Представь, что ты системный аналитик на проекте в Т-банке. В вашем продукте реализован сервис по отправке платежей юридическим лицам. Но периодически возникают ситуации, когда платежи надо приостановить, потому что клиент ненадежен (есть подозрение на мошенничество и нужно время разобраться), либо клиент предоставил не те реквизиты для перечисления и платежи отбиваются банком клиента (нужно время на выяснение верных реквизитов).

Запрос бизнеса следующий:

* нужна возможность заблокировать платежи конкретного клиента;
* нужна возможность разблокировать платежи клиента;
* нужна возможность проверить, заблокирован ли клиент;
* нужна возможность отличать блокировки мошенников от блокировок добропорядочных клиентов.

Тебе надо предложить решение по реализации этого запроса бизнеса:

* Подготовить спецификацию (желательно в формате OpenAPI / Swagger) endpoint’ов для реализации задач выше

**Описание структуры запроса**

**Запросы, связанные с клиентами**

**Привлечение клиентов (GET)**

* ***/клиент*** - все клиенты. Ответ - список клиентов
* ***/client/{id}*** - получение клиента по ID. Ответ - клиент, либо ошибка 404 - клиент не найден
* ***/client/code/{code}*** - получение всех клиентов, заблокированных по корректному коду (для кодов 100 и 200 возвращаются все клиенты, заблокированные по кодам вида 1?? и 2?? соответственно) Ответ - список клиентов, либо ошибка 404 - код нет в таблице

**PS проверка того, заблокирован ли клиент, находится в разделе "получение кодов"**

**Добавление клиентов (PUT)**

* ***/client/add (имя, ИНН)*** - добавление нового клиента в БД. Ответ - добавленный клиент, либо ошибка 409 - клиент с таким ИНН уже есть, в ней возвращается существующий клиент

**Изменение данных о клиентах (POST)**

* ***/client/{id} (имя, ИНН)*** - изменение данных клиента (имя и ИНН). Ответ - измененный клиет, либо ошибки 404 и 409 - (клиент не найден) и (новый ИНН занят) соответственно
* ***/client/block/{id} (код)*** - блокировка клиента по id (с учетом кода блокировки). Ответ - пользователь, которого заблокировали, либо ошибки 404, 406 и 409 - (клиент не найден), (код равен 0) и (кода нет в таблице) соответственно
* ***/client/unblock/{id}*** - разблокировка клиента по его id. Ответ - разблокированный пользователь, либо ошибка 404 - клиент не найден

**Запросы, связанные с платежами**

**Получение денежных средств (GET)**

* ***/pay*** - все платежи. Ответ - список платежей
* ***/pay/{id}*** - получение платежа по ID. Ответ - платеж, либо ошибка 404 - платеж не найден
* ***/pay/code/{code}*** - получение всех платежных поручений, заблокированных по конректному коду (для кодов 100 и 200 возвращаются все платежи, заблокированные по кодам вида 1?? и 2?? соответственно) Ответ - платежные системы, либо ошибка 404 - код нет в таблице
* ***/pay/client/{id}*** - все платежи в зависимости от клиента (получаются по id клиента). Ответ - список платежей, либо ошибка 404 - клиент не найден

**Добавление платежных средств (PUT)**

* ***/pay/add (реквизиты, client\_id)*** - добавление нового платежа в БД. Ответ - добавленный платеж, либо ошибки 404 и 409 - (клиент не найден) и (неверные реквизиты) соответственно

**Изменение данных о платежах (POST)**

* ***/pay/{id} (реквизиты)*** - изменение реквизитов платежа. Ответ - измененный платеж, либо ошибки 404 и 409 - (платеж не найден) и (некорректированные реквизиты) соответственно.

**Запросы, связанные с кодами состояния**

**Получение кодов (GET)**

* ***/code*** — код всех. Ответ - код списка
* ***/code/{code}*** - получение кода по номеру. Ответ - код, либо ошибка 404 - кода нет в таблице
* ***client/check/{id}*** - проверка деталей клиента. Ответ - код состояния данного клиента, либо ошибка 404 - клиент не найден

**Добавление кодов (PUT)**

* ***/code/{code} (описание)*** - добавление нового кода в БД. Ответ - добавленный код, либо ошибка 409 - код уже занят, в ответ она возвращает существующий код с таким номером.

**Изменение данных о кодах (POST)**

* ***/code/{code} (description)*** - изменение описания кода. Ответ - измененный код, либо ошибка 404 - нет кода с таким номером