**Promises Exercise:**

The following github repository is implementing a server which tries to send 3 requests to a different server, receives a number and aggregates the results:

<https://github.com/tamarstern/requestAggregatorNoPromises>

The aggregation is done with async parallel.

Pay attention that 2 tests fail –

* Both of the tests that fail mock an error received from the url we are sending requests to. Async parallel, while receiving an error, stops the execution of the tasks immediately. In that case, the server will stop to aggregate the results and will return the aggregated number he had collected so far.

Your Task is:

1. Re-write the async parallel to use promise chaining, make sure all tests pass as your success criteria. The server should behave as follows :
   1. send the requests one after another
   2. aggregate the results
   3. Handle errors – if one of the requests fail – continue aggregate the results from the other requests.
2. Re-write the promise chaining to use promise.all(). Make sure all the tests pass.
   1. Pay attention – if there is an error in resolving one of the promises, promise.all execution will stop and the promise that it returns will be rejected. Handle this issue – meaning that
      1. Use promise.all for parallel execution
      2. Make sure exception is handled and you continue to collect the results.