**Design Choices**

**Reminder:** You have limited understanding of http methods and Kotlin. Your preconceived notions may not be correct. Do some digging.

**My thoughts:**

**1.**

/purchase

POST is meant for creation but if there are repeating customers, their information should be updated not recreated.

PUT is meant for update

**2.**

Null safety. Very interesting feature of Kotlin to force better coding habits. Not sure if my current workaround is considered clean.

**3.**

Program ignores cents.

Ex: Customer purchasing of $200.50 then another $200.50 will yield reward total points of 400 not 401.

Personally I think including the cents would be a good idea but there may be retail reasons for doing so that I am just not aware of. I’ll accept the previous programmer’s decision to round down the values and leave it that way.

**4.**

Endpoint 3

The instruction doesn’t specify to include the customers’ email address along with their rewards but I believe it would be beneficial to include the email address in the return so they can see which customer corresponds to which rewards. I may be creating unnecessary additional work here but I currently can’t see the value of simply returning all customers’ rewards info without knowing which reward belongs to who.

**Research:**

**1.**

The link below suggests I am wrong and both can be used to create/update with the main difference being idempotent. In that case, the cleaner approach would be to stick with one “Post /purchase” function and simply add additional functionality within its’ implementation to allow for both creation/update of customer info.

<https://spring.io/understanding/REST>

**Retrospect:**

Learned a great deal about Kotlin and refreshed my knowledge on HTTP. Definitely a great deal left for me to learn but really liking.