

Confidential - this document is not allowed to be shared outside of Salt Security without a written consent

Hi,  
Thanks for taking the time to work on this assignment!

This assignment is a close simulation of the work we do at Salt in our day to day, and is meant to assess your FE development skills.

One of the existing pages in Salt is our route details page which displays all the fields per route in a customer’s API (similar to Swagger).  
Your task is to implement this page and the components inside it according to the following design (see explanations in the next page):

GET /V1/balance/history

All APIs > mobile-api.dnssf.com > V1/balance/history

RequestResponse

Search

Show PII only

Apply

Reset Filter

NAME	PII	MASKING	TYPE
URL Parameters			
fraud_details.amount	PII	MASKED	INTEGER
Query Parameters			
Headers			
Body			
description	PII	MASKED	STRING
fraud_details	PII	MASKED	OBJECT
fraud_details.account	PII	MASKED	IBAN
ph_number	PII	MASKED	PHONE NUMBER

The page consists of 4 main parts:

1. The header panel - displaying:
  - a. The method and path of the route
  - b. "Breadcrumbs": All APIs > {apiName} > {path} - non clickable.
2. Tabs - Request & Response.
  - a. Switching to the response tab displays a table in the exact same format as in the request showing response data, excluding url params and query params groups (see json example data)
3. Filter panel:
  - a. Search - searches by field name or parameter type
  - b. PII (private info) toggle Checkbox - filters the fields by pii attribute (true / false), in addition to the search above.
  - c. Reset filter - resets the filter immediately
  - d. Apply button - applies filters (a & b) on the table / list below.
4. Table / List (displays data for request / response):
  - a. Contains a list of headers: Name, PII, Masking, Type
  - b. List of rows divided into 4 **groups** (2 for response): URL Parameters, Query Parameters, Headers, Body.
  - c. Each row contains:
    - i. The name of the field - string
    - ii. PII - boolean, this is a button which can be clicked, when the click is performed the value is toggled and the background and font colors are inverted.
    - iii. Masked - boolean, same action on click as for PII.
    - iv. Type - string

You can find example data (json file) and the design file attached to this assignment.

#### Guidelines:

- This page must be implemented using React
- The implementation should follow the design, as much "pixel perfect" as possible
- The implementation should be as clean, readable and efficient as possible - you should be able to explain the division into components and any other design decisions
- Data that is changed in this page will be saved solely in-memory for the purpose of this assignment (will return to default after refresh).
- You can use FontAwesome or similar libraries for icons.
- **You have to implement the table / list component yourself**, other than that: you can use any tool and online resource you want, and reach out to us for consultation - as you would if you were building this page as FE developer at Salt Security :-)

Good Luck!