Notes

1. SOA = Service-Oriented Architecture is a software design paradigm where services are provided from one API resource to another API resource through a communication protocol over a network
   1. Properties
      1. SOA is a response to need for software that functions in the heterogeneous and distributed environments of today
         1. Loosely coupled services + service interface + protocol = essence of SOA
         2. Service bus = a common interface & set of protocols for services to communicate
         3. “Integrate diverse services through a common logic that guides users through the service chain/service use
         4. …
         5. An architecture oriented around services….
         6. A Service oriented architecture
      2. Focuses on services over components
      3. Interoperability & cross platform
      4. Loosely coupled & distributed
      5. Needs a high level of abstraction
2. SOAP
   1. Simple Object Access Protocol
3. Benefits of loosely coupling
4. .NET usage of git
   1. Just know that there are different SOP’s for using git
5. Recreate the actual MVC as an API
   1. Reference the Logic, Models, Repo (recall they are like their own projects)
6. If you want to create an MVC as an API you only need to change the front-end
7. Swagger & open API practically synonymous
   1. Why?
8. Copy +Pasting Special from successful (200 level) API output allows you to create class objects of the JSON structure in VS
9. Swagger also allows access to the JSON format of the output
   1. This enables us to create clients
10. Documenting REST APIs & OPEN OPI
    1. Swagger
    2. Postman
11. AJAX = “is a set of web development techniques that use the JS XMLHtttpRequest object to communicate with servers asynchronously.”
    1. Properties
       1. The XML object can receive data in multiple forms, such as JSON, HTML, XML, text etc.
       2. Two major features:
          1. Make requests to the server without reloading the page (due to its asynchronism)
          2. Receive and work with data from the server
       3. AJAX lifecycle (conceptual)
          1. An XHR object is made
          2. An XHR object is sent over the internet
          3. Server receives the XHR
          4. Server processes the XHR and returns a response over the internet
          5. Client’s browser receives the response and processes using JS
          6. Data is used accordingly (such as updating page content)
       4. AJAX lifecycle (programmatically)
12. JavaScript
    1. Classes
       1. Have no hoisting
       2. Must have one constructor in the method and only 1 declared
       3. Have Class Expressions & Class Declarations
       4. Getters & Setters (Accessor properties)
       5. Prototypes
          1. A default property in the constructor that allows for inheritance in JS
          2. It is outdated & deprecated, but some scripts still use it & still built into JS