**Modern Asynchronous Programming: Promise Chaining**

**Instructions**

To complete this Practice problem, you will need to refactor existing code so that promises are returned and are not deeply nested.

**Starter code**

To start, make sure you've forked and cloned the constellations server, linked below. Follow the instructions in that repository to get the project up and running.

* [GitHub: Constellations Server](https://github.com/Thinkful-Ed/starter-constellations-server)

**updateIfExists()**

The updateIfExists() function takes an id and a body, before doing the following:

* Make a request to the API to ensure that the specified constellation exists.
* If it does not exist, the error message is logged.
* If it does exist, make a PUT request to update that constellation. Then, the response is logged.

You need to update the code so that it does the following:

1. Instead of logging the response and the error, return each.
2. "Flatten" the code so that there are no calls to .then() inside of other .then() calls.

**Tips**

* You may complete this challenge on your own machine before uploading it to Qualified.
* Reference the related checkpoint for help on completing this Practice problem.
* If you need help, contact your mentor or speak with your peers in Slack.