Тестовый API сервер: https://api.gplatform.org/sol/api

Запросы осуществляются по HTTP(HTTPS) протоколу версии 1.1. Независимо от вызываемого API метода тип HTTP запроса всегда POST. В теле запроса данные передаются в json формате (в кодировке UTF-8).

Заголовок запроса должен содержать:

1. Host - адрес и порт куда идет обращение
2. Content-Type: application/json; charset=utf-8
3. Cache-Control: no-cache
4. Pragma: no-cache
5. Content-Length: длина передаваемых данных

Тело запроса должно содержать:

1. Данные в формате json

Пример данных запроса:

{

"method": "find\_client",

"data": {

"client\_id": "test\_id\_1234",

"is\_create\_if\_not\_found": true

}

}

Пример данных ответа:

{

"@": "57bcfaee-bc78-49bc-b19e-9be9610c78ab",

"result": {

"@": "636f424a-4af6-48df-a0fe-1ea09a7b94e0",

"res": "OK",

"msg": ""

},

"data": {

"@": "6ae53969-df92-47c5-a0b5-9ca4e85e03b3",

"id": "test\_id\_1234",

"create\_ts": 0,

"name": "",

"balance": {

"@": "db877116-caef-42e2-a1f3-81123b72cf2e",

"total": 0,

"unlocked": 0,

"locked": 0

}

}

}

Каждый запрос независимо от вызываемого метода содержит следующий набор полей:

1. **method** - имя вызываемого метода (обязательный)
2. **sid** - идентификатор сессии (только если явно указано что метод требует валидной сессии).
3. **data** - передаваемые данные. Набор конкретных полей зависит от того какой именно метод вызывается.

В ответ может прийти:

1. Ошибка на уровне протокола HTTP (все отличное от кода 200). Здесь могут быть ошибки связанные только с транспортным уровнем HTTP (например ошибка в формате запроса либо ошибки сервера).
2. Успешный ответ на запрос на уровне протокола HTTP (код 200). В этом случае надо проверить код результата на уровне протокола приложения.

Каждый ответ независимо от вызываемого метода содержит следующий набор обязательных

полей:

1. **result** - структура описывающая результат. Может содержать поля:
   1. res - enum кода результата запроса (возможные коды перечислены ниже).
   2. msg - текстовое пояснение к результату запроса (полезно использовать при отладке).
2. **data** - данные ответа. Формат зависит от того какой именно метод вызывается.

Поля "@" могут игнорироваться, это часть системы идентификации типов при сериализации/десериализации.

## Коды результата запроса

1. OK - успешный запрос
2. FAILED - ошибка
3. UNKNOWN\_METHOD - неизвестный метод
4. INVALID\_SID - невалидный идентификатор сессии
5. DB\_ENTITY\_NOT\_FOUND - запрашиваемый элемент не найден

## Методы для эмулятора кассы

### Поиск клиента по id (адрес, публичный ключ)

**Название**: find\_client

**Требует сессию**: нет

**Описание**: поиск клиента по id (адрес, публичный ключ) и если не найден, то регистрация нового.

**Параметры запроса**:

1. client\_id - идентификатор клиента
2. is\_create\_if\_not\_found - флаг, если true то новый клиент будет создан автоматически.

Пример запроса:  
{

"method": "find\_client",

"sid": "",

"ts": 0,

"args": null,

"data": {

"client\_id": "test\_id\_1234",

"is\_create\_if\_not\_found": true

}

}

**Параметры ответа**:

1. id - идентификатор клиента
2. create\_ts - время создания в unix timestamp (секунды)
3. name - имя
4. balance - данные о балансе:
   1. total - всего (unlocked + locked). В копейках
   2. unlocked - можно потратить
   3. locked - нельзя потратить
5. is\_on\_chain - создан ли аккаунт на блокчейне

Пример ответа:

{

"@": "57bcfaee-bc78-49bc-b19e-9be9610c78ab",

"result": {

"@": "636f424a-4af6-48df-a0fe-1ea09a7b94e0",

"res": "OK",

"msg": ""

},

"data": {

"@": "6ae53969-df92-47c5-a0b5-9ca4e85e03b3",

"id": "9u1uhp9M6pttQv6t2YY8WJxyjDdKhM4z4X9UY636xdqu",

"create\_ts": 0,

"name": "",

"balance": {

"@": "db877116-caef-42e2-a1f3-81123b72cf2e",

"total": 0,

"unlocked": 0,

"locked": 0

},

"is\_on\_chain":false

}

}

### Совершение транзакции покупки

**Название**: make\_buy\_tx

**Требует сессию**: нет

**Описание**: создает транзакцию покупки. Если не указан client\_id, то транзакция просто запишется в БД. Если client\_id указан, то ему будут начислены бонусы исходя из правила 5% на каждую товарную позицию. Также если указаны spend и spend\_sign, то возможна оплата бонусами.

**Параметры запроса**:

1. client\_id - идентификатор клиента
2. items - товарные позиции (состав чека)
   1. id - последовательный номер 0, 1, 2….
   2. article - артикул
   3. name - название
   4. price - цена за штуку в копейках
   5. qty - кол-во штук
3. spend - сумма списываемых бонусов
4. spend\_sign - подпись для списания
5. commit - флаг коммита транзакции. Если false, то транзакция будет рассчитана, но в БД запись не произойдет.

Пример запроса:

{

"method": "make\_buy\_tx",

"sid": "",

"ts": 0,

"args": null,

"data": {

"client\_id": "9u1uhp9M6pttQv6t2YY8WJxyjDdKhM4z4X9UY636xdqu",

"items": [

{

"id": 0,

"article": "xxBZWDI4",

"name": "Test",

"price": 4137,

"qty": 5

},

{

"id": 1,

"article": "IgEQ83hH",

"name": "Test",

"price": 6028,

"qty": 4

},

{

"id": 2,

"article": "CkRpAamD",

"name": "Test",

"price": 1956,

"qty": 5

},

{

"id": 3,

"article": "KDF5Lexy",

"name": "Test",

"price": 229,

"qty": 3

}

],

"spend": 0,

"spend\_sign": "",

"new\_account\_sign": "",

"commit":"false"

}

}

**Параметры ответа**:

1. id - идентификатор транзакции (uuid)
2. items - товарные позиции (состав чека)
   1. id - последовательный номер 0, 1, 2….
   2. article - артикул
   3. name - название
   4. price - цена за штуку в копейках
   5. qty - кол-во штук
3. create\_ts - время создания в unix timestamp (секунды)
4. client\_id - идентификатор клиента
5. total - общая сумма чека (в копейках)
6. earned - начислено бонусов (в копейках)
7. spend - списано бонусов (в копейках)
8. spend\_sign - подпись для списания
9. new\_account\_sign - подпись для создания нового аккаунта (требуется если is\_on\_chain == false).
10. spend\_raw\_tx - “сырая” транзакция которую надо подписать (на списание). Поле заполняется если spend > 0 и commit = false
11. new\_account\_raw\_tx - “сырая” транзакция которую надо подписать (на создание нового аккаунта на блокчейне). Поле заполняется если is\_on\_chain = false 0 и commit = false

Пример ответа:

{

"@": "0c0ef560-3650-42d6-9ae3-c97af0bd419d",

"result": {

"@": "636f424a-4af6-48df-a0fe-1ea09a7b94e0",

"res": "OK",

"msg": ""

},

"data": {

"@": "8572b0d0-a63e-459c-b308-c879298a454e",

"id": "d8d30ec3-e31b-5ff0-06c5-3d624be6fc9a",

"items": [

{

"@": "b616772b-113f-4909-a2f6-7552a36504c1",

"id": 0,

"article": "xxBZWDI4",

"name": "Test",

"price": 4137,

"qty": 5

},

{

"@": "b616772b-113f-4909-a2f6-7552a36504c1",

"id": 1,

"article": "IgEQ83hH",

"name": "Test",

"price": 6028,

"qty": 4

},

{

"@": "b616772b-113f-4909-a2f6-7552a36504c1",

"id": 2,

"article": "CkRpAamD",

"name": "Test",

"price": 1956,

"qty": 5

},

{

"@": "b616772b-113f-4909-a2f6-7552a36504c1",

"id": 3,

"article": "KDF5Lexy",

"name": "Test",

"price": 229,

"qty": 3

}

],

"create\_ts": 1614091193,

"client\_id": "test\_id\_1234",

"total": 55264,

"earned": 2762,

"spend": 0,

"spend\_sign": "",

"spend\_raw\_tx":"VjNwM1ozcEJRbmxvY0VGaFp6bG9PVFF5ZEhWTVVWb3lVM2xPTXpCd2FVTkpSbEpVTXpGaVZqRmlUbkpGUmtWeWRsWnBaMkpTUWt4R1ZURnlZVTFGVkdJME1qSm9iVWRtU2xSSVozVjRVRE0yY1VWcFdFWXpOVGd5T0c0NFdIaEJZelpyYzNsb09HRkVjek5ETWxoUVRHcFdSbTFPUjBNNU1W",

"new\_account\_raw\_tx":""

}

}