

614066, Пермский край, г. Пермь, Индустриальный район, ул. Стахановская, д. 54, литера «Р», офис 400/5 Контакты: Дарья Чиртулова отдел технического анализа и экспертизы +7 (922)-367-98-69 +7 (342) 240 40 25 доб. 138 или + 7 (342) 240 38 56 отдел технической поддержки E-mail: support@bisys.ru

# Сервис приема платежей PayForm

Руководство по развертыванию ПО



## Пререквизиты

Файлы установки находятся по ссылке <u>https://bisys.ru/files</u> (ссылка на экземпляр ПО). Для скачивания дистрибутива ПО потребуется авторизоваться.

Авторизационные данные высылаются на адрес электронной почты после подачи заявки на нашем сайте по адресу <u>https://bisys.ru/form</u>

В случае, если у вас возникнут сложности с получением доступа, распаковкой/установкой экземпляра ПО, необходимо связаться с нашим специалистами по adpecy support@bisys.ru



## Оглавление

Пререквизиты	2
Оглавление	3
Термины и определения	4
Требования к среде для установки	5
Системные требования	5
Аппаратные требования	6
Опции безопасности	6
Установка	7
Локальная установка на node.js	7
Дефолтный конфиг продукта	8
Установка контейнера на сервере Apache1	1
Обновление1	2
Процедуры получения и установки обновления 1	2
Проверка сборки 1	2
Откат1	3



## Термины и определения

Термин	Определение		
Frontend	Часть веб-сайта, с которой взаимодействует пользователь. Это все, что пользователь видит в своем браузере: тексты, изображения, кнопки, формы и т.д. Frontend включает в себя разработку пользовательского интерфейса с использованием технологий, таких как HTML, CSS и JavaScript		
Backend	Серверная часть веб-сайта, которая обрабатывает запросы, полученные от клиента (в данном случае от Frontend). Васkend занимается обработкой данных, взаимодействием с базами данных, выполнением бизнес-логики и формированием ответов на запросы. Включает проверку введенных данных, обработку платежа через платежный шлюз, сохранение информации о транзакции в базе данных и отправку подтверждения об успешной оплате на Frontend		
Торговая точка (мерчант, платежный агент, магазин, поставщик услуги)	Представляет собой юр. или физ. лицо, которое принимает оплату за предлагаемые им товары/услуги. Примеры: Интернет-магазины, вендинговые автоматы		
OC	Операционная система		
Зелёная зона	жим приёма платежей для рынка Телекоммуникаций, гда у абонента нулевой баланс и есть ограничения оступа к глобальной сети		
Фавикон (англ. favicon, сокращение от favorite icon)	Небольшая иконка, которая обычно отображается в адресной строке браузера рядом с названием веб-сайта или на вкладке браузера. Она также может появляться в закладках, истории посещений и других местах, где гребуется идентификация веб-сайта		



Title (тайтл)	Название (заголовок) страницы, отображающееся в виде ссылки на документ в поисковой выдаче, в закладках браузера, в постах соцсетей и мессенджеров. Он написан более крупным шрифтом, чем описание, как гиперссылка выделен синим цветом и подчёркивается при наведении курсора компьютерной мыши
Превью	Небольшое изображение или краткое описание веб- страницы, которое обычно отображается в результатах поиска или на социальных медиа. Оно предоставляет пользователям предварительное представление о содержании страницы и может включать заголовок, краткое описание и изображение
Обфускация кода	Процесс изменения исходного кода программы таким образом, чтобы он стал сложнее для понимания и анализа.

## Требования к среде для установки

### Системные требования

Язык программирования:

- JavaScript ECMAScript ES6
- PHP 7.3.3
- HTML

Framework:

• React (JS)

Системные требования:

- Для языка программирования JavaScript ECMAScript ES6 и фреймворка React (JS) необходимо учитывать совместимость с браузерами, такими как Google Chrome, Mozilla Firefox, Safari, и т.д.
- Для PHP 7.3.3 рекомендуется использовать совместимые версии веб-серверов, такие как Apache, Nginx или другие, поддерживающие PHP.
- Для HTML нет специфических системных требований, поскольку HTML интерпретируется браузером.



Для запуска Frontend веб-приложения **PayForm** с использованием Node.js, рекомендуется использовать Node.js LTS-версию. Node.js является кроссплатформенным и может работать на различных операционных системах, включая Windows, Linux, Unix, macOS и другие. Однако, для установки и использования Node.js на Windows, также можно использовать Windows Subsystem for Linux (WSL) для запуска Linux-версии Node.js на Windows.

Рекомендуемая версия Windows: Windows 10.

## Аппаратные требования

Для развертывания Frontend веб-приложения **PayForm** необходимы следующие минимальные аппаратные требования к серверу приложений:

- 1. Процессор: 1 ГГц двухъядерный процессор или выше.
- 2. Оперативная память: 2 ГБ ОЗУ или более.
- 3. Жесткий или твердотельный диск: Минимум 1 ГБ свободного места.
- 4. Видеокарта: Современная видеокарта с поддержкой HTML5 и CSS3.
- 5. Сетевое оборудование: Сетевая карта для подключения к Интернет. Минимальная скорость подключения 1 Мбит/с.
- 6. Современные драйверы устройств и обновления ОС.

Можно установить на одну машину или же развернуть на нескольких машинах (для горизонтального масштабирования программного продукта, распределения нагрузки, повышения отказоустойчивости).

### Опции безопасности

Пользовательские веб-интерфейсы **PayForm** не подразумевают использование учетных записей и ролей. Управление ПО **PayForm** осуществляется стандартными средствами ОС среды функционирования **PayForm**. В среде функционирования ПО **PayForm** на уровне ОС сервера рекомендуется реализовать систему дискреционного управления доступом.

#### Политика учетных записей:

У администратора **PayForm** для доступа к системе создается персонализированная учетная запись. В рамках документированных политик безопасности системы **PayForm** указано, что запрещено использование групповых учетных записей и паролей.

Перечень учетных записей:

- администратор ОС
- копирование конфигурационных файлов
- обновление приложения (деплой)
- взаимодействие между системами мониторинга (не обязательная)

Ко всем паролям предъявлены следующие требования парольных политик:

1) минимальная длина пароля должна составлять не менее 8 символов;



- 2) в числе символов пароля обязательно должны присутствовать строчные и прописные буквы, цифры и (или) специальные символы (@, #, \$, &, \*, % и т.п.);
- пароль не должен включать в себя легко вычисляемые сочетания символов (имена, фамилии, наименования рабочих станций и т.д.), а также общепринятые сокращения и термины (qwerty, pa\$\$w0rd, и т.п.);
- 4) при смене пароля новый пароль должен отличаться от старого не менее, чем двумя символами.
- 5) пароли учетных записей хранятся в файловой системе ОС в хэшированном виде алгоритм sha512

В среде функционирования ПО **PayForm** запрещено создание технологических учетных записей со стандартными паролями и иными механизмами аутентификации, использующими стандартный секрет для аутентификации, задаваемыми автоматически при установке программного обеспечения, на организационном уровне.

Для запуска приложений, являющихся компонентами **PayForm**, не требуется прав администратора OC.

## Установка

Описывается Ручной способ установки с базовыми конфигурациями.

ПО **Сервис приема платежей PayForm** обладает конфигурируемостью (масштабируется).

Специалист, разворачивающий данное ПО, должен следовать базовой настройке, но может подстраивать под свою инфраструктуру, выбранную для программной среды.

Сценарий установки (развертывания) комплекта ПО **Сервис приема платежей РауForm** в промышленной среде содержит следующие логические шаги:

- 1. Подготовить программную среду для установки (оборудование, ОС, драйвера).
- 2. Скопировать файлы ПО **PayForm**. Для установки в промышленной среде сборка формируется из Git, в которой программный код проходит обфускацию.
- 3. Прописать настройки конфигураций.
- 4. Запуск приложения. Проверка на наличие ошибок. При необходимости, настраивается логирование и производится проверка чтения логов.
- 5. Настройка Сетевых доступов.

#### Комплект установки

Архив с исходными файлами ПО PayForm для установки<sup>1</sup> payrofm.zip

<sup>&</sup>lt;sup>1</sup> Инструкция получения файлов экземпляра ПО находится в этом документе в разделе <u>Пререквизиты</u>



### Локальная установка на node.js

- 1. Установите Node.js: скачайте и установите Node.js с официального сайта <u>https://nodejs.org/en/download</u>. Следуйте инструкциям для вашей операционной системы.
- 2. Скачайте и извлеките файлы программы **PayForm** из архива, полученного по ссылке из заявления.
- 3. Создайте директорию для вашего веб-приложения. Скопируйте в неё файлы **PayForm**, распакованные на предыдущем шаге.
- 4. Инициализируйте проект: в командной строке перейдите в созданную на предыдущем шаге директорию и выполните команду npm install
- 5. В файле vite.config.js в строке №13 пропишите порт, на котором будет запущен локальный сервер, например port: 4010
- 6. Для запуска приложения на локальном сервере выполните npm run dev
- 7. Проверьте работу приложения: откройте веб-браузер и перейдите по адресу http://localhost:<порт>

, где <порт> - это порт, на котором запущен локальный сервер.

Для выполнения команд и навигации в Node.js можно воспользоваться терминалом IDE.

### Дефолтный конфиг продукта

Конфигурационные файлы:

- 1. \vite.config.js
- 2. \configs\<домен PayForm>\config.js
- 3. \configs\<домен PayForm>\preview.config.json

```
\vite.config.js
import { defineConfig,
                           loadEnv,
                                     splitVendorChunkPlugin } from 'vite';
import
                   react
                                     from
                                                       '@vitejs/plugin-react';
                                    from
                 basicSsl
                                                   '@vitejs/plugin-basic-ssl';
import
                                             }
import
                 {
                            resolve
                                                        from
                                                                       'path';
               default
                                          mode
export
                               ({
                                                       })
                                                                  =>
                                                                             {
    process.env = { ...process.env,
                                         ...loadEnv(mode, process.cwd())
                                                                            };
    return
                                                                defineConfig({
        root:
                                                                         './',
        server:
                                                                    '0.0.0.0',
            host:
```



```
port:
                                                                             4010,
             https:
                                                                              true
        },
        plugins:
                                                                                  ſ
             basicSsl(),
             react(),
             splitVendorChunkPlugin()
        ],
        build:
                                                                                  {
             manifest:
                                                                             true,
             outDir:
                                                                         './dist',
             sourcemap:
                                    mode
                                                                     'production',
                                                     !==
             rollupOptions:
                                                                                  {
                 input:
                                                                                  {
                                     resolve(__dirname,
                                                                    'index.html'),
                     main:
                                      resolve(__dirname,
                                                                   'version.html')
                     version:
                 },
             },
        },
        define:
                                                                                  {
             'import.meta.env.APP_VERSION':
JSON.stringify(process.env.npm_package_version),
        }
    });
};
Настроить нужно только строчку с портом<sup>2</sup>
```

port: 4010

\configs\<домен PayForm>\config.js

var		config	=	{
	cabinetUrl	:	"https://[cabinetU	RL]",
	cabinetLeg	alUrl:	"https://[cabinetLegalU	RL]",
	cabinetUrl	Offline:	"https://[offlineCabinetU	RL]",
	acqUrl:	["https://[acqURL]",	"https://[offlineAcqUR	L]"],
	yandexCoun	ter:	"[yandexCount	er]",
	acqApplePa	yToken:	"[acqApplePayTok	en]",
	currencies	:		{
	RUB:			"₽" <b>,</b>
	KZT:			<b>"⊤</b> ",
	USD:			"\$",

<sup>2</sup> См. раздел <u>Локальная установка на node.js</u>



"€",	EUR:
"£",	GPB:
"c",	KGS:
"So'm",	UZS:
"仇",	AZN:
, "۵"	GEL:
"も",	TRY:
"¥",	CNY:
" <b>4</b> ",	AMD:
	},
"bisysru",	footerLogoSvg:
"https://bisys.ru",	footerLogoLink:
"https://bisys.ru/docs/oferta",	offerLink:
"",	invoiceUrl:
["serviceCode"],	hideErrorBtnServices:
false,	receiptButtonVisibility:

```
};
```

#### Пояснение к значениям конфигов:

cabinetUrl: "<u>https://[cabinetURL</u>]", адрес апи личного кабинета для получения логотипа торговой точки. По умолчанию загружается предустановленный логотип категории услуги.

cabinetLegalUrl: "<u>https://[cabinetLegalURL</u>]", ссылка для юридической информации об основании приема оплаты в пользу торговой точки.

cabinetUrlOffline: "<u>https://[offlineCabinetURL</u>]", адрес апи личного кабинета из Зелёной зоны для получения логотипа торговой точки. По умолчанию загружается предустановленный логотип категории услуги.

acqUrl: ["https://[acqURL]", "https://[offlineAcqURL]"], адрес для получения Postmessage от приложения эквайринга <u>"Универсальная платформа</u> эквайринга ACQ4all" для проведения операций по банковской карте.

yandexCounter: "[yandexCounter]", идентификатор для сервиса Яндекс.Метрика.

acqApplePayToken: "[acqApplePayToken]", токен для проведения операций по ApplePay. По умолчанию отключен.

\configs\<домен PayForm>\preview.config.json

```
{
  "favicon":
                                                   "favicon_bisysru.png",
  "title":
            "оплатить
                        онлайн
                                 Bce
                                           платежи
                                                     вовремя с
                                                                  bisys",
  "siteName":
                          "Платежная
                                                форма
                                                                  bisys",
                                            "https://payframe.bisys.ru/",
  "mainUrl":
  "apiUrl":
                                                "https://[PayFormApiURL]"
```



#### }

#### Пояснение к значениям конфигов:

"favicon": "favicon bisysru.png", файл иконки для фавикона.

"title": "оплатить онлайн | Все платежи вовремя с bisys", значение для метатега title.

"siteName": "Платежная форма bisys", название сайта в метатеге свойств для подстановки в превью.

"mainUrl": "https://[PayFormDomain]/", для формирования адреса сайта в метатеге свойств для подстановки в превью.

"apiUrl": "https://[PayFormApiURL]" адрес для API бэкенда PayForm

## Обновление

### Процедуры получения и установки обновления

Передача собранного релиза производится средствами системы управления версиями Git и устанавливается на сервере с ПО PayForm специалистом с компетенциями в области информационной инфраструктуры согласно внутреннего регламента.

В зависимости от типа обновления может потребоваться дополнительная настройка конфигурационных файлов, влияющих на функционирование или безопасность программного обеспечения.

Для поддержки обновления промышленного модуля экземпляра ПО без прерывания его работы рекомендуется использование архитектуры, в которой все сервисы имеют дублирующие компоненты.

- 1. Готовится сборка.
- 2. В сборку разработчик вкладывает обновленный файл version.html
- 3. Действия специалиста, назначенного на обновление:
  - 3.1. Скачать дистрибутив с ПО
  - 3.2. Проверить контрольную сумму
  - 3.3. Создать новый каталог **www1** в репозитории приложения
  - 3.4. Файлы новой версии копируются в каталог www1
  - 3.5. Для каждого инстанса фронт отключается апстрим на всех параллельных нодах, кроме одной. Соответственно, трафик идет только на одну ноду.
  - 3.6. Вначале на каталог с новой версией **www1** направляется одна из нод, на которой остался трафик
  - 3.7. Убедиться по логам, что версия задеплоилась и нет ошибок
- 4. Если мониторинг стабильно без критических ошибок
  - Заливаем обновление в **каталог www**



- Переключаем ноду, используемую для проверки, на основной каталог www
- Включаем апстримы остальных нод

#### Проверка сборки

- 1. Проверить текущую версию сборки в проде https://payframe.ckassa.ru/version.html
- 2. Проверить наличие критических ошибок в консоли браузера
- 3. Проверить наличие в сборке файла checkout.js. Если нет взять с бэкапа.
- 4. Проверить настройку вызова формы с index.php, если вызов формы идет с index.html изменить.
- 5. Мониторить трафик оплаты с ПФ

### Откат

Если наблюдаются ошибки в приложении, то следует произвести откат обновления. Действия, в зависимости от этапа обновления, когда обнаружены ошибки, представлены ниже.

В процессе проверки на одной из нод:

- в работу подключаются ноды с предыдущей версией приложения
- снимается трафик с ноды с новой версией приложения
- на ноде с обновлением откатывая предыдущая рабочая версия приложения
- нода возвращается в эксплуатацию

После раскатки на все ноды:

- снимается трафик с одной из нод
- на ноде откатывая предыдущая рабочая версия приложения
- нода возвращается в эксплуатацию
- рабочий трафик направляется на эту ноду
- остальные параллельные ноды отключаются и откатываются обновления приложения
- все ноды возвращаются в эксплуатацию