API key authentication
Key: auth-key
Value:
590e3e17b6a26a8fcda726e2a91520e476
e2c894
In: header

// Example GET product_benchmarks
response

{
"product_benchmarks": [
{
"id": 1,
"product_name": "Data Stream Pro",
"provider_name": "MarketData Inc.",
"start_date": "2023-01-01",
"end_date": "2023-12-31",
"currency": {
"id": 840,
"name": "USD",
"symbol": "$"
},
"payment": 1500,
"benchmark": 1450,
},
...
]
}

Get exchange rates

Description
Return exchange rate information for all currency conversions by calendar year

Path
GET exchange_rates

Response properties
exchange_rates: Array < exchange_rate >
Exchange rates between two currencies for a calendar year. Currencies are identified by their key

exchange_rate : object
Properties
- exchange_rate : number
  Rate of currency exchange between from and to currency
- from_currency_id : integer
  Key of the currency to convert from
- to_curency_id : integer
  Key of the currency to convert to
- year : integer
  Period the exchange rate is valid for

Task

Substantive Research's clients provide payment information from their contracts and agreements with market data providers, such as index providers and ratings agencies. Substantive Research applies internal algorithms to these payments and returns price benchmarks for each client's unique provider usage so they can see whether they are over-paying or under-paying for products.

Clients may have multiple products with a provider, sometimes in different currencies, and each product may span variable length yearly payments. We have created a mock API to retrieve this data for one client and another API to return currency exchange rates. We explain both the structure of the APIs and how to retrieve them below.

1. As a tech test assessor, I want to assess that the tech test deliverable is using modern web technologies and techniques.

Conditions Of Satisfaction:
- All processing through JavaScript, using any modern JavaScript framework
- Deliverable includes a README file with installation instructions
- Web user interface that highlights key data points
- Code shared via a remote repository e.g. Github

2. Show over or under benchmark payment for each provider

Description
As a Substantive Research client user, I want to understand whether my data provider payments are over or under benchmark, so that I can make informed decisions about cost efficiency and negotiate better contracts with my providers.

Conditions Of Satisfaction
- Payment and benchmark totals over 4 years (2021-2024) presented for each provider.
- Total benchmark difference (benchmark minus payment) presented for each provider.
- Totals should be converted to the user's preferred currency, which is Euros.

3. Show product payment year on year trends

Description
As a Substantive Research client user I want to understand how my product payments are changing from year to year, so I can evaluate trends, make informed budgeting decisions, and identify any significant increases or decreases in costs.

Conditions Of Satisfaction
- A chart showing the payment trend by year for each product that has more than one year of payments

Nice to have:
- A nice UI that makes the most of the output
- Use of a css preprocessor
- Use of graphs/tables
- Sharing your plan and approach (any notes you made before coding started)
- Any future plans for how the application could be improved

stick with basic css or Sass preprocessor?

Chart.js for graphs and charts

Corrine Sing

npm install chart.js

https://www.chartjs.org/docs/latest/getting-started/usage.html

Corrine Sing

2.
Display information in a table

TABLE HEADERS:
provider name
total payment (eur)
total benchmark (eur)
difference (eur)
status (over or under benchmark)

start off with using basic table. if there's time look for 3rd party js package

Corrine Sing

1. ensure data can be fetched and rendered from api
2. Need to convert currency to euros using rates api
3. Calculate the total payment and benchmark for each provider
4. display the difference between the benchmark and payment (positive value = client is underpaying (payment less than benchmark)

Corrine Sing

3.
Need to store payment amounts by year for each product

need to iterate through the data and extract the year from the end date field

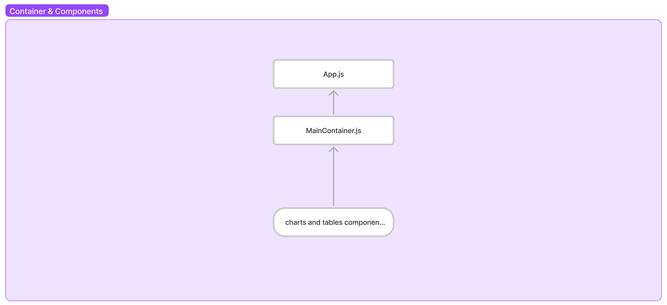
still need to convert to euros

trendData is object so will need an array of objects

each product will be an object
within that will be years and payment amounts.
the year will be an object

Corrine Sing

Container & Components



Wireframe

Future imrovements

if over bench mark have text display as red

if under then display text in green

Corrine Sing

better table option instead of html for better responsiveness

Corrine Sing

To Do 1

- Uss css preprocessor if theres time

In Progress 0

Done 7

- Test that api can be pulled through
- update readme with npm installs and instructions to run app
- Add table view
- Finish off styling
- Add line graph view
- nice ui that makes the most of the output
- Add table view as seperate component