

XSQUARE-MAIL

XSQUARE-MAIL (**XMAIL**) - это веб-сервис для отправки email по протоколу SMTP с помощью HTTP-запросов в формате JSON.

Для запуска и функционирования XMAIL необходимо инициализировать конфигурацию сервера одним из 3-х возможных способов:

- С помощью переменных окружения ENV
- С помощью ключей командной строки
- С помощью конфигурационного файла **config.json**

ПАРАМЕТРЫ КОНФИГУРАЦИИ

Пример файла конфигурации config.json

```
{
  "app": {
    "port": 8889,
    "debug": true
  },
  "smtp": {
    "host": "smtp.mail.ru",
    "port": 465,
    "email": "example@mail.ru",
    "password": "94ec390fe1",
    "skip_validate_recipients": false
  }
}
```

- **Port** – число, номер сетевого порта, на котором будет запущен сервис (по умолчанию - 9000)
- **Debug** - логическое значение. Включает режим отладки, при котором доступен подробный лог обработки запросов. Значение по умолчанию - false.
- **SMTP host** – строка, имя или IP-адрес SMTP сервера
- **SMTP port** – число, порт SMTP сервера
- **SMTP email** – строка, учетная запись на сервере для отправки почты
- **SMTP password** – строка, пароль от учетной записи на сервере для отправки почты
- **SMTP verify_cert** - логическое значение. Включает проверку сертификата. Значение по умолчанию - false.
- **SMTP from** - строка. Адрес FROM по умолчанию.
- **SMTP skip_validate_recipients** - логическое значение. Управляет предварительной проверкой доменов получателей через DNS (MX-записи) перед отправкой. Значение по умолчанию - false, означает, что проверка будет проводиться.

Далее в таблице приведено соответствие ключам командной строки и переменным окружения.

Параметр	Название переменной окружения	Ключ командной строки	Config.json
Port	PORT	-port	app.port

Параметр	Название переменной окружения	Ключ командной строки	Config.json
Debug	DEBUG	-debug	app.debug
SMTP host	SMTP_HOST	-smtpHost	smtp.host
SMTP port	SMTP_PORT	-smtpPort	smtp.port
SMTP email	SMTP_EMAIL	-smtpEmail	smtp.email
SMTP password	SMTP_PASSWORD	-smtpPassword	smtp.password
SMTP verify_cert	SMTP_VERIFY_CERT	-smtpVerifyCert	smtp.verify_cert
SMTP from	SMTP_FROM	-smtpFrom	smtp.from
SMTP skip_validate_recipients	SMTP_SKIP_VALIDATE_RECIPIENTS	-smtpSkipValidateRecipients	smtp.skip_validate_recipients

ENDPOINTS

В сервисе доступно 2 эндпоинта:

- **/send** – для отправки сообщения через метод POST
- **/status** – для проверки состояния сервиса через метод GET

ОТПРАВКА СООБЩЕНИЯ

Отправка сообщения происходит с помощью HTTP POST запроса в теле которого передан JSON с параметрами отправляемого письма.

Например:

```
curl -X POST http://localhost:9000/send -H "Content-Type: application/json" -d '{
  "From": "test@xsquare.ru",
  "Recipients": {
    "To": ["name@xsquare.ru"],
    "Cc": ["name1@xsquare.ru"],
    "Bcc": ["name2@xsquare.ru"],
  },
  "Subject": "Test Email subj",
  "ContentType": "text/plain",
  "Body": "Hello from XMAIL",
  "Headers": {
    "X-Custom-Header": "CustomValue",
    "Reply-To": "replyto@xsquare.ru"
  },
  "Attachments": {
    "file1.txt": "aGVsbG8geHNxdWFyZQ=="
  }
}
```

- **From** – строка, email отправителя
- **Recipients.To** – массив строк, список получателей
- **Recipients.Cc** – массив строк, список получателей копии письма
- **Recipients.Bcc** – массив строк, список скрытых получателей копии письма
- **Subject** – строка, тема письма
- **ContentType** – строка, формат тела письма. Возможные значения:

- text/plain – обычный текст
- text/html – письмо в формате HTML
- Body – строка, содержимое письма
- Headers – словарь, заголовки письма в формате: имя-значение.
- Attachments – словарь, вложения письма в формате: "имя файла – его содержимое в base64".

В ответ сервис возвращает код 200 в случае успеха или код ошибки и ее описание – в случае проблемы при отправке.

ОТПРАВКА БЕЗ АУТЕНТИФИКАЦИИ

Для взаимодействия с почтовым сервером без аутентификации необходимо указать пустые smtp.email и smtp.password, либо удалить эти параметры из конфига.

Например:

```
{
  "app": {
    "port": 9000
  },
  "smtp": {
    "host": "10.150.117.48",
    "port": 25
  }
}
```

ОТПРАВКА НА GOOGLE, YANDEX, MAIL.RU

Большая часть почтовых сервисов работает по o2auth, поэтому необходимо получить токен для почтового клиента и указать его в поле smtp.password

Например:

```
"smtp": {
  "host": "smtp.mail.ru",
  "port": 465,
  "email": "email@mail.ru",
  "password": "94df390fe15b780d90b5ef8198a5a5c79892b50937363830"
}
```

Для yandex необходимо открыть раздел «Почтовые программы» <https://mail.yandex.ru/?dpda=yes#setup/client>

Обязательно выберите опции Разрешить доступ к почтовому ящику с помощью почтовых клиентов → С сервера imap.yandex.ru по протоколу IMAP и Пароли приложений и OAuth-токены.

ВАЛИДАЦИЯ АДРЕСОВ И СТАТУС ДОСТАВКИ

Предварительная проверка доменов (DNS)

По умолчанию XMAIL выполняет проверку доменов всех получателей (To, Cc, Bcc) через DNS-запрос MX-записей до установления SMTP-сессии. Это позволяет:

- Мгновенно отсеивать опечатки и заведомо несуществующие домены (например, user@gmail.ru, user@yandex.r);
- Возвращать клиенту понятную ошибку 400 Bad Request без обращения к SMTP-шлюзу;
- Снижать нагрузку на почтовый сервер и экономить ресурсы на установке соединения и передаче тела письма.

Результаты DNS-запросов кэшируются в памяти сервиса на 4 часа, поэтому при массовой рассылке по одному и тому же домену повторные запросы к DNS не выполняются. Если DNS-сервер недоступен или не отвечает в пределах таймаута (3 секунды), сервис делает так: письмо не блокируется и передаётся на SMTP-шлюз для стандартной обработки. Блокировка происходит только при явно несуществующем домене (NXDOMAIN) или отсутствии MX-записей.

Отключение валидации

Для сценариев массовой рассылки, где критична максимальная скорость обработки запросов, предварительную проверку можно отключить:

```
{
  "smtp": {
    "host": "smtp.mail.ru",
    "port": 465,
    "email": "example@mail.ru",
    "password": "94ec390fe1",
    "skip_validate_recipients": true
  }
}
```

Статус доставки и DSN

Важно понимать, что протокол SMTP работает **асинхронно**. HTTP-ответ `200 OK` от XMAIL означает лишь то, что ваш SMTP-шлюз (например, корпоративный Exchange или почтовый провайдер) **успешно принял письмо в очередь на отправку**. Это не является гарантией того, что письмо физически доставлено в почтовый ящик конечного получателя.

XMAIL по умолчанию запрашивает у SMTP-сервера **DSN (Delivery Status Notifications)**. Если сервер получателя отклонит письмо (например, из-за переполненного ящика, несуществующего адреса или спам-фильтров), он должен отправить служебное письмо (Bounce/отбивку) на адрес, указанный в поле `From` (или `Return-Path`). Для отслеживания факта недоставки (Hard Bounce) на стороне вашей инфраструктуры необходимо настроить парсинг этих отбивок или использовать Webhooks транзакционных почтовых сервисов.

ПРИМЕЧАНИЕ: крупные почтовые провайдеры (Gmail, Mail.ru, Yandex, Microsoft) в целях защиты от спама часто игнорируют запросы DSN об успешной доставке, а корпоративные серверы Exchange могут подавлять генерацию отбивок политиками Anti-Backscatter. Предварительная DNS-валидация не определяет несуществующих пользователей внутри существующих доменов (например, `fake.user.12345@gmail.com`) — для таких случаев

рекомендуется использовать специализированные сервисы валидации или Webhooks ESP-провайдеров.

USER-AGENT и X-MAILER

По умолчанию в заголовки USER-AGENT и X-MAILER записывается значение XMAIL_version.

При необходимости можно указать кастомные заголовки

```
"Headers": {
  "X-Mailer": "CustomValue",
  "User-Agent": "CustomValue"
},
```

ОТПРАВКА СОБЫТИЯ КАЛЕНДАРЯ

К письму также можно прикрепить событие календаря в формате iCalendar (файл *.ics) Сам файл calendar.ics создается сервисом на основе запроса.

Например, минимальный запрос

```
{
  "From": "test@xsquare.ru",
  "Recipients": { "To": ["recipient1@xsquare.ru"] },
  "Subject": "Встреча: Проект X",
  "ContentType": "text/html",
  "Body": "<p>Приглашаем на встречу.</p>",
  "Calendar": {
    "Event": {
      "Summary": "Тестовая встреча по проекту X",
      "StartTime": "10.05.2026 11:00:00",
      "EndTime": "10.05.2026 12:00:00"
    }
  }
}
```

Структура Event

Поле (JSON Key)	Тип	Обязательно?	Описание	Пример
Summary	string	Да	Заголовок/тема встречи	"Еженедельный синк"
Description	string	Нет	Подробное описание повестки. Поддерживает HTML/текст	"Обсудим прогресс по спринту #12."
Location	string	Нет	Место проведения или ссылка на ВКС	"Переговорная А" / "https://zoom.us/j/..."
StartTime	string	Да	Дата и время начала.	"2026-05-12T14:00:00+03:00" или "12.05.2026 14:00:00"
EndTime	string	Да	Дата и время окончания. Должно быть > startTime	"2026-05-12T15:30:00+03:00"
TimeZone	string	Нет	Часовой пояс IANA. Применяется, если в startTime / endTime нет явного смещения	"Europe/Moscow" , "UTC"

Поле (JSON Key)	Тип	Обязательно?	Описание	Пример
Organizer	object	Нет	Данные организатора. Поля: Email (обяз.), Name (опц.) Если не указано, то будет взято из поля From основного запроса	{"Email": "boss@company.ru"}
Attendees	array	Нет	Список участников. Массив объектов: Email (обяз.), Name (опц.), RSVP (опц.) Если не указано, то будет взято из поля Recipients. То основного запроса	[{"Email": "ivan@co.ru", "RSVP": true}]
UID	string	△ Авто	Глобальный ID события (RFC 5545). Если пустой → генерируется автоматически	"evt-2026-05-12-001@xsquare.ru"
Sequence	int	Нет	Версия события. 0 для нового, увеличивается при обновлении/отмене	0, 1
Reminder	object	Нет	Напоминание. Поля: Trigger (смещение, напр. "-PT15M"), Action ("DISPLAY" / "EMAIL"), Description	{"Trigger": "-PT15M", "Action": "DISPLAY"}
Url	string	Нет	Ссылка на материалы, повестку или комнату встречи	"https://docs.company.com/agenda"

Важно по времени:

Если в StartTime / EndTime указан явный часовой пояс (например, +03:00 или Z), поле TimeZone игнорируется.

Если формат «наивный» (например, "12.05.2026 14:00:00"), время автоматически привязывается к TimeZone или к дефолтному поясу сервера (Europe/Moscow).

Вложенные структуры

```
"Organizer": { "Email": "string", "Name": "string?" }
"Attendees[]": { "Email": "string", "Name": "string?", "RSVP": "bool?" }
"Reminder": { "Trigger": "string?", "Action": "string?", "Description": "string?" }
```

Расширенный пример запроса:

```
{
  "From": "test@xsquare.ru",
  "Recipients": { "To": ["recipient1@xsquare.ru"] },
  "Subject": "Встреча: Проект X",
  "ContentType": "text/html",
  "Body": "<p>Приглашаем на встречу.</p>",
  "Calendar": {
    "Event": {
      "Summary": "Тестовая встреча по проекту X",
      "Description": "Тестовое описание",
      "Location": "Онлайн",
      "StartTime": "12.05.2026 14:00:00",
      "EndTime": "2026-05-12T15:00:00+03:00",
      "Organizer": { "Email": "info@xsquare.ru" },
      "Attendees": [
        {
          "Email": "recipient1@xsquare.ru",
          "RSVP": false
        }
      ],
    }
  }
}
```

```
{
  "Email": "recipient2@xsquare.ru",
  "RSVP": true
}
],
"UID": "proj-x-meeting-008@xsquare.ru",
"Reminder": {
  "Trigger": "-PT15M",
  "Action": "DISPLAY",
  "Description": "Напоминание: встреча через 15 минут"
}
}
}
```

ОТВЕТ СЕРВИСА

Сервис возвращает ответ в виде JSON. HTTP-статус ответа зависит от результата обработки запроса:

- **200 OK** — Письмо успешно сформировано и передано на SMTP-шлюз. (Гарантирует только факт отправки, но не факт доставки в ящик получателя, см. раздел *Валидация адресов и статус доставки*).
- **400 Bad Request** — Ошибка валидации входных данных. Возвращается при некорректном JSON, а также если DNS-проверка (включена по умолчанию) выявила несуществующий домен получателя.
- **500 Internal Server Error** — Внутренняя ошибка сервиса, проблемы с сетью, таймауты или отказ SMTP-сервера принимать письмо (ошибки авторизации, TLS и т.д.).

Пример успешного ответа:

```
{
  "status": "success",
  "message": "mail sent successfully",
  "message_id": "3dlEFw_GqlTI_RYyPwzW0.@helper",
  "event_id": "3a6f95d3-6902-4ab5-b787-e8cfb7c331f8"
}
```

Пример ответа с ошибкой валидации (400 Bad Request):

```
{
  "status": "error",
  "message": "Invalid recipient address provided",
  "message_id": "",
  "event_id": "",
  "error": "recipient validation failed: domain 'gmail.ru' does not exist (NXDOMAIN)"
}
```

Message-ID

Свойство Message-ID генерируется автоматически и возвращается в ответе, но при необходимости его также можно передать в запросе:

```
"Headers": {  
  "Message-ID": "<CustomValue@xsquare.ru>"  
}
```

UID события календаря (event_id)

Если поле UID не указано в запросе, то идентификатор события будет сгенерирован сервисом и возвращен в ответе, в поле event_id