

Спецификация универсального WEB – сервиса поставщика услуг.

Руководство разработчика.

Версия 1.0

История документа

Дата	Версия	Описание	Автор
01.04.2022	1.0	Первоначальная версия документа.	
20.09.2022	1.1		
28.11.2023	3.3	CheckTransaction (timestamp)	

Содержание

1. Введение.....	4
2. Общее описание.....	4
2.1. Назначение сервиса.....	4
2.2. Требования	5
2.3. Формат запроса	5
2.4. Формат ответа.....	5
2.5. Общие ошибки.....	7
3. Методы	8
3.1. GetInformation.....	8
3.2. PerformTransaction	9
3.3. CheckTransaction.....	10
3.4. CancelTransaction.....	11
3.5. GetStatement.....	12
3.6. ChangePassword	13

1. Введение

Документ «Спецификация универсального WEB – сервиса поставщика услуг» содержит описание взаимодействия биллинга поставщика услуг с информационной системой ООО «UZPAYNET»

Взаимодействие должно осуществляться посредством запросов и ответов по протоколу JSON-RPC 2.0. Запросы должны приниматься поставщиком услуг по протоколу HTTP 1.1 методом POST и должны быть защищены криптографическим протоколом TLS (HTTPS).

2. Общее описание

2.1. Назначение сервиса

WEB - сервис представляет собой механизм взаимодействия между информационной системой поставщика товаров и услуг и платежной системой Paynet. Сервис разворачивается на стороне поставщика и позволяет системе Paynet осуществлять перевод средств за предоставляемые товары и услуги от клиентов поставщика на счет поставщика товаров и услуг. Помимо проведения финансовых операций, WEB - сервис служит для получения сверочной и справочной информации.

Сервис позволяет осуществлять следующие методы:

- PerformTransaction - проведение финансовой операции в счет поставщика услуг.
- CheckTransaction – проверка состояния финансовой операции.
- CancelTransaction - отмена финансовой операции.
- GetStatement - получение сверочной информации по проведенным финансовым операциям за период.
- GetInformation - получение справочной информации.
- ChangePassword – изменение пароля (необязательный метод).

Схема работы: клиент поставщика обращается к агенту Paynet для проведения оплаты за предоставленные товары или услуги. Информационная платежная система Paynet обращается к WEB - сервису поставщика вызывая один из вышеперечисленных методов. Информационная система поставщика обрабатывает поступивший запрос и возвращает ответ системе Paynet.

2.2. Требования

Запросы в web сервис поставщика должны быть в виде HTTP POST запросов, содержащее JSON сообщение с данными транзакции.

Запрос:

- использует для аутентикации HTTP Basic authentication с username и password пользователя (логин и пароль предоставляется со стороны поставщика)
- имеет заголовки Content-Type: application/json и Accept: application/json
- использует кодировку UTF-8.
- использует формат даты YYYY-MM-dd HH:mm:ss

Важно!!! Если в запросе не передан заголовок с данными аутентификации или переданы неверные данные, система должна вернуть HTTP статус 401 – Unauthorized.

2.3. Формат запроса

Запрос — вызов определённого метода, предоставляемого удалённой системой. Запрос должен содержать следующие обязательные элементы:

- jsonrpc — строка, указывающая версию протокола JSON-RPC. Указывать версию "2.0"
- method — строка с именем вызываемого метода.
- params — массив данных, которые должны быть переданы методу, как параметры.
- id — значение любого типа, которое используется для установки соответствия между запросом и ответом.

Сервер должен вернуть правильный ответ на каждый полученный запрос.

Форма RPC запроса:

```
{
    "jsonrpc": "2.0",
    "method": "methodname",
    "id": 123,
    "params": {
        //Body of request //
    }
}
```

2.4. Формат ответа

Формат ответа при успешном выполнении метода должен содержать следующие свойства:

- `jsonrpc` — строка, указывающая версию протокола JSON-RPC. Указывать версию "2.0"
- `result` — данные, которые возвращает метод. Если произошла ошибка во время выполнения метода, то параметр отсутствует в структуре ответа.
- `id` — должен совпадать со значением элемента идентификатора в объекте запрос

Пример ответа при успешном выполнении метода:

```
{
  "jsonrpc": "2.0",
  "id": 123,
  "result": { "client_id": "1463398",
    "fio ": "Иавнов И. И."
  }
}
```

Формат ответа в случае ошибки

Данный тип ответа будет передан в случае получения некорректного запроса или если при обработке уведомления произошла ошибка. Все обязательные поля будут переданы в ответе:

- `jsonrpc` — строка, указывающая версию протокола JSON-RPC. Указывать версию "2.0"
- `error` — код ошибки, если произошла ошибка во время выполнения метода, иначе параметр не указывается в структуре ответа.
- `id` — должен совпадать со значением элемента идентификатора в объекте запрос

Пример ответа при неуспешном выполнении метода:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -253,
    "message": "Error message!"
  }
}
```

Наличие поле error означает что запрос **не выполнен**, и транзакция считается неуспешным. При наличии поле result и отсутствия поля error запрос считается **успешно** выполненным.

2.5. Общие ошибки

-32300	Ошибка возникает в том случае, если метод запроса не POST.
-32700	Ошибка парсинга JSON.
-32600	В RPC-запросе отсутствуют обязательные поля или тип полей не соответствует спецификации.
-32601	Запрашиваемый метод не найден.
-32602	Отсутствуют обязательные поля параметров.
-32603	Системная (внутренняя ошибка). Ошибку следует использовать в случае системных сбоев: отказа базы данных, отказа файловой системы, неопределенного поведения и т.д.
0	Проведено успешно
77	Недостаточно средств на счету клиента для отмены платежа
100	Услуга временно не поддерживается
101	Квота исчерпана
102	Системная ошибка
103	Неизвестная ошибка
113	Кошелёк не идентифицирован
140	The monthly limit is exceeded for this
141	The daily limit is exceeded for this account
201	Транзакция уже существует
202	Транзакция уже отменена
203	Транзакция не найдена
301	Номер не существует
302	Клиент не найден
304	Товар не найден
305	Услуга не найдена
401	Ошибка валидации параметра 1
402	Ошибка валидации параметра 2
403	Ошибка валидации параметра 3
404	Ошибка валидации параметра 4
405	Ошибка валидации параметра 5
406	Ошибка валидации параметра 6
407	Ошибка валидации параметра 7
408	Ошибка валидации параметра 8
409	Ошибка валидации параметра 9
410	Ошибка валидации параметра 10

411	Не заданы один или несколько обязательных параметров
412	Неверный логин или пароль
413	Неверная сумма
414	Неверный формат даты и времени
415	Сумма превышает максимальный лимит
501	Транзакции запрещены для данного плательщика
601	Доступ запрещен
603	Неправильный код команды

3. Методы

3.1. GetInformation

Метод GetInformation используется для получения справочной информации. Объект **params** содержит следующие записи:

Наименование	Обязательность	Тип	Описание
serviceld	✓	Number	Идентификатор сервиса на стороне поставщика
fields	✓	Object	Перечень заполненных полей сервиса

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "GetInformation",
  "id": 12350,
  "params": {
    "serviceld": 1,
    "fields": {
      "client_id": 634247
    }
  }
}
```

Клиент запрашивает информацию по сервису под номером 1, передавая в объекте **fields** ключ **client_id** со значением 634247. Пример: запрос происходит по **id** клиента, и в ответе возвращается имя клиента и его остаток.

Ответ содержит следующие поля:

Наименование	Обязательность	Тип	Описание
status	✓	Number	Состояние запроса
timestamp	✓	String	Дата и время обработки запроса на стороне поставщика (GMT+5)
fields	✓	Object	Перечень дополнительных параметров

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 12350,
  "result": {
    "status": "0",
    "timestamp": "2021-04-30 08:00:00",
    "fields": {
      "balance": 420000,
      "name": "Пушкин А.С."
    }
  }
}
```

3.2. PerformTransaction

Метод PerformTransaction используется для проведения финансовой операции в счет поставщика услуг. Объект params содержит следующие записи:

Наименование	Обязательность	Тип	Описание
amount	✓	Number	Сумма операции в тийинах
serviceld	✓	Number	Идентификатор сервиса на стороне поставщика
transactionId	✓	Number	Идентификатор транзакции PAYNET
fields	✓	Object	Перечень заполненных полей сервиса

Перечень дополнительных параметров представляет собой key – value параметров. Набор этих параметров определяется бизнес – логикой конкретного сервиса поставщика услуг.

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "PerformTransaction",
  "id": 12345,
  "params": {
    "amount": 100000,
    "serviceld": 1,
    "transactionId": 12345678900,
    "fields": {
      "client_id": "634247"
    }
  }
}
```

Ответ содержит следующие поля:

Наименование	Обязательность	Тип	Описание
providerTrnId	✓	Number	Идентификатор транзакции поставщика
fields	✓	Object	Перечень заполненных полей сервиса
timestamp	✓	String	Дата и время обработки запроса на стороне Поставщика (GMT+5)

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": "12345",
  "result": {
    "timestamp": "2021-06-16 12:41:54",
    "providerTrnId": 2323,
    "fields": {
      "client_id": "634247"
    }
  }
}
```

3.3. CheckTransaction

Метод CheckTransaction используется для проверки состояния финансовой операции. Объект params содержит следующие записи:

Наименование	Обязательность	Тип	Описание
serviceld	✓	Number	Идентификатор сервиса на стороне поставщика
transactionId	✓	Number	Идентификатор транзакции PAYNET
timestamp	✓	String	Дата и время обработки запроса на стороне PAYNET (GMT+5)

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "CheckTransaction",
  "id": 12346,
  "params": {
    "serviceld": 1,
    "transactionId": 12345678900,
    "timestamp": "2021-06-16 12:41:54"
  }
}
```

Ответ проверки состояния финансовой операции содержит следующие поля:

Наименование	Обязательность	Тип	Описание
providerTrnId	✓	Number	Идентификатор транзакции поставщика
timestamp	✓	String	Дата и время обработки запроса на стороне Поставщика (GMT+5)
transactionState	✓	Number	Состояние транзакции

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": "12346",
  "result": {
    "transactionState": 1,
    "timestamp": "2021-06-16 12:44:57",
    "providerTrnId": 2323
  }
}
```

3.4. CancelTransaction

Метод CancelTransaction используется для отмены финансовой операции. Объект params содержит следующие записи:

Наименование	Обязательность	Тип	Описание
serviceld	✓	Number	Идентификатор сервиса на стороне поставщика
transactionId	✓	Number	Идентификатор транзакции PAYNET
timestamp	✓	Number	Дата и время обработки запроса на стороне PAYNET (GMT+5)

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "CancelTransaction",
  "id": 12347,
  "params": {
    "serviceld": 1,
    "transactionId": 12345678900,
    "timestamp": "16.06.2021 12:44:57"
  }
}
```

Ответ отмены финансовой операции содержит следующие поля:

Наименование	Обязательность	Тип	Описание
providerTrnId	✓	Number	Идентификатор транзакции поставщика
timestamp	✓	String	Дата и время обработки запроса на стороне Поставщика (GMT+5)
transactionState	✓	Number	Состояние транзакции

Сервер отменяет транзакцию и возвращает в состояние «2»- Отмененная транзакция.

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 12347,
  "result": {
    "providerTrnId": 2323,
    "timestamp": "16.06.2021 12:41:54",
    "transactionState": 2
  }
}
```

3.5. GetStatement

Метод GetStatement используется для получения сверочной информации по проведенным финансовым операциям за период. Объект params содержит следующие записи:

Наименование	Обязательность	Тип	Описание
serviceld	✓	Number	Идентификатор сервиса на стороне поставщика
dateFrom	✓	String	Дата и время начала периода
dateTo	✓	String	Дата и время окончания периода

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "GetStatement",
  "id": 12348,
  "params": {
    "serviceld": 1,
    "dateFrom": "2021-04-20 08:00:00",
    "dateTo": "2021-04-30 08:00:00"
  }
}
```

Объект Statement определяет массив транзакций. Каждая транзакция имеет следующие параметры:

Наименование	Обязательность	Тип	Описание
amount	✓	Number	Сумма финансовой операции в тийинах
transactionId	✓	Number	Идентификатор транзакции PAYNET
providerTrnId	✓	Number	Идентификатор транзакции поставщика
timestamp	✓	String	Дата и время обработки транзакции на стороне Поставщика

При ответе на запрос необходимо учитывать часовой пояс «GMT+5»

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 12348,
  "result": {
    "statements": [
      {
        "amount": 120000,
        "providerTrnId": 23,
        "transactionId": 12345679800,
        "timestamp": "2021-04-23 17:04:22"
      },
      {
        "amount": 780000,
        "providerTrnId": 47,
        "transactionId": 12346578901,
        "timestamp": "2021-04-24 13:25:02"
      }
    ]
  }
}
```

3.6. ChangePassword

Необязательный метод ChangePassword используется для изменения пароля доступа к сервису.

Наименование	Обязательность	Тип	Описание
newPassword	✓	String	Новый пароль

Пример запроса:

```
{
  "jsonrpc": "2.0",
  "method": "ChangePassword",
  "id": 12351,
  "params": {
    "newPassword": "newDifficultPassword"
  }
}
```

Если пароль успешно изменен, то ответ содержит следующие поля:

Наименование	Обязательность	Тип	Описание
result	✓	String	Статус обработки запроса

Пример ответа:

```
{
  "jsonrpc": "2.0",
  "id": 12351,
  "result": "success"
}
```

4. Статусы транзакции (transactionState)

Транзакция может иметь один из следующих статусов:

- 1 – Успешно проведенная транзакция
- 2 – Отмененная транзакция
- 3 – Транзакция не найдена