ТЕХНИЧЕСКОЕ ЗАДАНИЕ К КОНКУРСУ НА ТЕМУ: РАЗРАБОТКА СЕРВИСОВ УПРАВЛЕНИЯ ДАННЫМИ, МАСКИРОВАНИЕ И АУДИТ

# Аннотация

В современных условиях ведения бизнеса долгосрочные и доверительные отношения с клиентами являются одним из главных рычагов в конкурентной борьбе и основой многих капиталов финансовых компаний.

Текущий уровень развития ИТ-технологий позволяет перевести большинство коммуникаций с клиентами в цифровые каналы и поддерживать необходимый уровень взаимоотношений с помощью специальных инструментов, например, CRM - Системы управления взаимоотношениями с клиентами.

В контексте задания необходимо смоделировать несколько стандартных операций рабочего процесса взаимодействия с Физическим лицом (далее по тексту - клиентом):

* Поиск клиента;
* Получение карточки клиента;
* Получение последних **N** операций по счету;
* Добавление контактных данных;
* Расчет Уровня клиента.

Задание содержит несколько этапов (уровней), которые можно выполнять последовательно или сразу переходить к реализации более высокого уровня, пропустив что-то из предшествующего уровня. К итоговому зачету принимается максимальное количество баллов, набранных при проверке одного из уровней задания.

Работы оцениваются экспертами с помощью принципов интегральной оценки каждого из решений: качество реализации, используемые технологии, удобство сопровождения.

Используемые в работе структуры данных описаны в Приложении 1.

Проверка задания будет проходит на другом наборе данных такой же структуры.

# Термины, аббревиатуры и определения

Перечисленные ниже термины, аббревиатуры и определения используются только в контексте текущей задачи. Любые упоминания и ссылки должны быть использованы вместе с оговоркой: “используется в контексте решаемой задачи, в рамках соревнования по созданию модели динамической маскировки данных”

| ***Термины*** | ***Описание*** |
| --- | --- |
| CRM | Customer Relationship Management. Система управления взаимоотношениями с клиентами. |
| Мастер-система | Информационная система в банке, отвечающая за хранение тех или иных данных, за их достоверность и сохранность. |
| Маскирование | Частичное сокрытие информации, упрощённый метод маскировки данных, заключающийся в замене символов нулями или астерисками ( **\*** ). |
| Динамическое маскирование | Маскировка в режиме реального времени (маскировка «на лету»), которая происходит в процессе передачи промышленных данных потребителям без промежуточной записи на какие-либо носители информации. Также может применяться вместе с шифрованием данных в режиме реального времени, в частности при использовании шифрования с сохранением формата. |
| Токенизация | Процесс замены конфиденциального элемента данных на неконфиденциальный эквивалент, называемый токеном, который не имеет самостоятельного смысла, значения для внешнего или внутреннего использования. |
| Счет Овердрафт | Расчетный счет в банке, где держатель счета может хранить денежные средства и в то же время производит перерасход денежных средств ниже дебетового баланса в пределах установленного лимита. При таком перерасходе клиенту засчитывается открытие кредита, который он должен погасить в период отсрочки погашения или позже, когда за пользование кредитом будут начисляться процентные платежи. |
| Среднедневной входящий остаток (СВО) по счету | Сумма на конец рабочего дня на счете, взятая в среднем за заданный период дней. |

# Нефункциональные требования

Требуется разработать систему приложений, выполняемых под JRE 11. Именно под этой версией стабильная работа Apache Ignite гарантирована производителем. Необходимый технологический стэк:

* Java 11, или например Kotlin при особом желании.
* Spring Framework 5, Spring Boot 2, запуск приложений на Embedded Tomcat сервере.
* Gradle или Maven для сборки проекта.
* Apache Ignite как встроенный кэш приложения.
* Apache Kafka для обмена сообщениями между приложениями.
* BCrypt алгоритм шифрования пароля.
* YAML как формат файлов данных.
* HTTP Base Authentication.
* Gitlab.

Предлагаем также воспользоваться Open API Specification для генерации REST интерфейсов, сущностей. Расположено в каталоге /resources/oas/ .

Также предоставляем файлы с тестовыми данными, находящиеся в каталоге /resources/datafiles/ .

В корневом каталоге репозитория должен располагаться многомодульный проект, состоящий из приложений в соответствующих каталогах:

* /crm-bh/ содержит приложение **CRM-BH**.
* /dm/ содержит приложение **DM**.
* /dms/ содержит приложение **DMS**.
* /aw/ содержит приложение **AW**.

При сборке проекта должны генерироваться в соответствующем подкаталоге (build/libs/ или target/) исполняемые jar-файлы приложений, с именами:

* crm-bh-1.0.jar для приложения **CRM-BH**.
* dm-1.0.jar для приложения **DM**.
* dms-1.0.jar для приложения **DMS**.
* aw-1.0.jar для приложения **AW**.

Main классом Spring Boot приложения должны быть классы с именами:

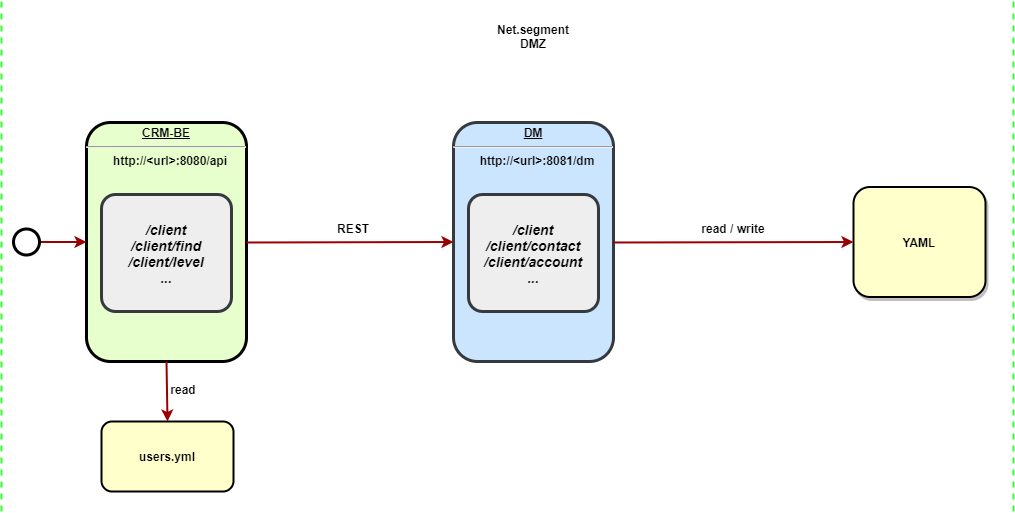
* ru.filit.mdma.crm.Application для приложения **CRM-BH**.
* ru.filit.mdma.dm.Application для приложения **DM**.
* ru.filit.mdma.dms.Application для приложения **DMS**.
* ru.filit.mdma.aw.Application для приложения **AW**.

Файл с параметрами для каждого приложения, application.yml, включающий параметры: порт запуска приложения, context-path приложения, а также параметры, описанные в Приложении 3, располагаться будет в подкаталоге config/ относительно каталога из которого запускается соответствующее приложение.

Файлы с данными следует расположить в подкаталоге datafiles/ относительно каталога, из которого запускается соответствующее приложение (**CRM-BH**, **DM**). Файлу аудита **dm-audit.txt** положено располагаться в подкаталоге auditfiles/ относительно каталога, из которого запускается приложение **AW**.

# [LEVEL-01] Разработка сервисов CRM-Системы

## Рабочий вариант решения



Разработать работающее решение согласно указанной схеме.

## Технические условия

1. Разрабатываемая подсистема должна состоять из двух приложений:
   * **CRM-BH** - CRM Business Hub, приложение, предоставляющее данные *Фронтальной системе*, context path: /api
   * **DM** - Data Manager, представляет доступ к данным Мастер-системы Банка, context path: /dm
2. Все сервисы приложений должны быть без состояния (*Stateless*).
3. При реализации решения использовать данные, предоставленные в файлах формата YAML.
4. В файлах данных значения атрибутов с типом дата (время) хранятся в формате Unix time в секундах. Часовым поясом считаем UTC.
5. Взаимодействия, помеченные меткой “REST”, необходимо реализовать в стиле REST API, протокол взаимодействия **http**, формат объектов JSON, кодировка UTF-8, все запросы осуществляются методом **POST**.
6. Обработка ошибок: соблюдение нотации по использованию HTTP-кодов.
   * Успешное выполнение запроса - код ответа 200.
   * Ошибка валидации параметров запроса - код ответа 400. (Этот же код ответа может быть возвращен при отдельных запросах, когда Клиент, соответствующий запросу, не найден).
   * Ошибка авторизации для приложния **CRM-BH**. В том случае, когда указаны неверные username или password в запросе - код ответа 401.
   * Приложение **CRM-BH** в данный момент не готово обработать запрос, например если приложение **DM** не отвечает - код ответа 503.
   * Прочая ошибка, возникшая при обработке запроса - код ответа 500.
7. Формат тела ответа при возникшей ошибке - стандартный от Spring Boot.
8. Использовать HTTP Basic Authentication - авторизация по логину и паролю - для каждого запроса к **CRM-BH**, создание пользовательских сессий не требуется. Данные о пользователях для авторизации хранятся в файле **users.yml**, вместо пароля пользователя используется его хэш, построенный алгоритмом BCrypt.
9. Адреса (*host*, *port*) всех компонентов (приложений) разрабатываемой подсистемы указываются в переменных окружения java, описанных в Приложении 3.
10. Приложение **CRM-BH** доступно по порту **8080**, приложение **DM** доступно по порту **8081**.
11. Алгоритм вычисления баланса по счету на дату: Значение баланса счета на начало месяца, плюс суммы всех приходных операций по счету, минус суммы всех расходных операций по счету. Операции по счету в этом случае выбираются за период от первого дня месяца, до дня, на который вычисляется баланс.
12. Алгоритм вычисления СВО по счету за 30 дней: Вычисляется значение баланса по счету на каждый из последних 30-ти дней (или меньше, если счет открыт менее 30-ти дней назад), находится среднее значение из полученных сумм баланса. Замечание: первый день из этих 30-и есть текущий день.
13. Алгоритм вычисления уровня клиента: Для всех счетов клиента, активных на текущий день, вычисляется СВО за 30 дней и берется максимальное СВО из них. Уровень клиента может принимать следующие значения в зависимости от величины СВО по счету:
    * до 30000 - “LOW”;
    * от 30000 до 300000 - “MIDDLE”;
    * от 300000 до 1000000 - “SILVER”;
    * от 1000000 - “GOLD”
14. Алгоритм вычисления процентных платежей за пользование кредитом по счету Овердрафт.
    * Параметры вычисления:
      + процент за пользование кредита - 0,07% от суммы задолженности в день (округлять до 2-х знаков после запятой на каждый день).
      + период отсрочки платежа - количество банковских дней (без суббот и воскресений) от дня возникновения задолженности до её погашения, в течении которого не начисляются процентные платежи за пользование кредитом.
    * Учитывая данные - Операции по счету - нужно вычислять на каждый день баланс на счете. При возникновении отрицательного баланса (задолженность) по истечении периода отсрочки платежа (учитывать только банковские дни, без суббот и воскресений) нужно вычислять процентные платежи по сумме просроченной задолженности, исходя из процентной ставки за пользование кредитом. Начисление процентов нужно делать только для банковских дней, без суббот и воскресений.
    * Уточнения: День возникновения задолженности - это первый день отсрочки начисления процентов по кредиту. Процентные платежи начисляются начиная со дня, следующего за последним днем отсрочки. Процентные платежи не входят в сумму задолженности и не вляют на начисления процентов по кредиту в следующих днях.
15. Генерация значений идентификаторов (id) осуществляется в приложении **DM**.
16. Вычисление остатка по счету, среднего остатка по счету, уровня клиента осуществляется в приложении **DM**.

## Функциональные требования

### Разработать следующие REST методы приложения CRM-BH

### FR01. Поиск клиентов

Запрос выполняет поиск клиентов по заданным в запросе параметрам.

***URI сервиса:*** http://<host>:8080/api/client/find

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| lastname | string | Нет |  |
| firstname | string | Нет |  |
| patronymic | string | Нет |  |
| birthDate | string | Нет | Дата в формате YYYY-MM-DD |
| passport | string | Нет | Серия и Номер паспорта, разделены пробелом |
| inn | string | Нет |  |

Значения любых приведенных атрибутов могут быть указаны или не указаны, но хотя бы одно из полей должно быть заполнено. Выполняется поиск клиентов по указанным атрибутам.

*Пример запроса:*

{  
 lastname: null,  
 firstname: null,  
 patronymic: null,  
 birthDate: "1900-04-28",  
 passport: null,  
 inn: null  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов client |
|  | id | string | Да |  |
|  | lastname | string | Да |  |
|  | firstname | string | Да |  |
|  | patronymic | string | Нет |  |
|  | birthDate | string | Да | Дата в формате YYYY-MM-DD |
|  | passportSeries | string | Да |  |
|  | passportNumber | string | Да |  |
|  | inn | string | Да |  |
|  | address | string | Нет |  |

В ответе список объектов сущности client, удовлетворяющих параметрам запроса.

*Пример ответа:*

[  
 {  
 id: "95471",  
 lastname: "Ворошилов",  
 firstname: "Клим",  
 patronymic: "Ефимович",  
 birthDate: "1900-04-28",  
 passportSeries: "6215",  
 passportNumber: "352617",  
 inn: "773626104512"  
 },  
 {  
 id: "258487",  
 lastname: "Торез",  
 firstname: "Морис",  
 birthDate: "1900-04-28",  
 passportSeries: "8403",  
 passportNumber: "116452",  
 inn: "774341922325",  
 address: "Москва, ул. Большая Якиманка, д. 45, кв. 11"  
 }  
]

### FR02. Получение информации о клиенте

Запрос возвращает информацию о клиенте по его id.

***URI сервиса:*** http://<host>:8080/api/client

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да | id клиента |

*Пример запроса:*

{  
 id: "95471"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| id |  | string | Да |  |
| lastname |  | string | Да |  |
| firstname |  | string | Да |  |
| patronymic |  | string | Нет |  |
| birthDate |  | string | Да | Дата в формате YYYY-MM-DD |
| passportSeries |  | string | Да |  |
| passportNumber |  | string | Да |  |
| inn |  | string | Да |  |
| address |  | string | Нет |  |
| contacts |  | array | Да | Список контактов клиента |
|  | id | string | Да |  |
|  | clientId | string | Да |  |
|  | type | string | Да |  |
|  | value | string | Да |  |
|  | shortcut | string | Да | Сокращенное значение контакта |
| accounts |  | array | Да | Список счетов клиента |
|  | number | string | Да |  |
|  | clientId | string | Да |  |
|  | type | string | Да |  |
|  | currency | string | Да |  |
|  | status | string | Да |  |
|  | openDate | string | Да | Дата в формате YYYY-MM-DD |
|  | closeDate | string | Нет | Дата в формате YYYY-MM-DD |
|  | deferment | string | Нет | Целое число |
|  | shortcut | string | Да | Сокращенный номер счета |
|  | balanceAmount | string | Да | Текущий баланс по счету, число с двумя знаками после запятой |

В ответе объект сущности client c расширенным набором атрибутов, включая:

* Список объектов сущности contact, с расширенным атрибутом shortcut (Сокращенное значение контакта, для типа PHONE - последние 4 цифры номера, для типа EMAIL - последняя буква имени учетной записи + “@” + имя домена почты).
* Список объектов сущности account, с расширенным атрибутами: shortcut (Сокращенный номер счета - последние 4 цифры номера), balanceAmount (Сумма текущего баланса на счете).

Если клиент с данным id не найден, то код ответа 400 с сообщением об ошибке в ответе.

*Пример ответа:*

{  
 id: "95471",  
 lastname: "Ворошилов",  
 firstname: "Клим",  
 patronymic: "Ефимович",  
 birthDate: "1900-04-28",  
 passportSeries: "6215",  
 passportNumber: "352617",  
 inn: "773626104512",  
 contacts: [  
 {  
 id: "38523",  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79161234567",  
 shortcut: "4567"  
 },  
 {  
 id: "63781",  
 clientId: "95471",  
 type: "EMAIL",  
 value: "voroshilovke@mail.ru",  
 shortcut: "e@mail.ru"  
 }  
 ],  
 accounts: [  
 {  
 number: "40817810452010063617",  
 clientId: "95471",  
 type: "PAYMENT",  
 currency: "RUR",  
 status: "CLOSED",  
 openDate: "2015-05-15",  
 closeDate: "2020-02-22",  
 shortcut: "3617",  
 balanceAmount: "10.51"  
 },  
 {  
 number: "40817810853110005823",  
 clientId: "95471",  
 type: "OVERDRAFT",  
 currency: "RUR",  
 status: "ACTIVE",  
 openDate: "2018-11-16",  
 deferment: "20",  
 shortcut: "5823",  
 balanceAmount: "150480.20"  
 }  
 ]  
}

### FR03. Получение информации о последних операциях

Запрос возвращает последние **три** операции по счету.

***URI сервиса:*** http://<host>:8080/api/client/account/last-operations

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| accountNumber | string | Да | Номер счета |

*Пример запроса:*

{  
 accountNumber: "40817810853110005823"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов operation |
|  | type | string | Да |  |
|  | accountNumber | string | Да |  |
|  | operDate | string | Да | Дата в формате YYYY-MM-DDThh:mm:ss |
|  | amount | string | Да | Число с двумя знаками после запятой |
|  | description | string | Нет |  |

В ответе массив объектов сущности operation.

*Пример ответа:*

[  
 {  
 type: "RECEIPT",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-10-19T12:36:13",  
 amount: "5000.00",  
 description: "VSP 5311 MOSKVA RUS"  
 },  
 {  
 type: "EXPENSE",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-10-12T18:15:12",  
 amount: "4926.59",  
 description: "YM OZON 1 GOROD MOSKVA RUS"  
 },  
 {  
 type: "EXPENSE",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-09-09T15:40:11",  
 amount: "215.00",  
 description: "МУП Водоканал. Оплата услуг"  
 }  
]

### FR04. Сохранение контакта клиента

Запрос направляет данные контакта клиента на сохранение, атрибуты запроса соответствуют атрибутам сущности contact и обязательны к заполнению, кроме атрибута id, что не указывается для нового контакта. Если в параметрах запроса не указан id контакта, то производится добавление контакта к клиенту. Если в параметрах запроса указан id контакта, то происходит обновление этого контакта клиента.

***URI сервиса:*** http://<host>:8080/api/client/contact/save

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да | id контакта, не заполняется для нового контакта |
| clientId | string | Да |  |
| type | string | Да |  |
| value | string | Да |  |

*Пример запроса:*

{  
 id: null,  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79267152685"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да |  |
| clientId | string | Да |  |
| type | string | Да |  |
| value | string | Да |  |
| shortcut | string | Да | Сокращенное значение контакта |

В ответе объект сущности contact, с расширенным атрибутом shortcut (Сокращенное значение контакта, для типа PHONE - последние 4 цифры номера, для типа EMAIL - последняя буква имени учетной записи + “@” + имя домена почты).

Если клиент с данным clientId не найден, то код ответа 400 с сообщением об ошибке в ответе.

*Пример ответа:*

{  
 id: "63782",  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79267152685",  
 shortcut: "2685"  
}

### FR05. Получение уровня клиента

Запрос возвращает уровень клиента, максимальный по всем его активным счетам.

***URI сервиса:*** http://<host>:8080/api/client/level

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да | id клиента |

*Пример запроса:*

{  
 id: "95471"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| level | string | Да | Значение уровня клиента |
| accountNumber | string | Да | Номер счета |
| avgBalance | string | Да | Средний баланс по счету, число с двумя знаками после запятой |

В ответе составной объект с указанными атрибутами. Атрибуты accuntNumber и avgBalance соответствуют счету, по которому определен уровень клиента.

Если клиент с данным id не найден, то код ответа 400 с сообщением об ошибке в ответе.

*Пример ответа:*

{  
 level: "MIDDLE",  
 accountNumber: "40817810853110005823",  
 avgBalance: "204815.11"  
}

### FR06. Получение суммы процентных платежей по счету Овердрафт

Запрос возвращает начисленную сумму платежей за пользование кредитом по счету Овердрафт за весь период пользования счетом клиента.

***URI сервиса:*** http://<host>:8080/api/client/account/loan-payment

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| accountNumber | string | Да | Номер счета |

*Пример запроса:*

{  
 accountNumber: "40817810853110005823"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| amount | string | Да | Сумма начисленных платежей, число с двумя знаками после запятой |

В ответе составной объект с суммой начисленных платежей.

*Пример ответа:*

{  
 amount: "15417.02"  
}

### Разработать следующие REST методы приложения DM

### FR11. Запрос клиентов

Запрос выполняет поиск клиента по заданным в запросе параметрам.

***URI сервиса:*** http://<host>:8081/dm/client

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Нет | id клиента |
| lastname | string | Нет |  |
| firstname | string | Нет |  |
| patronymic | string | Нет |  |
| birthDate | string | Нет | Дата в формате YYYY-MM-DD |
| passport | string | Нет | Серия и Номер паспорта, разделены пробелом |
| inn | string | Нет |  |

Значения любых приведенных атрибутов могут быть указаны или не указаны, но хотя бы одно из полей должно быть заполнено. Поиск осуществляется по полному совпадению значений атрибута и поля сущности client.

*Пример запроса:*

{  
 id: "95471"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов client |
|  | id | string | Да |  |
|  | lastname | string | Да |  |
|  | firstname | string | Да |  |
|  | patronymic | string | Нет |  |
|  | birthDate | string | Да | Дата в формате YYYY-MM-DD |
|  | passportSeries | string | Да |  |
|  | passportNumber | string | Да |  |
|  | inn | string | Да |  |
|  | address | string | Нет |  |

В ответе массив объектов сущности client, удовлетворяющих параметрам запроса.

*Пример ответа:*

[  
 {  
 id: "95471",  
 lastname: "Ворошилов",  
 firstname: "Клим",  
 patronymic: "Ефимович",  
 birthDate: "1900-04-28",  
 passportSeries: "6215",  
 passportNumber: "352617",  
 inn: "773626104512"  
 }  
]

### FR12. Запрос контактов клиента

Запрос выполняет поиск всех контактов клиента.

***URI сервиса:*** http://<host>:8081/dm/client/contact

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| clientId | string | Да | id клиента |

*Пример запроса:*

{  
 clientId: "95471"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов contact |
|  | id | string | Да |  |
|  | clientId | string | Да |  |
|  | type | string | Да |  |
|  | value | string | Да |  |
|  | shortcut | string | Да | Сокращенное значение контакта |

В ответе массив объектов сущности contact, с расширенным атрибутом shortcut (Сокращенное значение контакта, для типа PHONE - последние 4 цифры номера, для типа EMAIL - последняя буква имени учетной записи + “@” + имя домена почты).

*Пример ответа:*

[  
 {  
 id: "38523",  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79161234567",  
 shortcut: "4567"  
 },  
 {  
 id: "63781",  
 clientId: "95471",  
 type: "EMAIL",  
 value: "voroshilovke@mail.ru",  
 shortcut: "e@mail.ru"  
 }  
]

### FR13. Запрос счетов клиента

Запрос выполняет поиск всех счетов клиента.

***URI сервиса:*** http://<host>:8081/dm/client/account

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| clientId | string | Да | id клиента |

*Пример запроса:*

{  
 clientId: "95471"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов account |
|  | number | string | Да |  |
|  | clientId | string | Да |  |
|  | type | string | Да |  |
|  | currency | string | Да |  |
|  | status | string | Да |  |
|  | openDate | string | Да | Дата в формате YYYY-MM-DD |
|  | closeDate | string | Нет | Дата в формате YYYY-MM-DD |
|  | deferment | string | Нет | Целое число |
|  | shortcut | string | Да | Сокращенный номер счета |

В ответе массив объектов сущности account, с расширенным атрибутом: shortcut (Сокращенный номер счета - последние 4 цифры номера).

*Пример ответа:*

[  
 {  
 number: "40817810452010063617",  
 clientId: "95471",  
 type: "PAYMENT",  
 currency: "RUR",  
 status: "CLOSED",  
 openDate: "2015-05-15",  
 closeDate: "2020-02-22",  
 shortcut: "3617"  
 },  
 {  
 number: "40817810853110005823",  
 clientId: "95471",  
 type: "OVERDRAFT",  
 currency: "RUR",  
 status: "ACTIVE",  
 openDate: "2018-11-16",  
 deferment: "20",  
 shortcut: "5823"  
 }  
]

### FR14. Запрос баланса по счету

Запрос баланс на счете на текущую дату.

***URI сервиса:*** http://<host>:8081/dm/client/account/balance

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| accountNumber | string | Да | Номер счета |

*Пример запроса:*

{  
 accountNumber: "40817810853110005823"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| balanceAmount | string | Да | Текущий баланс по счету, число с двумя знаками после запятой |

В ответе составной объект с cуммой текущего баланса счета.

*Пример ответа:*

{  
 balanceAmount: "150480.20"  
}

### FR15. Запрос операций по счету

Запрос выполняет поиск операций по счету. Сортировка операций - по убыванию даты операции.

***URI сервиса:*** http://<host>:8081/dm/client/account/operation

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| accountNumber | string | Да | Номер счета |
| quantity | string | Да | Максимальное количество операций в ответе, целое число |

*Пример запроса:*

{  
 accountNumber: "40817810853110005823",  
 quantity: 3  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов operation |
|  | type | string | Да |  |
|  | accountNumber | string | Да |  |
|  | operDate | string | Да | Дата в формате YYYY-MM-DDThh:mm:ss |
|  | amount | string | Да | Число с двумя знаками после запятой |
|  | description | string | Нет |  |

В ответе массив объектов сущности operation.

*Пример ответа:*

[  
 {  
 type: "RECEIPT",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-10-19T12:36:13",  
 amount: "5000.00",  
 description: "VSP 5311 MOSKVA RUS"  
 },  
 {  
 type: "EXPENSE",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-10-12T18:15:12",  
 amount: "4926.59",  
 description: "YM OZON 1 GOROD MOSKVA RUS"  
 },  
 {  
 type: "EXPENSE",  
 accountNumber: "40817810853110005823",  
 operDate: "2021-09-09T15:40:11",  
 amount: "215.00",  
 description: "МУП Водоканал. Оплата услуг"  
 }  
]

### FR16. Сохранение контакта клиента

Запрос направляет данные контакта клиента на сохранение, атрибуты запроса соответствуют атрибутам сущности contact и обязательны к заполнению, кроме атрибута id, что не указывается для нового контакта.

***URI сервиса:*** http://<host>:8081/dm/client/contact/save

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да | id контакта, не заполняется для нового контакта |
| clientId | string | Да |  |
| type | string | Да |  |
| value | string | Да |  |

*Пример запроса:*

{  
 id: null,  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79267152685"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да |  |
| clientId | string | Да |  |
| type | string | Да |  |
| value | string | Да |  |
| shortcut | string | Да | Сокращенное значение контакта |

В ответе объект сущности contact, с расширенным атрибутом shortcut (Сокращенное значение контакта, для типа PHONE - последние 4 цифры номера, для типа EMAIL - последняя буква имени учетной записи + “@” + имя домена почты).

Если клиент с данным clientId не найден, или контакт с указанным id не найден (для случая, когда указан не null), то код ответа 400 с сообщением об ошибке в ответе.

*Пример ответа:*

{  
 id: "63782",  
 clientId: "95471",  
 type: "PHONE",  
 value: "+79267152685",  
 shortcut: "2685"  
}

### FR17. Получение уровня клиента

Запрос уровня клиента, рассчитанного как максимальный уровень по всем его активным счетам.

***URI сервиса:*** http://<host>:8081/dm/client/level

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| id | string | Да | id клиента |

*Пример запроса:*

{  
 id: "95471"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| level | string | Да | Значение уровня клиента |
| accountNumber | string | Да | Номер счета |
| avgBalance | string | Да | Средний баланс по счету, число с двумя знаками после запятой |

В ответе составной объект с указанными атрибутами. Атрибуты accuntNumber и avgBalance соответствуют счету, по которому определен уровень клиента.

Если клиент с данным id не найден, то код ответа 400 с сообщением об ошибке в ответе.

*Пример ответа:*

{  
 level: "MIDDLE",  
 accountNumber: "40817810853110005823",  
 avgBalance: "204815.11"  
}

### FR18. Получение суммы процентных платежей по счету Овердрафт

Запрос возвращает начисленную сумму платежей за пользование кредитом по счету Овердрафт за весь период пользования счетом клиента.

***URI сервиса:*** http://<host>:8081/dm/client/account/loan-payment

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| accountNumber | string | Да | Номер счета |

*Пример запроса:*

{  
 accountNumber: "40817810853110005823"  
}

***Параметры ответа:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| amount | string | Да | Сумма начисленных платежей, число с двумя знаками после запятой |

В ответе составной объект с cуммой начисленных платежей.

*Пример ответа:*

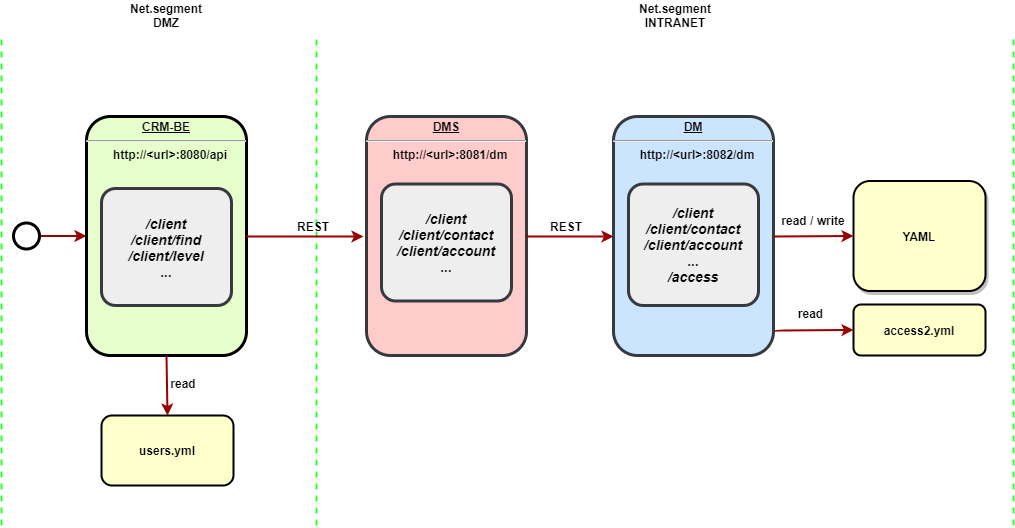
{  
 amount: "15417.02"  
}

## Критерии приемки

1. Работающее решение позволяет получить корректный (верный) ответ на все описанные запросы к приложению **CRM-BH**.

# [LEVEL-02] Маскирование чувствительных данных

## Рабочий вариант решения



Как можно видеть на схеме, компоненты подсистемы эмулируют работу в разных сегментах сети: **CRM-BH** - сеть DMZ (DeMilitary Zone), **DM** - сеть INTRANET. Таким образом, шлюзом между двумя сегментами сети служит приложение **DMS**, что проксирует работу **DM** для **CRM-BH**, добавляя новую функциональность при обработке запросов.

## Технические условия

Технические условия задания [Level-01] выполняются, за исключением переопределенных ниже:

1. Разрабатываемая подсистема должна состоять из трех приложений:
   * **CRM-BH** - CRM Business Hub, приложение, предоставляющее данные *Фронтальной системе*, context path: /api
   * **DM** - Data Manager, представляет доступ к данным Мастер-системы Банка, context path: /dm
   * **DMS** - Data Masking System, приложение реализует шлюз данных, прокладывающий доступ между разными сегментами сети, маскирует чувствительные данные, идущие из внутреннего сегмента сети во внешний, согласно роли пользователя, context path: /dm

…

1. Приложение **CRM-BH** доступно по порту **8080**, приложение **DMS** доступно по порту **8081**, приложение **DM** доступно по порту **8082**.

…

1. Все HTTP запросы, исходящие от **CRM-BH** к **DM** (а по факту к **DMS**) должны сопровождаться HTTP заголовком **CRM-User-Role**, в котором передается роль пользователя CRM системы, авторизованного в запросе к ней.
2. Маскирование данных - замена реальных значений атрибутов объектов в ответе на HTTP запрос (в виде JSON-объекта) на значение “**\*\*\*\***”.
3. План маскирования - набор полей в бизнес-сущностях, подлежащих маскированию. Определяется так: маскированию подлежат все поля бизнес-объектов, на которые отсутствуют права доступа у текущего пользователя. В рамках данного задания следует использовать план маскирования версии **2**.

## Функциональные требования

### FR21. Добавление HTTP заголовка к запросам из приложения CRM-BH

Для всех HTTP запросов приложения **CRM-BH**, направленных к приложению **DM** необходимо добавлять HTTP заголовок “CRM-User-Role”, в котором указывать роль пользователя, авторизовавшегося в **CRM-BH**.

### FR22. Разработать сетевой шлюз Data Masking System

Требуется разработать приложение **DMS**, выступающее шлюзом взаимодействия между **CRM-BH** и **DM**. Приложение реализует интерфейс REST API, полностью повторяющий **DM**, с аналогичными сигнатурами, context path и прочими параметрами. Каждый запрос принимает также роль пользователя, от имени кого сделан запрос в CRM систему. Если данную роль определить не удалось - должен возвращаться HTTP код ответа 403. Все запросы перенаправляются в соответствующий метод **DM**, полученные оттуда ответы отправляются в ответ **CRM-BH**. Приложение **DMS** запускается на порту 8081, таким образом, полностью подменяя **DM** для **CRM-BH**.

### FR23. Запрос прав доступа к полям сущностей для Роли

Запрос к приложению **DM**, возвращает все Права доступа (сущность access) по заданной роли пользователя.

***URI сервиса:*** http://<host>:8082/dm/access

***Параметры запроса:***

| **Поле** | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- |
| role | string | Да | Роль пользователя |
| version | string | Да | Версия полномочий |

Параметр version может принимать значения “2” и “3”, что регулирует из какого файла следует считать права доступа - **access2.yml** и **access3.yml** соответственно.

*Пример запроса:*

{  
 role: "MANAGER",  
 version: "2"  
}

***Параметры ответа:***

| **Поле** |  | **Тип** | **Обяз.** | **Примечание** |
| --- | --- | --- | --- | --- |
| <root> |  | array | Да | Список объектов access |
|  | entity | string | Да |  |
|  | property | string | Да |  |

В ответе массив объектов сущности access.

*Пример ответа:*

[  
 {  
 entity: "client",  
 property: "lastname"  
 },  
 {  
 entity: "client",  
 property: "firstname"  
 },  
 {  
 entity: "client",  
 property: "patronymic"  
 }  
]

### FR24. Разработать механизм маскирования данных

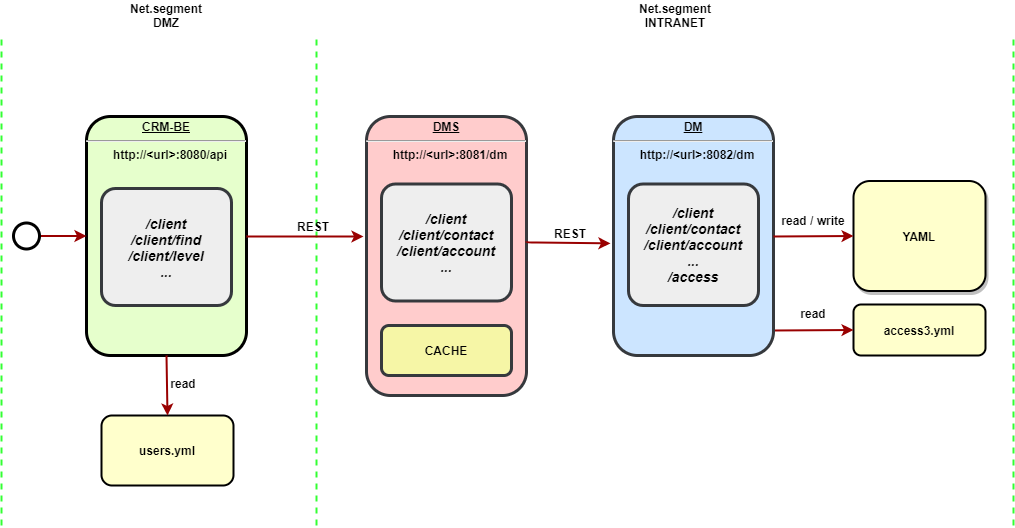
В приложении **DMS** со всеми ответами, полученными от **DM**, в рамках запроса от **CRM-BH**, нужно произвести маскирование данных в атрибутах ответа согласно плану маскирования по роли пользователя. Поля объектов, не подлежащие маскированию (согласно роли пользователя), содержатся в файле данных **access2.yml**, а также эти правила описаны в Приложении 2. Маскировать следует все поля в ответах, которые не числятся в списке видимых для данного пользователя. Сведения о правах доступа получаются в **DMS** вызовом сервиса ‘/dm/access’ из **DM**. При этом в параметре version передается значение “2”.

## Критерии приемки

1. Работающее решение позволяет получить корректный (верный) ответ на все описанные запросы к приложению **CRM-BH**.
2. Данные, получаемые в ответах, замаскированы согласно правам доступа пользователя.

# [LEVEL-03] Токенизация чувствительных данных

## Рабочий вариант решения



## Технические условия

Технические условия задания [Level-02] выполняются, за исключением переопределенных ниже:

…

1. Сервисы приложений **CRM-BH**, **DM** должны быть без состояния (*Stateless*). Приложение **DMS** должно содержать внутренний кэш данных, храня в нем замаскированные данные, а также любые другие данные, какие потребуется.

…

1. Маскирование данных - замена реальных значений атрибутов объектов в ответе на HTTP запрос (в виде JSON-объекта) на значение токена. Иными словами, этот процесс носит название Токенизация. Токен являет собой GUID без символов-разделителей (только буквы и цифры), обрамленный с обеих сторон символом “#”. Пример токена: #802FF7AF4DE340ED924F3415E8071CA0# (длина 34 символа). При этом маскируемые данные помещаются в кэш с ключом - токен. Время жизни токена в кэше нужно ограничить 30 минутами.
2. План маскирования - набор полей в бизнес-сущностях, подлежащих маскированию. Определяется так: маскированию подлежат все поля бизнес-объектов, на которые отсутствуют права доступа у текущего пользователя. В рамках данного задания следует использовать план маскирования версии **3**.
3. Демаскирование данных - поиск в данных значений токенов, указывающих на ранее маскированные данные и подмена их этими данными, выбранными из кэша по токену.
4. В качестве кэша в приложении **DMS** следует использовать Apache Ignite.

## Функциональные требования

### FR31. Встраивание кэша данных

В приложение **DMS** нужно внедрить кэш данных (In Memory Data Grid). В качестве реализации кэша нужно использовать Apache Ignite.

### FR32. Разработать механизм токенизации данных

В приложении **DMS** со всеми ответами, полученными от **DM**, в рамках запроса от **CRM-BH**, нужно произвести маскирование (токенизацию) данных в атрибутах ответа согласно плану маскирования по роли пользователя. Поля объектов, не подлежащие маскированию (согласно роли пользователя), содержатся в файле данных **access3.yml**, а также эти правила описаны в Приложении 2. Сведения о правах доступа получаются в **DMS** вызовом сервиса /dm/access из **DM**. При этом в параметре version передается значение “3”. Токенизированные таким образом данные сохраняются в кэш с ключом - токен.

### FR33. Разработать механизм демаскирования данных

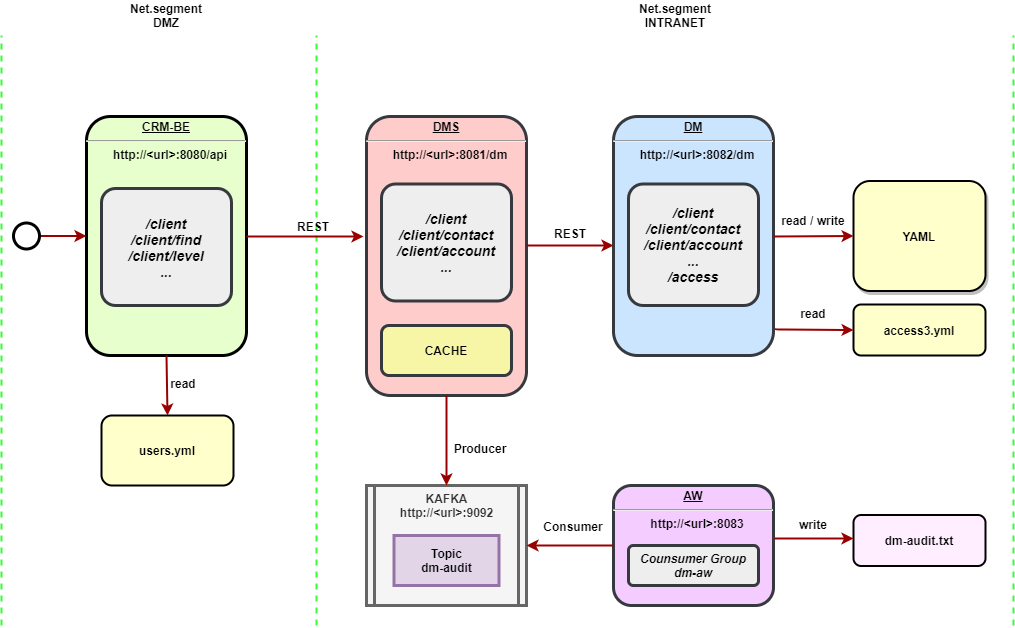
В приложении **DMS** все запросы, поступающие в REST сервисы, следует проверять на наличие в полях тела запроса токена, в формате, описанном в Технических условиях. По этим токенам нужно выбирать данные в кэше и подставлять эти данные в тело запроса. Уже измененное тело запроса перенаправлять в соответствующий сервис **DM**.

## Критерии приемки

1. Работающее решение позволяет получить корректный (верный) ответ на все описанные запросы к приложению **CRM-BH**.
2. Данные, получаемые в ответах, замаскированы токенами согласно правам доступа пользователя.

# [LEVEL-04] Разработка системы Аудита

## Рабочий вариант решения



## Технические условия

Технические условия задания [Level-03] выполняются, за исключением переопределенных ниже:

1. Разрабатываемая подсистема должна состоять из четырех приложений:
   * **CRM-BH** - CRM Business Hub, приложение, предоставляющее данные *Фронтальной системе*, context path: /api
   * **DM** - Data Manager, представляет доступ к данным Мастер-системы Банка, context path: /dm
   * **DMS** - Data Masking System, приложение реализует шлюз данных, прокладывающий доступ между разными сегментами сети, маскирует чувствительные данные, идущие из внутреннего сегмента сети во внешний, согласно роли пользователя, context path: /dm
   * **AW** - Audit Writer, производит запись событий аудита в файл (dm-audit.txt), выбирая их из Topic-а брокера сообщений.

…

1. Все HTTP запросы, исходящие от **CRM-BH** к **DM** (а по факту к **DMS**) должны сопровождаться HTTP заголовком **CRM-User-Role**, в котором передается роль пользователя CRM системы, авторизованного в запросе к ней. Также эти запросы должны сопровождаться HTTP заголовком **CRM-User-Name**, в котором передается имя авторизованного пользователя CRM системы.

…

1. Приложение **DMS** должно вести аудит всех запросов, поступающих к нему в рамках сервиса /dm/. Запись аудита являет собой JSON объект, содержащий:
   * Время запроса, с точностью до секунд (текущее время на сервере);
   * Уникальный идентификатор запроса, генерируемый при поступлении этого запроса (8-и значное число);
   * Имя пользователя, сделавшего запрос;
   * Полное имя вызываемого метода;
   * Тело запроса после демаскирования;
   * Отметка, указывающая, что запись аудита относится к запросу.
2. Приложение **DMS** должно вести аудит всех ответов на поступающие запросы в рамках сервиса /dm/. Запись аудита являет собой JSON объект, содержащий:
   * Время поступления ответа, с точностью до секунд (текущее время на сервере);
   * Уникальный идентификатор запроса;
   * Имя пользователя, под кем сделан запрос;
   * Полное имя вызываемого метода;
   * Тело ответа, не маскированное;
   * Отметка, указывающая, что запись аудита относится к ответу.
3. Все записи аудита в приложении **DMS** должны направляться в брокер сообщений Apache Kafka, в topic с именем dm-audit. Тела запросов и ответов в записи аудита должны маскироваться по плану маскирования для служебной роли **AUDITOR**.

## Функциональные требования

### FR41. Выполнить настройку брокера сообщений Kafka

Необходимо:

* Установить брокер сообщений Apache Kafka и требуемое окружение.
* Создать и настроить Topic с именем dm-audit.
* Указать необходимые реквизиты адреса Kafka-и в параметрах приложения.

Результат данного Требования не нужно как-либо отражать в репозитории с выполненным заданием. На стенде, где будет проходить проверка задания брокер сообщений Apache Kafka, содержащийся в нем нужный Topic будут установлены и настроены.

### FR42. Добавление HTTP заголовка к запросам из приложения CRM-BH

Для всех HTTP запросов приложения **CRM-BH**, направленных к приложению **DM** необходимо добавлять HTTP заголовок “CRM-User-Name”, в котором указывать имя пользователя, авторизовавшегося в **CRM-BH**.

### FR43. Разработать модуль Аудит

В приложении **DMS** для каждого запроса, поступающего в рамках сервиса /dm/ нужно формировать две записи аудита. Первая запись - с телом запроса, вторая запись - с телом ответа, согласно Техническим условиям. Записи аудита (представляющие JSON объект) должны поступать в topic брокера сообщений Kafka - dm-audit. Тела ответов перед попаданием в запись аудита должны быть маскированы по плану маскирования для служебной роли **AUDITOR**. Тела запросов при попадании в запись аудита маскировать не нужно.

### FR44. Разработать приложение Audit Writer

Разработать приложение, которое должено получать сообщения из topic dm-audit по Consumer Group dm-aw. Эти сообщения, содержащие JSON-объект, нужно записывать в файл **dm-audit.txt** (методом append). При записи сообщений в файл для сообщения записи аудита запроса перед полным именем метода нужно ставить символ “>”; для сообщения записи аудита ответа перед полным именем метода нужно ставить символ “<”. Запись сообщения в файле должна располагаться в одной строке и содержать следующую информацию позиционно:

* Дата запроса, позиции с 1 по 19
* Идентификатор запроса, позиции с 21 по 28
* Имя пользователя, позиции с 30 по 44
* Полное имя вызываемого метода, позиции с 46 по 79
* Тело запроса / ответа, позиции с 81

Пример записей аудита в файле приведен ниже.

29.10.2021 18:41:33 85444519 piivanov >/dm/client/account/balance {accountNumber: "40817820460103613657"}  
29.10.2021 18:41:36 85444519 piivanov </dm/client/account/balance {balanceAmount: "#27D088B90C58475599F640891AB46263#"}

## Критерии приемки

1. Работающее решение позволяет получить корректный (верный) ответ на все описанные запросы к приложению **CRM-BH**.
2. Данные, получаемые в ответах, замаскированы токенами согласно правам доступа пользователя.
3. Информация о всех запросах, направленных в **DM**, и ответах записаны в файле **dm-audit.txt** .
4. Записи аудита проходили через соответствующий topic Kafka при фиксации в файл.

# Приложение 1. Структуры данных (сущности)

### Бизнес-сущности, используемые в проекте:

| ***Сущность*** / ***Атрибут*** | ***Тип данных*** | ***Обяз.*** | ***Описание*** |
| --- | --- | --- | --- |
| **client** |  |  | **Клиент банка, Физ. лицо** |
| id | string | Да | Идентификатор клиента |
| lastname | string | Да | Фамилия |
| firstname | string | Да | Имя |
| patronymic | string |  | Отчество |
| birthDate | long | Да | Дата рождения, момент времени (UTC, с) |
| passportSeries | string | Да | Серия паспорта |
| passportNumber | string | Да | Номер паспорта |
| inn | string | Да | ИНН клиента |
| address | string |  | Адрес регистрации |
| **contact** |  |  | **Контактные данные клиента** |
| id | string | Да | Идентификатор контакта |
| clientId | string | Да | Идентификатор клиента |
| type | string | Да | Тип контакта: **PHONE**, **EMAIL** |
| value | string | Да | Значение контакта |
| **account** |  |  | **Банковские счета клиента** |
| number | string | Да | Номер счета |
| clientId | string | Да | Идентификатор клиента |
| type | string | Да | Тип счета: |
|  |  |  | - **PAYMENT** (Расчётный), |
|  |  |  | - **BUDGET** (Бюджетный), |
|  |  |  | - **TRANSIT** (Транзитный), |
|  |  |  | - **OVERDRAFT** (Овердрафт) |
| currency | string | Да | Валюта счета (**RUR**) |
| status | string | Да | Состояние счета: |
|  |  |  | - **INACTIVE** (Неактивный), |
|  |  |  | - **ACTIVE** (Активный), |
|  |  |  | - **LOCKED** (Заблокированный), |
|  |  |  | - **CLOSED** (Закрытый) |
| openDate | long | Да | Дата открытия счета, момент времени (UTC, с) |
| closeDate | long |  | Дата закрытия счета, момент времени (UTC, с) |
| deferment | integer |  | Отсрочка платежа (дни) для счета Овердрафт |
|  |  |  |  |
|  |  |  |  |
| **operation** |  |  | **Банковские операции по счету** |
| type | string | Да | Тип операции: |
|  |  |  | - **RECEIPT** (Приходная), |
|  |  |  | - **EXPENSE** (Расходная) |
| accountNumber | string | Да | Номер счета |
| operDate | long | Да | Дата операции, момент времени (UTC, с) |
| amount | number | Да | Сумма операции |
| description | string |  | Описание операции |
| **accountBalance** |  |  | **Подбитые балансы на счетах на начало месяца** |
| accountNumber | string | Да | Номер счета |
| balanceDate | long | Да | Дата баланса (00:00:00 1-го числа месяца), момент времени (UTC, с) |
| amount | number | Да | Сумма баланса |

### Служебные сущности, используемые в проекте:

| ***Сущность*** / ***Атрибут*** | ***Тип данных*** | ***Обяз.*** | ***Описание*** |
| --- | --- | --- | --- |
| **user** |  |  | **Пользователь системы** |
| username | string | Да | Имя пользователя |
| password | string | Да | Пароль, точнее его хэш-код |
| role | string | Да | Основная роль пользователя: |
|  |  |  | - **MANAGER** (Офис-менеджер банка), |
|  |  |  | - **SUPERVISOR** (Старший смены), |
|  |  |  | - **AUDITOR** (Служебная роль) |
| **access** |  |  | **Права доступа к полям сущностей** |
| role | string | Да | Роль пользователя |
| entity | string | Да | Сущность, к чему дается право доступа |
| property | string | Да | Свойство сущности |

### Файлы данных сущностей

| ***Сущность*** | ***Имя файла*** | ***Где хранится*** |
| --- | --- | --- |
| client | clients.yml | DM |
| contact | contacts.yml | DM |
| account | accounts.yml | DM |
| operation | operations.yml | DM |
| accountBalance | balances.yml | DM |
| user | users.yml | CRM-BH |
| access | access2.yml, access3.yml | DM |

# Приложение 2. Планы маскирования данных

### Атрибуты сущностей, доступные Ролям пользователей

| ***Роль*** | ***Сущность*** | ***Атрибут*** | ***Примечание*** |
| --- | --- | --- | --- |
| **MANAGER** | client | id | Доступен на уровне level-02 |
|  |  | lastname |  |
|  |  | firstname |  |
|  |  | patronymic |  |
|  |  | birthDate |  |
|  | contact | id | Доступен на уровне level-02 |
|  |  | clientId | Доступен на уровне level-02 |
|  |  | type |  |
|  |  | shortcut |  |
|  | account | number | Доступен на уровне level-02 |
|  |  | clientId | Доступен на уровне level-02 |
|  |  | type |  |
|  |  | currency |  |
|  |  | status |  |
|  |  | openDate |  |
|  |  | closeDate |  |
|  |  | shortcut |  |
|  | operation | type |  |
|  |  | accountNumber | Доступен на уровне level-02 |
|  |  | operDate |  |
|  | clientLevel | level |  |
|  | loanPayment | amount |  |
| **SUPERVISOR** | client | id | Доступен на уровне level-02 |
|  |  | lastname |  |
|  |  | firstname |  |
|  |  | patronymic |  |
|  |  | birthDate |  |
|  |  | inn |  |
|  |  | address |  |
|  | contact | id | Доступен на уровне level-02 |
|  |  | clientId | Доступен на уровне level-02 |
|  |  | type |  |
|  |  | value |  |
|  |  | shortcut |  |
|  | account | number |  |
|  |  | clientId | Доступен на уровне level-02 |
|  |  | type |  |
|  |  | currency |  |
|  |  | status |  |
|  |  | openDate |  |
|  |  | closeDate |  |
|  |  | deferment |  |
|  |  | shortcut |  |
|  | operation | type |  |
|  |  | accountNumber |  |
|  |  | operDate |  |
|  |  | amount |  |
|  |  | description |  |
|  | currentBalance | balanceAmount |  |
|  | clientLevel | level |  |
|  |  | accountNumber |  |
|  |  | avgBalance |  |
|  | loanPayment | amount |  |
| **AUDITOR** | client | id |  |
|  |  | lastname |  |
|  |  | passportNumber |  |
|  | contact | id |  |
|  |  | clientId |  |
|  |  | type |  |
|  |  | shortcut |  |
|  | account | number |  |
|  |  | clientId |  |
|  |  | type |  |
|  |  | currency |  |
|  |  | status |  |
|  |  | openDate |  |
|  |  | closeDate |  |
|  |  | shortcut |  |
|  | operation | type |  |
|  |  | accountNumber |  |
|  |  | operDate |  |
|  |  | amount |  |
|  | clientLevel | level |  |
|  |  | accountNumber |  |
|  | loanPayment | amount |  |

# Приложение 3. Параметры приложений подсистемы

### Атрибуты сущностей, доступные Ролям пользователей

| ***Приложение*** | ***Параметр*** | ***Описание*** | ***Примечание*** |
| --- | --- | --- | --- |
| **CRM-BH** | system.element.dm.host | Адрес сервера с приложением **DM** | С level-02 здесь адрес приложения **DMS** |
|  | system.element.dm.port | Порт на сервере приложения **DM** | С level-02 здесь порт приложения **DMS** |
| **DMS** | system.element.dm.host | Адрес сервера с приложением **DM** |  |
|  | system.element.dm.port | Порт на сервере приложения **DM** |  |
|  | system.element.queue.host | Адрес брокера сообщеий Kafka |  |
|  | system.element.queue.port | Порт брокера сообщеий Kafka |  |
| **AW** | system.element.queue.host | Адрес брокера сообщеий Kafka |  |
|  | system.element.queue.port | Порт брокера сообщеий Kafka |  |