

# BONUS CHECKS

## Спецификация

**Проверяем корректность конвертации суммы в баллы.** Используем техники анализа граничных значений и эквивалентных разбиений. На основании спецификации, получаются следующие группы значений:

- -1
- 999.99
- 1000
- 100000
- 149000.99
- 150000
- 190000
- 199000.99
- 200000
- 300000
- 399000.99
- 400000

☐ **AP-0.** Отпр запрос на `api/customer/001/purchase` с `-1` в качестве `amount`. ОР – `response.code: 400, response.error: 'Amount cannot be negative'`.

☐ **AP-1.** Отпр запрос на `api/customer/001/purchase` с `999.99` в качестве `amount`. ОР – `response.code: 200, response.points: 0`.

- ☐ **AP-2.** Отпр запрос на `api/customer/002/purchase` с `1000` в качестве `amount`. ОР – `response.code: 200, response.points: 1`.
- ☐ **AP-3.** Отпр запрос на `api/customer/003/purchase` с `100000` в качестве `amount`. ОР – `response.code: 200, response.points: 100`.
- ☐ **AP-4.** Отпр запрос на `api/customer/004/purchase` с `149000.99` в качестве `amount`. ОР – `response.code: 200, response.points: 149`.
- ☐ **AP-5.** Отпр запрос на `api/customer/005/purchase` с `150000` в качестве `amount`. ОР – `response.code: 200, response.points: 150`.
- ☐ **AP-6.** Отпр запрос на `api/customer/006/purchase` с `190000` в качестве `amount`. ОР – `response.code: 200, response.points: 190`.
- ☐ **AP-7.** Отпр запрос на `api/customer/007/purchase` с `199000.99` в качестве `amount`. ОР – `response.code: 200, response.points: 199`.
- ☐ **AP-8.** Отпр запрос на `api/customer/008/purchase` с `200000` в качестве `amount`. ОР – `response.code: 200, response.points: 200`.
- ☐ **AP-9.** Отпр запрос на `api/customer/009/purchase` с `300000` в качестве `amount`. ОР – `response.code: 200, response.points: 300`.
- ☐ **AP-10.** Отпр запрос на `api/customer/010/purchase` с `399000.99` в качестве `amount`. ОР – `response.code: 200, response.points: 399`.
- ☐ **AP-11.** Отпр запрос на `api/customer/011/purchase` с `400000` в качестве `amount`. ОР – `response.code: 200, response.points: 400`.
- ☐ **AP-12.** Отпр запрос на `api/customer/011/purchase` с `1000` в качестве `amount`. ОР – `response.code: 200, response.points: 401`.

**Проверяем, что баллы начисляются.**

- ☐ **GP-0.** Отпр запрос на `api/customer/2024/points` (юзера с таким ID не существует в Базе). ОР – `response.code: 404, response.error: 'There is no user with specified ID'`.
- ☐ **GP-1.** Отпр запрос на `api/customer/001/points`. ОР – `response.code: 200, response.points: 0`.

- ☐ **GP-2.** Отпр запрос на `api/customer/002/points`. OP – `response.code: 200, response.points: 1`.
- ☐ **GP-3.** Отпр запрос на `api/customer/003/points`. OP – `response.code: 200, response.points: 100`.
- ☐ **GP-4.** Отпр запрос на `api/customer/004/points`. OP – `response.code: 200, response.points: 149`.
- ☐ **GP-5.** Отпр запрос на `api/customer/005/points`. OP – `response.code: 200, response.points: 100`.
- ☐ **GP-6.** Отпр запрос на `api/customer/006/points`. OP – `response.code: 200, response.points: 190`.
- ☐ **GP-7.** Отпр запрос на `api/customer/007/points`. OP – `response.code: 200, response.points: 199`.
- ☐ **GP-8.** Отпр запрос на `api/customer/008/points`. OP – `response.code: 200, response.points: 200`.
- ☐ **GP-9.** Отпр запрос на `api/customer/009/points`. OP – `response.code: 200, response.points: 300`.
- ☐ **GP-10.** Отпр запрос на `api/customer/010/points`. OP – `response.code: 200, response.points: 399`.
- ☐ **GP-11.** Отпр запрос на `api/customer/011/points`. OP – `response.code: 200, response.points: 401`.

**Проверяем корректность расчета скидки.** Снова воспользуемся техниками анализа граничных значений и эквивалентных разбиений. На основании спецификации, получаются следующие группы значений:

- 0
- 100
- 149
- 150

- 190
- 199
- 200
- 300
- 399
- 400

- ☐ **GD-1.** Отпр запрос на `/api/discount/calculate` с `0` в качестве points. OP – `response.code: 200, response.discount: 1.`
- ☐ **GD-2.** Отпр запрос на `/api/discount/calculate` с `100` в качестве points. OP – `response.code: 200, response.discount: 1.`
- ☐ **GD-3.** Отпр запрос на `/api/discount/calculate` с `149` в качестве points. OP – `response.code: 200, response.discount: 1.`
- ☐ **GD-4.** Отпр запрос на `/api/discount/calculate` с `150` в качестве points. OP – `response.code: 200, response.discount: 5.`
- ☐ **GD-5.** Отпр запрос на `/api/discount/calculate` с `190` в качестве points. OP – `response.code: 200, response.discount: 5.`
- ☐ **GD-6.** Отпр запрос на `/api/discount/calculate` с `199` в качестве points. OP – `response.code: 200, response.discount: 5.`
- ☐ **GD-7.** Отпр запрос на `/api/discount/calculate` с `200` в качестве points. OP – `response.code: 200, response.discount: 7.`
- ☐ **GD-8.** Отпр запрос на `/api/discount/calculate` с `300` в качестве points. OP – `response.code: 200, response.discount: 7.`
- ☐ **GD-9.** Отпр запрос на `/api/discount/calculate` с `399` в качестве points. OP – `response.code: 200, response.discount: 7.`
- ☐ **GD-10.** Отпр запрос на `/api/discount/calculate` с `400` в качестве points. OP – `response.code: 200, response.discount: 10.`



Я намеренно не концентрировался на негативных проверках. Насколько я понял, такой задачи в тестовом не стояло. Написал 2 негативных проверки (AP-0, GP-0) исключительно с целью демонстрации навыков написания негативных проверок при тестировании API и сокращения времени на выполнение тестового.

Такие проверки с перебором параметров имеет смысл автоматизировать при помощи Постмана.