DOCUMENTATION SOLUTION

OBI Project

Requirement

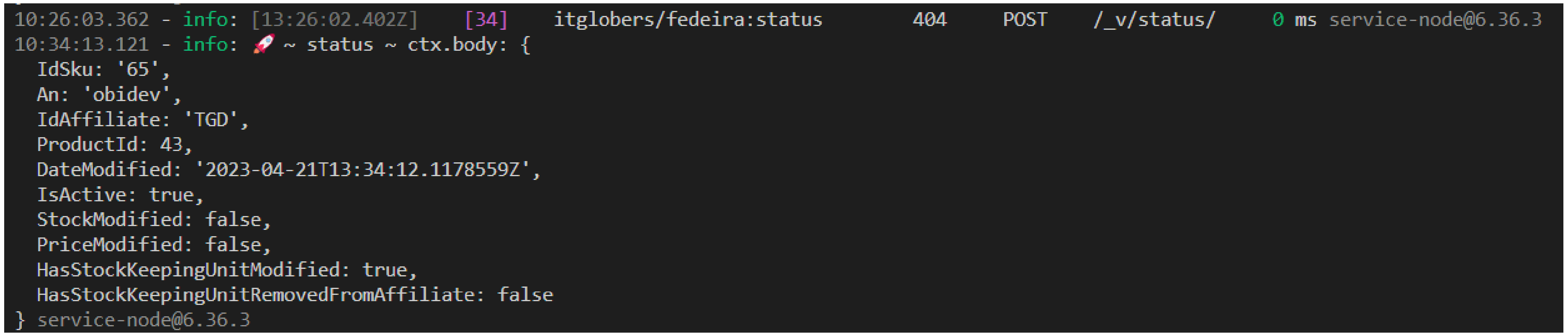
We were required to receive any sku inventory/stock and price updates from external sellers to be processed through kafka.

Solution

Through vtex affiliate feature, we created a custom hook that listens to any sku inventory or price updates and triggers a notification of such modifications.

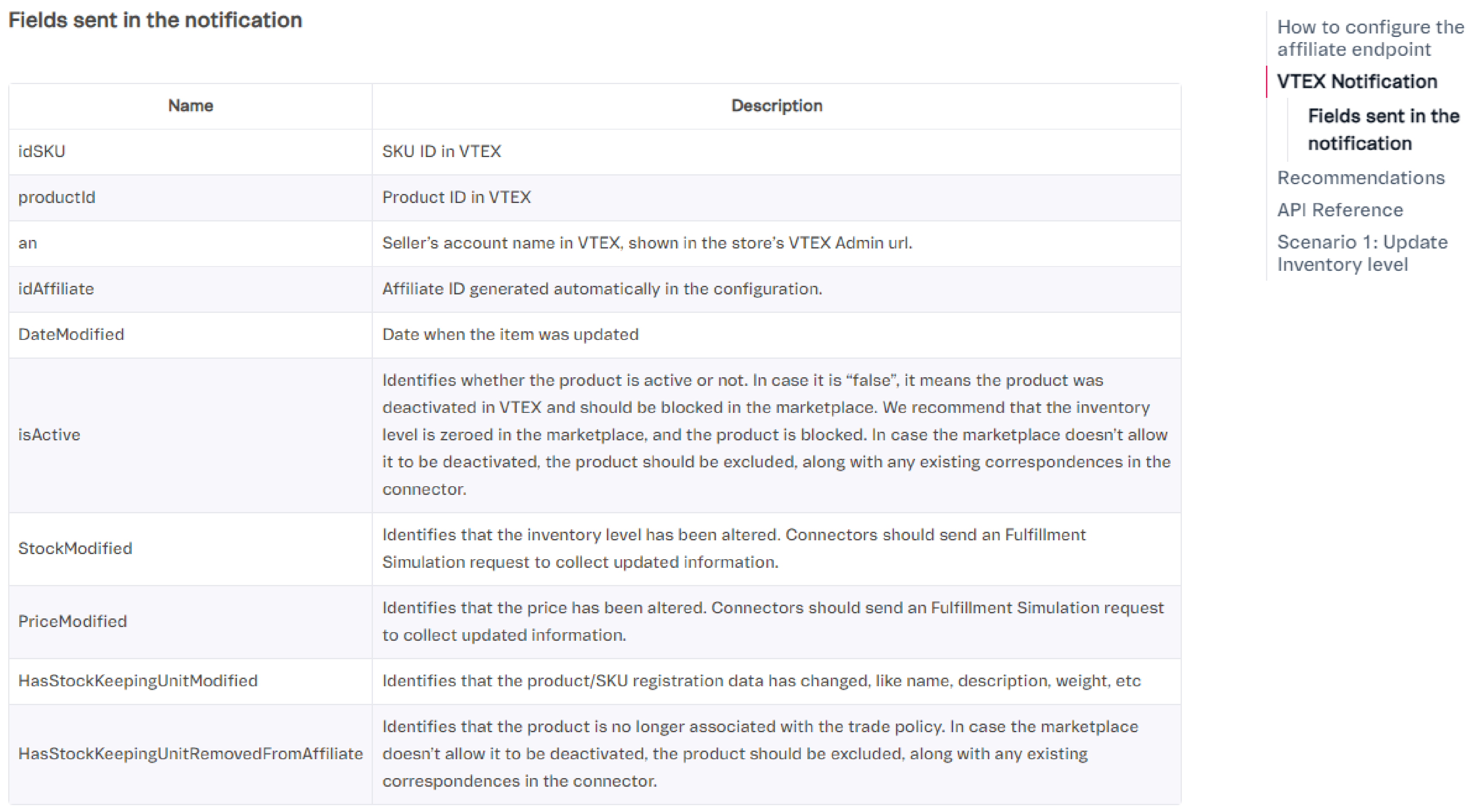
Results

Through service url, notification of these updates are sent our application with the following fields:



```
10:26:03.362 - info: [13:26:02.402Z]    [34]    itglobers/fedeira:status        404     POST    /_v/status/     0 ms service-node@6.36.3
10:34:13.121 - info: 🚀 ~ status ~ ctx.body: {
  IdSku: '65',
  An: 'obidev',
  IdAffiliate: 'TGD',
  ProductId: 43,
  DateModified: '2023-04-21T13:34:12.1178559Z',
  IsActive: true,
  StockModified: false,
  PriceModified: false,
  HasStockKeepingUnitModified: true,
  HasStockKeepingUnitRemovedFromAffiliate: false
} service-node@6.36.3
```

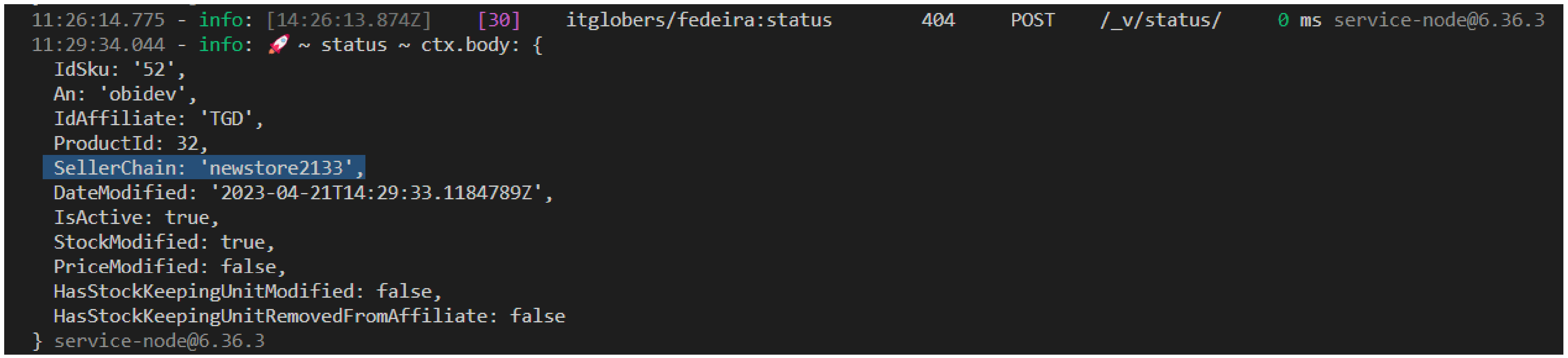
Below, an explanation of each field from vtex documentation (link provided on last title of this document: “Vtex Documentation”):



Fields sent in the notification

Name	Description
idSKU	SKU ID in VTEX
productId	Product ID in VTEX
an	Seller's account name in VTEX, shown in the store's VTEX Admin url.
idAffiliate	Affiliate ID generated automatically in the configuration.
DateModified	Date when the item was updated
isActive	Identifies whether the product is active or not. In case it is "false", it means the product was deactivated in VTEX and should be blocked in the marketplace. We recommend that the inventory level is zeroed in the marketplace, and the product is blocked. In case the marketplace doesn't allow it to be deactivated, the product should be excluded, along with any existing correspondences in the connector.
StockModified	Identifies that the inventory level has been altered. Connectors should send an Fulfillment Simulation request to collect updated information.
PriceModified	Identifies that the price has been altered. Connectors should send an Fulfillment Simulation request to collect updated information.
HasStockKeepingUnitModified	Identifies that the product/SKU registration data has changed, like name, description, weight, etc
HasStockKeepingUnitRemovedFromAffiliate	Identifies that the product is no longer associated with the trade policy. In case the marketplace doesn't allow it to be deactivated, the product should be excluded, along with any existing correspondences in the connector.

Please note that when an updated sku corresponds to an external seller, the field "SellerChain" is added to the notification:

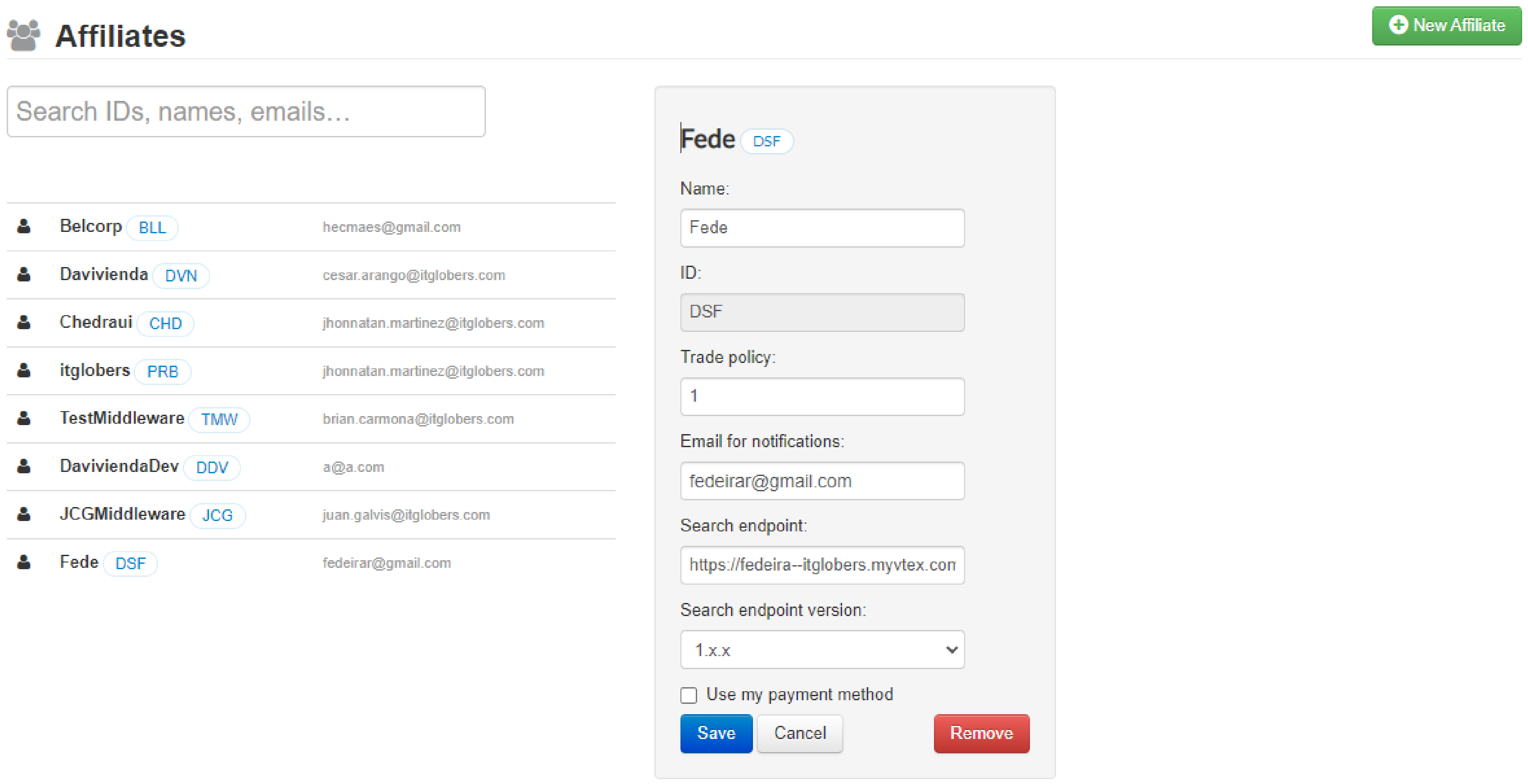


```
11:26:14.775 - info: [14:26:13.874Z]    [30]    itglobers/fedeira:status        404     POST    /_v/status/     0 ms service-node@6.36.3
11:29:34.044 - info: 🚀 ~ status ~ ctx.body: {
  IdSku: '52',
  An: 'obidev',
  IdAffiliate: 'TGD',
  ProductId: 32,
  SellerChain: 'newstore2133',
  DateModified: '2023-04-21T14:29:33.1184789Z',
  IsActive: true,
  StockModified: true,
  PriceModified: false,
  HasStockKeepingUnitModified: false,
  HasStockKeepingUnitRemovedFromAffiliate: false
} service-node@6.36.3
```

Step by step

Below, you will find a description of the steps carried out to obtain these results:

1) Set up an affiliate on vtex account. On the following example, we have used itGlobers environment:



Affiliates

New Affiliate

Search IDs, names, emails...

Belcorp	BLL	hecmaes@gmail.com
Davivienda	DVN	cesar.arango@itglobers.com
Chedraui	CHD	jhonnatan.martinez@itglobers.com
itglobers	PRB	jhonnatan.martinez@itglobers.com
TestMiddleware	TMW	brian.carmona@itglobers.com
DaviviendaDev	DDV	a@a.com
JCGMiddleware	JCG	juan.galvis@itglobers.com
Fede	DSF	fedeirar@gmail.com

Fede DSF

Name: Fede

ID: DSF

Trade policy: 1

Email for notifications: fedeirar@gmail.com

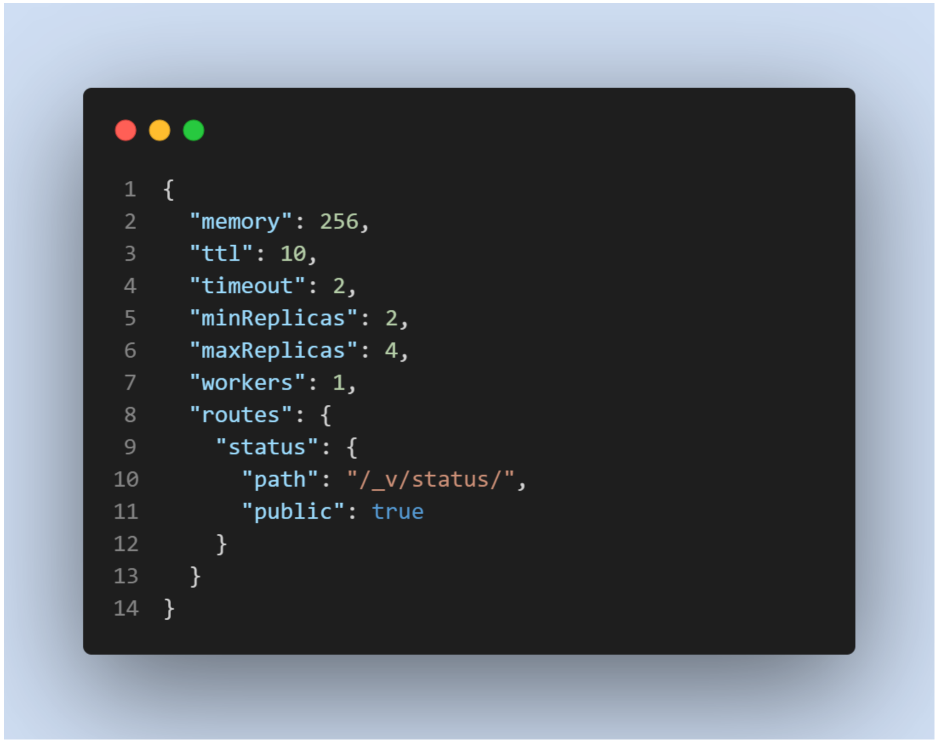
Search endpoint: https://fedeira--itglobers.myvtex.com

Search endpoint version: 1.x.x

☐ Use my payment method

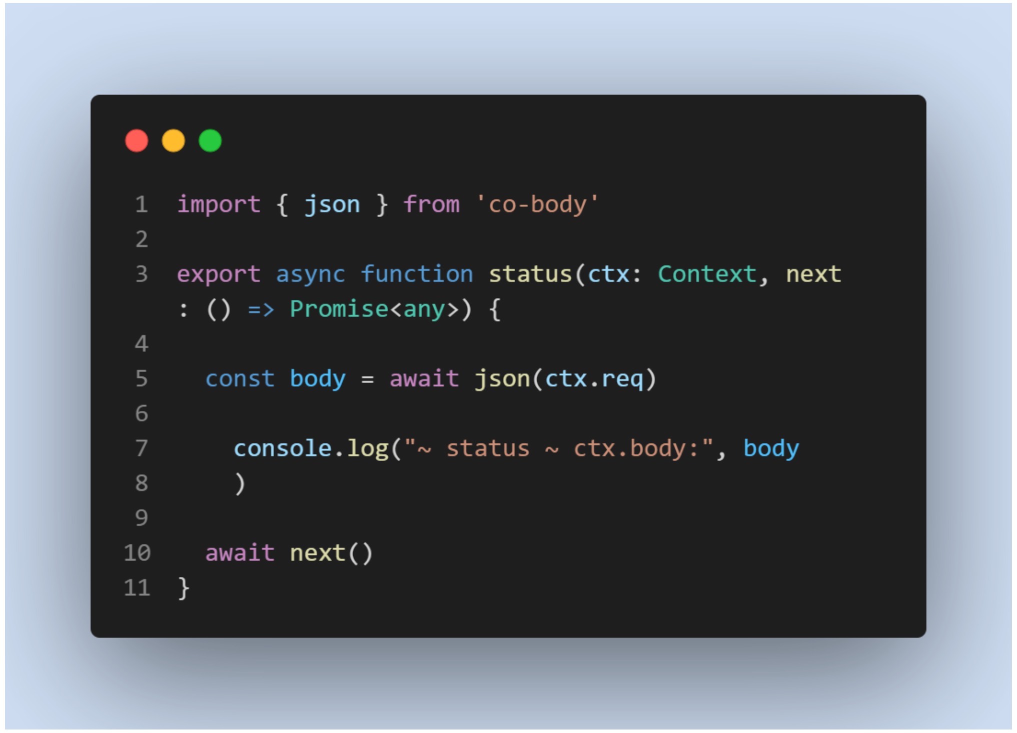
Save Cancel Remove

2) Match declared affiliate endpoint with service url (https://fedeira--itglobers.myvtex.com/_v/status/):



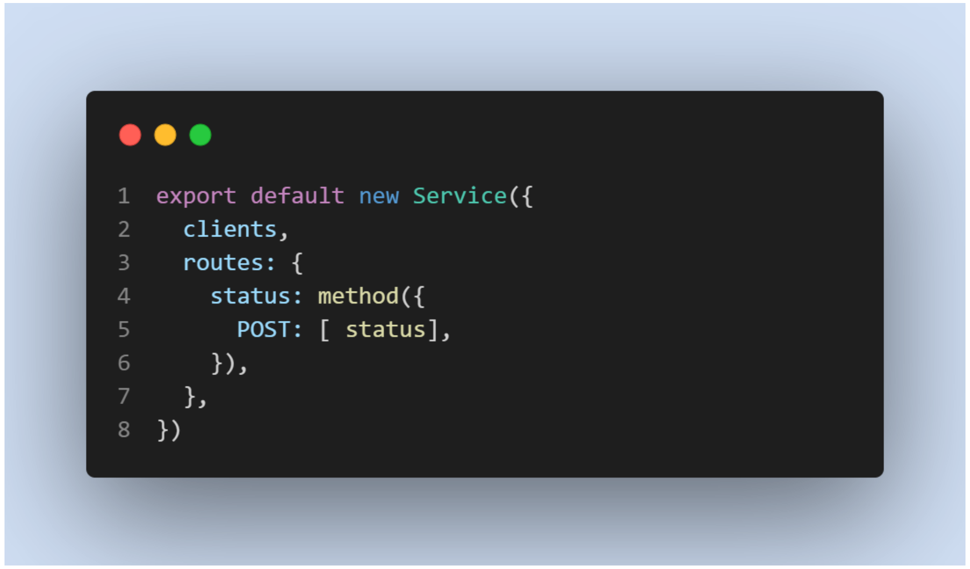
```json
{
  "memory": 256,
  "ttl": 10,
  "timeout": 2,
  "minReplicas": 2,
  "maxReplicas": 4,
  "workers": 1,
  "routes": {
    "status": {
      "path": "/_v/status/",
      "public": true
    }
  }
}
```

3) Notification fields are on context request. Add the following to node/middlewares/status.ts:



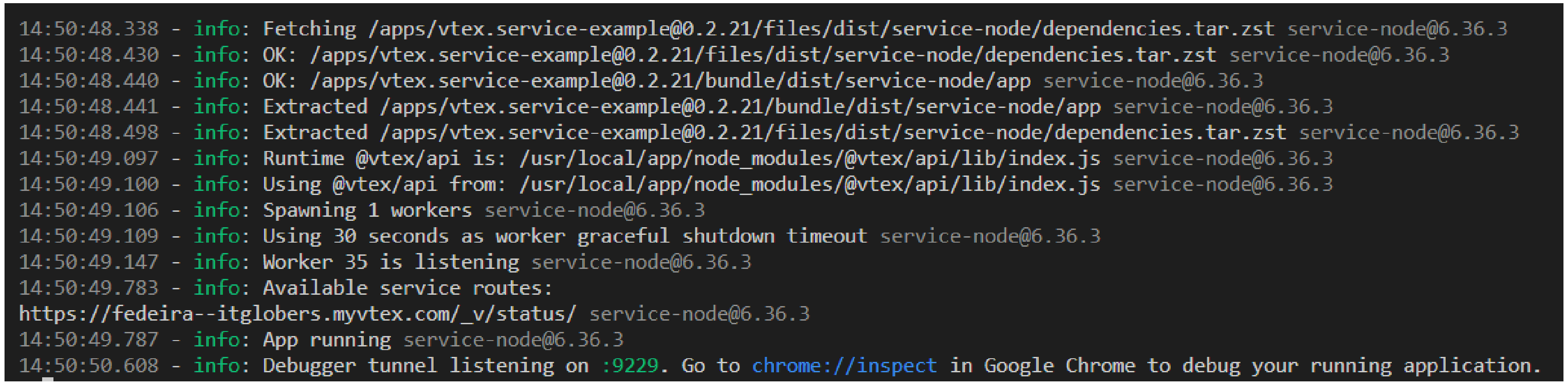
```
import { json } from 'co-body'

export async function status(ctx: Context, next
: () => Promise<any>) {

  const body = await json(ctx.req)

    console.log("~ status ~ ctx.body:", body
    )

  await next()
}
```

Also, for the purpose of this test: a) we need a POST HTTP method; and b) there is no need to use validate middleware, so said middleware can be removed from service:



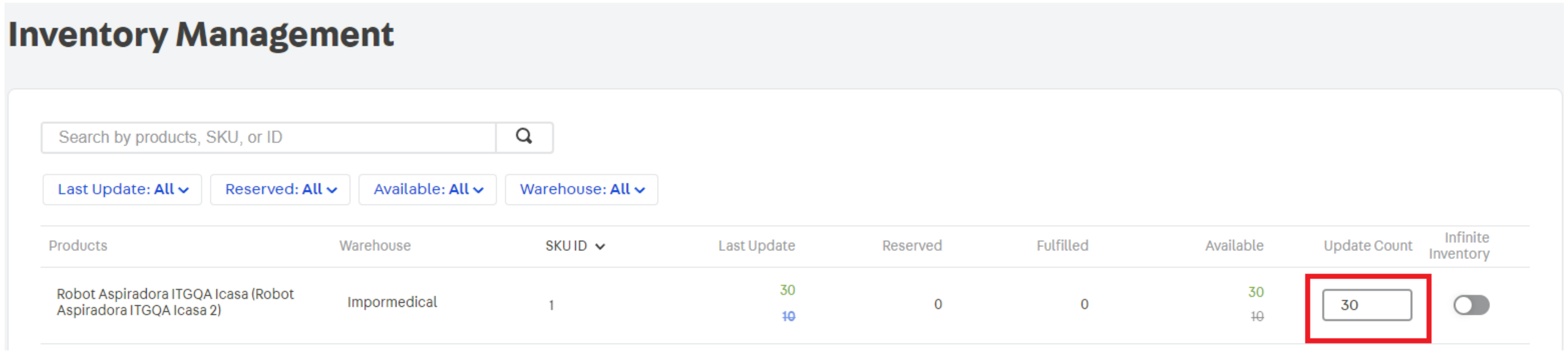
```
export default new Service({
  clients,
  routes: {
    status: method({
      POST: [ status],
    }),
  },
})
```

4) Link application to vtex account:



```
14:50:48.338 - info: Fetching /apps/vtex.service-example@0.2.21/files/dist/service-node/dependencies.tar.zst service-node@6.36.3
14:50:48.430 - info: OK: /apps/vtex.service-example@0.2.21/files/dist/service-node/dependencies.tar.zst service-node@6.36.3
14:50:48.440 - info: OK: /apps/vtex.service-example@0.2.21/bundle/dist/service-node/app service-node@6.36.3
14:50:48.441 - info: Extracted /apps/vtex.service-example@0.2.21/bundle/dist/service-node/app service-node@6.36.3
14:50:48.498 - info: Extracted /apps/vtex.service-example@0.2.21/files/dist/service-node/dependencies.tar.zst service-node@6.36.3
14:50:49.097 - info: Runtime @vtex/api is: /usr/local/app/node_modules/@vtex/api/lib/index.js service-node@6.36.3
14:50:49.100 - info: Using @vtex/api from: /usr/local/app/node_modules/@vtex/api/lib/index.js service-node@6.36.3
14:50:49.106 - info: Spawning 1 workers service-node@6.36.3
14:50:49.109 - info: Using 30 seconds as worker graceful shutdown timeout service-node@6.36.3
14:50:49.147 - info: Worker 35 is listening service-node@6.36.3
14:50:49.783 - info: Available service routes:
https://fedeira--itglobers.myvtex.com/_v/status/ service-node@6.36.3
14:50:49.787 - info: App running service-node@6.36.3
14:50:50.608 - info: Debugger tunnel listening on :9229. Go to chrome://inspect in Google Chrome to debug your running application.
```

5) Updated sku stock or price on master admin dashboard (this exercise was carried out both on itGlobers and ovidev environments):



Products	Warehouse	SKU ID	Last Update	Reserved	Fulfilled	Available	Update Count	Infinite Inventory
Robot Aspiradora ITGQA Icasa (Robot Aspiradora ITGQA Icasa 2)	Impormedical	1	30 ~~10~~	0	0	30 ~~10~~	30	

Vtex Documentation and service repository link:

You can find vtex documentation of the steps previously explained on the following links:

1) Configuration of affiliate: https://help.vtex.com/en/tutorial/configuring-affiliates--tutorials_187

2) vtex service-example repo git: https://github.com/vtex-apps/service-example

3) Modification of sku's stock and prices, and notification fields:
   a) Stock: https://developers.vtex.com/docs/guides/external-marketplace-integration-stock-update
   b) Prices: https://developers.vtex.com/docs/guides/external-marketplace-integration-price-update