Home Exercise

## General Description

The task simulates a simple version to one of our platform’s most basic *evidence[[1]](#footnote-0)* data processing.

We want to implement a simple service that asynchronously receives “events”/”requests” with “evidence”objects in the format of JSON and knows how to convert them into a structured table based on configuration.

Assume the following JSON structured payload (attached as a JSON file):

{

"evidence\_id": 1,

"evidence\_data": [

{

"login\_name": "anecdotes-exercise", "role": "owner", "user\_details": { "updated\_at":"2021-07-26T09:41:56Z", "id": 120000, "email": "[exercise@anecdotes.ai](mailto:exercise@anecdotes.ai)", "first\_name": "anec", "last\_name": "dotes"}, "security": {"mfa\_enabled": true, "mfa\_enforced": true}},

{

"login\_name": "anecdotes-exercise-test", "role": "owner", "user\_details": { "updated\_at": "2021-09-24T09:41:56Z", "id": 120001, "email": "[exercise-test@anecdotes.ai](mailto:exercise-test@anecdotes.ai)", "first\_name": "be", "last\_name": "exc"}, "security": { "mfa\_enabled": false, "mfa\_enforced": false}},

{

"login\_name": "anecdotes-dev", "role": "owner", "user\_details": {"updated\_at": "2021-06-26T09:41:56Z", "id": 120003, "email": "[exercise-dev@anecdotes.ai](mailto:exercise-dev@anecdotes.ai)", "first\_name": "bac", "last\_name": "kend"}, "security": {"mfa\_enabled": true, "mfa\_enforced": true}}

]

}

← (This is the “**evidence payload**”)

That outputs the following structured table (as JSON or any similar):

| **id** | **Full name** | **Email** | **Updated at** | **MFA enabled** |
| --- | --- | --- | --- | --- |
| 120000 | anec dotes | exercise@anecdotes.ai | 2021-07-26T09:41:56Z | True |
| 12001 | be exc | exercise-test@anecdotes.ai | 2021-09-24T09:41:56Z | False |
| … |  |  |  |  |
| 12002 | bac kend | exercise-dev@anecdotes.ai | 2021-06-26T09:41:56Z | True |

While the following JSON structured payload:

**{**

**"evidence\_id": 2,**

**"evidence\_data": [**

**{**

**"created": "2022-10-19T07:47:14.000Z", "description": "Single fa", "id": "00p6wqv6avsLDtEBC5d7", "lastUpdated": "2022-10-19T07:47:14.000Z", "name": "England policy", "priority": 1, "settings": {"authenticators": [{"enroll": {"self": "OPTIONAL"**

**},"key": "okta\_email"}],"type": "AUTHENTICATORS"},"status": "ACTIVE","system": false, "type": "MFA\_ENROLL"},**

**{**

**"created": "2022-10-19T07:47:49.000Z", "description": "MFA", "id": "00p6wqtt3dIcc1eKs5d7", "lastUpdated": "2022-10-19T07:47:49.000Z", "name": "Disney policy", "priority": 2, "settings": {"authenticators": [{"enroll": {"self": "OPTIONAL"**

**},"key": "okta\_email"}],"type": "AUTHENTICATORS"}, "status": "ACTIVE", "system": false, "type": "MFA\_ENROLL"**

**},**

**{**

**"created": "2022-05-17T13:57:44.000Z","description": "The default policy applies in all situations if no other policy applies.", "id": "02jkdn3jnf3wofew", "lastUpdated": "2022-10-19T07:47:49.000Z", "name": "Default Policy", "priority": 3, "settings": {"authenticators": [{"enroll": {"self": "REQUIRED"},"key": "okta\_email"}], "type": "AUTHENTICATORS"}, "status": "ACTIVE", "system": true, "type": "MFA\_ENROLL"**

**}**

**]}**

← (This is the “**evidence payload**”)

Outputs the following structured table (again, as JSON or any similar):

| **Name** | **ID** | **Authentication status** |
| --- | --- | --- |
| Default Policy | 02jkdn3jnf3wofew | ACTIVE |
| … | … | … |
| England policy | 00p6wqv6avsLDtEBC5d7 | ACTIVE |

**!!!** Please note that those are **only two examples** for evidence payloads and outputs. The payload can vary between different kinds of evidence. This means that the representation of the data and its structure are configurable. In a generic manner, the code should be able to convert a JSON payload into a structured table based on configurations per evidence. **No code changes should be required to add new evidence.**

### Step 1

Implement a **simple** RESTful/event based service that knows how to receive a collected *raw*evidence

### Step 2

For every **evidence payload and ID**, the service reads a relevant “*parsing* configuration” that you have defined (**the instructions of how to convert the evidence payload**) and outputs a new table-structured JSON (or any other table format you’d like).  
*Please note that the* ***“parsing configuration”*** *definition is a crucial part.*

### Step 3 (Bonus, Just theoretical)

Assuming that we receive a lot of evidence and every evidence includes tons of rows (for instance, evidence\_data with 1M entries) - are you extracting the evidence payload efficiently?   
Imagine this was a real task assigned to you, would you have done something differently or do you feel comfortable with your suggested solution?

## 

## Expected outcome:

We expect to receive a zip file containing the source code for the service you have just written and a simple README markdown file with some basic how-tos (e.g. how to run the service, how to feed it with an evidence payload, instructions for adding new evidence support).

## Just before you start 👾:

1. No need to deal with UI/UX anyhow.
2. You may assume that *evidence\_id* is unique
3. Bonus is just a bonus, feel free to skip this part.
4. Please do not pay this exercise more than 2-3 hours, your time is important to us and whatever your outcome is, share it with us 🙏🏼
5. Good luck!✌🏼

1. The most basic building block that exists in anecdotes, this is the data we collect from various SaaS products to automate compliance. [↑](#footnote-ref-0)