

Тестовое задание. API для динамических правил начисления бонусов

Описание задачи

Вы разрабатываете сервис для начисления бонусных баллов пользователям за различные действия в системе.

Правила:

- **За каждые потраченные \$10** начисляется 1 бонус.
- **В выходные и праздничные дни** начисляется x2 бонусов.
- **VIP-клиенты** получают +40% к любым начислениям.

Важные нюансы:

- Все правила начисления должны быть легко изменяемыми и должно быть легко добавить новые правила — завтра могут появиться новые механики, поэтому их нельзя жестко прописывать в коде. Все правила должны храниться либо **в конфигурации**, либо **в базе данных**.
 - То есть в конфиге или БД должны быть все цифры: сколько начислять, когда, при каких условиях.
- Правила должны комбинироваться.
 - Базовое правило всегда применяется первым (например, 1 бонус за каждые \$10).
 - Далее к результату последовательно применяются все дополнительные правила.
 - Порядок применения дополнительных правил должны быть настраиваемыми — должно быть возможно менять последовательность расчета без изменения кода.
- Пример расчета:
 - Базовое правило: $\$150 / \$10 = 15$ бонусов
 - Выходной день (x2): $15 \times 2 = 30$ бонусов
 - VIP-бонус (+40%): $30 \times 1.4 = 42$ бонуса
 - Итог: 42 бонуса

Что нужно реализовать?

Всего **один API-метод** (`POST /calculate-bonus`):

- **На входе:**
 - `transaction_amount` (сумма покупки)
 - `timestamp` (дата покупки)
 - `customer_status` (обычный или VIP)
 - **На выходе:**
 - Количество начисленных бонусов
 - Какие правила были применены
-

Пример запроса

```
POST /calculate-bonus
{
  "transaction_amount": 150,
  "timestamp": "2025-03-08T14:30:00Z",
  "customer_status": "vip"
}
```

Пример ответа

```
{
  "total_bonus": 42,
  "applied_rules": [
    {"rule": "base_rate", "bonus": 15},
    {"rule": "holiday_bonus", "bonus": 15},
    {"rule": "vip_boost", "bonus": 12}
  ]
}
```

Дополнительные условия

- Вы **сами** должны продумать, как хранить правила (например, в JSON/YAML-файле или БД).
- API **не должно зависеть** от конкретных правил, а только от их структуры.
- Графический или другой интерфейс для управления правилами в рамках тестового задания **не требуется**. Добавление и редактирование правил можно выполнять вручную.

- В рамках данного задания покрытие тестами не требуется.
- README-файл должен содержать инструкции:
 - Как запустить сервис.
 - Как его настроить.
 - Как использовать.

Остальные детали реализации остаются на усмотрение исполнителя.