Part 7 Lecture Review Questions

1. Stream abstraction – streams are an abstraction used to model asynchronous data sources. They are an abstraction used when reading or writing files, or communicating over network sockets.

Relationship between streams and the observer pattern – Streams implement observer pattern. They implement where data is realised using the subscribe operation. Once a stream is created over some future data, the stream’s data can be operated on and transformed into new streams.

What streams are useful for modelling and when might you use them for Rich Web Development – in the context of rich web app development, an obvious application for streams is as an abstraction of asynchronous events.

1. Describe in detail how you could use RxJS library to handle asynchronous network responses to API requests – first you install it [ $ npm install –save rxjs]. Then you import ( Observable ) from ‘rxjs/Rx’. The first constant gets element by getting the id, and then another constant is combined with the first constant using Observable. There’s error handling and events are dealt with separately.

What are the benefits to using a streams library for networking over promises and what are the disadvantages – Promises solve the specific drawbacks with callbacks and are fine but they don’t really help with the larger data synchronisation problem we face in UI design and implementation.