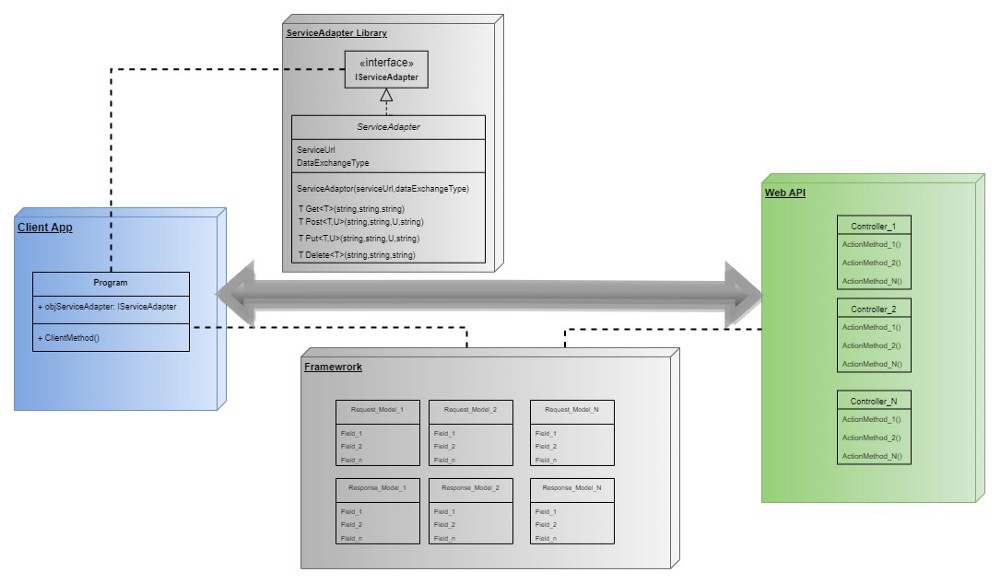
### Building Simple Web API Service Adapters in .NET

Author: **Sanish Abraham**

No doubt that the introduction of REST API revolutionized the technology landscape in a short span of time. It triggered the metamorphosis in the information exchange systems, especially the transition from Web Services to Web APIs in the software development. In the .NET world, WCF still got its place considering the support for multiple transport protocols, encodings, transactions, message security and exchange patterns. But Web API is an open source light-weight HTTP only framework which supports wide variety of media types with a broad range of clients including browsers, mobiles, tablets, IoT devices etc.

But for those early migrators of Web APIs, a major hindrance was the extra hardship on parsing and reading the xml or json response manually. No need to ask the component creators how much they might have missed the Web API equivalent of WSDL proxy creation in .NET projects. Life was so easy with that ‘Add Web Reference’ proxy creation tool within the Visual Studio IDE or the command line wsdl.exe utility. Consequently, they had to go through the hassle of manually parsing and processing the XML or JSON responses. And that was exactly the reason, why back in those days, I had no choice other than to create our own simple adapter for connecting clients to REST API handling all the cumbersome tasks under the hood.

Time has changed and now there are many solutions available on the market based on the structured metadata for OData integrated APIs. Swagger based on Open API standard provides some client generator tools. If we are creating the public REST API, we can rely on swagger and make use of it to publish your service interface. Even though to use that within the same project or domain, is definitely an overkill. And more importantly, if the service response data classes change frequently during the development and the client need to update the code accordingly, then using our own light weight adapter is efficient in terms of time and effort.

A Basic ServiceAdapter Usage Architecture

**Read the full article here:**

Medium: <https://medium.com/@sanish.abraham/building-simple-web-api-service-adapters-in-net-72df98620d83>

Blogspot: <https://saneecodes.blogspot.com/>

LinkedIn: <https://www.linkedin.com/in/sanishabraham/>

GitHub: <https://github.com/mailsanish/ServiceAdapter>