1. API методы для работы с заявкой.

* Формат взаимодействия: RESTful API;
* Заголовок: Content-Type: application/json - указывает, что тело запроса/ответа содержит JSON-данные;
* x-client-id: <идентификатор\_клиента> - идентификатор клиента, отправляющего запрос;
* x-request-id: <идентификатор\_запроса> - отслеживание запроса по идентификатору;
* x-application-id: <идентификатор\_приложения> - идентификатор приложения, которое отправляет запрос.

JSON схема draft-4:

{

"type": "object",

"properties": {

"applicationId": {

"type": "string",

"description": "Идентификатор заявки"

},

"externalClientId": {

"type": "string",

"description": "Внешний идентификатор клиента в системе"

},

"status": {

"type": "string",

"description": "Статус заявки"

},

"amount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма по заявке в формате 999,99"

},

"term": {

"type": "string",

"description": "Срок кредита в месяцах"

},

"interestRate": {

"type": "number",

"multipleOf": 0.01,

"description": "Процентная ставка по кредиту"

},

"createDate": {

"type": "string",

"description": "Дата создания заявки в формате DD.MM.YYYY HH:MM:SS"

},

"updateDate": {

"type": "string",

"description": "Дата обновления заявки в формате DD.MM.YYYY HH:MM:SS"

},

"monthlyPayment": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж в формате 999,99"

},

"lastName": {

"type": "string",

"description": "Фамилия клиента"

},

"firstName": {

"type": "string",

"description": "Имя клиента"

},

"middleName": {

"type": "string",

"description": "Отчество клиента"

},

"passportSeries": {

"type": "string",

"description": "Серия паспорта",

"pattern": "[0-9]{4}"

},

"passportNumber": {

"type": "string",

"description": "Номер паспорта",

"pattern": "[0-9]{6}"

},

"insuranceAmount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма страховки в формате 999,99"

},

"monthlyPaymentWithInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж со страховкой в формате 999,99"

},

"monthlyPaymentAfterInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж после вычета страхового платежа в формате 999,99"

},

"externalId": {

"type": "string",

"description": "Дополнительный внешний идентификатор"

},

"bankId": {

"type": "string",

"description": "Идентификатор отделения банка"

}

},

"additionalProperties": true,

"required": [

"passportSeries",

"passportNumber",

"bankId"

]

}

1. Создание заявки:

Метод: POST

URL: /applications/create

Схема сообщения:

POST /applications/create

Content-Type: application/json

x-application-id: <идентификатор\_приложения>

x-client-id: <идентификатор\_клиента>

x-request-id: <идентификатор\_запроса>

{

"type": "object",

"description": "Схема для сообщения по созданию заявки",

"properties": {

"status": {

"type": "string",

"description": "Статус заявки"

},

"amount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма по заявке в формате 999,99"

},

"term": {

"type": "string",

"description": "Срок кредита в месяцах"

},

"interestRate": {

"type": "number",

"multipleOf": 0.01,

"description": "Процентная ставка по кредиту"

},

"createDate": {

"type": "string",

"description": "Дата создания заявки в формате DD.MM.YYYY HH:MM:SS"

},

"updateDate": {

"type": "string",

"description": "Дата обновления заявки в формате DD.MM.YYYY HH:MM:SS"

},

"monthlyPayment": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж в формате 999,99"

},

"lastName": {

"type": "string",

"description": "Фамилия клиента"

},

"firstName": {

"type": "string",

"description": "Имя клиента"

},

"middleName": {

"type": "string",

"description": "Отчество клиента"

},

"passportSeries": {

"type": "string",

"description": "Серия паспорта",

"pattern": "[0-9]{4}"

},

"passportNumber": {

"type": "string",

"description": "Номер паспорта",

"pattern": "[0-9]{6}"

},

"insuranceAmount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма страховки в формате 999,99"

},

"monthlyPaymentWithInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж со страховкой в формате 999,99"

},

"monthlyPaymentAfterInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж после вычета страхового платежа в формате 999,99"

},

"bankId": {

"type": "string",

"description": "Идентификатор отделения банка"

}

},

"additionalPropenties": true,

"required": [

"passportSeries",

"passportNumber",

"bankId"

]

}

Ответ:

201 Created: Заявка успешно создана.

{

"success": "The application has been added"

}

400 Bad Request: Некорректные данные в запросе.

{

"error": "Incorrect data in the request"

}

1. Получение заявки по ID:

Метод: POST

URL: /applications/get

Параметр: applicationId

Схема сообщения:

POST /applications/get

Content-Type: application/json

x-application-id: <идентификатор\_приложения>

x-client-id: <идентификатор\_клиента>

x-request-id: <идентификатор\_запроса>

{

"type": "object",

"description": "Схема для сообщения по получению заявки",

"properties": {

"applicationId": {

"type": "string",

"description": "Идентификатор заявки"

}

},

"additionalPropenties": true,

"required": [

"applicationId"

]

}

Ответ:

200 OK: Заявка найдена, возвращается JSON с данными заявки.

{

"passportSeries": "4500",

"passportNumber": "123456",

"bankId": "some\_bank\_id",

}

404 Not Found: Заявка с указанным ID не найдена.

{

"error": "The application with the specified ID was not found"

}

1. Обновление заявки:

Метод: POST

URL: /applications/update

Параметр: applicationId

Схема сообщения:

POST /applications/update

Content-Type: application/json

x-application-id: <идентификатор\_приложения>

x-client-id: <идентификатор\_клиента>

x-request-id: <идентификатор\_запроса>

{

"type": "object",

"description": "Схема для сообщения по обновлению заявки",

"properties": {

"applicationId": {

"type": "string",

"description": "Идентификатор заявки"

},

"status": {

"type": "string",

"description": "Статус заявки"

},

"amount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма по заявке в формате 999,99"

},

"term": {

"type": "string",

"description": "Срок кредита в месяцах"

},

"interestRate": {

"type": "number",

"multipleOf": 0.01,

"description": "Процентная ставка по кредиту"

},

"createDate": {

"type": "string",

"description": "Дата создания заявки в формате DD.MM.YYYY HH:MM:SS"

},

"updateDate": {

"type": "string",

"description": "Дата обновления заявки в формате DD.MM.YYYY HH:MM:SS"

},

"monthlyPayment": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж в формате 999,99"

},

"lastName": {

"type": "string",

"description": "Фамилия клиента"

},

"firstName": {

"type": "string",

"description": "Имя клиента"

},

"middleName": {

"type": "string",

"description": "Отчество клиента"

},

"passportSeries": {

"type": "string",

"description": "Серия паспорта",

"pattern": "[0-9]{4}"

},

"passportNumber": {

"type": "string",

"description": "Номер паспорта",

"pattern": "[0-9]{6}"

},

"insuranceAmount": {

"type": "number",

"multipleOf": 0.01,

"description": "Сумма страховки в формате 999,99"

},

"monthlyPaymentWithInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж со страховкой в формате 999,99"

},

"monthlyPaymentAfterInsurance": {

"type": "number",

"multipleOf": 0.01,

"description": "Ежемесячный платеж после вычета страхового платежа в формате 999,99"

},

"bankId": {

"type": "string",

"description": "Идентификатор отделения банка"

}

},

"additionalPropenties": true,

"required": [

"applicationId"

]

}

Ответ:

200 OK: Заявка успешно обновлена.

{

"success": "The application has been successfully updated"

}

400 Bad Request: Некорректные данные в запросе.

{

"error": "Incorrect data in the request"

}

404 Not Found: Заявка с указанным ID не найдена.

{

"error": "The application with the specified ID was not found"

}

1. Удаление заявки:

Метод: DELETE

URL: /applications/delete

Параметр: applicationId

Схема сообщения:

DELETE /applications/delete

Content-Type: application/json

x-application-id: <идентификатор\_приложения>

x-client-id: <идентификатор\_клиента>

x-request-id: <идентификатор\_запроса>

{

"type": "object",

"description": "Схема для сообщения по удалению заявки",

"properties": {

"applicationId": {

"type": "string",

"description": "Идентификатор заявки"

}

},

"additionalPropenties": false,

"required": [

"applicationId"

]

}

Ответ:

200 OK: Заявка успешно удалена.

{

"success": "The application has been successfully deleted"

}

404 Not Found: Заявка с указанным ID не найдена.

{

"error": "The application with the specified ID was not found"

}

1. Получение списка заявок:

Метод: POST

URL: /applications/search

POST /applications/search

Content-Type: application/json

x-application-id: <идентификатор\_приложения>

x-client-id: <идентификатор\_клиента>

x-request-id: <идентификатор\_запроса>

{

"type": "object",

"description": "Схема для сообщения по поиску заявки по критериям",

"properties": {

"externalId":{

"type": "string",

"description": "1231231",

"maxLength": "37"

},

"createdDateMin": {

"type":"string",

"description": "Наименьшая дата создания"

},

"createdDateMax": {

"type": "string",

"description": "Наибольшая дата создания"

},

"updatedDateMin"：{

"type": "string",

"description": "Наименьшая дата обновления"

},

"updatedDateMax": {

"type": "string",

"description": "Наибольшая дата обновления"

},

"status": {

"type": "string",

"description": "Статус заявки"

},

"bankId": {

"type": "string",

"description": "Идентификатор отделения банка"

}

},

"additionalPropenties": false,

"required": [

"createdDateMin",

"status"

]

}

Ответ:

200 OK: Возвращается список заявок, удовлетворяющих условиям фильтрации.

[

{

"applicationId": "1234567890",

"externalClientId": "9876543210",

"externalId": "some\_external\_id",

"createDate": "2024-07-08T12:34:56.789Z",

"updateDate": "2024-07-08T13:45:01.234Z",

"status": "APPROVED",

"bankId": "12345"

},

{

"applicationId": "9876543210",

"externalClientId": "1234567890",

"externalId": "another\_external\_id",

"createDate": "2024-07-07T10:11:12.345Z",

"updateDate": "2024-07-08T11:22:33.456Z",

"status": "APPROVED",

"bankId": "54321"

}

]

404 Not Found: Заявки с заданными параметрами не найдены.

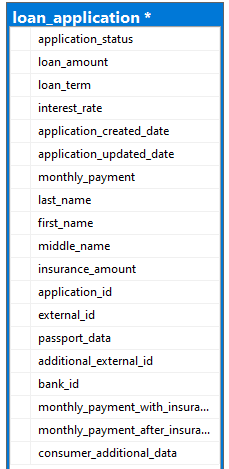
{

"error": "The applications with the specified parameters was not found"

}

1. Таблица БД.

Диаграмма полей:



Модель данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Таблица** | **Атрибут** | **Тип данных** | **Ограничения** | **Ключ** | **Поле для индекса** |
| loan\_application | application\_id | int | not null | PK | 1 |
| external\_id | int |  |  | 3 |
| application\_status | varchar | not null | FK | 2 |
| loan\_amount | varchar |  |  | 4 |
| loan\_term | date |  |  |  |
| interest\_rate | varchar |  |  |  |
| application\_created\_date | datetime | <= NOW()  not null |  | 2 |
| application\_updated\_date | datetime | <= NOW()  not null |  |  |
| monthly\_payment | varchar |  |  |  |
| last\_name | varchar |  |  |  |
| first\_name | varchar |  |  |  |
| middle\_name | varchar |  |  |  |
| passport\_data | varchar | not null |  |  |
| insurance\_amount | varchar |  |  |  |
| monthly\_payment\_with\_insurance | varchar |  |  |  |
| monthly\_payment\_after\_insurance\_deduction | varchar |  |  |  |
| additional\_external\_id | int |  |  | 6 |
| bank\_id | int | not null | FK | 7 |
| consumer\_additional\_data | CLOB |  |  |  |

Маппинг:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Параметр** | **API** | **Тип данных JSON** | **БД** | **Тип данных БД** |
| Идентификатор заявки | applicationId | string | application\_id | int |
| Идентификатор клиента во внешней системе | externalClientId | string | external\_id | int |
| Статус заявки | status | string | application\_status | varchar |
| Сумма по заявке | amount | number | loan\_amount | varchar |
| Срок по заявке | term | string | loan\_term | date |
| Процентная ставка по заявке | interestRate | number | interest\_rate | varchar |
| Дата создания | createDate | string | application\_created\_date | datetime |
| Дата обновления | updateDate | string | application\_updated\_date | datetime |
| Ежемесячный платёж | monthlyPayment | number | monthly\_payment | varchar |
| Фамилия | lastName | string | last\_name | varchar |
| Имя | firstName | string | first\_name | varchar |
| Отчество | middleName | string | middle\_name | varchar |
| Серия паспорта | passportSeries | string | passport\_data | varchar |
| Номер паспорта | passportNumber | string |
| Сумма страховки | insuranceAmount | number | insurance\_amount | varchar |
| Ежемесячный платёж со страховкой | monthlyPaymentWithInsurance | number | monthly\_payment\_with\_insurance | varchar |
| Ежемесячный платёж после вычета страхового платежа | monthlyPaymentAfterInsurance | number | monthly\_payment\_after\_insurance\_deduction | varchar |
| Дополнительный внешний идентификатор | externalId | string | additional\_external\_id | int |
| Идентификатор банковского отделения | bankId | string | bank\_id | int |
| Дополнительные данные | additionalProperties | string | consumer\_additional\_data | clob |

1. Сценарий работы системы.

Таблица ошибок:

|  |  |
| --- | --- |
| **Код ошибки** | **Описание** |
| 200 OK | Успешное выполнение запроса. |
| 400 Bad Request | Некорректные данные в запросе. |
| 409 Conflict | Заявка с таким внешним идентификатором уже существует. |
| 404 Not Found | Заявки с заданными параметры не найдены. |
| 503 Service Unavailable | Сервис временно недоступен. |
| 504 Gateway Timeout | Время ожидания ответа от сервера истекло. |

1. Прием запроса.

В приложение из внешней системы поступает запрос с телом JSON на добавление или манипуляцию с данными о заявках.

1. Проверка запроса.

В приложении осуществляется валидация запроса на соответствие заданному формату, а также на корректность значений полей.

1. Обработка запроса.

Если запрос содержит данные о добавлении заявки, то осуществляется добавление новой записи в БД (см. маппинг).

Если запрос содержит данные об обновлении, то происходит изменение данных соответствующей заявке в БД.

Если запрос содержит параметры поиска, система выполняет поиск в базе данных по заданным критериям.

1. Формирование ответа.

Успешный ответ (200 OK): в случае успешного сохранения/обновления заявки или поиска.

Неуспешный ответ (400 Bad Request): в случае некорректного запроса (неверный формат, невалидные данные).

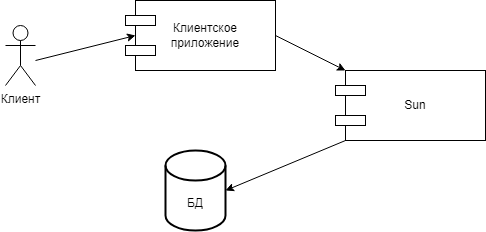
Неуспешный ответ (404 Not Found): в случае, если заявка с заданным ID не найдена.

Неуспешный ответ (409 Conflict): в случае, если заявка с заданным ID уже существует.

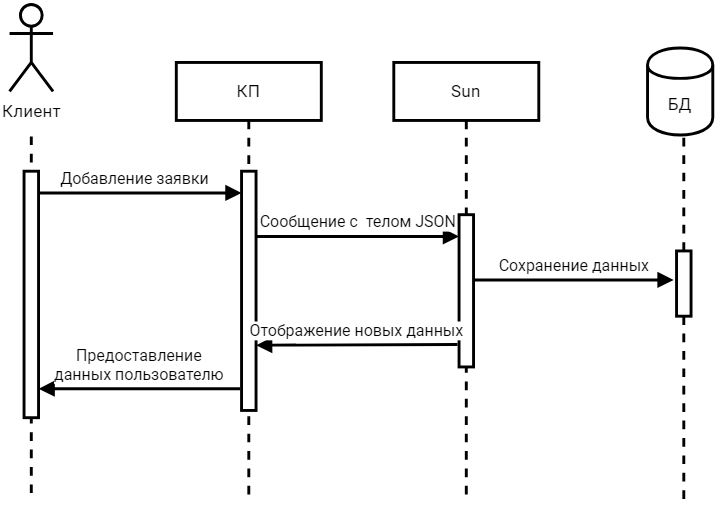
1. Отправка ответа.

Система отправляет соответствующий ответ с телом JSON в результате своей работы.

1. Диаграмма компонентов.



1. Диаграмма последовательности.



1. НФТ.

1. Журналирование:

Система должна отслеживать все действия системы, включая создание, обновление и удаление заявок, а также изменения статусов заявок.

В журнале событий должны быть записаны:

Идентификатор клиента, совершившего действие;

Идентификатор запроса;

Идентификатор приложения, с которого был отправлен запрос;

Дата и время действия;

Тип действия (создание, обновление, удаление);

Измененные данные заявки;

Поля журнала:

|  |  |
| --- | --- |
| **Данные** | **Поле журнала** |
| Идентификатор пользователя, совершившего действие | x-client-id |
| Идентификатор запроса | x-request-id |
| Идентификатор приложения, с которого был отправлен запрос | x-application-id |
| Дата и время действия | dateTime |
| Тип действия (создание, обновление, удаление) | actionType |
| Измененные данные заявки | changedData |

2. Аудит:

Система должна предоставлять возможность для проведения аудита по журналу событий.

Поля аудита:

|  |  |
| --- | --- |
| **Данные** | **Поле аудита** |
| ID заявки | applicationId |
| Фамилия клиента | lastName |
| Имя клиента | firstName |
| Отчество клиента | middleName |
| Серия паспорта | passportSeries |
| Номер паспорта | passportNumber |
| Сумма по кредиту | amount |
| Срок по кредиту | term |
| Процент по кредиту | interestRate |

3. Мониторинг:

Система должна обеспечивать непрерывную работу, своевременное обнаружение и устранение ошибок.

Система мониторинга должна отслеживать следующее:

Доступность системы;

Нагрузка на систему;

Ошибки в работе системы.

При возникновении ошибок или проблем с доступностью системы необходима генерация соответствующих уведомлений.