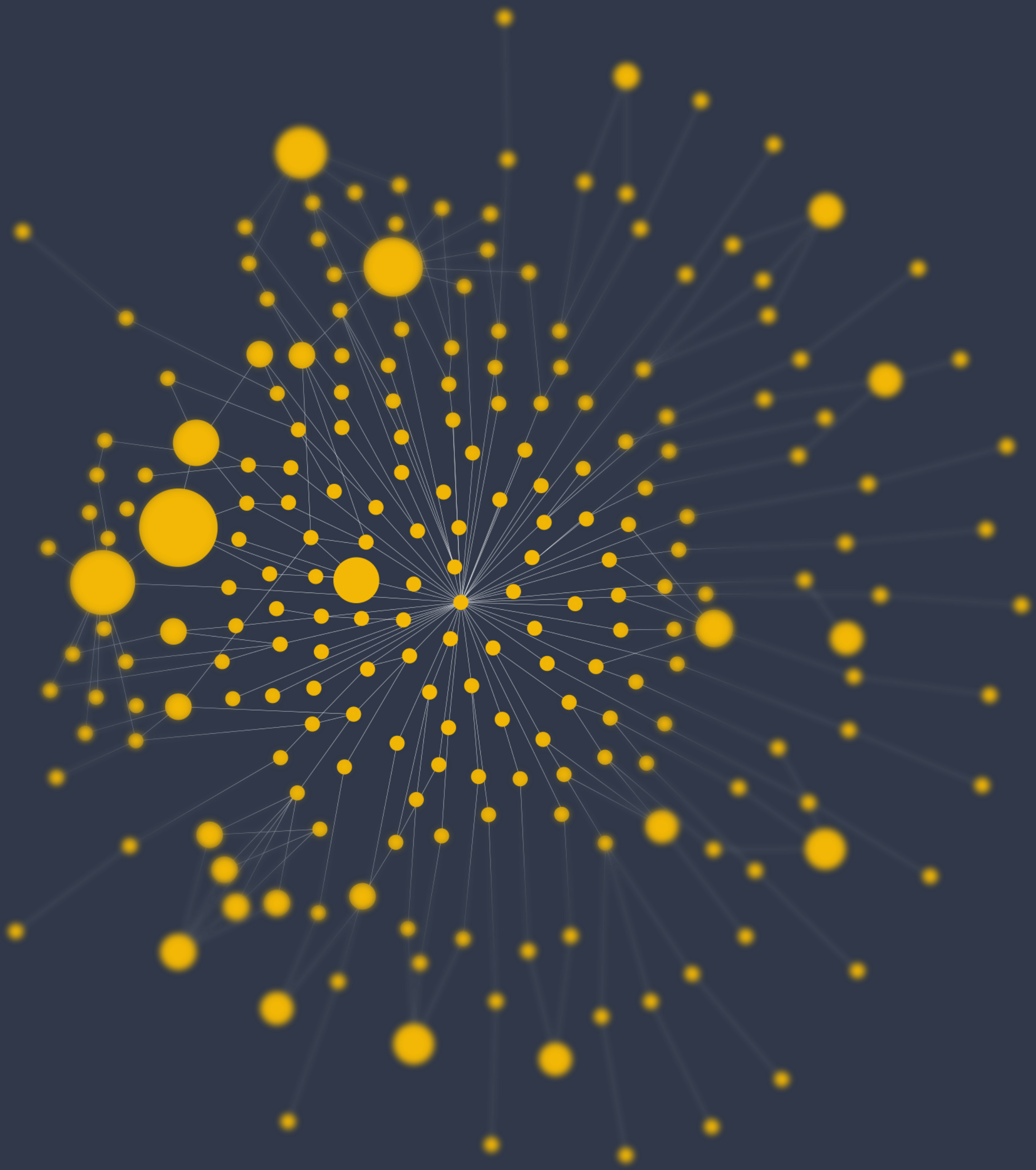
MALTEGO
MINE, MERGE, MAP DATA.

Interview Challenge

Munich, May 2022



The Practical Presentation

We would like to get to know you better and learn about your fit to the role of Senior Front End Engineer at Maltego.

These challenges should not take you longer than a couple of hours. Please submit your solution via email in a zip archive, without including dependencies. Include a readme with instructions on dependency installation and running the app. It is not necessary to include production ready configuration.

We are looking forward to getting to know you,

The Maltego team

Task 1 - Server

Write an API that provides the following functionality:

- Return a list of graphs
- Return a specific graph by graph ID
- Delete a graph by ID

You can find a seed file for the graph data in the attached file “graphs.json”. For this challenge it is not necessary to store the graphs in a database. All modifications can be done in memory. This is a front-end position after all.

Pick any framework based on Java, JavaScript or Python.

Task 2 - Client

Write a React web app that consumes the API that you created in challenge 1. The app should have two routes:

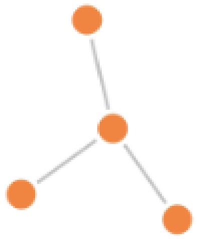
GRAPH LIST

Display a list of available graphs. Feel free to use your own design. In addition to viewing the graph list, a user should be able to perform the following actions:

- Filter the graph list based on node label and graph name. The user should enter the filter in a text input. Graphs with a matching graph name should be ranked on top, while graphs without a matching name but a matching node label should follow. The filter should be case insensitive. Filtering can be done in the client.
- Delete a graph
- Create a new graph with empty data (graph.data = { nodes: [], edges: [] }). The only data that should be stored is the graph name. Ensure that the user cannot create graphs with empty names. The graph id should be assigned on the server side.
- Open a graph. When clicking a graph, the user should be sent to a graph view route with a parameter that contains the graph id.

GRAPH VIEW

- This route renders the graph data using a force directed graph drawing algorithm. Feel free to use any of the many open-source libraries. The user should be able to navigate back to the graph list route. The graph view route should load the graph from the API, making sure that the link can be opened directly.



Task 3 – Bonus Points

- Write unit tests for the graph filtering functionality
- Make sure the client is responsive
- Render node labels on the graph view route
- Write a feature for adding nodes to existing nodes. Be creative with the details.

Background information

Information on how to use Maltego

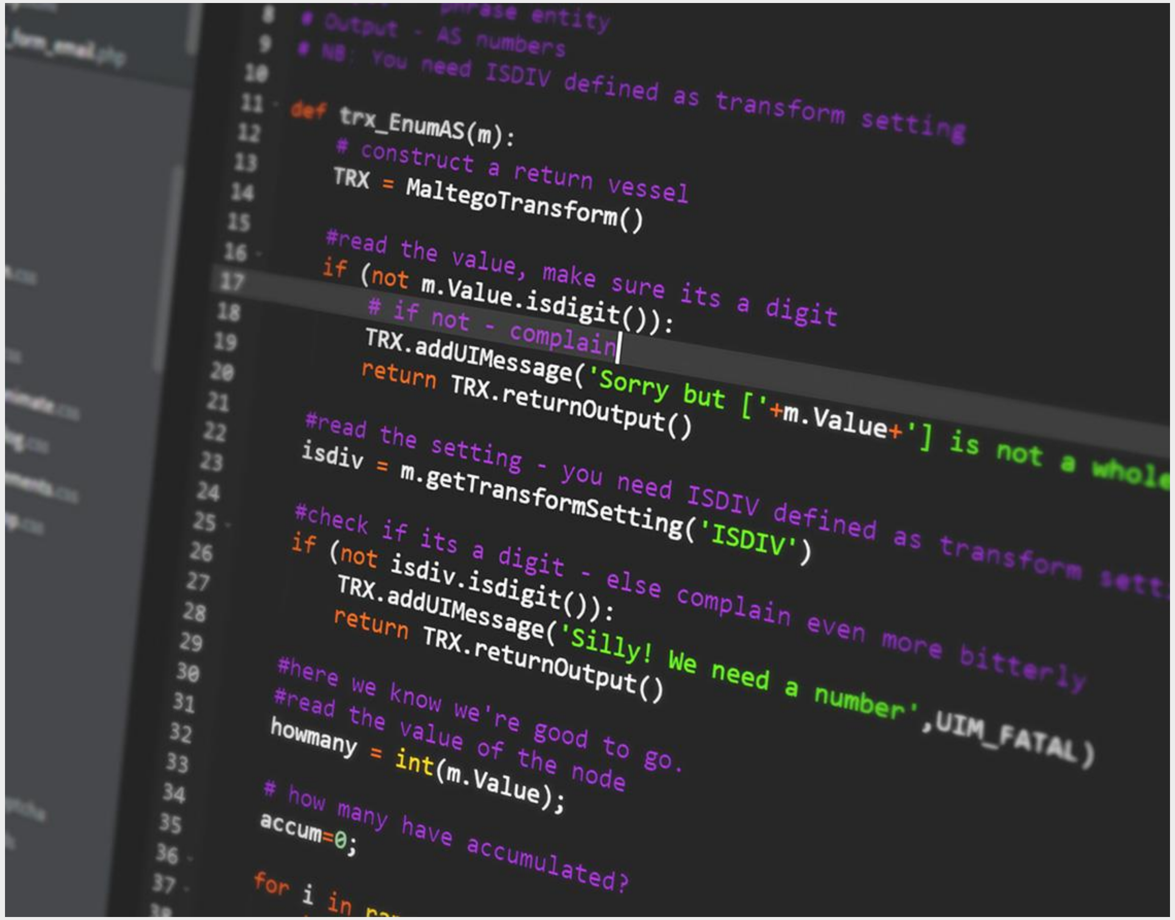
Docs.maltego.com

https://courses.maltego.com/users/sign_in

About us

- **Maltego Technologies GmbH is your partner for all Maltego products and provider of all Maltego related services. It was founded in 2017 and is based in Munich, Germany.**
- The Maltego application is a visual link analysis tool that, out of the box, comes with open source intelligence (OSINT) plugins called Transforms. The tool offers real time data mining and information gathering, as well as the representation of this information on a node-based graph, making patterns and multiple order connections between said information easily identifiable.

Maltego combines the creative exploration capabilities of humans with the automation potential of machines. Maltego is a much loved and widely used tool for open source intelligence and graphical link analyses.



```
    phrase entity
# Output - AS numbers
# NB: You need ISDIV defined as transform setting

def trx_EnumAS(m):
    # construct a return vessel
    TRX = MaltegoTransform()

    #read the value, make sure its a digit
    if (not m.Value.isdigit()):
        # if not - complain
        TRX.addUIMessage('Sorry but ['+m.Value+'] is not a whole
        return TRX.returnOutput()

    #read the setting - you need ISDIV defined as transform sett
    isdiv = m.getTransformSetting('ISDIV')

    #check if its a digit - else complain even more bitterly
    if (not isdiv.isdigit()):
        TRX.addUIMessage('Silly! We need a number',UIM_FATAL)
        return TRX.returnOutput()

    #here we know we're good to go.
    #read the value of the node
    howmany = int(m.Value);

    # how many have accumulated?
    accum=0;

    for i in ra
```

Since 2008 Maltego has empowered a million investigations worldwide.

Mine.

Easily gather information from dispersed data sources.

Merge.

Automatically link and combine all information into one graph.

Map.

Visually explore relationships in your data.

The Power of Transforms

- **Transforms are small pieces of code that automatically fetch data from different sources and return the results as visual entities in the desktop client. Transforms are the central elements of Maltego which enable its users to unleash the full potential of the software whilst using a point-and-click logic to run analyses.**

- Explore step-by-step. Transforms are designed to build on each other, so you can create complex graphs.

Automate in a click. Execute a set of Transforms in a pre-defined sequence to automate routines and workflows.

Create your own queries. Use the Transform Development Toolkit to write and customize your own Transforms, and to integrate new data sources.

How Investigators Use Maltego

Cyber Threat Intelligence

Visualize and understand the attack surface. Help prevent potential threats to the infrastructure.

Digital Risk Protection

Investigate and help protect against unwanted asset exposure resulting from Digital Transformation efforts.

Business Risk Intelligence

Reduce your organizational risk by visualizing intel from illicit communities where threat actors discuss how to access and monetize proprietary data.

Cryptocurrency & the Blockchain

Gather and link digital evidence on suspicious use of cryptocurrencies.