Module 3 Vocabulary

**MVC:** <https://www.tutorialsteacher.com/mvc/mvc-architecture>

MVC stands for Model-View-Controller and is an architectural software pattern for implementing user interfaces. The MVC works by separating the application into three components; Model, which represents the shape of data and business logic as well as maintaining data and using objects to retrieve it. The View is the user interface, this allows users to see data and modify it. The Controller handles user requests; it typically does this by getting information from the View and then doing work to generate the appropriate response.

**Routing:** <https://www.tutorialsteacher.com/webapi/web-api-routing>

Routing in Web API is used to route incoming HTTP requests to a particular action method on the Web API controller. It does this in one of two ways; Convention-based Routing has Web API use route templates to determine which controller and method to use. This style only works if at least one template has been added to the route table. Attribute Routing, supported in Web API 2, uses the *[Route()]* attribute to define routes.

**Status Codes:** <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

Status Codes are HTTP responses that are sent when a user performs a request using HTTP. Since Web API relies on HTTP, it also uses status codes to relay information to the user. There are five levels of status codes 100, 200, 300, 400, and 500. Level 100 status codes are information responses, but they aren’t used in Web API. Level 200 status codes are used to tell the user their request did what it was supposed to; codes often used for that include 200-OK, 201-Created, and 204-No Content. Level 300 status codes are used to redirect the user elsewhere when necessary, but are generally not used in Web API. Level 400 status codes tell the user they did something wrong; this is commonly done with the 400-Bad Request, 401-Unauthorized, 403-Forbidden, 404-Not Found, and 409-Conflict. Lastly, level 500 status codes are used when server has a problem, usually this is done with 500-Internal Server Error.

**Inversion of Control:** <https://www.tutorialsteacher.com/ioc/inversion-of-control>

Inversion of Control (IoC) is a design principle used to invert different controls in object-oriented design for loose coupling. This means that a class will transfer any responsibilities it has other than its main one to another class. This could include control over the flow of an application or maybe object creation.

**Dependency Injection:** <https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/>

Dependency Injection is a technique where one object/static method supplies the dependencies of a different object; dependencies are objects that can be used. This is done by using dependency injection as an intermediary that can inject necessary dependencies at runtime instead of compile time.