

Описание API продукта НСЖ

Версия 1.2 от 17.11.2025

Общие принципы работы API

Вызов api осуществляется путем осуществления POST http запроса на необходимый адрес endpointa. Полезная нагрузка запроса передается в теле запроса в формате JSON.

Необходимым условием является наличие http заголовка запроса Content-Type: application/json

В теле http ответа содержится строка JSON с полезной нагрузкой (ответом сервера на запрос).

Авторизация

Ключ API необходимо передавать во всех методах api в http заголовке Authorization: Bearer {ключ_api}

Ключ API для тестовой среды: ede88df2c022e810fedc09d4

Endpoints

Среда	url
Тест	https://demo.avinfors.ru/api-life/api/flow/

Форматы передаваемых данных

- Поля в которых необходимо передавать дату-время должны содержать строку в формате "DD-MM-YYYY hh:mm:ss", например "12.04.2025 02:25:03". Передачу времени можно игнорировать, если необходимо.
- Поля с номером телефона должны иметь следующий формат: "+7(999)999-99-99" или "+79999999999".
- Поля с типом float должны иметь не более 2х знаков после запятой (если иное не указано отдельно)
- Поля с типом bool должны содержать либо true либо false
- Если какое то поле должно содержать значение (структуру данных) из определенного справочника, то достаточно передать только поле code (уникальный код значения справочника).

Например, необходимо передать поле type, формат которого (по модели данных) - значение из определенного справочника:

```
{
  "type":{
    "code":"firstValue",
    "name":"Первое значение справочника",
    "desc":"Описание значения",
    "k":1
  }
}
```

Такое поле можно передать следующими способами:

1) Передать только поле code внутри type

```
{
  "type":{
    "code":"" firstValue
  }
}
```

2) Передать поле type не объектом, а строкой, содержащей уникальный идентификатор значения:

```
{
  " " type ":" firstValue
}
```

При сохранении данных, система сама подставит необходимые поля из найденного по идентификатору значения справочника. Если переданный идентификатор не найден в справочнике, то система вернёт ошибку. В случае получения данных из системы вернется полный объект, как описано в модели данных.

Общий формат запроса к api и ответа от него

Формат JSON, передаваемый в теле запроса, должен иметь следующий вид:

```
{
  "operation": "Код метода",
  "data": {
    // объект с полезной нагрузкой (согласно описанию метода)
  }
}
```

Ответ api будет иметь следующий вид (JSON в теле ответа):

```
{
  "success": true,
  "data": {
    // объект с полезной нагрузкой (согласно описанию метода)
  }
}
```

Ответ api в случае наличия ошибок при выполнении метода:

```
{
  "success": false,
  "errors": [
    { "code": "код ошибки", "text": "Текст ошибки", "path": " путь к данным внутри полезной
нагрузки, содержащим неверные значения" }
  ]
}
```

Ошибок может быть одна или несколько.

Коды ошибок

Общие коды ошибок вне зависимости от операции api

Код ошибки	Описание
dataNotFound	Не передано обязательное поле
invalidDataType	Поле имеет неправильный тип данных
valueOutOfRange	Значение поля выходит за границы корректного диапазона
arrayElementsExceed	Число значений в массиве выходит за границы корректного диапазона
valueLengthExceed	Длина строки не соответствует требованиям
registerValueNotFound	Значение не найдено в справочнике
registerIdNotFound	Не передан идентификатор значения справочника

Пример запроса к api

```
curl --location --request POST 'https://test.somedomain.ru/api/flow/' \
--header 'Authorization: Bearer БАШ_TOKEN_API' \
--header 'Content-Type: application/json' \
--data-raw '{
  "operation": "Contract.LifeEndowment.getProducts",
  "data": {}
}'
```

Список методов API

Код метода	Наименование метода
Contract.LifeEndowment.getProducts	Получение списка продуктов НСЖ

Contract.LifeEndowment.calculate	Расчет страховой премии по продукту
----------------------------------	-------------------------------------

Диаграмма последовательности вызовов API

TODO

1. Договоры страхования НСЖ

1.1. Получение списка продуктов НСЖ

Код метода: **Contract.LifeEndowment.getProducts** Пример

запроса

```
curl --location --request POST 'https://test.somedomain.ru/api/flow/' \
--header 'Authorization: Bearer БАШ_TOKEN_API' \
--header 'Content-Type: application/json' \
--data-raw '{
  "operation": "Contract.LifeEndowment.getProducts",
  "data": {}
}'
```

Ответ:

```
{
  "success": true,
  "data": [
    {
      "code": "sampleProduct",
      "name": "Дожитие на 10 лет"
    }
  ]
}
```

Модель данных ответа:

Поле	Обязательно	Тип данных	Название	Описание	Пример заполнения
success	Да	bool	Признак успешности операции		
data	Да	array	Список продуктов	Массив объектов Количество элементов [0..∞]	
code	Да	string	Код продукта	Буквенно-числовой уникальный код продукта Длина строки [3..64]	sampleProduct
name	Да	string	Наименование продукта	Длина строки [3..256]	Дожитие на 10 лет

1.2. Расчет котировки НСЖ

Код метода: **Contract.LifeEndowment.calculate** Пример

запроса

```
curl --location --request POST 'https://test.somedomain.ru/api/flow/' \
--header 'Authorization: Bearer БАШ_TOKEN_API' \
--header 'Content-Type: application/json' \
--data-raw '{
  "operation": "Contract.LifeEndowment.calculate",
  "data": {
```

```

"beginDate": "10.11.2025 00:00:00",
"insConditions": {
  "program": "base",
  "currency": "RUR",
  "paymentVariant": 0,
  "term": 5
},
"policyHolder": {
  "dob": "01.01.1980",
  "age": 45,
  "sex": "male"
},
"insuredPerson": {
  "dob": "01.01.1981",
  "age": 44,
  "sex": "female",
  "isPolicyHolder": false
},
"calcData": {
  "valuationType": "byPremium",
  "premium": 250000
},
"client": {
  "name": "Иванов Иван Иванович",
  "phone": "+7(121)121-22-12",
  "email": "vkrylov@avinfors.ru"
}
}
}'

```

Модель данных полезной нагрузки запроса (поле data):

Поле	Обязательно	Тип данных	Название	Описание	Пример заполнения
beginDate	Да	datetime	Дата начала страхования		
insConditions	Да	object	Блок условий страхования		
program	Да	string	Уникальный код продукта НСЖ		sample
currency	Да	string	Код валюты договора		RUR
paymentVariant	Нет	integer	Вариант оплаты	Единовременно или рассрочка	0
term	Да	integer	Срок страхования в годах		5
policyHolder	Нет	object	Блок данных Страхователя		
dob	Нет	datetime	Дата рождения Страхователя		
age	Нет	integer	Возраст Страхователя	Если не передана дата рождения, то используется возраст	
sex	Да	string	Пол Страхователя		male

insuredPerson	Да	object	Блок данных Застрахованного		
dob	Нет	datetime	Дата рождения Застрахованного		
age	Нет	integer	Возраст Застрахованного		
sex	Нет	string	Пол Застрахованного		
isPolicyHolder	Да	bool	Является Страхователем	Признак того, что Застрахованный является Страхователем. В случае истины поля dob, age и sex передать не надо	
calcData	Нет	object	Блок исходных данных для расчета		
valuationType	Да	string	Тип расчета	От страхового взноса или от страховой суммы	byPremium
premium	Нет	float	Страховая премия по договору	В случае выбора типа расчета от страховой премии	250000
limit	Нет	float	Страховая сумма по договору	В случае выбора типа расчета от страховой суммы	
client	Нет	object	Блок данных по Клиенту	Контактная информация клиента	
name	Нет	string	ФИО Клиента		
phone	Нет	string	Телефон		
email	Нет	string	Email		

Варианты значений полей

Пол

Код	Наименование
male	Мужской
female	Женский

Валюта

Код	Наименование
RUR	Российский рубль
USD	Доллар США
EUR	Евро

Вариант оплаты

Код	Наименование
0	Единовременно

1	Ежегодно
2	Раз в полгода
4	Ежеквартально
12	Ежемесячно

Тип расчета	
Код	Наименование
byPremium	От страховой премии
byLimit	От страховой суммы

Ответ:

```
{
  "success": true,
  "data": {
    "results": {
      "success": true,
      "term": 5,
      "risks": [
        {
          "code": "endowment",
          "name": "Дожитие",
          "type": "endowment",
          "participant": "insuredPerson",
          "term": 5,
          "tariff": 0.0197,
          "limit": 4980356.9,
          "premium": 980,
          "limitRUR": 4980356.9,
          "premiumRUR": 980
        }, {
          "code": "death",
          "name": "Смерть ЛП",
          "type": "death",
          "participant": "insuredPerson",
          "term": 5,
          "tariff": 0.00197,
          "limit": 49803.57,
          "premium": 1,
          "limitRUR": 49803.57,
          "premiumRUR": 1
        }
      ],
      "premium": 250000,
      "limit": 4980356.92,
      "payTerm": 5,
      "premiumRUR": 250000,
      "paymentsList": [{
        "i": 1,
        "date": 1762732800000,
        "endDate": 1794268799999,
        "premium": 250000,
        "dueDate": 1762732800000
      }]
    },
    "warnings": [],
    "date": 1762791129091
  }
}
```

Модель данных ответа (поле data)

Поле	Обязательно	Тип данных	Название	Описание	Пример заполнения
results	Нет	object	Блок результатов расчета		
success	Да	bool	Признак успешности расчета		
term	Да	integer	Срок страхования		5
risks	Нет	array	Блок страховых рисков по договору		
code	Да	string	Код риска		
name	Да	string	Название риска		
type	Да	string	Тип риска		
participant	Да	string	Участник договора, к которому привязывается риск		
term	Да	integer	Срок действия риска		
tariff	Да	float	Тариф по риску		
limit	Да	float	Страховая сумма по риску	В валюте договора	
premium	Да	float	Страховая премия по риску	В валюте договора	
limitRUR	Да	float	Страховая сумма по риску	В рублях	
premiumRUR	Да	integer	Страховая премия по риску	В рублях	
premium	Да	integer	Общая премия по договору	В валюте договора	
limit	Да	float	Общая страховая сумма по договору	В валюте договора	
premiumRUR	Да	integer	Общая премия по договору	В рублях	
paymentsList	Нет	array	График оплаты договора		
i	Да	integer	Порядковый номер платежа		
date	Да	datetime	Дата платежа		
endDate	Да	datetime	Дата окончания оплаченного периода		

premium	Да	float	Сумма платежа		
dueDate	Да	integer	Оплатить до		
warnings	Да	array	Список возможных ошибок расчета или предупреждений		
date	Нет	datetime	Дата проведения расчета	UNIX timestamp ms	