

VTEX - Cron Job implementation

March 2023

Contents

1	Introduction	2
2	Create a cron job	3
2.1	Expected body	3
2.2	How to create a cron job on app installation?	4
3	Retrieve a cron job	4
4	Retrieves all Cron jobs	5
5	Delete a Cron job	5
6	Cron Client Full Example	6

1 Introduction

VTEX offers OOTB APIs for managing custom cron jobs. The base URL is **`http://{accountName}.{environment}.com.br/api/scheduler/:workspace/:app`** where `:workspace` is your custom workspace or master and `:app` is the name of the application. *(i.e. `crm-async-integration`, this value can be easily retrieved from the `APP.NAME` importable from `"@vtex/api"`)*

```
constructor(context: IOContext, options?: InstanceOptions) {
  super(`http://${context.workspace}--${context.account}.
vtexcommercestable.com.br/api/scheduler/${context.workspace}/${
APP.NAME}`, context, {
  ...options,
  headers: {
    ...(options?.headers ?? {}),
    'Content-Type': 'application/json',
    'X-Vtex-Use-Https': 'true',
    VtexIdclientAutCookie: context.authToken,
  },
})
}
```

Listing 1: Constructor

2 Create a cron job

- Method: **POST**
- Base URL: `http://{workspace}-{account}.vtexcommercestable.com.br`
- Path: `/api/scheduler/:ws/:appName`
- Params:
 - version: 4

2.1 Expected body

```
{
  request: {
    uri: string,
    method: "GET" | "POST" | "PUT" | "PATCH" | "DELETE",
    headers: Record<string, string>, // it's an object
    body: Record<string, string>, // it's an object
  },
  scheduler: {
    endDate: string;
    expression: string;
  }
}
```

Listing 2: Create Cron Body Structure

Field	Description
uri	URL to call
method	REST API method to use
headers	headers to send at each request
body	body to send at each request
endDate	end date of the cron job in ISO string
expression	cron scheduled expression, see for example https://crontab.guru/ to understand how expressions are defined

Table 1: Caption

```
{
  request: {
    uri: 'https://alberto--replyqa1.myvtex.com/v1/pick-and-
collect/pickup',
    method: 'POST',
    headers: {
```

```

        'headerKey': 'headerValue'
    },
    body: {}
  },
  scheduler: {
    endDate: '2025-01-01T01:00:00+00:00';
    expression: '* /5 * * * *';
  }
}

```

Listing 3: Example of a cron job that triggers the endpoint every 5th minute

2.2 How to create a cron job on app installation?

To create a cron job, if it has not already been created, every time the application is installed, we can put our application listening for the following event:

```

"onAppInstalled": {
  "sender": "apps",
  "keys": ["setup"]
}

```

in the *service.json* file and then in the *index.ts* handle the creation of the cron job upon receipt of the event.

3 Retrieve a cron job

- Method: **GET**
- Base URL: `http://{workspace}-{account}.vtexcommercestable.com.br`
- Path: `/api/scheduler/ws/:appName/:cronId`
- Params:
 - version: 4

```

export interface CronResponse {
  id: string;
  workspace: string;
  app: string;
  request: Request;
  retry: Retry;
  attempt: number;
  endDate: string;
  expression: string;
  NextExecution: string;
}

```

```

}

export interface Retry {
  delay: Delay;
  times: number;
  backOffRate: number;
}

export interface Delay {
  addMinutes: number;
  addHours: number;
  addDays: number;
}

```

Listing 4: Response Structure

4 Retrieves all Cron jobs

- Method: **GET**
- Base URL: `http://{workspace}-{account}.vtexcommercestable.com.br`
- Path: `/api/scheduler/:ws/:appName`
- Params:
 - version: 4

5 Delete a Cron job

- Method: **DELETE**
- Base URL: `http://{workspace}-{account}.vtexcommercestable.com.br`
- Path: `/api/scheduler/:ws/:appName/:cronId`
- Params:
 - version: 4

6 Cron Client Full Example

```
import { APP, ExternalClient, InstanceOptions, IOContext } from "@vtex/api";
import { CronBody, CronResponse, Scheduler } from "../typings/cron";
import { EXPECTED_VALUE_HEADER_VTEX_APP_KEY, HEADER_VTEX_APP_KEY } from "../utils/constants";

export default class CronJob extends ExternalClient {
  constructor(context: IOContext, options?: InstanceOptions) {
    super(`http://${context.workspace}--${context.account}.vtexcommercestable.com.br/api/scheduler/${context.workspace}/${APP.NAME}`, context, {
      ...options,
      headers: {
        ...(options?.headers ?? {}),
        'Content-Type': 'application/json',
        'X-Vtex-Use-Https': 'true',
        VtexIdclientAutCookie: context.authToken,
      },
    });
  }

  public async getCronJobs() {
    return this.http.get('', {
      params: {
        version: 4
      }
    });
  }

  public async getCronJob(cronJobId: string) {
    return this.http.get(`/${cronJobId}`, {
      params: {
        version: 4
      }
    });
  }

  private async createCronJob(cronBody: CronBody) {
    return this.http.post('', cronBody, {
      params: {
        version: 4,
      },
    });
  }
}
```

```

    });
}

public createCronBody(uri: string, scheduler: Scheduler):
CronBody {
    const headers: Record<string, string> = {}
    headers[HEADER_VTEX_APP_KEY] =
EXPECTED_VALUE_HEADER_VTEX_APP_KEY
    return {
        request: {
            uri: `https://${this.context.workspace}--${this.context.
account}.myvtex.com${uri}`,
            method: "POST",
            headers: headers,
            body: {}
        },
        scheduler: {
            endDate: new Date(new Date().getFullYear() + 2, 0, 1, 1).
toISOString(), //first day of the year in 2 years
            expression: "* /5 * * * *" //at every 5th minute
        }
    }
}

public async createCronIfNotExists(uri: string, cronBody:
CronBody) {
    const crons: CronResponse[] = await this.getCronJobs();
    const isActiveCron = crons.find((cron) => cron.request.uri.
includes(uri));
    //if the cron doesn't exist then we create it
    if (!isActiveCron) {
        return await this.createCronJob(cronBody);
    }
    return isActiveCron;
}

public async deleteCronJob(cronJobId: string) {
    return this.http.delete(`/${cronJobId}`, {
        params: {
            version: 4,
        },
    })
}
}

```

CronJobClient.ts