Promises

To make it easier to work with asynchronous calls (like requests to the server), we decided we would use Promises (with the help of the RSVP.js library). Promises are objects that contain function(s) that are asynchronous. A promise can help you solve the so called callback hell.

For example, you have an asynchronous function called “get” (for example, this will send a get request to a server). You can wrap this function call within a Promise object. When the function is done, you can call the “fulfill” function (this is a parameter within the promise constructor) with the results you want to pass on. You can see an example of this in the OctopeerService.js file.

When you have created your promise object, you can call the “then” function on that object. This function is called when the fulfill method is called in the promise object. You can also add a catch method, which will be called when an error is thrown.

Then “when” function is used when you have several function calls, and fulfill will be fired when all of them are done. An array of promises can also be handled, using the all(<array of promises>) functions.

For more clear information about promises in general: https://promisesaplus.com/