

Создание мини-аппа

Инструкция администраторов

Оглавление

Назначение документа	3
Дополнительная документация	3
Шаг 1. Создайте приложение	4
Шаг 2. Создайте сервер	5
Шаг 3. Зарегистрируйте домен	8
Шаг 4. Сгенерируйте самоподписанный сертификат	9
Шаг 5. Зарегистрируйте мини-апп в Метаботе	11

Назначение документа

В данном документе описано создание мини-аппа с использованием библиотеки VK Teams Bridge.

Чтобы создать мини-апп, вам потребуется:

1. Приложение с подключенной библиотекой VK Teams Bridge. Для успешной загрузки приложения достаточно одной HTML-страницы с подключенной библиотекой.
2. Собственный сервер, который будет раздавать статический контент приложения через Nginx с самоподписанным сертификатом.
3. Открытый домен, по которому будет доступно приложение.
4. Самоподписанный сертификат для вашего домена.
5. Доступ к Супераппу VK WorkSpace, чтобы зарегистрировать приложение.

Подробное описание создания мини-аппа представлено ниже.

Документ предназначен для использования администраторами организации.

Дополнительная документация

[Общая информация](#) — в документе описано, что такое мини-аппы, и как они выглядят в приложении.

[Техническое описание](#) — в документе описаны принципы разработки мини-аппа, выполнение запросов, авторизация и отладка мини-аппа.

[Библиотека VK Teams Bridge](#) — в документе описаны методы взаимодействия мини-аппа с нативным клиентом приложения.

[Регистрация мини-аппа](#) — в документе описана регистрация мини-аппа при помощи Метабота приложения VK WorkSpace.

[Публикация мини-аппа в панель сервисов Супераппа VK WorkSpace](#) — в документе описаны способы публикации мини-аппа для отображения в панели сервисов приложения.

[FAQ](#) — в разделе собраны часты вопросы при работе с мини-аппами.

Архитектура и описание системы — в документе представлена информация о сервисах Супераппа, обеспечивающих функциональность мини-аппов. Не является частью публичной документации, обратитесь к представителю VK Tech, чтобы ознакомиться с документом.

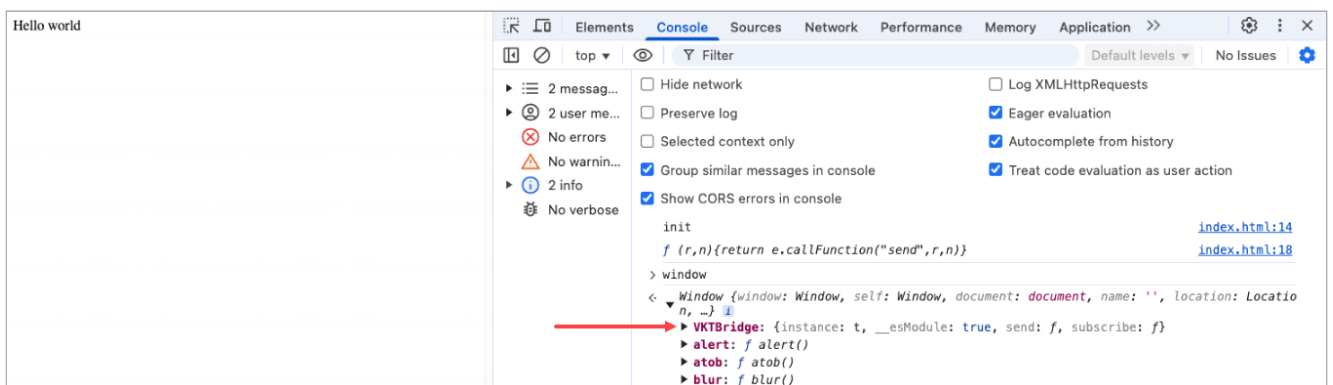
Шаг 1. Создайте приложение

1. Создайте директорию для проекта и скопируйте в нее [файл с библиотекой VK Teams Bridge](#).

2. Создайте в директории файл index.html со следующим содержимым:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>
  <div>Hello world</div>
</body>
<script>
  console.log('init')
</script>
<script src="./package/dist/iife/index.min.js"></script> # В теге src укажите путь до
директории с файлом библиотеки VK Teams Bridge
<script>
  window.VKTBridge.send('LoadingCompleted', { ok: true })
    .then((bridgeResponse) => {
      console.log({ bridgeResponse });
    })
    .catch((bridgeError) => {
      console.log({ bridgeError });
    });
</script>
</html>
```

3. Откройте в браузере index.html и убедитесь, что на странице есть Hello world. Далее перейдите в консоль разработчика и убедитесь, что нет ошибок подключения библиотеки, и в объекте Window есть VK Teams Bridge:



Шаг 2. Создайте сервер

Для подключения мини-аппа нужен собственный сервер, который будет раздавать статический контент приложения. На этот сервер нужно загрузить собранное приложение и самоподписанный сертификат для домена. Затем необходимо установить на сервере Nginx и настроить его конфигурацию так, чтобы он раздавал контент и использовал сертификат.

Далее в качестве примера мы рассмотрим создание сервера на мощностях VK Cloud. Если у вас уже есть собственный сервер, вы можете использовать его для раздачи статического контента.

1. Зарегистрируйтесь и авторизуйтесь в VK Cloud— <https://cloud.vk.com/>
2. Перейдите в раздел **Облачные вычисления** и нажмите на кнопку **Создать инстанс**.
3. В отобразившемся окне создайте необходимую конфигурацию сервера. Вы можете выбрать любую операционную систему, но далее в инструкции в качестве примера все команды будут для Ubuntu 20.04.
4. На следующем шаге оставьте настройки по-умолчанию кроме пункта **Настройки Firewall**. Нажмите на **Добавить группу безопасности** и выберите **Всё разрешено**.
5. Далее нажмите на кнопку **Создать инстанс**.

Если вы создаете виртуальную машину впервые, то на ваш компьютер скачается файл с ключом для подключения к машине через SSH. Название будет разным в зависимости от того, какие настройки были указаны на шаге выше. Например, название ключа для ОС Ubuntu 20.04 может быть `ubuntu-STD2-1-1-10GB-eNw7UeS1.pem`.

6. Проверьте, установлен ли SSH-клиент на вашем компьютере. Для этого выполните в терминале команду `ssh`. Если SSH-клиент установлен, в выводе команды будет:

```
usage: ssh [...] [-B bind_interface] [-b bind_address]
       [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
       [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
       [-J destination] [-L address] [-l login_name] [-m mac_spec]
       [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
       [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
       destination [command [argument ...]]
ssh [-Q query_option]
```

Если SSH-клиент не установлен, установите его на ваш компьютер.

7. После установки SSH-клиента и создания VM выполните в терминале команды:

```
chmod 400 <путь к SSH-ключу из шага выше> # устанавливаем ограничения на чтение ключа
ssh -i <путь к SSH-ключу> <название системы VM>@<IP-адрес VM> # подключаемся к виртуальной машине
```

Имя и IP-адрес виртуальной машины отображаются в таблице созданных VM.

Инструкция по подключению к VM также отображается в интерфейсе VK Cloud в разделе «Информация» для VM.

Примечание

Далее используются примеры команд для операционной системы Ubuntu 20.04.

8. Обновите пакеты Ubuntu:

```
sudo apt update
```

9. Установите Nginx командой:

```
sudo apt install nginx
```

10. Перейдите в директорию с Nginx:

```
cd ../etc/nginx/
```

далее в директорию **/var/www** — из нее забирается статический контент приложения:

```
cd ../../var/www/
```

и удалите в ней директорию с HTML:

```
sudo rm -r html
```

11. Перейдите в директорию с приложением и библиотекой VK Teams Bridge и загрузите их на сервер:

```
scp -i<путь к SSH-ключу> -r <директория с проектом> <имя ВМ>@<IP-адрес ВМ>:/var/www
```

Пример команды, если директория с проектом называется «test»:

```
sudo scp -i ../Downloads/ubuntu-STD2-1-1-10GB-eNw7UeS1.pem -r test ubuntu@212.233.91.193:~/
```

12. На виртуальной машине переместите папку с проектом в директорию **/var/www**:

```
sudo mv ~/<директория с проектом> /var/www/
```

13. Настройте Nginx, чтобы он раздавал статический контент приложения:

```
sudo nano ../../etc/nginx/sites-enabled/default
```

Найдите строку **root /var/www/html;** и замените ее на **root /var/www/<путь до index.html>;**

Пример строки, если репозиторий называется «test» и файл index.html лежит внутри еще одной папки «test» — **root /var/www/test/test**

Сохраните изменения.

14. Перезапустите сервер:

```
sudo service nginx restart
```

15. Введите в адресную строку браузера `http://<IP-адрес виртуальной машины>/`. Пример — `http://31.128.50.35/`

В браузере должна отображаться HTML-страница приложения.

Шаг 3. Зарегистрируйте домен

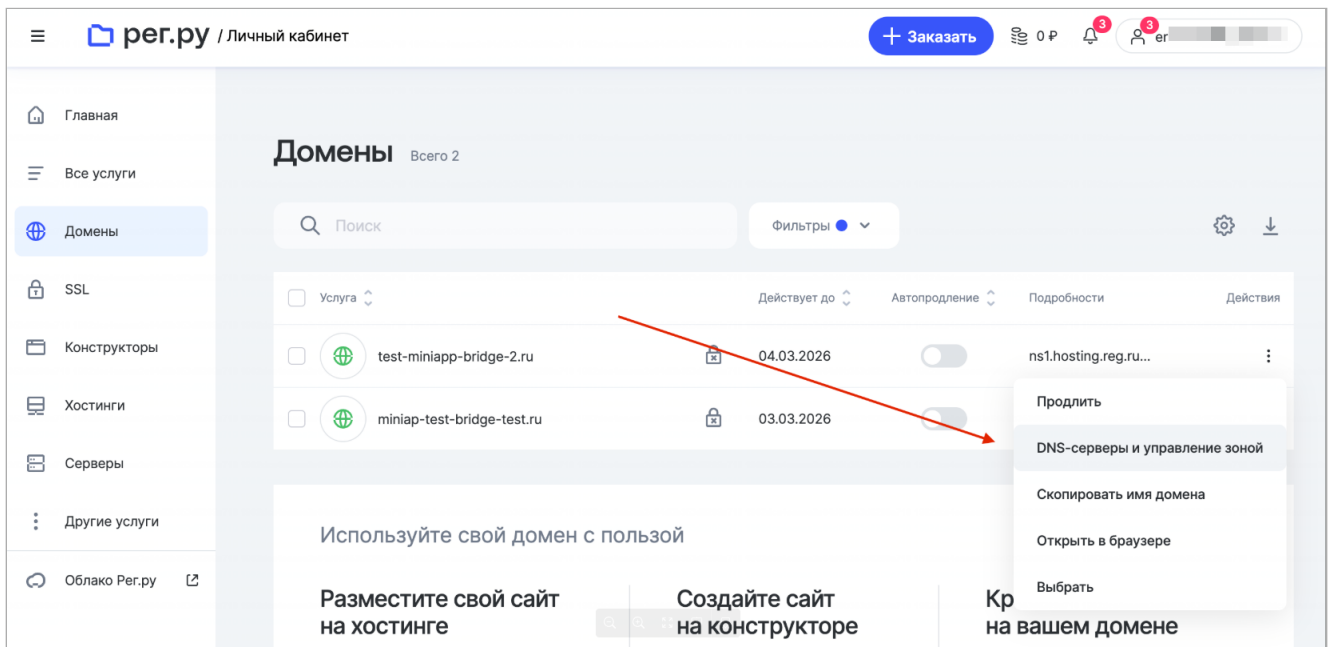
Для регистрации мини-аппа, вам потребуется собственный домен. Он должен быть открытым, чтобы Суперапп смог «забирать» приложение.

Вы можете использовать имеющийся у вас домен или зарегистрировать новый на одном из удобных для вас сервисов. Далее в качестве примера будет рассмотрена регистрация домена при помощи сервиса <https://www.reg.ru/>.

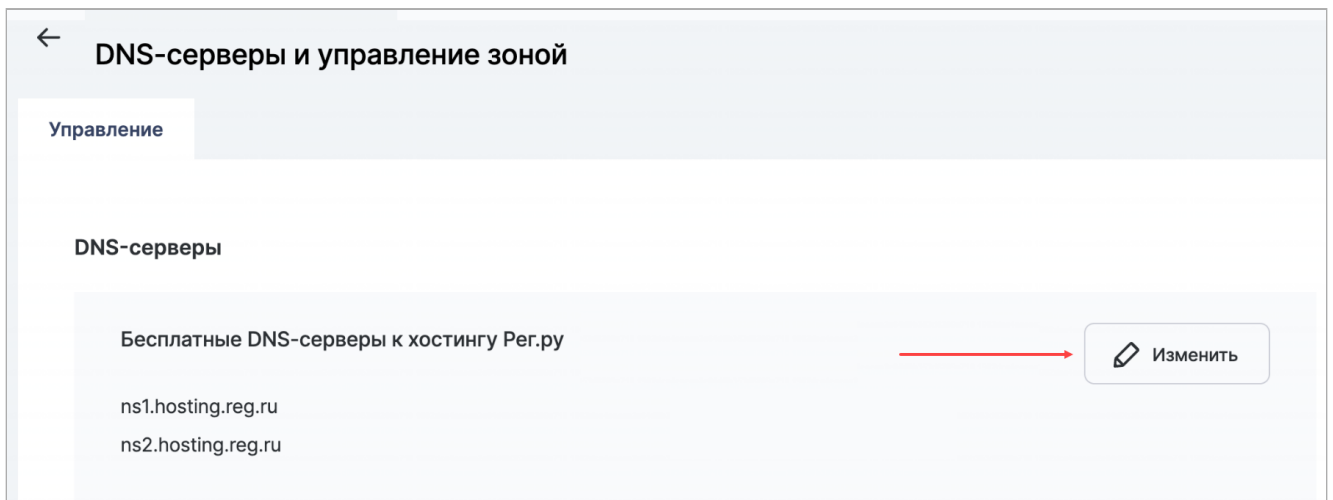
Зарегистрируйтесь на сайте <https://www.reg.ru/> и в верхнем меню нажмите на кнопку **Заказать**. Укажите домен, который хотите зарегистрировать. Выберите и оплатите его.

Далее настройте DNS-хостинг:

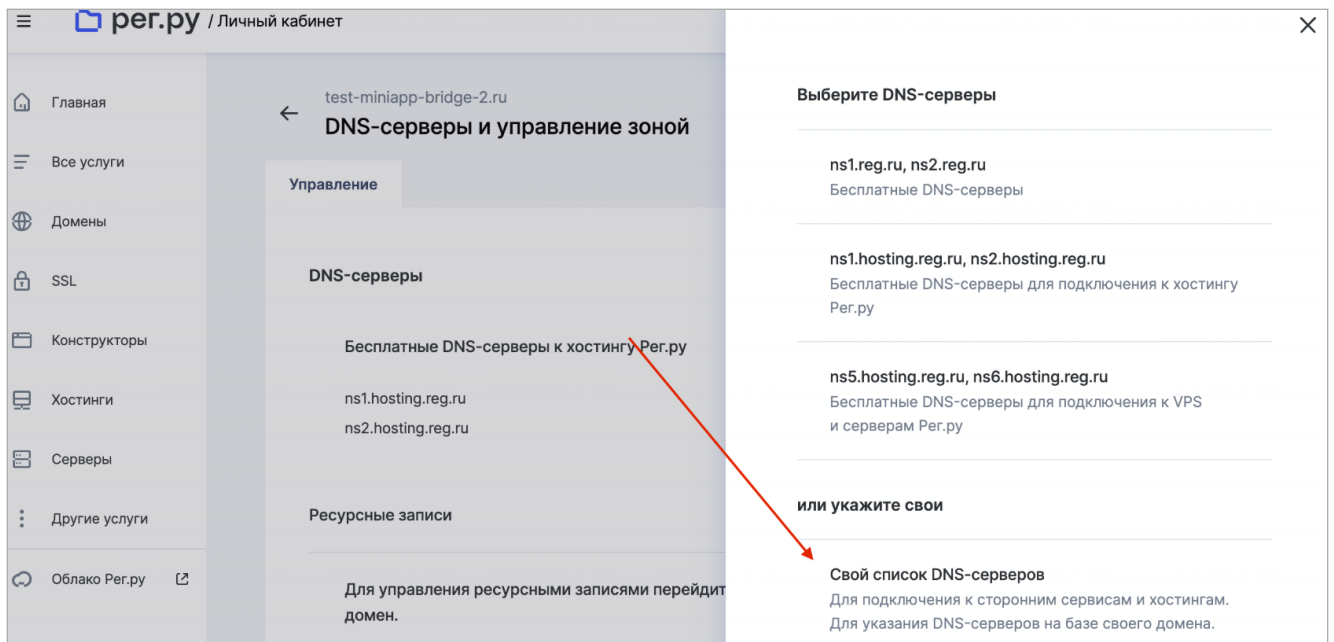
1. Делегируйте домен на сайте, где регистрировали домен. Перейдите на <https://www.reg.ru/> → **Личный кабинет** → **Домены** → **DNS серверы и управление зоной**:



2. В разделе **DNS серверы и управление зоной** нажмите **Изменить**:



В открывшемся меню нажмите **Свой список DNS серверов**:



Добавьте два DNS-сервера:

ns1.mcs.mail.ru

ns2.mcs.mail.ru

и нажмите **Продолжить**.

3. Свяжите домен с IP-адресом сервера в VK Cloud:

3.1. В разделе **DNS** нажмите на кнопку **Создать DNS-зону** — оставьте настройки по-умолчанию, в поле **DNS-зона** укажите зарегистрированный домен.

3.2. Нажмите на созданную DNS-зону. На отобразившейся странице добавьте запись — укажите IP-адрес виртуальной машины в VK Cloud и префикс к DNS. Нажмите на кнопку **Добавить запись**.

Подождите обновление DNS-серверов. После этого приложение будет доступно по зарегистрированному домену+префикс.

Шаг 4. Сгенерируйте самоподписанный сертификат

Взаимодействие между сервером Супераппа и веб-сервером мини-аппа осуществляется по mTLS. Поэтому веб-сервер мини-аппа должен использовать самоподписанный сертификат с соответствующим CN, который будет указываться при регистрации мини-аппа в приложении.

Для генерации сертификата должен быть установлен OpenSSL. Чтобы проверить установлен ли он, выполните в терминале команду `openssl -v`. Если OpenSSL не установлен, установите его.

1. Сгенерируйте сертификат командой:

```
openssl req -x509 -days 36500 -subj "/CN=HOST DOMAIN NAME" -newkey rsa:4096 -nodes -keyout key.pem -out crt.pem
```

где CN — префикс + зарегистрированный домен.

В результате выполнения команды в директории должны появиться файлы `key.pem` и `crt.pem`. Данные файлы необходимы для запуска веб-сервера мини-аппа. Файл `crt.pem` будет использоваться при регистрации мини-аппа в Метаботе на шаге 5.

2. Переместите сертификат и ключи на сервер командой:

```
scp -i<путь к SSH-ключу из шага 2> -r</путь к crt.pem> </путь к key.pem> <имя VM>@<IP-адрес VM>:~/
```

3. На виртуальной машине выполните команду:

```
sudo mv crt.pem key.pem /etc/ssl/private
```

4. Подключитесь к серверу и проверьте, что сертификат и ключи скопировались:

```
sudo bash #выполняем команды с привилегиями root-пользователя
cd /etc/ssl/private/ # переходим в директорию с ключами
ls -a # выводим список файлов в директории
exit # выходим из под root-пользователя
```

5. Измените конфигурацию Nginx.

Выполните команду:

```
sudo nano /etc/nginx/sites-enabled/default
```

и замените содержимое файла на код:

Внимание

Ниже в коде приведены в качестве примера тестовые данные. Укажите ваш домен и пути до сертификата и приватного ключа

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    server_name test-miniapp-bridge-2.ru; # укажите ваш домен

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl;

    server_name test-miniapp-bridge-2.ru; # укажите ваш домен

    # Указываем путь к сертификатам
    ssl_certificate /etc/ssl/private/crt.pem; # путь к сертификату
    ssl_certificate_key /etc/ssl/private/key.pem; # путь к приватному ключу

    # Настройки SSL
    ssl_protocols TLSv1.2 TLSv1.3; # используем только безопасные версии TLS
    ssl_ciphers 'HIGH:!aNULL:!MD5'; # безопасные шифры для SSL

    root /var/www/test/test; # путь к корневой директории приложения, в котором лежит
index.html
    index index.html index.htm index.nginx-debian.html;

    ssl_client_certificate /etc/ssl/private/crt.pem; # путь к сертификату, который будет
использоваться для проверки клиентских сертификатов

    location / {
        try_files $uri $uri/ =404; # ищем запрашиваемый файл
    }
}
```

Сохраните изменения.

6. Перезапустите Nginx командой:

```
sudo systemctl reload nginx
```

Перейдите на сайт. Браузер отобразит предупреждение о самоподписанном сертификате, но он позволит перейти, если согласиться с условиями.

Шаг 5. Зарегистрируйте мини-апп в Метаботе

Теперь нужно зарегистрировать мини-апп в VK WorkSpace через специального бота — [Metabot](#).

1. Перейдите в Суперапп VK WorkSpace и найдите Metabot.
2. Отправьте боту команду `/newapp` и следуйте инструкциям бота.

Для регистрации мини-аппа необходимы следующие атрибуты:

- Название мини-аппа.
- Описание мини-аппа.
- Картинка в формате PNG.

Требования к картинке:

- разрешение не мене 1024x1024px.
- квадрат.

[Скачать пример картинки](#)

- Иконка в формате SVG для невыбранного таба.

Требования:

- монохромная;
- без встроенных стилей и скриптов.

[Скачать пример иконки](#)

- Иконка в формате SVG для выбранного таба.

Требования:

- монохромная;
- без встроенных стилей и скриптов.

[Скачать пример иконки](#)

При успешном завершении регистрации Метабот отправляет сообщение со ссылкой на мини-апп, которой можно пользоваться и пересылать коллегам.

Ссылка на мини-апп содержит базовую часть, а также ID нового мини-аппа.

Пример ссылки на мини-апп:

<https://u.internal.myteam.mail.ru/miniapp/miniapp-16c6c83e-075f-4d92-b840-ad1991f3445c>

3. Добавьте URL-адрес на ваш домен и сгенерированный сертификат crt.pem. Нажмите на **Set backend URL and certificate** и отправьте боту эти данные.

Если все прошло успешно, бот выводит сообщение «The mini app backend info is updated». При нажатии на ссылку мини-аппа, он отображается в Супераппе.

 Технический писатель: Белова Ирина

 18 февраля 2026 г.