

Backend developer challenge

Build an API in Node.js following the paradigm and using the libraries of your choice.

The goal of this API is to process events, augment them and to be able to get a summary of the events processed.

Requirements:

- A first endpoint should accept a list of events in JSON form, see data sample below. One or many events can be posted to this endpoint. The sample events describe the minimum amount of data you can expect for each type of events (2 types only).

The events should be augmented to contain product details instead of just containing a SKU. You will have to use the provided Products API to retrieve a merchant's products.

Available merchant ids:

- Merchant 1 ID: YcxOCwj0jg
- Merchant 2 ID: iWU4p9dJ9m

Swagger doc here: <https://www.usehero.com/static/tech-itw/>

Sample calls:

Retrieve a specific product

```
curl -X GET "https://dev.backend.usehero.com/products/887447521318" -H "x-hero-merchant-id: YcxOCwj0jg"
```

Retrieve a list of all products:

```
curl -X GET "https://dev.backend.usehero.com/products" -H "x-hero-merchant-id: YcxOCwj0jg"
```

Sample events (2 types only):

```
[{
  "type": "product-view",
  "merchant": "iWU4p9dJ9m",
  "user": "e5d54832-52ea-44c8-9307-a98a347d15bc",
```

```
"time": "2018-03-23T18:25:43.511Z",

"data": {

  "product": {

    "sku_code": "220309"

  },

  "location": "https://website.com/product/220309"

}

},

{

  "type": "transaction",

  "merchant": "yyy",

  "user": "dqw-dwww-wdqwq",

  "time": "2018-03-23T18:30:43.511Z",

  "data": {

    "transaction": {

      "order_id": "xxxxx1",

      "subtotal": 50,

      "total": 55,

      "line_items": [{

        "product": {

          "sku_code": "887447498177",

          "price": 20

        },

        "quantity": 1,

        "subtotal": 20

      },

      {

        "product": {

          "sku_code": "887447498139",
```

```
        "price": 10
      },
      "quantity": 3,
      "subtotal": 30
    }
  ]
}
}
```

- A second endpoint should return a summary of the events for a merchant.

Sample response:

```
{
  "total_events": 2,
  "number_of_customers": 2,
  "events_summary": [{
    "type": "product-view",
    "total_events": 1,
    "number_of_customers": 1
  },
  {
    "type": "transaction",
    "total_events": 1,
    "number_of_customers": 1,
    "total_value": 55
  }
]
```

- A third endpoint should list all events processed so far.

Data persistence in a database is not required, you can use in-memory data structures.

You can deploy this API somewhere or just provide instructions to run it locally. Please provide the source code either way.