

Библиотека VK Teams Bridge

Инструкция для разработчиков

Оглавление

Назначение документа	4
Дополнительная документация	4
Общая информация	5
Использование библиотеки	5
Методы	5
Общие коды ошибок	5
Запросы метода Send	7
CreateChat	7
DownloadFile	7
GetAuth	8
GetLanguage	8
GetMiniAppShareLink	9
GetMiniappConfig	10
GetPlatformVersion	10
GetSelfId	11
GetThemeSettings	11
LoadingCompleted	12
OpenAuthModal	12
OpenContactSelectionDialog	14
OpenDialog	15
OpenLink	16
OpenPopUp	16
OpenProfile	17
OpenThread	18
SetBadge	18
SetButtonGroup	18
SetCanGoBack	20
SetCanNotGoBack	20
SetTitle	21

ShowToast	21
StartAudioCall	21
StartVideoCall	22
StorageClear	22
StorageDelete	22
StorageGet	23
StoragePut	23
Запросы метода Subscribe	25
ThemeSettings	25
Language	25
BackButtonPressed	25
ButtonPressed	26

Назначение документа

В данном документе описаны методы взаимодействия мини-аппа с нативным клиентом VK WorkSpace.

Документ предназначен для использования администраторами организации.

Дополнительная документация

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

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

[Создание мини-аппа](#) — в документе описано создание мини-аппа в VK WorkSpace с использованием библиотеки VK Teams Bridge.

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

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

[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.

Общая информация

Взаимодействие мини-аппа с нативным клиентом VK WorkSpace осуществляется через библиотеку VK Teams Bridge <https://www.npmjs.com/package/@instant-messengers/vk-teams-bridge>. Она отвечает за предоставление JavaScript-коду мини-аппа удобного интерфейса для вызова функций нативного клиента.

Библиотека позволяет обращаться к нативному клиенту за различными данными или действиями для плотной интеграции с продуктом.

Код библиотеки абстрагирован от протокола JS SDK и осуществляет лишь передачу запроса-ответа между мини-аппом и нативным приложением.

Использование библиотеки

```
import { Bridge } from '@instant-messengers/vk-teams-bridge';
const bridge = new Bridge()

useEffect(() => {
  bridge.send('LoadingCompleted', { ok: true }).then(() => {
    console.log('LoadingCompleted success');
  }).catch((err: any) => {
    console.error('LoadingCompletedError', err)
  });
}, [])
```

Методы

В библиотеке есть два метода:

1. Send — принимает название и параметры запроса.
2. Subscribe — принимает название подписки, функцию для обработки событий и параметры подписки.

Общие коды ошибок

Метод Send:

- BAD_REQUEST
- NETWORK_ERROR
- INTERNAL_ERROR

Метод Subscribe:

- UNKNOWN_SUBSCRIPTION_NAME

Запросы метода Send

CreateChat

Открывает экран для создания группового чата.

Входные данные

```
bridge.send('CreateChat', {
  "name": "Рабочая группа",
  "members": [...employee.emails]
})
.then((data) => {
  const { chatId, url } = data;
  // your code
})
.catch((err) => {
  console.error('CreateChatError', err)
});
```

- name: string — название чата.
- members: string[] — список участников чата. Содержит уникальные идентификаторы (sn/email) пользователей внутри сервиса.

Ответ

```
{
  "chatId": "123@chat.agent",
  "url": "https://u.vkteams-domain.team/profile/chat_id"
}
```

- chatId: string — идентификатор созданного чата.
- url: string — ссылка на чат.

Специфичные коды ошибок

- BAD_REQUEST — не выполнены ограничения на длину названия (по умолчанию — не более 255 символов) или количество участников чата (по умолчанию — не более 25000).
- MODAL_CLOSED — модальное окно было закрыто.
- CREATE_FAILED — ошибка создания чата. Чат не был создан из-за ошибок на стороне клиентского приложения, либо из-за внутренних ошибок на стороне сервера.

DownloadFile

Обрабатывает процедуру скачивания файла из мини-аппа на стороне клиентского приложения.

Входные данные

```
bridge.send('DownloadFile', {
  "url": "https://biz-test-kuber.hb.bizmrg.com/k8s-test/static/2025-02-27-07-46-23-116/
static/media/vk-tech-img.48cbe6c4.webp"
})
```

- url: string — ссылка на файл.

Специфичные коды ошибок

- BAD_REQUEST — ошибка валидации url-адреса (должен соответствовать общепринятому синтаксису согласно стандарту RFC 3986).

GetAuth

Позволяет получить silent token для использования внутри мини-аппа. Для успешного получения токена, в конфигурации мини-аппа должен быть параметр "needs_auth": true.

Входные данные

```
bridge.send('GetAuth', {})
.then(({ token_type, silent_token }) => {
  // your code
})
.catch((err) => {
  console.error('GetAuthError', err)
});
```

Ответ

```
{
  "token_type": "silent",
  "silent_token": "f328afaeaf81424390b193b1afba411413d04657ace"
}
```

- token_type: string — тип токена.
- silent_token: string — токен.

Специфичные коды ошибок

- 401 — Unauthorized.
- 403 — Forbidden.
- 500 — Server error.

GetLanguage

Позволяет мини-аппу узнать язык, выбранный в данный момент в клиентском приложении.

Входные данные

```
bridge.send('GetLanguage', {})  
.then(({ language }) => {  
  // your code  
})
```

Ответ

```
{  
  "language": "ru"  
}
```

- language: string — код языка в соответствии со стандартом [RFC5646](#).

GetMiniAppShareLink

Позволяет получить внутреннюю ссылку на объект мини-аппа по его ID.

Входные данные

```
bridge.send('GetMiniAppShareLink', {  
  "miniappId": "miniapp-0757f779-90e7-4579-add3-07ad98e660c3"  
})  
.then(({ url }) => {  
  // your code  
})
```

- miniappId: string — ID мини-аппа.

Ответ

```
{  
  "url": "https://u.vkteams-domain.team/miniapp/miniapp-0757f779-90e7-4579-  
add3-07ad98e660c3"  
}
```

- url: string — ссылка на мини-апп.

GetMiniappConfig

Позволяет получить конфигурационные параметра мини-аппа.

Входные данные

```
bridge.send('GetMiniappConfig', {})
.then((data) => {
  const { id, url, name, description, config } = data;
  // your code
})
.catch((err) => {
  console.error('GetMiniappConfigError', err)
});
```

Ответ

Ответ содержит уникальные для текущего мини-аппа конфигурационные параметры.

```
{
  "id": "miniapp-123",
  "url": "https://miniapp-123.sandbox.my-domain.ru",
  "name": "My miniapp",
  "description": "Miniapp description",
  "config": {
    "needs_auth": true
  }
}
```

GetPlatformVersion

Возвращает версию клиентского приложения, в котором запущен мини-апп.

Входные данные

```
bridge.send('GetPlatformVersion', {})
.then(({ platform, version }) => {
  // your code
})
```

Ответ

```
{
  "platform": "web",
  "version": "24.7.0"
}
```

- platform: string — одно из значений: web | desktop | mobile.
- version: string — версия клиентского приложения в формате YY.MM.MINOR.

GetSelfId

Позволяет получить email текущего пользователя.

Входные данные

```
bridge.send('GetSelfId', {})
  .then(({ userId }) => {
    // your code
  })
```

Ответ

```
{
  "userId": "usere@mycompany.ru"
}
```

- `userId`: string — sn текущего пользователя.

GetThemeSettings

Позволяет получить настройки темы оформления, выбранной в клиентском приложении.

Тема оформления представляет собой набор цветов, используемых в продукте, а также признак светлого/тёмного режима.

Входные данные

```
bridge.send('GetThemeSettings', {})
  .then(({ colorsScheme, colors }) => {
    const { colorTextPrimary, colorAccentBlue, colorBackground } = colors;
    // your code
  })
```

Ответ

```
{
  "colorsScheme": "light",
  "colors": {
    "colorTextPrimary": "#111111ff",
    "colorBackgroundContent": "#ffffff",
    "colorAccentBlue": "#1b85f1ff",
    "colorIconAccent": "#2d90f5ff",
    "colorBackgroundSecondary": "#f3f5f8ff",
    "colorIconSecondary": "#838690ff",
    "colorTextSecondary": "#838690ff",
    "colorSeparatorPrimaryAlpha": "rgba(0,0,0,0.05)",
    "colorBackground": "#f6f6f6ff",
    "colorSeparatorPrimary": "#cccfd8ff"
  }
}
```

- `colorsScheme`: 'light' | 'dark' — значение светлой/тёмной темы.

- colors: Record — содержит объект `ColorsDescriptionStruct` со значениями типа `ColorDescriptionStatic`.

LoadingCompleted

Информирование клиентского приложения об успешной загрузке мини-аппа. Является обязательным для успешного открытия мини-аппа в клиентском приложении.

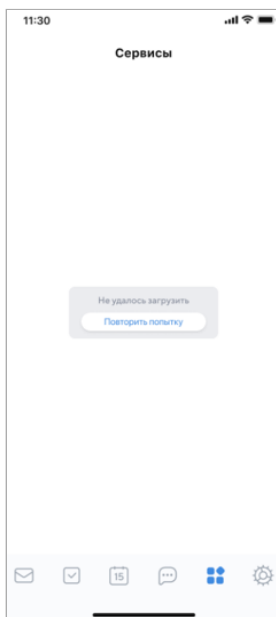
Входные данные

```
bridge.send('LoadingCompleted', {  
  "ok": true  
})
```

- ok: boolean — признак загрузки мини-аппа.

При передаче значения `ok: true` мини-апп успешно открывается.

При передаче значения `ok: false`, либо, если запрос `LoadingCompleted` не получен клиентским приложением в течение 10 секунд после открытия мини-аппа, отображается заглушка с текстом «Не удалось загрузить»:



Ответ

- ALREADY_CALLED — запрос выполнен дважды.

OpenAuthModal

Позволяет открыть браузер для авторизации пользователя во внешнем сервисе.

По окончании выполняется перенаправление на указанный в параметрах адрес для закрытия браузера и передачи информации мини-аппу.

Входные данные

```
bridge.send('OpenAuthModal', {
  "url": "https://u-ianferov.v3.im-sandbox.devmail.ru/test-redirect",
  "redirectUrlParamName": "redirect_url"
})
.then(({ querystring }) => {
  // your code
})
.catch((err) => {
  console.error('OpenAuthModalError', err)
});
```

- url: string — адрес, который будет открыт в браузере.
- redirectUrlParamName: string — имя, под которым в url будет добавлен адрес для возврата результата из браузера в мини-апп путём выполнения перенаправления на этот адрес.

Ответ

```
{
  "querystring": "https://u-ianferov.v3.im-sandbox.devmail.ru/test-redirect?
  redirect_url=https://webim.best-company.onprem.ru/auth.html"
}
```

- querystring: string — параметры, переданные при выполнении перенаправления (в место ответа будет открыта новая вкладка с переданным адресом).

Специфичные коды ошибок

- MODAL_CLOSED — модальное окно было закрыто пользователем.

OpenContactSelectionDialog

Позволяет открыть экран для выбора контактов. Экран содержит контакты из мессенджера. Выбранные контакты возвращаются в мини-апп.

Входные данные

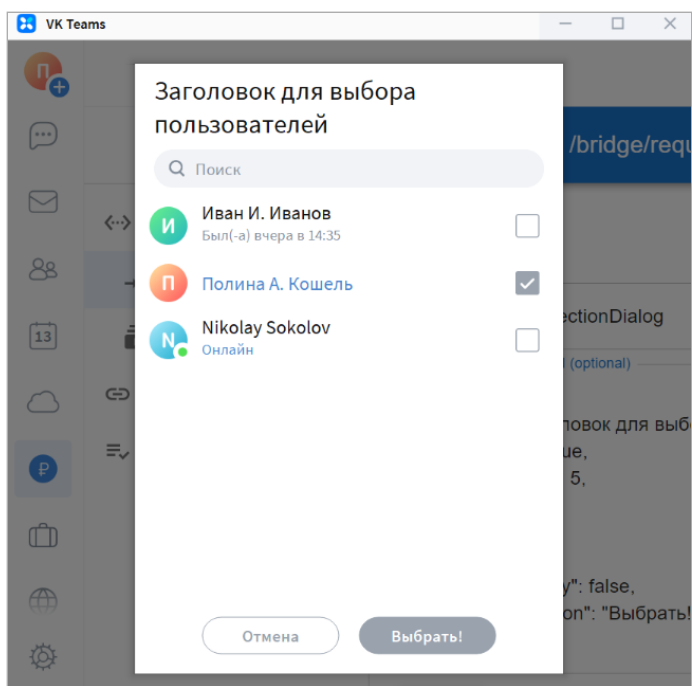
```
bridge.send('OpenContactSelectionDialog', {
  "title": "Заголовок",
  "multiple": true,
  "maxCount": 5,
  "filter": [
    "users"
  ],
  "writableOnly": false,
  "confirmButton": "Выбрать!"
})
.then(({ contacts }) => {
  const bots = contacts.filter(({ type }) => type === 'bot');
  // your code
})
.catch((err) => {
  console.error('OpenContactSelectionDialogError', err)
});
```

- title: string — заголовок окна.
- multiple: boolean — разрешить выбор нескольких контактов.
- maxCount: number (integer, >= 1)— максимальное количество контактов для выбора в случае multiple=true. При multiple=false значение maxCount игнорируется и принимает значение «1».
- filter: string[] — типы контактов, доступных к выбору пользователем. Если параметр отсутствует, то к выбору доступны все типы контактов:
 - users — пользователи;
 - bots — боты;
 - groups — группы;
 - channels — каналы.
- writableOnly: boolean — отображать только контакты, которым можно написать сообщение. Группы и каналы с ролью readonly не отображаются в диалоге. При отсутствии параметра значение по умолчанию false.
- confirmButton: string, not empty — текст, отображаемый на кнопке подтверждения выбора контактов в случае multiple=true. Если параметр отсутствует, отображается дефолтный текст «ОК». При multiple=false кнопка отсутствует.

Ответ

```
{
  "contacts": [
    {
      id: 'anna@mycompany.ru',
      type: 'user',
    },
    ...
  ]
}
```

- contacts: [{ id: string; type: string; }] — выбранные пользователем контакты:
 - id — sn пользователя, бота, группы или канала.
 - type — один из: user, bot, group, channel.



Специфичные коды ошибок

- BAD_REQUEST — поле title или confirmButton содержит пустую строку, filter содержит пустой массив или в массиве filter есть пустые строки, или не выполнены прочие ограничения на типы данных или домены значений.
- MODAL_CLOSED — пользователь закрыл экран.

OpenDialog

Открывает чат в мессенджере с контактом. Также есть возможность открытия чата на позиции конкретного сообщения при наличии его ID.

Входные данные

```
bridge.send('OpenDialog', {
  "chatId": "anna@mycompany.ru",
  "messageId": "1856720213"
})
```

- chatId: string — ID контакта или чата.
- messageId: string — ID сообщения, на котором необходимо сфокусироваться. При отсутствии чат открывается на первом непрочитанном сообщении.

Специфичные коды ошибок

- DIALOG_NOT_FOUND — диалог не найден.

OpenLink

Обработывает открытие внешних ссылок внутри мини-аппа. Рекомендуется использовать для всех внешних ссылок, чтобы гарантировать их корректное открытие.

Входные данные

```
bridge.send('OpenLink', {
  "url": "https://vk.com/"
})
```

- url: string — адрес на ресурс.

OpenPopUp

Открывает стандартное всплывающее окно для подтверждения с двумя кнопками: основная и дочерняя.

Входные данные

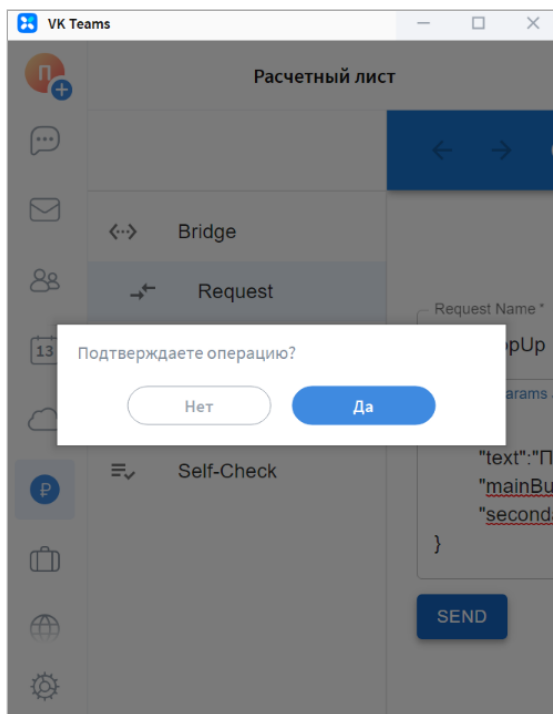
```
bridge.send('OpenPopUp', {
  "title": "Заголовок",
  "text": "Подтверждаете операцию?",
  "mainButton": "Да",
  "secondaryButton": "Нет"
})
.then(({ feedback }) => {
  // your code
})
```

- title: string — заголовок окна.
- text: string — основной текст.
- mainButton: string — текст основной кнопки.
- secondaryButton: string — текст неосновной кнопки.

Ответ

```
{
  "feedback": "secondary"
}
```

- feedback : string — "main" или "secondary" в зависимости от того, какую кнопку нажал пользователь.



Специфичные коды ошибок

- BAD_REQUEST — поле text, mainButton или secondaryButton пусто.
- POPUP_CLOSED — всплывающее окно было закрыто.

OpenProfile

Открывает профиль контакта.

Входные данные

```
bridge.send('OpenProfile', {
  "chatId": "anna@mycompany.ru"
})
```

- hatId: string — ID контакта или чат.

Специфичные коды ошибок

- PROFILE_NOT_FOUND — профиль не найден.

OpenThread

Открывает в мессенджере ветку с обсуждением. Также есть возможность открыть обсуждение на позиции конкретного сообщения при наличии его ID.

Входные данные

```
bridge.send('OpenThread', {  
  "threadId": "56578645647",  
  "messageId": "1856720213"  
})
```

- threadId: string — ID обсуждения.
- messageId: string — ID сообщения, на котором необходимо сфокусироваться. При его отсутствии обсуждение открывается на первом непрочитанном сообщении.

Специфичные коды ошибок

- THREAD_NOT_FOUND — обсуждение не найдено.

SetBadge

Устанавливает счётчик на иконке мини-аппа.

Входные данные

```
bridge.send('SetBadge', {  
  "count": 12  
})
```











- count: integer, ≥ 0 — значение счётчика.

SetButtonGroup

Добавляет в заголовок мини-аппа от одной до трёх иконок из набора доступных: максимум одна слева и две справа.

Полезно для вывода активных действий в заголовок мини-аппа. Иконки автоматически приобретают тему оформления, выбранную в клиентском приложении. Иконки отображаются в заголовке мини-аппа на всех клиентских платформах VK WorkSpace. Отрисовка разного набора иконок на разных платформах не предусмотрена.

Наименование	Иконка
add	+

Наименование	Иконка
edit	
menu	
event	
send	
done	
search	
more	
tune	
home	
orgstructure	

Входные данные

```
bridge.send('SetButtonGroup', {
  "group": "headerRight",
  "text": [
    {
      icon: 'send'
      callbackData: 'sendMessage',
      tooltip: 'Hold for other ways of sending'
    },
    ...
  ]
})
```

- `group`: string — название панели, для которой устанавливается набор кнопок. Доступные значения:
 - `headerLeft`. Максимальное количество кнопок для панели — 1.
 - `headerRight`. Максимальное количество кнопок для панели — 2.
- `buttons` — описание кнопок, устанавливаемых для выбранной панели:
 - `icon` — одна из набора, указанного выше.
 - `callbackData` — ASCII-строка, задаваемая мини-аппом для возврата в коллбэке при нажатии на кнопку (максимум — 1КБ).

- tooltip — текст до 50 символов, отображаемый в подсказке при наведении на иконку в десктоп- и веб-приложениях. Если данное поле отсутствует или его значение равно null или пустой строке, подсказка не отображается. Если длина строки превышает 50 символов, она будет обрезана и дополнена троеточием так, что ее итоговая длина не будет превышать ограничение в 50 символов.

Результат



Специфичные коды ошибок

- UNKNOWN_GROUP — неизвестное имя группы.
- UNKNOWN_ICON — одна из переданных иконок не поддерживается.
- CALLBACK_DATA_TOO_LARGE — размер callbackData для одной из кнопок превышает максимально допустимый размер.
- TOO_MANY_BUTTONS — превышено ограничение на максимальное количество кнопок для панели.

SetCanGoBack

Уведомляет клиентское приложение о возможности возврата в предыдущий экран мини-аппа для отрисовки иконки «Назад» в заголовке.

Используется в паре с подпиской BackButtonPressed, которая срабатывает при нажатии пользователя на иконку «Назад». Мини-апп самостоятельно обрабатывает нажатие на иконку «Назад» и осуществляет переход на предыдущий экран.

Если пользователю некуда возвращаться, мини-апп должен **обязательно** вызвать запрос SetCanNoGoBack для сбрасывания иконки «Назад». Иначе она останется висеть в интерфейсе и будет некликабельна.

Входные данные

```
bridge.send('SetCanGoBack', {})
```

SetCanNotGoBack

Уведомляет клиентское приложение о необходимости сбросить иконку «Назад» в заголовке.

Запрос **обязателен** для использования, если мини-апп использует SetCanGoBack. Иначе стрелка «Назад» останется висеть в интерфейсе и будет некликабельна.

Входные данные

```
bridge.send('SetCanNotGoBack', {})
```

SetTitle

Меняет текст заголовка мини-аппа. По умолчанию заголовком является имя мини-аппа.

Входные данные

```
bridge.send('SetTitle', {  
  "title": "Заголовок моего сервиса"  
})
```

- title: string — текст для заголовка.



ShowToast

Позволяет отрисовать всплывающее уведомление с текстом.

Входные данные

```
bridge.send('ShowToast', {  
  "text": "Важное уведомление"  
})
```

- text: string — текст для уведомления.

Специфичные коды ошибок

- BAD_REQUEST — поле text пустое.

StartAudioCall

Начинает аудиозвонок пользователю.

Входные данные

```
bridge.send('StartAudioCall', {
  "userId": "anna@mycompany.ru"
})
```

- `userId`: string — ID пользователя.

Специфичные коды ошибок

- `USER_NOT_FOUND` — пользователь с таким ID не найден.

StartVideoCall

Начинает видеозвонок пользователю.

Входные данные

```
bridge.send('StartVideoCall', {
  "userId": "anna@mycompany.ru"
})
```

- `userId`: string — ID пользователя.

Специфичные коды ошибок

- `USER_NOT_FOUND` — пользователь с таким ID не найден.

StorageClear

Работа с персистентным хранилищем мини-аппа. Очищает персистентное хранилище данных мини-аппа.

Входные данные

```
bridge.send('StorageClear', {})
```

StorageDelete

Работа с персистентным хранилищем мини-аппа. Удаляет пару ключ-значение из персистентного хранилища.

Входные данные

```
bridge.send('StorageDelete', {
  "key": "field_key"
})
```

- `key`: string — ключ.

Специфичные коды ошибок

- KEY_NOT_FOUND — ключ не найден.
- INVALID_KEY — ключ не валидный UTF-8.

StorageGet

Работа с персистентным хранилищем мини-аппа. Получение значения из персистентного хранилища.

Входные данные

```
bridge.send('StorageGet', {  
  "key": "field_key"  
})  
.then(({ value }) => {  
  // your code  
})
```

- key: string — ключ.

Ответ

```
{  
  "value": "123"  
}
```

- value: string — значение.

Специфичные коды ошибок

- KEY_NOT_FOUND — ключ не найден.
- INVALID_KEY — ключ не валидный UTF-8.

StoragePut

Работа с персистентным хранилищем мини-аппа. Добавляет пару ключ-значение в персистентное хранилище.

Входные данные

```
bridge.send('StoragePut', {  
  "key": "field_key",  
  "value": "123"  
})
```

- key: string — ключ. Валидный UTF-8. Максимальный размер ограничен 1024Б.
- value: string — значение. Base64. Максимальный размер ограничен 10М.

Специфичные коды ошибок

- KEY_IS_TOO_LARGE — превышено ограничение на размер ключа.
- VALUE_IS_TOO_LARGE — превышено ограничение на размер значения.
- TOTAL_LIMIT_EXCEEDED — превышено ограничение на суммарный размер всех записей, хранящихся в персистентном хранилище мини-аппа. Максимальный размер ограничен значением `omicron-` переменной `miniapp-storage-total-max-size` (в байтах, по умолчанию— 10M).
- INVALID_KEY 1 ключ не валидный UTF-8.
- INVALID_VALUE — значение не валидное Base64.

Запросы метода Subscribe

ThemeSettings

Возвращает настройки темы оформления клиентского приложения в случае её изменения.

Формат аналогичен запросу GetThemeSettings метода send.

Входные данные

```
bridge.subscribe('ThemeSettings', (data) => {
  setTheme({
    theme: data.colorsScheme,
    colors: data.colors
  })
})
```

Language

Возвращает язык клиентского приложения в случае его изменения.

Формат аналогичен запросу GetLanguage метода send.

Входные данные

```
bridge.subscribe('Language', (data) => {
  setLocale(data.language)
})
```

BackButtonPressed

Пользователь нажал кнопку «Назад», установленную при помощи запроса SetCanGoBack.

Входные данные

```
bridge.subscribe('BackButtonPressed', () => {
  const backup = someCustomState.history.pop();
  if (backup) {
    // your code
    if (!someCustomState.history.length) {
      bridge.send('SetCanNotGoBack', {});
    }
  }
})
```

ButtonPressed

Пользователь нажал на одну из кнопок, установленных при помощи запроса SetButtonGroup.

Входные данные

```
bridge.subscribe('ButtonPressed', (data) => {
  if (data.callbackData === 'sendMessage') {
    sendMessage();
  } else {
    // do something else
  }
})
```

Результат

- callbackData: string — callbackData, установленная для нажатой кнопки при добавлении кнопки.

 Технический писатель: Белова Ирина

 18 февраля 2026 г.