

Pamatuzdevums

Nepieciešams izveidot express.js API, kurš atlasītu produktus un veikt datu transformāciju.

1. Nepieciešams izveidot REST POST tipa JSON mikroservisu:

Ienākošais ziņojuma JSON saturs:

```
1  {
2    "query": "string",
3    "page": 1
4  }
```

Ienākošā ziņojumu parametru kritēriji:

query - meklēšanas frāze (type: string, min: 3, max: 10, required: true);

page - rezultātu lapa, kura ir jāatgriež (type: number, min: 1, required: false, default: 1);

Izejošais ziņojuma JSON saturs:

```
1  [
2    {
3      "title": "string",
4      "description": "string",
5      "final_price": "double",
6    }
7  ]
```

2. Dati par produktiem ir iegūstami no online API: GET

- <https://dummyjson.com/products/search?q=phone&limit=1&skip=0>

2.1 ir nepieciešams izveidot produktu lapu sadalījumu, kur vienā lapā ir maksimums 2 produkti;

2.2 lauks "final_price" veidojas no online API atbildes lauka "price" cenas, noņemot "discountPercentage" procentus;

2.3 laukam "final_price" ir jāsaturs diviem cipariem aiz komata.

3. Izveidot ienākošā ziņojumu validāciju, kura pārbaudītu ienākošos parametrus vadoties pēc 1. punktā norādītiem kritērijiem.

3.1 Gadījumos, kad notiek kļūda atbildes HTTP statusam ir jābūt 400 - "Bad Request response".

4. Ja mikroservisa izsaukums beidzās ar kļūdu, tad servisam ir jāatgriež informāciju par šo kļūdu:

4.1 Kļūdas "code" lauka saturam ir jāsakrīt ar HTTP atbildes statusa kodu.

Kļūdas JSON ziņojuma JSON saturs:

```
1 {  
2   "code": "number",  
3   "message": "string",  
4 }
```

Cietais rieksts.

5. Ir jāpievieno logging mehānisms izmantojot express.js middleware, kura izmantojot console.log() komandu izvadītu ienākošos un izejošos ziņojumu saturu konsolē.

5.1 Atbildes Saturam ir jāatrodas konsolē arī gadījumos, ja mikroserviss beidzās ar kļūdu;

5.2 Log ierakstam sastāv no divām daļām;

5.3 Kļūdu gadījumos izejošam ziņojumam "fault" laukam vajag saturēt detalizētu informāciju par kļūdu (stack).

Ienākošais ziņojuma JSON saturs:

```
1 {  
2   type: 'messageIn',  
3   body: 'string',  
4   method: 'string'  
5   path: 'url'  
6   dateTime: 'dateTime'  
7 }
```

Izejošais ziņojuma JSON saturs:

```
1 {  
2   type: 'messageOut',  
3   body: 'string',  
4   dateTime: 'dateTime'  
5   fault: 'string'  
6 }
```

Es ticu veiksmi.

6. Ir jāspēj saņemt ziņojums gan JSON formātā, gan XML datu formātā.

6.1 Ienākošā ziņojuma formāts tiek apstrādāts un pārveidots JS objektu, izmantojot middleware funkcionalitāti vadoties pēc pieprasījuma header parametra "Content-Type";

6.2 Izejošā ziņojuma formāts tiek pārveidots no JS objekta vadoties pēc pieprasījuma header parametra "Accept";

6.3 Mikroservisu gadījuma ir jārespektē vēlamās atbildes formāts un kļūda jāatgriež "Accept" prasītā formātā.