
Руководство пользователя
“Сервис по созданию и
управлению балансами”

Москва, 2023 г.

Содержание

Содержание	2
1. Введение	3
1.1 Область применения	3
1.2 Краткое описание возможностей	3
1.3 Уровень подготовки пользователей	3
1.4 Перечень программной документации	3
1.5 Термины и сокращения	4
2. Назначение и требования для эксплуатации ПО	4
2.1 Функциональное назначение	4
2.2 Эксплуатационное назначение	4
2.3 Требования к эксплуатации серверной части	5
3. Подготовка к работе	6
4. Описание операций	7
5. Аварийные ситуации	26
6. Рекомендации по освоению	27

1. Введение

Наименование программного обеспечения “Сервис по созданию и управлению балансами”.

1.1 Область применения

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) “Сервис по созданию и управлению балансами”.

“Сервис по созданию и управлению балансами” представляет собой техническое решение, предоставляющее возможность управления денежными балансами.

1.2 Краткое описание возможностей

- Масштабирование благодаря наличию ресурсов для запуска нескольких экземпляров сервиса;
- Штатный функционал позволяет подписать **N инстансов** на одну и ту же очередь, сообщения из которой будут случайным образом приходить в тот или иной инстанс;
- Передача данных в виде запросов в Базу данных и получение этих данных;
- Управление балансами:
 - Списание средств;
 - Пополнение средств;
 - Заморозка средств;
 - Разморозка средств;
 - Подсчет баланса.

1.3 Уровень подготовки пользователей

Для интеграции API пользователь должен иметь квалификацию разработчика не ниже уровня Regular Middle.

1.4 Перечень программной документации

- “Сервис по созданию и управлению балансами”. Техническое задание (ГОСТ 19.201-78);
- “Сервис по созданию и управлению балансами”. Руководство пользователя;
- “Сервис по созданию и управлению балансами”. Инструкция по развертыванию экземпляра ПО;

-
- “Сервис по созданию и управлению балансами”. Описание жизненного цикла разработки ПО.

1.5 Термины и сокращения

API – описание способов взаимодействия одной компьютерной программы с другими.

БД – база данных – набор структурированных данных, хранящихся в виде таблицы.

Инстанс – экземпляр класса (объекта) в объектно-ориентированном программировании.

Мерчант – партнер, пользователь программного обеспечения “Сервис по созданию и управлению балансами”.

Очередь – это некая структура данных, которая обеспечивает хранение и передачу двоичных данных между различными участниками системы.

Сервис – независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Сервисы обычно выполняются в виде библиотек общего пользования.

ПО – программное обеспечение.

2. Назначение и требования для эксплуатации ПО

Программное обеспечение “Сервис по созданию и управлению балансами” предназначено для использования на стороне мерчанта.

2.1 Функциональное назначение

“Сервис по созданию и управлению балансами” представляет собой техническое решение, предоставляющее возможность управления денежными балансами.

2.2 Эксплуатационное назначение

ПО интегрируется и эксплуатируется на стороне Заказчика и предоставляет следующие преимущества при использовании сервиса:

-
- **Независимость** компонентов системы друг от друга. Благодаря использованию очереди, компоненты взаимодействуют через некий общий интерфейс, но ничего не знают о существовании друг друга.
 - **Экономия ресурсов** достигается вследствие возможности разумно распределять информацию, поступающую в очередь от одних процессов, между другими процессами, осуществляющими ее обработку. Кроме того, благодаря тому, что нет необходимости промежуточного хранения необработанных данных, достигается дополнительная экономия ресурсов.
 - **Надежность** очередей достигается благодаря возможности накапливать сообщения, амортизируя недостаток вычислительных возможностей системы, а также благодаря независимости компонентов. Помимо этого очередь может аккомодировать сбои отдельных компонентов, осуществляя доставку «опоздавших» сообщений после восстановления.
 - И, наконец, **гарантия последовательной обработки**, позволяющая точно контролировать потоки данных в системе и запускать асинхронную обработку там, где это необходимо, не беспокоясь, что одна операция выполнится раньше другой, от результата которой она зависит.

2.3 Требования к эксплуатации серверной части

Обеспечение функционирования ПО серверной части “Сервис по созданию и управлению балансами” реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей являются:

- Современная ОС: Linux;
- Оперативная память: 2 ГБ;
- Свободное дисковое пространство: не менее 20 Gb;
- Количество логических ядер процессора: 2;
- Частота процессора: 3.50 GHz.

Возможно разворачивание экземпляра ПО и на других ОС, поддерживающих платформу для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker, например Windows 10 (Профессиональная или Корпоративная).

3. Подготовка к работе

3.1 Формат запросов

Оповещения передаются на POST запросов на url в формате json rpc 2.0, предоставляемый при подключении платформы. Важно проверять X-Data-Hash запроса.

3.1.1 HTTP-заголовки

Обязательные к передаче заголовки HTTP запроса:

```
Content-Type: application/json
```

```
X-Data-Hash
```

3.1.2 X-Data-Hash

Используется для подтверждения легитимности и подлинности запроса. Передается в HTTP заголовке **X-Data-Hash**.

```
X-Data-Hash = sha512(%request_body% + %secret_key%)
```

- **request_body** – объект запроса, сериализованный в виде JSON строки (берем исходную строку).
- **secret_key** – секретный ключ, выдаваемый при подключении приложения.

На выходе получаем ЭЦП к запросу.

Проверка производится следующим образом:

```
isValidRequest = X-Data-Hash == sha512(%request_body% + %secret_key%)
```

- **isValidRequest == true** - запрос легитимный;
- **isValidRequest == false** - запрос следует проигнорировать.

4. Описание операций

4.1 balance.in

Данный метод необходим для пополнения валюты баланса.

4.1.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.value</code>	Number	true	Сумма
<code>balance.amount.currency</code>	String	true	Валюта
<code>operation</code>	Object	false	Объект дополнительной информации об операции
<code>operation.external_id</code>	String	false	Внешний идентификатор платежа
<code>operation.service_id</code>	Number	false	Идентификатор сервиса, создавшего операцию

4.1.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.in",
  "params": {
    "balance": {
      "id": 5,
      "amount": {
        "value": 1000000000,
        "currency": "RUB"
      }
    }
  },
}
```

```
"operation": {  
  "external_id": "12345",  
  "service_id": 123  
}  
},  
"id": 1  
}
```

4.1.3 Формат ответа (пример)

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "balance": {  
      "amount": {  
        "id": 1,  
        "balance_id": 1,  
        "currency": "RUB",  
        "amount": 1009990700,  
        "amount_freezing": 0,  
        "amount_blocking": 0,  
        "created_at": "2020-06-14T19:42:33.949Z",  
        "updated_at": "2020-09-01T06:55:29.617Z",  
        "enabled": true  
      }  
    },  
  },  
}
```



```

"operation": {
  "id": "2164",
  "balance_amount_id": 1,
  "amount": 1000000000,
  "operation_type": "in",
  "created_at": "2020-09-01T06:55:29.608Z",
  "success": true,
  "updated_at": "2020-09-01T06:55:29.632Z",
  "service_id": 123,
  "external_id": "12345",
  "balance_amount": 1009990700,
  "balance_amount_freezing": 1000000000,
  "balance_amount_blocking": 0
}
}
}

```

4.2 balance.out

Данный метод необходим для уменьшения суммы валюты баланса.

4.2.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.value</code>	Number	true	Сумма

<code>balance.amount.currency</code>	String	true	Валюта
<code>operation</code>	Object	false	Объект дополнительной информации об операции
<code>operation.external_id</code>	String	false	Внешний идентификатор платежа
<code>operation.service_id</code>	Number	false	Идентификатор сервиса, создавшего операцию

4.2.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.out",
  "params": {
    "balance": {
      "id": 1,
      "amount": {
        "value": 1000000000,
        "currency": "RUB"
      }
    },
    "operation": {
      "external_id": "12345",
      "service_id": 123
    }
  },
  "id": 1
}
```

4.2.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 1,
```

```
"result": {
  "balance": {
    "amount": {
      "id": 1,
      "balance_id": 1,
      "currency": "RUB",
      "amount": 1009990700,
      "amount_freezing": 0,
      "amount_blocking": 0,
      "created_at": "2020-06-14T19:42:33.949Z",
      "updated_at": "2020-09-01T06:55:29.617Z",
      "enabled": true
    }
  },
  "operation": {
    "id": "2164",
    "balance_amount_id": 1,
    "amount": 1000000000,
    "operation_type": "out",
    "created_at": "2020-09-01T06:55:29.608Z",
    "success": true,
    "updated_at": "2020-09-01T06:55:29.632Z",
    "service_id": 123,
    "external_id": "12345",
    "balance_amount": 1009990700,
    "balance_amount_freezing": 0,
    "balance_amount_blocking": 0
  }
}
```

4.3 balance.freezing

Данный метод необходим для заморозки суммы валюты баланса.

4.3.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.value</code>	Number	true	Сумма
<code>balance.amount.currency</code>	String	true	Валюта
<code>operation</code>	Object	false	Объект дополнительной информации об операции
<code>operation.external_id</code>	String	false	Внешний идентификатор платежа
<code>operation.service_id</code>	Number	false	Идентификатор сервиса, создавшего операцию

4.3.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.freezing",
  "params": {
    "balance": {
      "id": 1,
      "amount": {
        "value": 1000000000,
        "currency": "RUB"
      }
    }
  }
}
```

```
    },
    "operation": {
      "external_id": "12345",
      "service_id": 123
    }
  },
  "id": 1
}
```

4.3.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "balance": {
      "amount": {
        "id": 1,
        "balance_id": 1,
        "currency": "RUB",
        "amount": 1009990700,
        "amount_freezing": 1000000000,
        "amount_blocking": 0,
        "created_at": "2020-06-14T19:42:33.949Z",
        "updated_at": "2020-09-01T06:55:29.617Z",
        "enabled": true
      }
    },
    "operation": {
      "id": "2164",
      "balance_amount_id": 1,
      "amount": 1000000000,
      "operation_type": "freezing",
      "created_at": "2020-09-01T06:55:29.608Z",
      "success": true,
      "updated_at": "2020-09-01T06:55:29.632Z",
      "service_id": 123,
    }
  }
}
```

```

    "external_id": "12345",
    "balance_amount": 1009990700,
    "balance_amount_freezing": 1000000000,
    "balance_amount_blocking": 0
  }
}
}

```

4.4 balance.unfreezing

Данный метод необходим для разморозки суммы валюты баланса.

4.4.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.value</code>	Number	true	Сумма
<code>balance.amount.currency</code>	String	true	Валюта
<code>operation.external_id</code>	String	false	Внешний идентификатор платежа
<code>operation.service_id</code>	Number	false	Идентификатор сервиса, создавшего операцию

4.4.2 Формат запроса (пример)

```
{
```

```
"jsonrpc": "2.0",
"method": "balance.unfreezing",
"params": {
  "balance": {
    "id": 1,
    "amount": {
      "value": 1000000000,
      "currency": "RUB"
    }
  },
  "operation": {
    "external_id": "12345",
    "service_id": 123
  }
},
"id": 1
}
```

4.4.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "balance": {
      "amount": {
        "id": 1,
        "balance_id": 1,
        "currency": "RUB",
        "amount": 1009990700,
        "amount_freezing": 1000000000,
        "amount_blocking": 0,
        "created_at": "2020-06-14T19:42:33.949Z",
        "updated_at": "2020-09-01T06:55:29.617Z",
        "enabled": true
      }
    }
  },
}
```

```
"operation": {
  "id": "2164",
  "balance_amount_id": 1,
  "amount": 1000000000,
  "operation_type": "unfreezing",
  "created_at": "2020-09-01T06:55:29.608Z",
  "success": true,
  "updated_at": "2020-09-01T06:55:29.632Z",
  "service_id": 123,
  "external_id": "12345",
  "balance_amount": 1009990700,
  "balance_amount_freezing": 0,
  "balance_amount_blocking": 0
}
}
```

4.5 balances.get

Данный метод необходим для получения всех балансов.

4.5.1 Основные параметры

Параметры отсутствуют

4.5.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balances.get",
  "params": {},
  "id": 1
}
```

4.5.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 1,
```



```
"result": {
  "balances": [
    {
      "id": 1,
      "created_at": "2020-07-14T20:03:47.583Z",
      "updated_at": "2020-07-14T20:03:47.583Z",
      "enabled": false
    },
    {
      "id": 2,
      "created_at": "2020-07-14T20:03:55.632Z",
      "updated_at": "2020-07-14T20:03:55.632Z",
      "enabled": false
    },
    {
      "id": 3,
      "created_at": "2020-08-24T15:21:53.207Z",
      "updated_at": "2020-08-24T15:21:53.207Z",
      "enabled": false
    },
  ]
}
```

4.6 balance.create

Данный метод необходим для создания баланса.

4.6.1 Основные параметры

Параметры отсутствуют

4.6.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.create",
  "params": {},
}
```

```
"id": 1
}
```

4.6.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "balance": {
      "id": 15,
      "created_at": "2020-09-01T06:00:51.580Z",
      "updated_at": "2020-09-01T06:00:51.580Z",
      "enabled": false
    }
  }
}
```

4.7 balance.amount.create

Данный метод необходим для создания валюты баланса.

4.7.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.currency</code>	String	true	Валюта

4.7.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.amount.create",
  "params": {
    "balance": {
      "id": 2,
      "amount": {
        "currency": "EUR"
      }
    }
  },
  "id": 10
}
```

4.7.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 10,
  "result": {
    "balance": {
      "amount": {
        "id": 4,
        "balance_id": 2,
        "currency": "EUR",
        "amount": 0,
        "amount_freezing": 0,
        "amount_blocking": 0,
        "created_at": "2020-09-01T06:39:24.474Z",
        "updated_at": "2020-09-01T06:39:24.474Z",
        "enabled": false
      }
    }
  }
}
```

4.8 balance.amount.get

Данный метод необходим для получения валюты баланса.

4.8.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	true	Объект баланса
<code>balance.id</code>	Number	true	Идентификатор баланса
<code>balance.amount</code>	Object	true	Объект валюты
<code>balance.amount.currency</code>	String	true	Валюта

4.8.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.amount.get",
  "params": {
    "balance": {
      "id": 2,
      "amount": {
        "currency": "EUR"
      }
    }
  },
  "id": 10
}
```

4.8.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
```

```

    "id": 10,
    "result": {
      "balance": {
        "amount": {
          "id": 4,
          "balance_id": 2,
          "currency": "EUR",
          "amount": 0,
          "amount_freezing": 0,
          "amount_blocking": 0,
          "created_at": "2020-09-01T06:39:24.474Z",
          "updated_at": "2020-09-01T06:39:24.474Z",
          "enabled": false
        }
      }
    }
  }
}

```

4.9 balance.amount.operations.get

Данный метод необходим для получения операций по валюте баланса.

4.9.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>balance</code>	Object	false	Объект баланса
<code>balance.id</code>	Number Number[]	false	Идентификатор баланса
<code>balance.amount</code>	Object	false	Объект валюты

<code>balance.amount.currency</code>	String String[]	false	Валюта
<code>conditions</code>	Object	false	Объект условий
<code>conditions.id</code>	BigInt BigInt[]	false	Идентификатор(ы) операции
<code>conditions.external_id</code>	BigInt BigInt[]	false	Внешний(е) идентификатор операции
<code>conditions.service_id</code>	Number Number[]	false	Идентификатор(ы) сервиса, создавшего операцию
<code>conditions.operation_type</code>	'in' 'out' 'freezing' 'unfreezing'	false	Тип операции
<code>conditions.success</code>	Boolean	false	Флаг успешности операции
<code>conditions.created_at</code>	Object	false	Объект с условиями для даты создания операции
<code>conditions.created_at.from</code>	Date	false	Дата с которой должна начинаться выборка
<code>conditions.created_at.to</code>	Date	false	Дата на которой должна заканчиваться выборка
<code>conditions.updated_at</code>	Object	false	Объект с условиями для даты изменения операции
<code>conditions.updated_at.from</code>	Date	false	Дата с которой должна начинаться выборка
<code>conditions.updated_at.to</code>	Date	false	Дата на которой должна заканчиваться выборка

<code>conditions.limit</code>	Number	false	Максимальный размер выборки
<code>conditions.offset</code>	Number	false	Сдвиг выборки
<code>conditions.sort</code>	'id' 'external_id' 'service_id' 'currency' 'operation_type' 'balance_id' 'created_at' 'updated_at'	false	Поле, по которому необходима сортировка
<code>conditions.sort_kind</code>	asc' 'desc'	false	Тип сортировки

4.9.2 Формат запроса (пример)

```
{
  "jsonrpc": "2.0",
  "method": "balance.amount.operations.get",
  "params": {
    "balance": {
      "id": [2],
      "amount": {
        "currency": ["USD"]
      }
    },
    "conditions": {
      "id": ["341", "342", "343"],
      "operation_type": "in",
      "success": true,
      "created_at": {
        "from": "2021-04-06T12:55:03.448Z",
        "to": "2021-04-06T16:55:03.448Z"
      }
    }
  }
}
```

```
    },
    "updated_at": {
      "from": "2021-04-06T12:55:03.448Z",
      "to": "2021-04-06T16:55:03.448Z"
    },
    "limit": 10,
    "offset": 0,
    "sort": "asc",
    "sort_kind": "id"
  }
},
"id": 10
}
```

4.9.3 Формат ответа (пример)

```
{
  "jsonrpc": "2.0",
  "id": 10,
  "result": {
    "conditions": {
      "offset": 0,
      "limit": 10,
      "count": "3",
      "sort": "id",
      "sort_kind": "asc"
    },
    "operations": [
      {
        "id": "341",
        "balance_amount_id": 55,
        "amount": 123,
        "operation_type": "freezing",
        "created_at": "2021-05-31T15:54:12.080Z",
        "updated_at": "2021-05-31T15:54:12.093Z",
        "error": {
          "code": 9,
          "message": "Not enough funds"
        }
      }
    ]
  }
}
```



```
    },
    "success": false,
    "service_id": null,
    "external_id": null,
    "balance_id": 72,
    "currency": "USD",
    "balance_amount": 123,
    "balance_amount_freezing": 123,
    "balance_amount_blocking": 0
  },
  {
    "id": "342",
    "balance_amount_id": 55,
    "amount": 123,
    "operation_type": "in",
    "created_at": "2021-05-31T15:54:23.825Z",
    "updated_at": "2021-05-31T15:54:23.842Z",
    "error": null,
    "success": true,
    "service_id": null,
    "external_id": null,
    "balance_id": 72,
    "currency": "USD",
    "balance_amount": 256,
    "balance_amount_freezing": 123,
    "balance_amount_blocking": 0
  },
  {
    "id": "343",
    "balance_amount_id": 55,
    "amount": 123,
    "operation_type": "in",
    "created_at": "2021-05-31T15:55:39.855Z",
    "updated_at": "2021-05-31T15:55:39.875Z",
    "error": null,
    "success": true,
    "service_id": null,
    "external_id": null,
```

```
"balance_id": 72,  
"currency": "USD",  
"balance_amount": 369,  
"balance_amount_freezing": 123,  
"balance_amount_blocking": 0  
}  
]  
}  
}
```

5. Аварийные ситуации

В случае аварийных ситуаций обращаться в техническую поддержку.

6. Рекомендации по освоению

Для успешной работы с ПО “Сервис по созданию и управлению балансами” необходимо:

- Иметь квалификацию разработчика не ниже уровня Regular Middle;
- Иметь оборудованное рабочее место с подключением к сети Интернет;
- Рабочее место должно соответствовать минимальным требованиям, указанным в разделе “2.3 Требования к эксплуатации серверной части” данного документа;
- Ознакомиться с документом “Руководство пользователя “Сервис по созданию и управлению балансами””;
- Ознакомиться с документом “Разворачивание экземпляра ПО “Сервис по созданию и управлению балансами””.