For this discussion post I want to discuss Angular Observables and Observable Streams. I will define these two terms as well as give a brief explanation to what they actually are and how they operate in Angular. So, what are Angular Observables?

“Observables provide support for passing messages between parts of your application. They are used frequently in Angular and are a technique for event handling, asynchronous programming, and handling multiple values.” (*Angular*, n.d.) The observer pattern is a software design pattern that involves an object, called the subject, maintaining a list of its dependents and notifying them automatically of state changes. This pattern is like the publish/subscribe design pattern but is not identical. Observables are declarative, meaning you define a function for publishing values. The function is not executed until a consumer subscribes to it and publishes values, at which point it will complete. An observable can deliver multiple values of any type —literals, messages, or events—depending on the context. The API for consuming values is the same whether the values are delivered synchronously or asynchronously. Because setup and teardown logic are both handled by the observable, your application code only needs to worry about subscribing to consume values and unsubscribing when that is done. (*Angular*, n.d.)

To understand how observables work, we need to go back to the fundamentals of what a stream is. A stream is a sequence of events, where each event consists of different data. In an observable, you define a stream as an ordinary function that converts data into an observable stream. When you want to use it as an observable, you pass it an observer function as its argument. Just like any other function, the observer function gets called whenever you want something to happen—in this case, when someone subscribes to the observable. The observer function can then do anything it wants with that data; for example, it could print it out or extract the first few lines. As observables are declarative: You define them much like any other variable in your codebase (e.g., by declaring them in variables). This means that they won't start emitting values until someone subscribes them (i.e., calls their observers). (*Angular Observable Tutorial*, 2021)

In conclusion we now have a basic understanding and have defined what Angular Observables are and what an Observable Stream is. Observables are declarative, meaning you define a function for publishing values. In an observable, you define a stream as an ordinary function that converts data into an observable stream.

References:

*Angular*. (n.d.). Retrieved January 23, 2023, from <https://angular.io/guide/observables>

*Angular Observable Tutorial*. (2021, August 20). TekTutorialsHub. Retrieved January 23, 2023, from <https://www.tektutorialshub.com/angular/angular-observable-tutorial-using-rxjs/>