**Explain what is meant by the stream abstraction. What is the relationship between streams and the observer pattern? What are streams useful for modelling and when might you use them in Rich Web development?**

Streams are an abstraction used to model asynchronous data sources, as it is a sequence of data, sequences known as arrays and lists, stream implement the observer pattern where data is realise using the subscribe operation. Streams allows to process data one by one, the data shows in memory, and forget it after passing it further.

The difference between them is that the stream may not be available all at once, as what makes the streams powerful when working with large amount of data is that they don’t have fit in memory. The observer pattern is objects interface provide a generalised mechanism for push-based notification, the observable object shows that the object that sends notifications the provider, the observer object shows that the class that receives them.

**Assume that you are building an interface to an API in your Rich Web App. Describe in detail how you could use the RxJS library to handle asynchronous network responses to API requests. In your opinion, what are the benefits to using a streams library for networking over, say, promises? And what do you think are the downsides?**

The RxJS is offering a strong approach of dealing with events and with integration into a high number of frameworks, utilities and libraries. The good thing about using the Rx it can be used for asynchronously integrating any event based process with any other process, it uses Rx to make asynchronous calls to a web service from a button click in a web page, the button click is such a sequence of one event to use the Rx in the client as also side web environment, it will use Rx for JavaScript(RxJS).

Basically, Rx used for monitoring a sequence of events from a long running task. The API calls is a staple on with many benefits from offloading work of designing reliable network behaviour.