

---

Руководство пользователя  
**“Сервис по оптимизации  
длины URL”**

Москва, 2023 г.

---

## Содержание

<b>Содержание</b>	<b>2</b>
<b>1. Введение</b>	<b>3</b>
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
<b>2. Назначение и требования для эксплуатации ПО</b>	<b>4</b>
2.1 Функциональное назначение	4
2.2 Эксплуатационное назначение	4
2.3 Требования к эксплуатации серверной части	5
<b>3. Подготовка к работе</b>	<b>6</b>
<b>4. Описание операций</b>	<b>7</b>
<b>5. Аварийные ситуации</b>	<b>8</b>
<b>6. Рекомендации по освоению</b>	<b>8</b>

---

# 1. Введение

Наименование программного обеспечения “Сервис по оптимизации длины URL”.

## 1.1 Область применения

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) “Сервис по оптимизации длины URL”.

“Сервис по оптимизации длины URL” представляет собой техническое решение, предоставляющее возможность оборачивать(укорачивать) **http-ссылки**.

## 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 приходит в тот или иной инстанс;
- Укорачивание http ссылки;
- Передача данных в виде запросов в Базу данных и получение этих данных;
- Редирект (переход) по сокращенной ссылки.

## 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 оптимизации длины URL”. Техническое задание (ГОСТ 19.201-78);
- “Сервис по оптимизации длины URL”. Руководство пользователя;
- “Сервис по оптимизации длины URL”. Инструкция по развертыванию экземпляра;
- “Сервис по оптимизации длины URL”. Описание жизненного цикла разработки.

---

## 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 отправки уведомлений при изменении статусов операций”.

**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 оптимизации длины URL” предназначено для использования на стороне мерчанта.

### 2.1 Функциональное назначение

ПО представляет собой техническое решение, с возможностью оборачивания (укорачивания) http ссылок.

### 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 оптимизации длины URL” 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, 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 Оборачивание (укорачивание) ссылки

Укорачивание массива ссылок

#### 4.1.1 Основные параметры

Параметр	Тип	Обязательность	Описание
<code>method</code>	String	+	Метод для оборачивания ссылок - links.wrap
<code>params</code>	Object	+	Параметры для метода
<code>params.links</code>	Array	+	Параметры для укорач
<code>params.links.name</code>	String	+	Имя ссылки
<code>params.links.original</code>	String	+	Оригинальная ссылка
<code>params.links.metadata</code>	Object/String	-	Любые дополнительные данные
<code>service_id</code>	Number	+	Для данного метода service_id = 0

#### 4.1.2 Формат запроса (пример)

```
{ "service_id": 0, "method": "links.wrap", "params": { "links": [ { "name": "test", "original": "http://test.com", "metadata": { "test": true } } ] } }
```

#### 4.1.3 Формат ответа (пример)

```
{ "success": true, "result": { "links": [ { "name": "test", "wrapped": "http://link-wrapper/sdgdsjk5ms3535" } ] }, "next": null, "request_id": "wPv5yQwerIx320DuU6xX", "processing_time": 155 }
```

## 4.2 Перенаправление по ссылке

По укороченной ссылке перенаправляет на оригинальную ссылку.

---

#### 4.2.1 Формат запроса (пример)

<https://base-url/:id>, где :id идентификатор ссылки в сервисе, приходит в ответе на запрос по обертыванию ссылки.

### 5. Аварийные ситуации

В случае аварийных ситуаций обращаться в техническую поддержку.

### 6. Рекомендации по освоению

Для успешной работы с ПО “Сервис по оптимизации длины URL” 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 оптимизации длины URL””;
- Ознакомиться с документом “Разворачивание экземпляра ПО “Сервис по оптимизации длины URL””.