Questions & Answers

Q. What do you think $PFC.props$ is doing?

Answer: Creating an Object with reference parameters and map the keys. In other word, pass functions that should be executed and define dependencies that need to be resolved before execution.

Q. What is the concurrency variable good for?

Answer: Concurrency variable good for,

- To limit the number of functions/promise tasks that should be executed at the same time, which mean can limit the number of Promise created.

Therefore, if we set value of concurrency variable to 1, then processing only one function/promise task at the same time, i.e. only one promise is created and runs in a single moment.

However, Concurrency variable helps:

- To achieve pre-emptive multithreading, means that two computations can both make progress and advance regardless of the other.
- Also, make program more usable and conveniently used by adjusting the number of concurrency.
- If there are two threads, for example, then both make progress independently. The second computation doesn't need to wait for the first one to complete before it can be advanced.

Q. $sendOutStatusEmails$ ignores errors (except for logging them). Let's say your task was to make sure $updateExceededDaysOfOrganizations\,fails$ as soon sending out any email fails. How would you adjust the code?

Answer: As written in ZenkitCodeSamples.js, I will adjust the code. Refer the code from line number 59 to 77. By adding arrow function which throw an Error says that "updateExceededDaysOfOrganizations failed", which make sure that updateExceedDaysOfOrganizations fails means whole function fails as soon sending out any email failed.

Q. Do you think synchronizing the different resources could be done in parallel? If so, how would you do it and would there be a downside?

Answer: Yes, synchronizing the different resources could be done in parallel. By running two or more computations at the same time which known as MIMD (Multiple Instruction Multiple Data). That means could be used all function in $Promise.all$. But there would be a downside is:

- It's not possible with single-core CPU; instead of it required multiple-core architecture.
- And do not grant the chains of functions.

Q. Why use $\_.get(err, ['data', 'error', 'code'])$ if you could simply do $err.data.error.code$? And can you think of a better name for err here?

Answer: To gets the value at path of object. If the resolved value is undefined, then the default Value is returned in its place. As $\_.get$ () contains three arguments:

1. object($Object$): The object to query.
2. path($Array \mid string$): The path of the property to get.
3. [$defaultValue$] ($*$): The value returned for **undefined** resolved values and Returns
4. ($*$): Returns the resolved value.

Therefore, we cannot simply do $err.data.error.code$ because; the path Object is an Array. And In order to access it by referring to an index number. Due to that, it simply returns undefined, instead of throwing an error. That's why, we use $\_.get()$.

Challenge

Yes, I think of a better name for err here. Could be $e, E, \sim e!, Err!, e^{*}$, object or E^{*} and could be name, which referred others errors by its action, name and behavior or by took its responsibility.

Q. When you call $syncList(\{\ listId\colon 1\ \}).then(console.log)$, what will be logged?

Answer: It will be logged the first 1st lists.

Q. Imagine we added the $created\_at$ and $updated\_at$ columns for "$categorySortOrders$" as the TODO suggests. Please adjust the code in a way that prevents synchronizing this resource if there was no change.

Answer: As written in ZenkitCodeSamples.js, I will adjust the code. Refer the code from line number 129 to 140.