

---

# Руководство пользователя

## **“Сервис по приему и верификации запросов”**

Москва, 2023 г.

---

<b>1. Введение</b>	<b>4</b>
1.1 Область применения	4
1.2 Краткое описание возможностей	4
1.3 Уровень подготовки пользователей	4
1.4 Перечень программной документации	4
1.5 Термины и сокращения	5
<b>2. Назначение и требования для эксплуатации ПО</b>	<b>5</b>
2.1 Функциональное назначение	6
2.2 Эксплуатационное назначение	6
2.3 Требования к эксплуатации серверной части	6
<b>3. Подготовка к работе</b>	<b>7</b>
3.1 Формат запросов	7
3.1.1 X-Data-Application-ID	7
3.1.2 X-Data-Hash	7
<b>4. Описание операций</b>	<b>7</b>
4.1 Пополнение	8
4.1.1 Основные параметры	8
4.1.2 Формат запроса (пример)	10
4.1.3 Формат ответа (пример)	11
4.2 Выплата	12
4.2.1 Основные параметры	12
4.2.2 Формат запроса (пример)	15
4.2.3 Формат ответа (пример)	15
4.3 Получение статусов	16
4.3.1 Параметры запроса	16
4.3.2 Формат запроса (пример)	16
4.3.3 Формат ответа (пример)	16
<b>4.4 Получение списка доступных стран</b>	<b>17</b>
4.4.1 Параметры запроса	17
4.4.2 Формат запроса (пример)	17
4.4.3 Формат ответа (пример)	18
<b>4.5 Получение списка доступных документов</b>	<b>18</b>
4.5.1 Параметры запроса	19
4.5.2 Формат запрос (пример)	19
4.5.3 Формат ответа (пример)	20
<b>4.6 Получение списка регионов по стране</b>	<b>21</b>
4.6.1 Параметры запроса	21
4.6.2 Формат запроса (пример)	22
4.6.3 Формат ответа (пример)	22
<b>4.7 Получение списка банков</b>	<b>22</b>
4.7.1 Параметры запроса	22

---

4.7.2 Формат запроса (пример)	23
4.7.3 Формат ответа (пример)	23
<b>5. Аварийные ситуации</b>	<b>23</b>
6. Рекомендации по освоению	23

---

## 1. Введение

Наименование программного обеспечения “Сервис по приему и верификации запросов”.

### 1.1 Область применения

“Сервис по приему и верификации запросов” должен представлять собой техническое решение, *B2B сервис*, с возможностью получения заголовков HTTP запросов, верификации этих запросов и обработке.

### 1.2 Краткое описание возможностей

- Возможность приема запросов в унифицированном формате;
- Возможность приемки шлюзом API и дальнейшей автоматической разбивки на несколько запросов;
- Наличие штатного функционала позволяет подписать *N инстансов* на одну и ту же очередь, сообщения из которой случайным образом приходят в тот или иной инстанс;
- Возможность проверки запросов по заданным параметрам;
- Наличие собственной базы данных для хранения информации о платежных провайдерах с которыми работает сервис;
- Передача данных в виде запросов в Базу данных и получение этих данных.

### 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 приему и верификации запросов”. Техническое задание (ГОСТ 19.201-78);
- “Сервис по приему и верификации запросов”. Руководство пользователя;
- “Сервис по приему и верификации запросов”. Инструкция по развертыванию экземпляра ПО;
- “Сервис по приему и верификации запросов”. 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 приему и верификации запросов”.

**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.

**Платежный провайдер** – компания, которая предоставляет онлайн-сервисы по осуществлению электронных платежей различными способами. Некоторые поставщики платежных сервисов предоставляют различные инновационные сервисы.

**ПО** – программное обеспечение.

## 2. Назначение и требования для эксплуатации ПО

Программное обеспечение “Сервис по приему и верификации запросов” предназначено для использования на стороне мерчанта.

### 2.1 Функциональное назначение

ПО представляет собой техническое решение, с возможностью получения заголовков HTTP запросов, верификации этих запросов и обработке.

---

## 2.2 Эксплуатационное назначение

ПО интегрируется и эксплуатируется на стороне мерчанта и предоставляет следующие возможности:

- Прием запросов в унифицированном формате;
- Приемка шлюзом API и автоматическая разбивка на несколько запросов;
- Поднятие нескольких одинаковых шлюзов благодаря наличию инстансов;
- Выполнение проверки запросов на параметры:
  - Можно ли мерчанту отправлять запрос;
  - Корректны ли данные в подписи;
  - Доступен ли сервис мерчанту;
  - От кого поступил запрос;
  - По какой операции;
  - На какой сервис или плагин перенаправить запрос;
  - По какому платежному методу.

## 2.3 Требования к эксплуатации серверной части

Обеспечение функционирования ПО серверной части “Сервис по приему и верификации запросов” 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

Взаимодействие происходит посредством отправки запросов и ответов на них. Запросы отправляются по протоколу HTTPS POST-запросами по умолчанию, если для запроса не задан другой метод. Запросы должны содержать обязательные заголовки: `Content-Type: application/json`, `X-Data-Application-ID` и `X-Data-Hash`.

#### 3.1.1 X-Data-Application-ID

Статичный параметр, выдаваемый при подключении приложения. Целое число.

#### 3.1.2 X-Data-Hash

Используется для подтверждения легитимности и подлинности запроса. Передается в формате `X-Data-Hash = sha512(%request_body% + %secret_key%)`

- `request_body` - объект запроса, сериализованный в виде JSON строки.
- `secret_key` - секретный ключ, выдаваемый при подключении приложения.
- `sha512` - алгоритм однонаправленного хэширования. На выходе получаем ЭЦП к запросу.

Формат содержит заголовок `Content-Type: "application/json"` и объекты в JSON

## 4. Описание операций

### 4.1 Пополнение

Перевод средств со счета клиента на счет платежного сервиса.

#### 4.1.1 Основные параметры

Параметр	Тип	Обязательный	Описание
----------	-----	--------------	----------

<code>method</code>	string	+	Название метода
<code>params.payment</code>	Payment	+	Информация о транзакции
<code>params.payment.service_id</code>	integer	+	Идентификатор платежной системы
<code>params.payment.amount</code>	Amount	+	Объект суммы и валюты платежа
<code>params.payment.amount.value</code>	integer	+	Сумма платежа
<code>params.payment.amount.currency</code>	string(3)	+	Валюта платежа
<code>params.payment.identifiers</code>	Identifier	+	Объект с идентификаторами платежа
<code>params.payment.identifiers.c_id</code>	integer	+	Идентификатор платежа на стороне Мерчанта
<code>params.payment.payer</code>	Payer	-	Объект с платежными данными пользователя
<code>params.payment.payer.card</code>	Card	-	Карточные данные при платеже
<code>params.payment.payer.card.id</code>	string	-	Карточный токен при платеже
<code>params.payment.payer.card.mask</code>	string	-	Маска карты при платеже
<code>params.payment.payer.card.holder</code>	string	-	Имя Фамилия владельца карты при платеже
<code>params.payment.payer.payment_system_account</code>	Account	-	Объект с аккаунтом пользователя платежной системы

			при платеже
<code>params.payment.payer.payment_system_account_id</code>	string	+	Идентификатор аккаунта пользователя платежной системы при платеже
<code>params.payment.payer.person</code>	Person	-	Объект с персональными данными
<code>params.payment.payer.person.first_name</code>	string	-	Имя пользователя
<code>params.payment.payer.person.last_name</code>	string	-	Фамилия пользователя
<code>params.payment.payer.person.birth_date</code>	string	-	Дата рождения пользователя (DD-MM-YYYY)
<code>params.payment.payer.person.documents</code>	Array	-	Список документов пользователя
<code>params.payment.payer.person.documents[0].type</code>	string	-	Тип документа
<code>params.payment.payer.person.documents[0].number</code>	string	-	Номер документа
<code>params.payment.payer.customer_account</code>	object	-	Данные по пользователю у Мерчанта
<code>params.payment.payer.customer_account_id</code>	string	+	Идентификатор привязанного пользователя
<code>params.payment.redirect</code>	Redirect	-	Объект с информацией о переадресациях пользователя

<code>params.payment.redirect.on_success</code>	string	-	ссылка для редиректа в случае успеха
<code>params.payment.redirect.on_fail</code>	string	-	ссылка для редиректа в случае ошибки/отказа/
<code>params.params.payment.redirect.on_return</code>	string	-	ссылка для редиректа в случае возврата со страницы
<code>params.payment.client</code>	object	-	Объект с данными устройства пользователя
<code>params.payment.client.ip</code>	string	-	IP адрес пользователя
<code>params.payment.client.browser</code>	string	-	Браузер пользователя
<code>params.payment.client.country</code>	string	-	Страна пользователя
<code>params.payment.client.language</code>	string	-	Язык пользователя

#### 4.1.2 Формат запроса (пример)

```
{ "service_id": 233, "method": "payment.in", "params": { "payment": { "identifiers": { "c_id": 213 }, "description": "комментарий", "amount": { "value": 10000, "currency": "BRL" }, "payer": { "email": "test@test.com", "person": { "first_name": "Pablo", "last_name": "FSF", "birth_date": "12-12-1990", "documents": [ { "type": "BR_CPF", "number": "12345678946" } ] }, "customer_account": { "id": "test@gmail.com" }, "payment_system_account": { "id": "23" } }, "client": { "language": "RU" } } } }
```

### 4.1.3 Формат ответа (пример)

```
{ "success": true, "result": { "payment": { "amount": { "value": 10000, "currency": "BRL" }, "description": "комментарий", "identifiers": { "c_id": 50312, "h_id": 600, "p_id": null }, "redirect": { "to": "https://cashier.paywallk.com/payment?payment_amount=10000&payment_currency=RUB&payment_id=50312&language_code=EN&payment_description=%D0%BA%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B9&force_payment_method=neteller&project_id=1960&signature=4btWWpdcT6OYOSmJDLmHg9dNBCTG%2BE36vlVw7BYrvEq5mec6gSyhJBDmpoF4eCgO7C%2FO9tISO2b6EzrH4BmTBQ%3D%3D" }, "status": { "status": "await_redirect", "final": false, "success": null, "error": null, "history": [ { "status": "created", "final": false, "success": null, "created": "2020-02-26T14:20:09.127Z", "reason": null, "amount": 10000 }, { "status": "await_redirect", "final": false, "success": null, "created": "2020-02-26T14:20:09.545Z", "reason": null, "amount": 10000 } ] }, "timestamps": { "created": "2020-02-26T14:20:09.127Z", "updated": "2020-02-26T14:20:09.476Z", "finished": null }, "destination": "in", "operations": [ { "id": "617", "operation_type": "payment.in", "amount": { "value": 10000, "currency": "RUB" }, "timestamps": { "created": "2020-02-26T14:20:09.200Z", "updated": "2020-02-26T14:20:09.571Z", "finished": null }, "status": { "status": "await_user_action", "final": false, "success": null, "error": null } } ] }, "operation": { "id": "617", "operation_type": "payment.in", "amount": { "value": 10000, "currency": "RUB" }, "timestamps": { "created": "2020-02-26T14:20:09.200Z", "updated": "2020-02-26T14:20:09.571Z", "finished": null }, "status": { "status": "await_user_action", "final": false, "success": null, "error": null } } }, "next": null, "request_id": "gMUy6lExaRfzYjHIRY1dl", "processing_time": 1135 }
```

## 4.2 Выплата

Перевод средств со счета платежного сервиса на счет клиента.

## 4.2.1 Основные параметры

Параметр	Тип	Обязательный	Описание
<code>method</code>	string	+	Название метода
<code>params.payment</code>	Payment	+	Информация о транзакции
<code>params.payment.service_id</code>	integer	+	Идентификатор платежной системы
<code>params.payment.amount</code>	Amount	+	Объект суммы и валюты платежа
<code>params.payment.amount.value</code>	integer	+	Сумма платежа
<code>params.payment.amount.currency</code>	string(3)	+	Валюта платежа
<code>params.payment.identifiers</code>	Identifier	+	Объект с идентификаторами платежа
<code>params.payment.identifiers.c_id</code>	integer	+	Идентификатор платежа на стороне Мерчанта
<code>params.payment.receiver</code>	Receiver	-	Объект с данными пользователя при выплате
<code>params.payment.receiver.card</code>	Card	-	Карточные данные при выплате
<code>params.payment.receiver.card.id</code>	string	-	Карточный токен при выплате
<code>params.payment.receiver.card.mask</code>	string	-	Маска карты при выплате

<code>params.payment.receiver.card_holder</code>	string	-	Имя Фамилия владельца карты при выплате
<code>params.payment.receiver.payment_system_account</code>	string	-	Объект с аккаунтом пользователя платежной системы при выплате
<code>params.payment.receiver.payment_system_account.id</code>	string	+	Идентификатор аккаунта пользователя платежной системы при выплате
<code>params.payment.receiver.person</code>	Person	-	Объект с персональными данными
<code>params.payment.receiver.person.firts_name</code>	string	-	Имя пользователя
<code>params.payment.receiver.person.last_name</code>	string	-	Фамилия пользователя
<code>params.payment.receiver.person.birth_date</code>	string	-	Дата рождения пользователя (DD-MM-YYYY)
<code>params.payment.receiver.person.documents</code>	Array	-	Список документов пользователя
<code>params.payment.receiver.person.documents[0].type</code>	string	-	Тип документа
<code>params.payment.receiver.person.documents[0].number</code>	string	-	Номер документа
<code>params.payment.receiver.customer_account</code>	object	-	Данные по пользователю у Мерчанта
<code>params.payment.receiver.customer_account.id</code>	string	+	Идентификатор привязанного пользователя

<code>params.receiver.redirect</code>	Redirect	-	Объект с информацией о переадресациях пользователя
<code>params.receiver.redirect.on_success</code>	string	-	ссылка для редиректа в случае успеха
<code>params.receiver.redirect.on_fail</code>	string	-	ссылка для редиректа в случае ошибки/отказа/
<code>params.receiver.redirect.on_return</code>	string	-	ссылка для редиректа в случае возврата со страницы
<code>params.payment.client</code>	object	-	Объект с данными устройства пользователя
<code>params.payment.client.ip</code>	string	-	IP адрес пользователя
<code>params.payment.client.browser</code>	string	-	Браузер пользователя
<code>params.payment.client.country</code>	string	-	Страна пользователя
<code>params.payment.client.language</code>	string	-	Язык пользователя

#### 4.2.2 Формат запроса (пример)

```
{ "service_id": 233, "method": "payment.out", "params": {
"payment": { "description": "Some description", "identifiers": {
"c_id": 146 }, "receiver": { "person": { "last_name": "Dis",
"first_name": "Dis", "birth_date": "21-01-2021", "documents": [ {
"type": "BR_CPF", "number": "12345678946" } ] }, "email":
```

```
"test@test.com", "customer_account": { "id": "123" },  
"payment_system_account": { "id": "23" } }, "amount": { "value":  
20000, "currency": "BRL" }, "client": { "ip": "192.168.1.1" } } } }
```

### 4.2.3 Формат ответа (пример)

```
{ "success": true, "result": { "payment": { "amount": { "value":  
15000, "currency": "BRL" }, "description": "комментарий",  
"identifiers": { "c_id": 50311, "h_id": 599, "p_id": null },  
"status": { "status": "processing", "final": false, "success":  
null, "error": null, "history": [ { "status": "created", "final":  
false, "success": null, "created": "2020-02-26T13:17:03.608Z",  
"reason": null, "amount": 15000 }, { "status": "processing",  
"final": false, "success": null, "created": "2020-02-  
26T13:17:05.271Z", "reason": null, "amount": 15000 } ] },  
"timestamps": { "created": "2020-02-26T13:17:03.608Z", "updated":  
"2020-02-26T13:17:05.222Z", "finished": null }, "destination":  
"out", "operations": [ { "id": "616", "operation_type":  
"payment.out", "amount": { "value": 15000, "currency": "BRL" },  
"timestamps": { "created": "2020-02-26T13:17:03.681Z", "updated":  
"2020-02-26T13:17:05.291Z", "finished": null }, "status": {  
"status": "processing", "final": false, "success": null, "error":  
null } } ] }, "operation": { "id": "616", "operation_type":  
"payment.out", "amount": { "value": 15000, "currency": "BRL" },  
"timestamps": { "created": "2020-02-26T13:17:03.681Z", "updated":  
"2020-02-26T13:17:05.291Z", "finished": null }, "status": {  
"status": "processing", "final": false, "success": null, "error":  
null } } }, "next": null, "request_id": "bx2murjp45nO2kfKrNTJx",  
"processing_time": 2345 }
```

## 4.3 Получение статусов

Получение статуса платежа.

### 4.3.1 Параметры запроса

Параметр	Тип	Обязательность	Описание
payment	Object	true	Объект содержащий информацию о платеже
payment.identifiers	Object	true	Объект с идентификаторами платежа
payment.identifiers.c_id	Number	true	Идентификатор платежа на стороне платформы

### 4.3.2 Формат запроса (пример)

```
{ "service_id": 0, "method": "payment.status", "params": {
"payment": { "identifiers": { "c_id": 1 } } } }
```

### 4.3.3 Формат ответа (пример)

```
{ "success": true, "result": { "payment": { "amount": { "value":
99992, "currency": "RUB" }, "description": null, "identifiers": {
"c_id": 9999, "h_id": 38, "p_id": null }, "status": { "status":
"success", "final": true, "success": true, "error": null,
"history": [ { "status": "created", "final": false, "success":
null, "created": "2019-06-06T13:49:47.470Z", "reason": null,
"amount": "99992" }, { "status": "success", "final": true,
"success": true, "created": "2019-06-06T13:49:47.516Z", "reason": {
"code": 2000, "message": "Wrote..." }, "amount": "99992" } ] },
"timestamps": { "created": "2019-06-06T13:49:47.470Z", "updated":
"2019-06-06T13:49:47.516Z", "finished": "2019-06-06T13:49:47.516Z"
}, "destination": "in", "operations": [ { "id": "20",
"operation_type": "payment.in", "amount": { "value": 1000,
"currency": "RUB" }, "timestamps": { "created": "2019-06-
06T13:49:47.554Z", "updated": "2019-06-06T13:49:47.554Z",
"finished": null }, "status": { "status": "success", "final": true,
"success": true, "error": null } }, { "id": "21", "operation_type":
"payment.hold", "amount": { "value": 1000, "currency": "RUB" },
"timestamps": { "created": "2019-06-06T13:49:47.571Z", "updated":
```

```
"2019-06-06T13:49:47.576Z", "finished": "2019-06-06T13:49:47.576Z"
}, "status": { "status": "error", "final": true, "success": false,
"error": null } } ] } }, "next": null, "request_id":
"gIPmI9LmvJh2mTemS4MeB", "processing_time": 45 }
```

## 4.4 Получение списка доступных стран

### 4.4.1 Параметры запроса

Параметр	Описание
method	Для получения данных по всем странам - <a href="#">reference.countries</a>
params	Для данного метода отправляется пустой объект
service_id	Для данного метода service_id = 0

### 4.4.2 Формат запроса (пример)

```
{
  "method": "reference.countries",
  "params": {},
  "service_id": 0
}
```

### 4.4.3 Формат ответа (пример)

```
{
  "success": true,
```

```

"result": {
  "countries": [
    {
      "name": "Russian Federation",
      "code": "RU",
    },
    {
      "name": "Canada",
      "code": "CA",
    }
  ],
}

```

## 4.5 Получение списка доступных документов

Список документов по стране и платежной системе.

### 4.5.1 Параметры запроса

Параметр	Тип	Обязательность	Описание
method	String	+	Для получения регионов по стране - <code>reference.documents</code>
params	Object	+	Параметры для получения банка
params.servic	Num	+	Идентификатор платежной системы для

e_id	number		которой нужны документы
params.country.code	String	-	ISO 3166-2 код страны
service_id	Number	+	Для данного метода service_id = 0

#### 4.5.2 Формат запрос (пример)

```
{
  "method": "reference.documents",
  "params": {
    "country": {
      "code": "PE"
    },
    "service_id": 301
  },
  "service_id": 0
}
```

#### 4.5.3 Формат ответа (пример)

```
{
  "success": true,
  "result": {
    "documents": [
      {
```

```
    "doc_name": "PE_CE",
    "country_id": "PE",
    "regexp": "/^\\d{9}$/",
    "example": "111111111"
  },
  {
    "doc_name": "PE_DNI",
    "country_id": "PE",
    "regexp": "/^\\d{8,9}$/",
    "example": "157406116"
  },
  {
    "doc_name": "PE_PASS",
    "country_id": "PE",
    "regexp": "/^[a-zA-Z0-9]{12}$/",
    "example": "111111111111"
  },
  {
    "doc_name": "PE_RUC",
    "country_id": "PE",
    "regexp": "/^[a-zA-Z0-9]{11}$/",
    "example": "11111111111"
  }
]
},
```

```
}
```

## 4.6 Получение списка регионов по стране

Список регионов по стране.

### 4.6.1 Параметры запроса

Параметр	Тип	Обязательность	Описание
method	String	+	Для получения регионов по стране - <code>reference.country.regions</code>
params	Object	+	Параметры для получения банка
params.country.code	String	+	ISO 3166-2 код страны для получения списка регионов
service_id	Number	+	Для данного метода <code>service_id = 0</code>

### 4.6.2 Формат запроса (пример)

```
{  
  "method": "reference.country.regions",  
  "params": {  
    "country": {  
      "code": "BR"  
    }  
  },  
  "service_id": 0  
}
```

### 4.6.3 Формат ответа (пример)

```
{ "success": true, "result": { "regions": [ { "name": "Acre",  
"code": "BR-AC" }, { "name": "Alagoas", "code": "BR-AL" }, {  
"name": "Amapá", "code": "BR-AP" }, { "name": "Amazonas", "code":  
"BR-AM" }, { "name": "Bahia", "code": "BR-BA" }, ... ] } }
```

## 4.7 Получение списка банков

Список банков по стране.

### 4.7.1 Параметры запроса

Параметр	Тип	Обязательность	Описание
method	String	+	Для получения банка по стране - <code>reference.country.banks</code>
params	Object	+	Параметры для получения банка
params.country.code	String	+	ISO 3166-2 код страны для получения списка банков
params.method	String/null	-	Значение может принимать null/"payment.in"/"payment.out". Если не задан или null, выдает список банков для всех методов
params.bank.providers	Array	+	Провайдеры к которому привязан банки
service_id	Number	+	Для данного метода service_id = 0

---

#### 4.7.2 Формат запроса (пример)

```
{ "method": "reference.country.banks", "params": { "country": {  
"code": "BR" }, "bank": { "providers": [3] }, "method":  
"payment.in" }, "service_id": 0 }
```

#### 4.7.3 Формат ответа (пример)

```
{ "success": true, "result": { "banks": [ { "id": 321, "name":  
"Royal Bank of Canada" }, { "id": 999, "name": "Toronto-Dominion  
Bank" }] } }
```

### 5. Аварийные ситуации

В случае аварийных ситуаций обращаться в техническую поддержку.

### 6. Рекомендации по освоению

Для успешной работы с ПО "Сервис по приему и верификации запросов" 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 приему и верификации запросов"";
- Ознакомиться с документом "Разворачивание экземпляра ПО "Сервис по приему и верификации запросов".