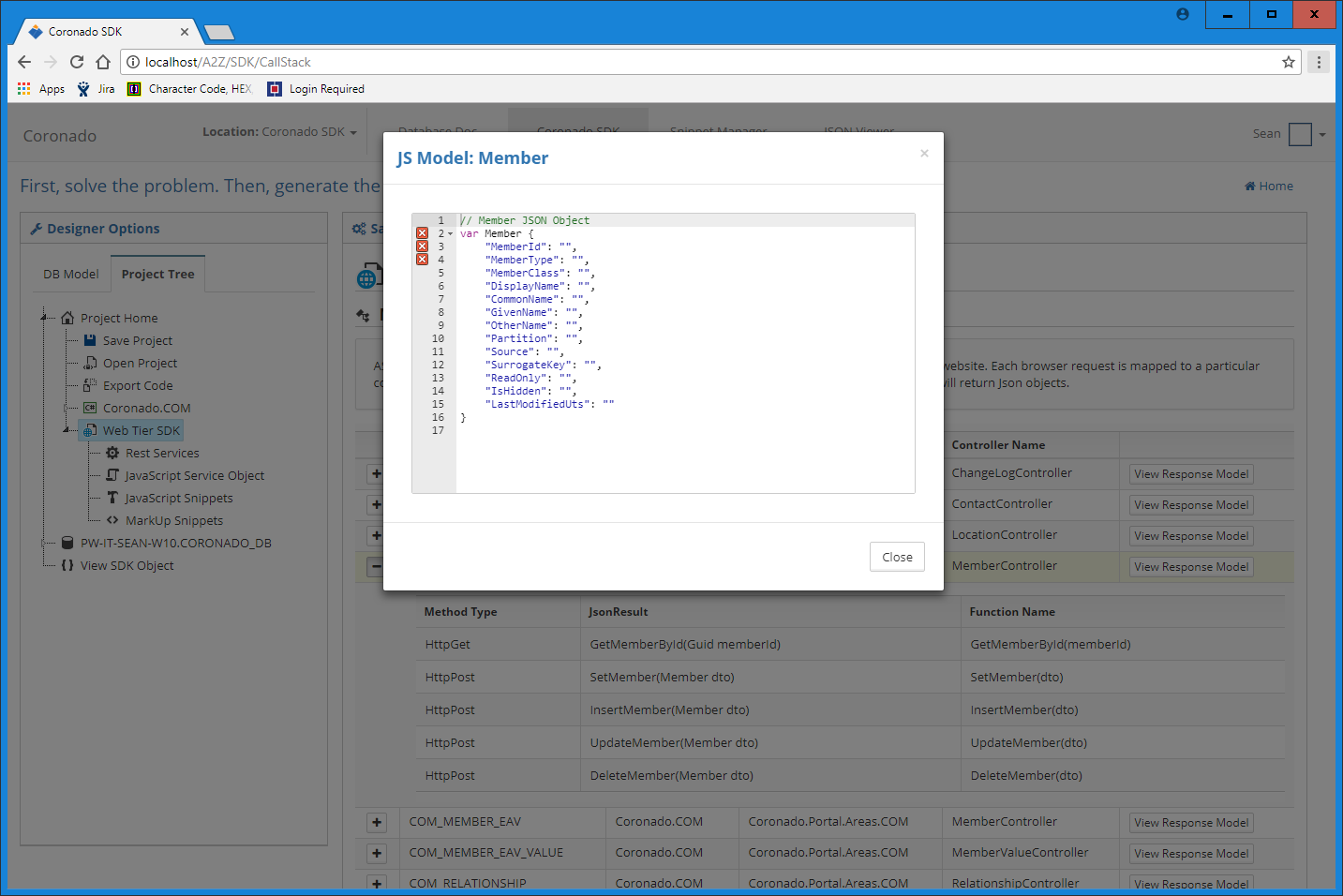
The SDK also serves as your point of API definition and documentation



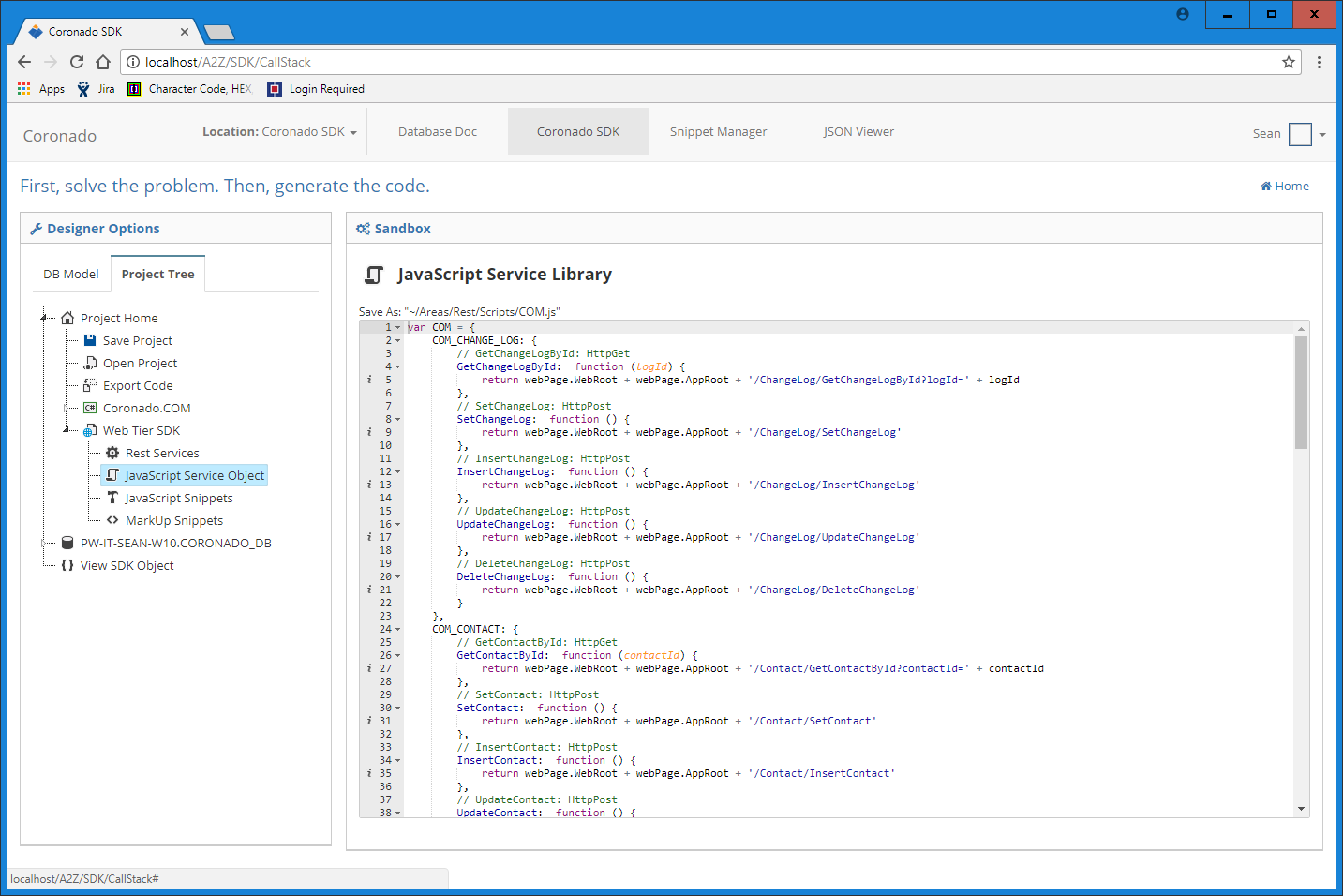
Lastly, the source code is made available for your review. Changes can be made at this stage before the code is written to disk, methods can be added, object names can be changed. Full refactor capabilities



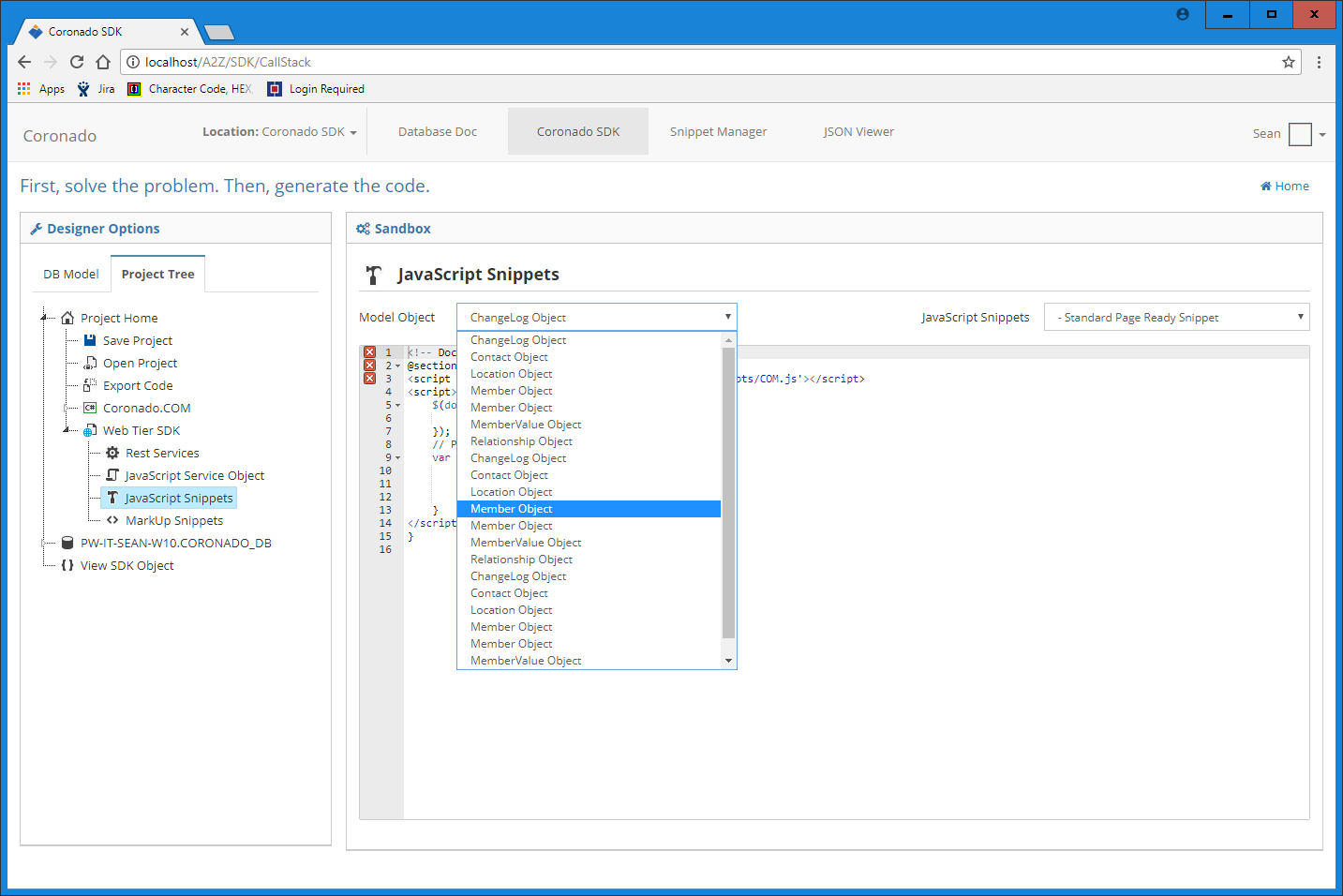
In addition to the back end call stack, we also assemble building blocks for the web tier, specifically ASP.NET MVC w/Rest Services and jQuery libraries into the API



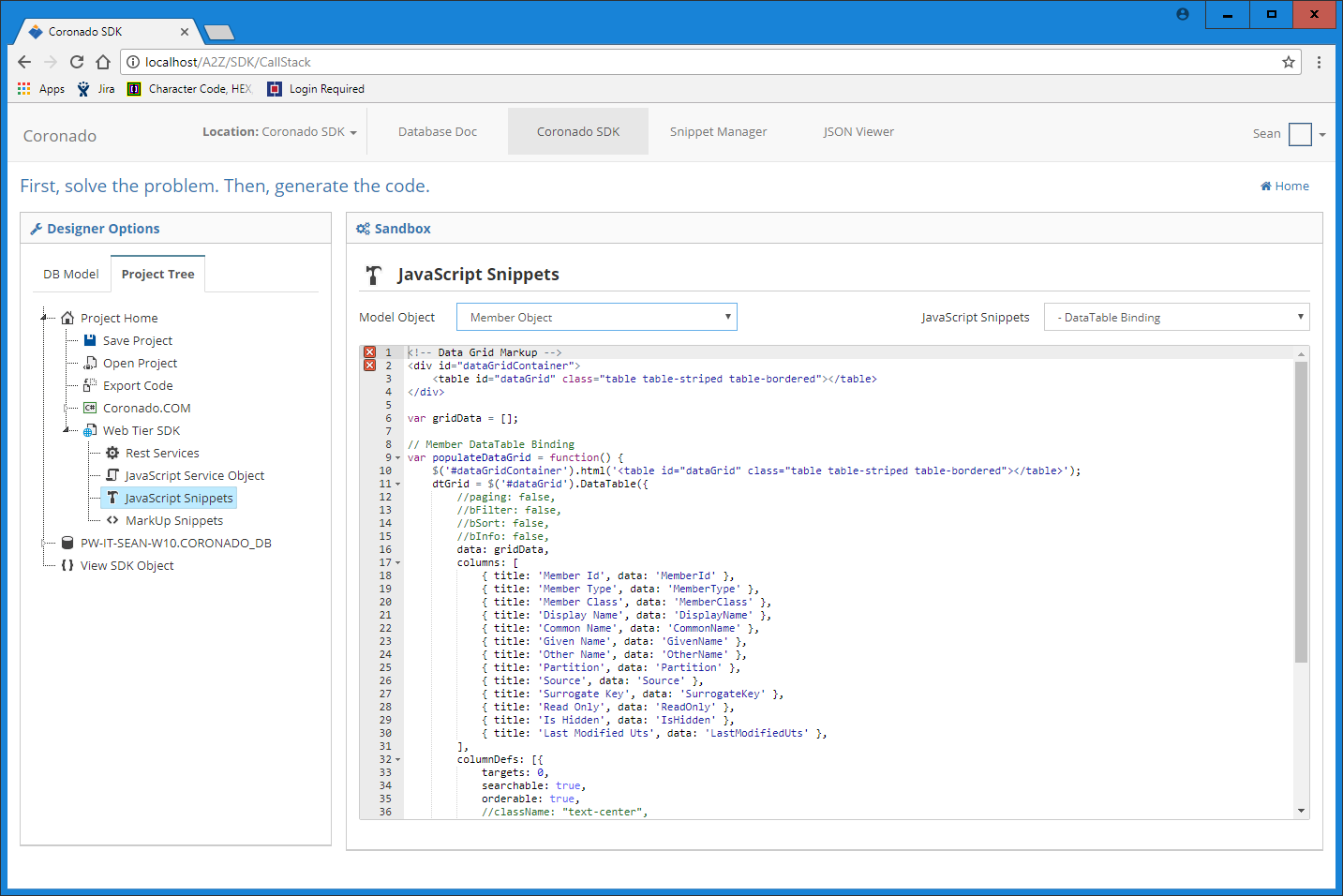
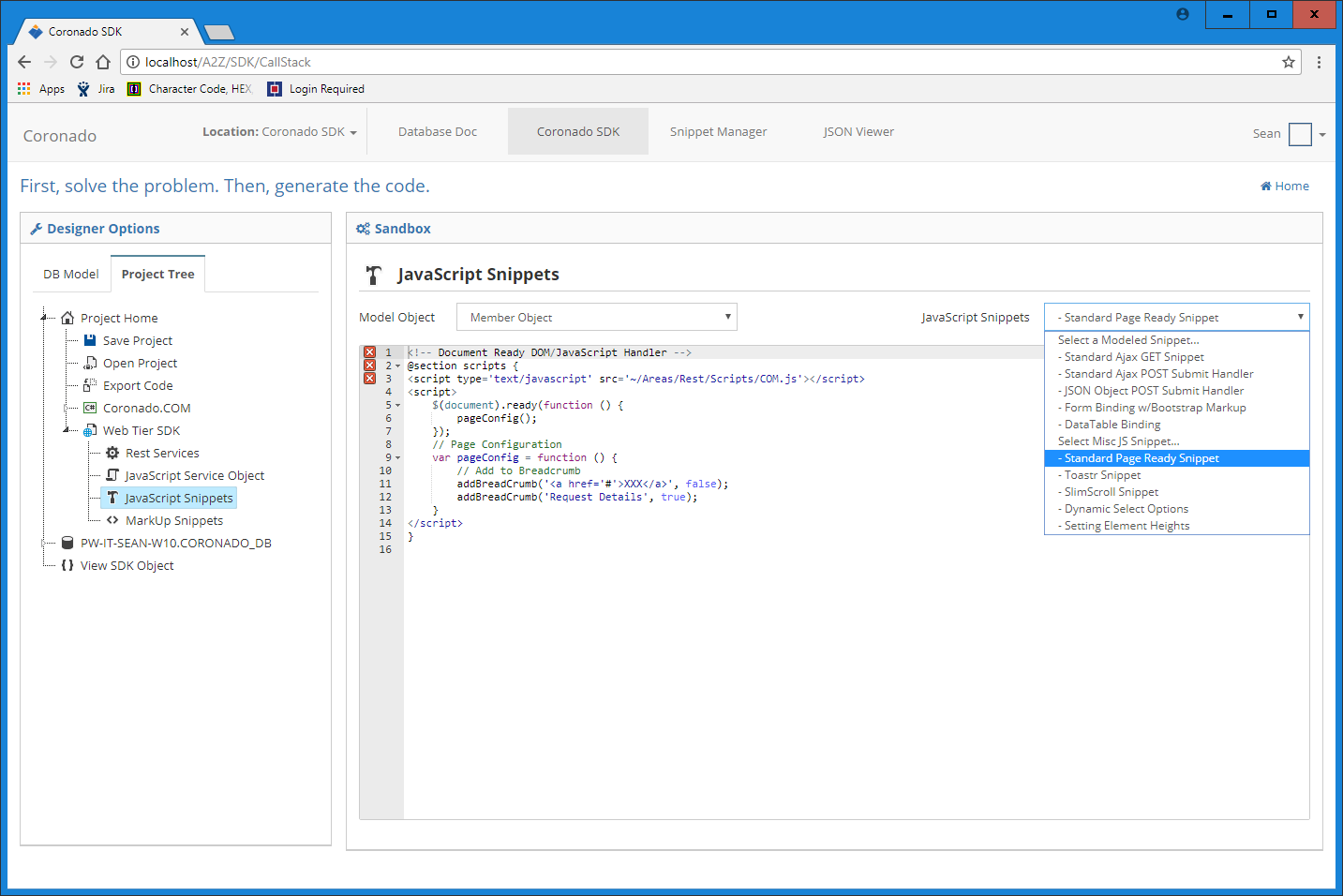
Here you can copy JavaScript or just review what the objects look like that you are interfacing with



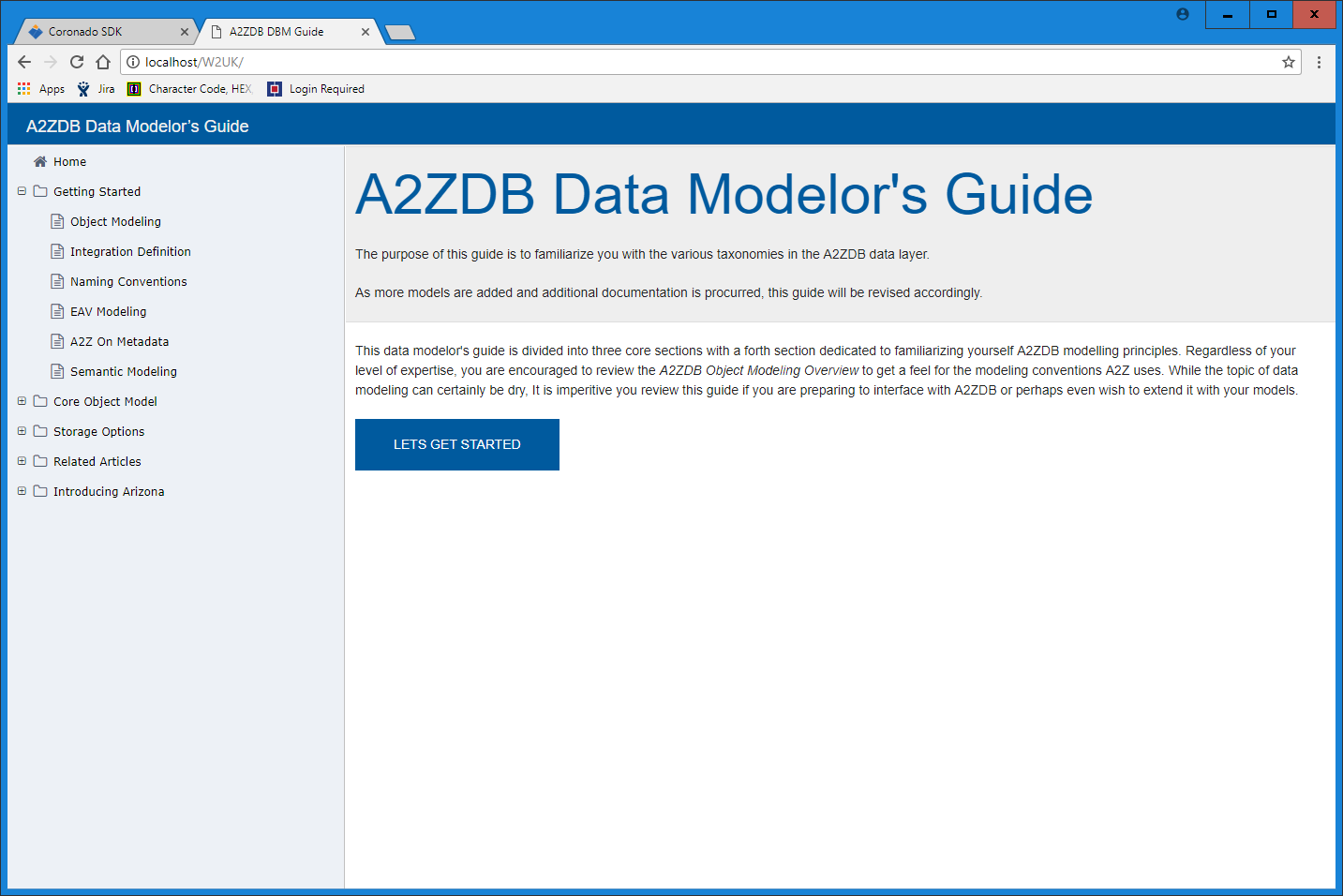
The SDK will also write a service layer JS file to serve as a pointer to your rest services  
Example: url: COM.COM\_CONTACT.GetContactById(‘XXX-000-999’);



Object sensitive JavaScript generation as demonstrated in the next few screens



// End of Screen Captures of Working SDK Prototype



My Prototype of W2UI – Simple Documentation. I am looking into creating a state factory to manage how your containers unfold based on user preference. I am also going to switch over to css sensitive SVG iconography with png fallback as well as a theme engine that supports (MS Cobalt, Amazon, Google Material, GitHub, Bootstrap and Twitter styles) as well as design templates that are maintained in a CMS (Content Management System). Basically, eventually the CMS component will spit out docs like this with even more component options as well as serve as the basis for auto generated application interfaces using pure JSON. Even cooler is these docs will be able to stand alone with static json options or run with live db connections via rest. Everything you saw in the SDK is going into w2ui

