

Необходимо заполнить.

Наименование	Формат	Длина
url сервиса		
Логин авторизации		
Пароль авторизации		
serviceId	NUMBER	14
field_name		
field_value (not found)		
field_value (found)		
transaction_id (1-method)	NUMBER	11-20
transaction_id (2-method)	NUMBER	11-20
transaction_id (3-method)	NUMBER	11-20
transaction_id (4-method)	NUMBER	11-20
transaction_id (6-method)	NUMBER	11-20
amount	NUMBER	
amount (wrong amount)	NUMBER	
timestamp (7-method)	EEE MMM dd HH:mm:ss z yyyy	
dateFrom	yyyy-mm-dd hh-mm-ss	
dateTo	yyyy-mm-dd hh-mm-ss	
timestamp	yyyy-mm-dd hh-mm-ss	

**'https://endpoint' \**

**--header 'Content-Type: application/json' \**

**--header 'Authorization: \*\*\*\*\*' \**

**1. Метод (GetInformation):** Получение информации:  
НЕ существующий клиент (Номер телефон, Товар, Услуга).  
request:

```
body {
  "id": transaction_id, (1-method)
  "method": "GetInformation",
  "jsonrpc": "2.0",
  "params": {
    "serviceId": serviceId,
    " fields ": {
      "field_name": "field_value"
    }
  }
}
```

Не найден клиент (Номер телефон, Товар, Услуга):

```
response: {
  "jsonrpc": "2.0",
  "error": {
    "code": 302, (301, 304, 305)
    "message": "text"
  }
}
```

**2. Метод (GetInformation):** Получение информации:  
Существующий клиент (Номер телефон, Товар, Услуга).  
request:

```
body {
  "id": transaction_id, (2-method)
  "method": "GetInformation",
  "jsonrpc": "2.0",
  "params": {
    "serviceId": serviceId,
    " fields ": {
      "field_name": "field_value"
    }
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "id": value,
  "result": {
    "status": 0,
    "timestamp": "timestamp",
    "fields": {
      "field_name": "field_value",
      "field_name": "field_value",
      "field_name": "field_value",
      "field_name": "field_value",
      "field_name": "field_value"
    }
  }
}
```

**3. Метод (PerformTransaction): Проведение платежа:**  
НЕ существующий клиент (Номер телефон, Товар, Услуга).  
request:

```
body {
  "id": transaction_id, (3-method)
  "method": "PerformTransaction",
  "jsonrpc": "2.0",
  "params": {
    "amount": value*100,
    "serviceId": serviceId,
    "transactionId": transaction_id,
    "fields": {
      "field_name": "field_value"
    }
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "error": {
    "code": 302, (301, 304, 305)
    "message": "text"
  }
}
```

**4. Метод (PerformTransaction): Проведение платежа:**  
Неверная сумма.  
request:

```
body {
  "id": transaction_id, (6-method)
  "method": "PerformTransaction",
  "jsonrpc": "2.0",
  "params": {
    "amount": amount (wrong amount) *100,
    "serviceId": serviceId,
    "transactionId": transaction_id,
    "fields": {
      "field_name": "field_value"
    }
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "error": {
    "code": 413,
    "message": "Неверная сумма"
  }
}
```

**5. Метод (PerformTransaction): Проведение платежа:**  
Существующий клиент (Номер телефон, Товар, Услуга).  
request:

```
body {  
  "id": transaction_id, (3-method)  
  "method": "PerformTransaction",  
  "jsonrpc": "2.0",  
  "params": {  
    "amount": value*100,  
    "serviceId": serviceId,  
    "transactionId": transaction_id,  
    "fields": {  
      "field_name": "field_value"  
    }  
  }  
}
```

```
response: {  
  "jsonrpc": "2.0",  
  "id": value,  
  "result": {  
    "providerTrnId": value,  
    "timestamp": "timestamp",  
    "fields": {  
      "field_name": "field_value",  
      "field_name": "field_value",  
      "field_name": "field_value",  
      "field_name": "field_value",  
      "field_name": "field_value"  
    }  
  }  
}
```

## 6. Метод (PerformTransaction): Проведение платежа:

Дубликат transaction\_id.

request:

```
body {
  "id": transaction_id, (3-method)
  "method": "PerformTransaction",
  "jsonrpc": "2.0",
  "params": {
    "amount": value*100,
    "serviceId": serviceId,
    "transactionId": transaction_id,
    "fields": {
      "field_name": "field_value"
    }
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "error": {
    "code": 201,
    "message": "Транзакция уже существует"
  }
}
```

## 7. Метод (CheckTransaction) Проверка статуса успешной транзакции:

request:

```
body {
  "jsonrpc": "2.0",
  "method": "CheckTransaction",
  "id": value,
  "params": {
    "serviceId": serviceId,
    "transactionId": transaction_id, (3-method),
    "timestamp": "timestamp"
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "id": value,
  "result": {
    "transactionState": 1,
    "timestamp": "timestamp",
    "providerTrnId": value
  }
}
```

## 8. Метод (GetStatement) Получение списка успешных транзакций:

request:

```
body {
  "jsonrpc": "2.0",
  "method": "GetStatement",
  "id": value,
  "params": {
    "serviceId": serviceId,
    "dateFrom": "dateFrom",
    "dateTo": "dateTo"
  }
}
```

response: {

```
  "jsonrpc": "2.0",
  "id": value,
  "result": {
    "statements": {
      "amount": value*100,
      "providerTrnId": value,
      "transactionId": transaction_id, (3-method),
      "timestamp": "timestamp"
    }
  }
}
```

## 9. Метод (CancelTransaction) Отмена успешной транзакции:

request:

```
body {
  "jsonrpc": "2.0",
  "method": "CancelTransaction",
  "id": value,
  "params": {
    "serviceId": serviceId,
    "transactionId": transaction_id, (3-method),
    "timestamp": " timestamp "
  }
}
```

response: {

```
  "jsonrpc": "2.0",
  "id": value,
  "result": {
    "providerTrnId": value,
    "timestamp": "timestamp",
    "transactionState": 2
  }
}
```

## 10. Метод (CancelTransaction) Отмена, отменённый транзакции:

request:

```
body {
  "jsonrpc": "2.0",
  "method": "CancelTransaction",
  "id": value,
  "params": {
    "serviceId": serviceId,
    "transactionId": transaction_id, (3-method),
    "timestamp": " timestamp "
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "error": {
    "code": 202,
    "message": " Транзакция уже отменена"
  }
}
```

## 11. Метод (CheckTransaction) Проверка статуса, отменённый транзакции:

request:

```
body {
  "jsonrpc": "2.0",
  "method": "CheckTransaction",
  "id": value,
  "params": {
    "serviceId": serviceId,
    "transactionId": transaction_id, (3-method),
    "timestamp": "timestamp"
  }
}
```

```
response: {
  "jsonrpc": "2.0",
  "id": value,
  "result": {
    "transactionState": 2,
    "timestamp": "timestamp",
    "providerTrnId": value
  }
}
```

## 12. Метод (GetStatement) После отмены транзакций:

request:

```
body {  
  "jsonrpc": "2.0",  
  "method": "GetStatement",  
  "id": value,  
  "params": {  
    "serviceId": serviceId,  
    "dateFrom": "dateFrom",  
    "dateTo": "dateTo"  
  }  
}
```

```
response: {  
  "jsonrpc": "2.0",  
  "id": "value",  
  "result": {  
    "statements": []  
  }  
}
```