

# [ГХ] 6.2.1 Конфигурации Яндекс.Расписания

## Оглавление

- [Фрагмент файла конфигурации](#)
- [Описание параметров](#)

## Фрагмент файла конфигурации [↗](#)

```
1 ...
2   "integrations": {
3   ...
4     "yandexRasp": {
5       "url": "https://api.rasp.yandex.net/v3.0",
6       "apiKey": "6f8c4e2d-9b1d-4a7f-b3c5-82d1e5f7a9c4",
7       "timeout": 10000,
8       "retryPolicy": {
9         "retries": 2,
10        "backoffMs": 1000
11      }
12    },
13  ...
```

## Описание параметров [↗](#)

Параметр	Описание	Влияние на функциональность
<code>integrations.yandexRasp.url</code>	Базовый URL API Яндекс.Расписания	Куда идут запросы для поиска рейсов на транспорт
<code>integrations.yandexRasp.apiKey</code>	API-ключ для доступа	Используется для авторизации запросов во внешнюю систему
<code>integrations.yandexRasp.timeout</code>	Таймаут HTTP-запросов (в миллисекундах)	Обеспечивает стабильную работу при

		задержках со стороны внешней системы
<code>integrations.yandexRasp.retryPolicy</code>	Политика повтора запросов	Обеспечивает выполнение повторных запросов при временных ошибках со стороны внешней системы
<code>integrations.yandexRasp.retryPolicy.retries</code>	Количество повторных запросов	
<code>integrations.yandexRasp.retryPolicy.backoffMs</code>	Задержка между повторными запросами (в миллисекундах)	

## [ГХ] 6.2.2 Аутентификация и авторизация в Яндекс.Расписания

### Оглавление

- [Общее описание](#)
- [Первичная аутентификация и авторизация](#)
  - [Предусловия](#)
  - [Алгоритм работы](#)
- [Выполнение запросов к API Яндекс.Расписания](#)
- [Реавторизация при получении HTTP-401](#)
- [Обработка ошибки доступа при получении HTTP-403](#)
- [Деавторизация](#)

## Общее описание [↗](#)

Тип авторизации: API Key

API-ключ:

- Время жизни: постоянный, не имеет срока действия, не требует обновления
- Хранение: в файле конфигураций TWM

Алгоритм получения API-ключа в документации Яндекс.Расписания:  [Yandex](#)

## Первичная аутентификация и авторизация [↗](#)

### Предусловия [↗](#)

1. API-ключ сохранён в файл конфигураций TWM

### Алгоритм работы [↗](#)

1. Получить API-ключ из файла конфигураций среды выполнения
2. Использовать полученный ключ в каждом запросе к API Яндекс.Расписания
3. Если ключ отсутствует или недействителен:
  - a. Вернуть пользователю сообщение: **“Извините, произошла непредвиденная ошибка. Попробуйте позднее или свяжитесь с технической поддержкой.”**
  - b. Зафиксировать событие в журнале ошибок и отправить уведомление в систему мониторинга

## Выполнение запросов к API Яндекс.Расписания [↗](#)

Во всех запросах к API Яндекс.Расписания необходимо передавать API-ключ в качестве значения параметра `apikey` запроса (в Query Params):

```
1 apikey: <API_ключ>
```

Пример запроса:

```
1 curl --location 'https://api.rasp.yandex.net/v3.0/search/?apikey=<API_ключ>'
```

## Реавторизация при получении HTTP-401 [↗](#)

В API Яндекс.Расписания не предусмотрена отправка в ответ кода состояния HTTP 401 Unauthorized .

Если в запросе отсутствует или недействительный API-ключ, то Яндекс.Расписания возвращает код состояния HTTP 400 Bad Request с описанием ошибки. Описание ошибки передается в элементе `error.text` .

Пример тела ответа с описанием ошибки “Не указан ключ”:

```
1 {
2   "error": {
3     "text": "Не указан ключ",
4     "http_code": 400,
5     "error_code": "v3.0_26-424a816c69ba0d6cf3626c216ec44c9f264983500d847ed2d0e80a8d9273d3d5_72",
6     "request": "http://api.rasp.yandex.net/v3.0/thread/?uid=RT-542_250531_c56935_12"
7   }
8 }
```

Пример тела ответа с описанием ошибки “Неверный ключ...”:

```
1 {
2   "error": {
3     "text": "Неверный ключ \"12345\"",
4     "http_code": 400,
5     "error_code": "v3.0_26-424a816c69ba0d6cf3626c216ec44c9f264983500d847ed2d0e80a8d9273d3d5_113",
6     "request": "http://api.rasp.yandex.net/v3.0/search/?from=c50&to=c43&apikey=12345"
7   }
8 }
```

Если от API Яндекс.Расписания приходит ошибка HTTP 400 Bad Request :

1. Проверить значение элемента `error.text`
2. Если в описании ошибки присутствует слово “ключ”, то:
  - а. Повторно получить ключ из файла конфигураций
  - б. Повторно выполнить исходный запрос пользователя
- с. Если ошибка повторяется:
  - Вернуть пользователю сообщение: **“Извините, сервис временно недоступен. Попробуйте позднее или свяжитесь с технической поддержкой.”**
  - Зафиксировать событие в журнале ошибок с деталями запроса и ответа и отправить уведомление в систему мониторинга
3. Если в описании ошибки отсутствует слово “ключ”, то:
  - а. Реавторизация не нужна

Максимальное кол-во попыток реавторизации равно 2.

## Обработка ошибки доступа при получении HTTP-403 [↗](#)

В API Яндекс.Расписания не предусмотрена отправка в ответ кода состояния HTTP 403 Forbidden .

## Деавторизация [↗](#)

Отсутствует.