**Дано**:

1. Бизнес-объект "Клиент".

Атрибуты: ФИО, телефоны (дом, раб., моб.), дата рождения, ДУЛ (номер/серия, кем выдан, когда выдан), ID клиента в мастер-системе, адреса (дом, раб.),

признак согласия на обработку перс. данных, признак согласия на получение рекламы.

2. Мастер-система для ведения клиентов.

Предоставляет API для операций с клиентами:

 - search (Client), где Client - структура вида (имя, фамилия, отчество, тип документа, серия документа, номер документа, дата выдачи документа).

Возвращает карточку клиента.

 - create (карточка клиента) - метод создания клиента. Возвращает id созданного клиента в мастер-системе.

 - retrieve (clientId) - возвращает карточку клиента.

**Необходимо** написать спецификацию микросервиса-адаптера к мастер-системе.

 Спецификация должна содержать:

 1) swagger-контракт;

 2) текстовое описание внутренней логики методов сервиса в отдельном документе в виде пошаговых сценариев.

 API микросервиса должно предоставлять методы:

 1) Получение данных клиента по идентификатору клиента.

 2) Поиск клиентов по клиентским атрибутам

 3) Создание нового клиента.

При поиске клиента на стороне мастер-системы осуществляется поиск по любому параметру и их сочетаниям (фамилия, имя, отчество, дата рождения, серия, номер, тип ДУЛ, телефон... а также по конкретному id). . Предусмотрен поиск по маске ('%иванов%' и т.п.). Это функция мастер-системы. Мы анализируем ее ответы.

Возвращает все найденные значения (все экземпляры, удовлетворяющие критериям поиска).

При создании нового клиента на стороне мастер-системы осуществляются проверки, что такой клиент уже зареристрирован.

2.1. Жесткая проверка: совпали серия, номер, дата, тип ДУЛ. Возвращает ошибку.

2.2. Мягкие проверки. Возвращает предупредительное сообщение с указанием информации о потенциально совпадающих клиентах (напр. ID) и позволяет зарегистрировать клиента, если пользователь подтвердил, что совпадения некритичны.

Например, совпали фио и дата рождения. Возможно, что-то еще, например, уже есть клиент с таким номером телефона.

В общем случае у сервиса проверки должны быть следующими: проверка входных параметров на соответствие протоколу, проверка обязательных параметров, проверка http-статусов, полученных от API мастер-системы (давайте будем считать, что мастер-система также предоставляет нам RESTful API).

Опять же в общем случае:

http status 200: успешное выполнение;

http status 400: бизнес-ошибка. Ответ должен также содержать поля с кодом и описанием ошибки.

http status 500: внутренняя ошибка сервиса;

Нужно, чтобы по ответу было понятно, в чем ошибка (жесткая, мягкая проверка не прошла).

Необходимо протестировать корректность работы сервиса в swagger.

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

# 1. Описание мастер-системы

## 1.1 Используемые сущности

Идентификатор клиента (ClientCardId) – уникальный номер клиента в мастер-системе.

Карточка клиента (ClientCard) – полная информация о клиенте.

Атрибуты

1. Имя (обязательный при создании), строка до 250 символов
2. Фамилия (обязательный при создании), строка до 250 символов
3. Отчество, строка до 250 символов
4. Телефон: Номер, строка до 20 символов, Phone
5. Дата рождения (обязательный при создании), дата в формате DD.MM.YYYY DateBirth
6. ДУЛ (хотя бы 1 обязательный при создании)
   * Тип (значение из списка в пункте 1.1.2)
   * Номер (Строка до 10 символов)
   * серия (Строка до 10 символов)
   * кем выдан (строка до 200 символов)
   * когда выдан (дата в формате DD.MM.YYYY)
7. ID клиента в мастер-системе,
8. Адрес: (строка до 10000 символов) Address
9. Признак согласия на обработку перс. данных (обязательный при создании) PersFlag
10. Признак согласия на получение рекламы AdvFlag

Критерии поиска (ClientCriteria) – ограниченный набор атрибутов, используемый для поиска. Type Object

Атрибуты:

1. Имя Name
2. Фамилия Surname
3. Отчество ParentName
4. Документ, удостоверяющий личность PersonDoc

Тип документа TypeDoc

Серия документа SDoc

Номер документа NumDoc

Кем выдан (строка до 200 символов) Issuer

Дата выдачи документа IssueDate

Документ, удостоверяющий личность PersonDoc

1. Тип документа TypeDoc
2. Серия документа SDoc
3. Номер документа NumDoc
4. Кем выдан (строка до 200 символов) Issuer
5. Дата выдачи документа DateDoc

1.1.2 Типы ДУЛ

1 – Паспорт

2 – Зарубежный паспорт

3 – Паспорт моряка

1.1.3. Response\_on\_error – структура ответа при ошибке

Код ошибки

1.1.4. Response\_on\_success – структура ответа при успешном поиске

CardClient (одна либо много) <ДОБАВИТЬ СТРУКТУРУ ОТВЕТОВ>

## 1.2 Методы API

Мастер-система предоставляет API для операций с клиентами.

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование метода** | **Принимаемые данные** | **Возвращаемое значение** | **Описание** |
| Search (Client)  Get | ClientCriteria | Множество ClientCard либо Null с кодом ошибки | * Возвращает одну или несколько ClientCard по совпадению одного или нескольких критериев поиска (ClientCriteria) * либо возвращает пустое множество с кодом ошибки |
| Create (ClientCard)  Post | ClientCard | id созданного клиента в мастер-системе  (ClientCardId либо Null с кодом ошибки либо ClientCardId с предупредительным сообщением о совпадении параметров) |  |
| Retrieve (ClientCardId)  Get | ClientCardId | возвращает карточку клиента  Одну ClientCard с полным набором полей либо Null с кодом ошибки |  |

### 1.2.1 Описание метода Search(Client)

Возвращает ClientCard.

Возвращает карточки клиента по совпадению всех указанных значений полей.

### 1.2.2 Описание метода Create(ClientCard)

Метод создания клиента

Возвращает ClientCardId созданного клиента. При создании нового клиента осуществляются проверки, что такой клиент уже зарегистрирован.

#### 1.2.2.1 Жесткая проверка

При жесткой проверке проверяется одновременное совпадение следующих полей:

1. Серия
2. Номер
3. Дата
4. Тип дул

Возвращает ошибку «Клиент не создан».

#### 1.2.2.2. Мягкая проверка

Возвращает предупредительное сообщение с указанием информации о потенциально совпадающих клиентах (напр. ID) и позволяет зарегистрировать клиента, если пользователь подтвердил, что совпадения некритичны.

Например, совпали фио и дата рождения. Возможно, что-то еще, например, уже есть клиент с таким номером телефона.

Возвращает соответствующую ошибку «

### 1.2.3 Описание метода Retrieve(clientId)

Возвращает карточку клиента

Возвращает карточку клиента, либо ошибку «Клиент не найден».

# 2 Описание микросервиса-адаптера

## 2.1 Используемые сущности

Микросервис-клиент использует те же сущности, что и мастер-сервис. <????? А Client?>

## 2.2 Методы API

Микросервис-адаптер предоставляет API для операций с клиентами.

|  |  |  |
| --- | --- | --- |
| **Наименование метода** | **Принимаемые данные** | **Возвращаемое значение** |
| Search\_Adapt | ClientCriteria | Множество ClientCard |
| Create\_Adapt | ClientCard | ClientCardId |
| Retrieve\_Adapt | ClientCardId | ClientCard |

### 2.2.1 Получение данных по идентификатору клиента

Происходит вызов метода Retrieve мастер-системы (описано в п.1.2.3)

Пользователю возвращается информация о клиенте ClientCard

### 2.2.2 Поиск клиентов по клиентским атрибутам

Происходит вызов метода Search мастер-системы (описано в п.1.2.1

### 2.2.3 Создание новой карточки клиента

**Необходимо** написать спецификацию микросервиса-адаптера к мастер-системе.

 Спецификация должна содержать:

 1) swagger-контракт;

 2) текстовое описание внутренней логики методов сервиса в отдельном документе в виде пошаговых сценариев.

В общем случае у сервиса проверки должны быть следующими:

1. Проверка входных параметров на соответствие протоколу,
2. Проверка обязательных параметров,
3. Проверка http-статусов, полученных от API мастер-системы (давайте будем считать, что мастер-система также предоставляет нам RESTful API).

В общем случае:

http status 200: успешное выполнение;

http status 400: бизнес-ошибка. Ответ должен также содержать поля с кодом и описанием ошибки.

http status 500: внутренняя ошибка сервиса;

1. Поиск по любому параметру и их сочетаниям (фамилия, имя, отчество, серия, номер, тип ДУЛ, а также по конкретному id).

Возвращает все найденные значения (все экземпляры, удовлетворяющие критериям поиска).