REACT Technical Test

Introduction

This test will demonstrate your technical ability as a REACT developer and should highlight your design ability to produce High Quality UX with the 'User' at the forefront of your thinking. You should demonstrate knowledge of consuming RESTful API endpoints to present data in an easy-to-use intuitive application fulfilling the requirements set out in the detailed design brief.

We also want to introduce some of the more basic data concepts we have within Taylor Wessing for you to present data back to users in a clean easy to use interface. You will be provided access to a test RESTful API in azure which is accessed via an API Key*. The details of are documented below.

The RESTful API is self-documented as a SWAGGER endpoint** is available for you to understand the API endpoints you will have available to you, these are documented with the respective routing calls and responses.

**Details of the API location and how to authenticate will be provided in the detailed brief below*

***Swagger Endpoint details will be provided in the detailed brief below*

Detailed Brief

We would like you to build a to REACT web application to surface client data and matters from an API that you have been provided access provided to. This is an API for built for demonstration purposes and the data contained within is mock data. In some cases, data is synthetically generated. Therefore, this exercise is about trusting the API to perform its business logic and for you to surface the data appropriately in line with the brief's requirements. The API will respond with appropriate data matching the respective endpoints, but in terms of the synthetic data you may get different data when repeating the same calls.

This API will introduce you to two basic concepts used within Taylor Wessing (Clients) and (Matters). For the purposes of this exercise, we are not looking for securing the application with a login mechanic. The only security in place is that afforded by the API key for the application to communicate with the API endpoint.

Clients

Clients are the clients of Taylor Wessing, those individuals or entities that require the use of Taylor Wessing's Legal services. The data will comprise of address data and people who are known to Taylor Wessing. These are likely people who can act on behalf of the individual or company in relation to the work Taylor Wessing performs for an individual or company. Details such as email address and phone exist against a person as these tend to be details specific to that person rather than a company.

Matters

Matters in brief are the container that describes the work that Taylor Wessing does for a client. Larger organisations will tend to have many matters as Taylor Wessing carries out legal services for them. Individuals will likely have a few or maybe even one matter. We have not gone into too much detail with matters for this exercise, other than to understand that a client who is an individual or company may have one or more matters associated with them. Matters underpin how Taylor Wessing can bill a client for services carried out.

API Endpoint

The API endpoints are available at the following URL:

https://az-weu-webapp-34-techtest-f0g0fvdkgugabfb7.westeurope-01.azurewebsites.net/

Authorisation

For this exercise the endpoint is protected with the use of an API key which must be presented in a request header 'Authorization'.

API Key

Your API Key is shown below this is a time sensitive API Key and it will be revoked on the deadline date for your code submission.

02efd881-d2c2-4b03-869f-c206f6456ce5

Endpoints

Client Search

This is and endpoint used to search for clients:

{URL}/clientdata/clientsearch/{search term}/{column order}/{sort}/{index}/{offset}

{URL} as provided above in the section API Endpoint

{search term} is the filter you wish to apply to search by client's name

{column order} this is which column you wish to perform as sort against **NAME** or **DATE**. The default value if not supplied is **NAME**.

{sort} this is how you wish to sort the column described in *{column order}* **ASCENDING** or **DESCENDING**

{index} this is the point in the result set you wish to take you offset records from. This must be **0** or greater. (Used for building pagination)

{offset} the count of records you wish to return from the index. This must be a value of between **1** and **50**. (Used for building pagination)

A more detailed description of the endpoint, its behaviours and responses can be found in the Swagger endpoint detailed below.

Get Client Data

This is the endpoint to fetch client data by a given client id:

{URL}/clientdata/client/{client id}

{URL} as provided above in the section API Endpoint

{client id} is returned when a successful search has been performed in the Client Search endpoint.

A more detailed description of the endpoint, its behaviours and responses can be found in the Swagger endpoint detailed below.

Matter Search

This is the endpoint used to return matters for a given client Id:

{URL}/clientdata/mattersearch/{client id}/{column order}/{sort}/{index}/{offset}

{URL} as provided above in the section API Endpoint

{client id} is used to return matters for the given client id as returned from the initial Client Search endpoint.

{column order} this is which column you wish to perform as sort against **NAME** or **DATE**. The default value if not supplied is DATE.

{sort} this is how you wish to sort the column described in *{column order}* **ASCENDING** or **DESCENDING**

{index} this is the point in the result set you wish to take you offset records from. This must be **0** or greater. (Used for building pagination)

{offset} the count of records you wish to return from the index. This must be a value of between **1** and **50**. (Used for building pagination)

A more detailed description of the endpoint, its behaviours and responses can be found in the Swagger endpoint detailed below.

Get Matter Data

This is an endpoint to fetch a specific matter by its matter id:

{URL}/clientdata/matter/{matter id}

{URL} as provided above in the section API Endpoint

{matter id} is returned when a successful search has been performed in the Matter Search Endpoint.

A more detailed description of the endpoint, its behaviours and responses can be found in the Swagger endpoint detailed below.

Swagger

For a detailed description of the API and its endpoints the following Swagger endpoint exists:

{URL}/swagger/index.html

Open API json is also returned at the following URL:

{URL} swagger/v1/swagger.json

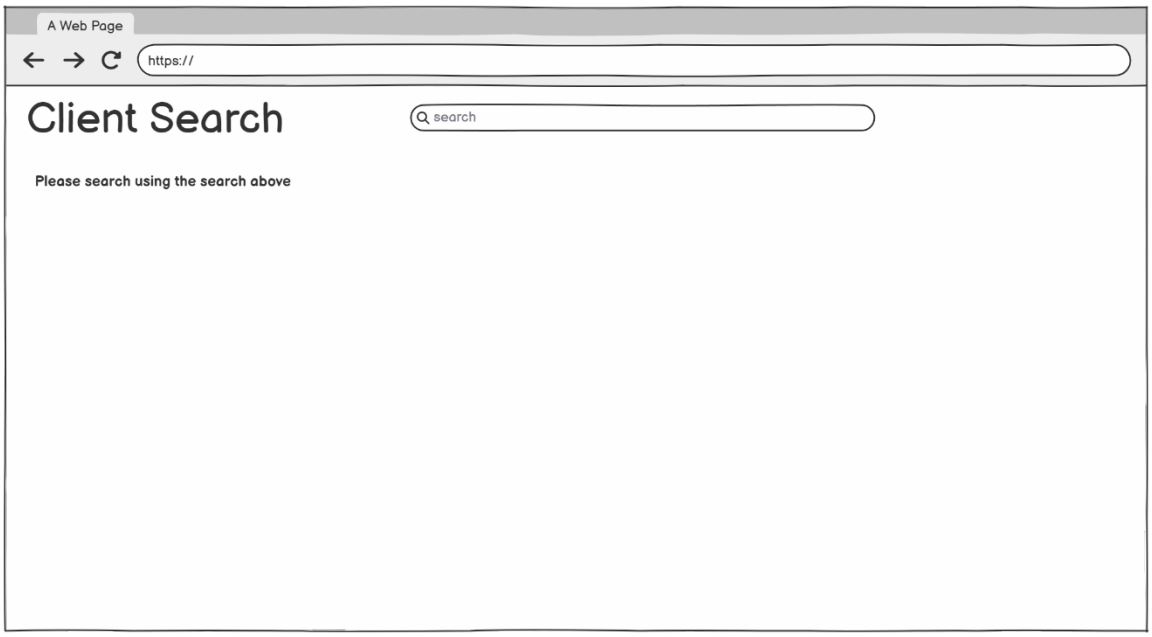
{URL} as provided above in the section API Endpoint

User Requirements

The following images describe a series of wireframes that describe how a user group thinks the application should look and feel. These are by no means how we want the product to look like, you should use your own initiative to come up with the design and function that best satisfies what is depicted in the wireframes.

Client Search Page Without Results

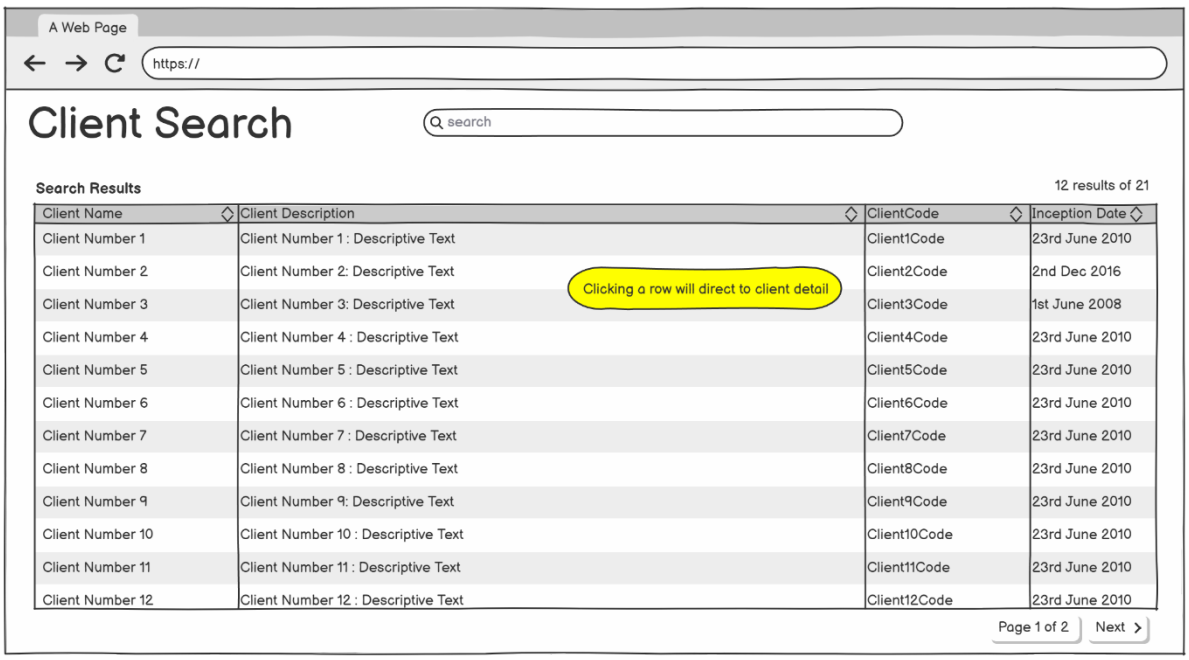
This is the initial landing page from which the user will search for a client. How the search is built is entirely up to you. Ensure the basics are adopted and its functional. Again, it must be functional, and the end design is up to you as long as it adheres closely to the ask.



Empty Results / Error handling is not covered in the wireframes, but you should consider it in your build.

Client Search with Results

This is the page depicting a successful set of client results after a successful search has been carried out. How this looks is up to you. The API search mechanic has been built to be as flexible as possible to allow whatever variation you want in terms of search results and pagination.



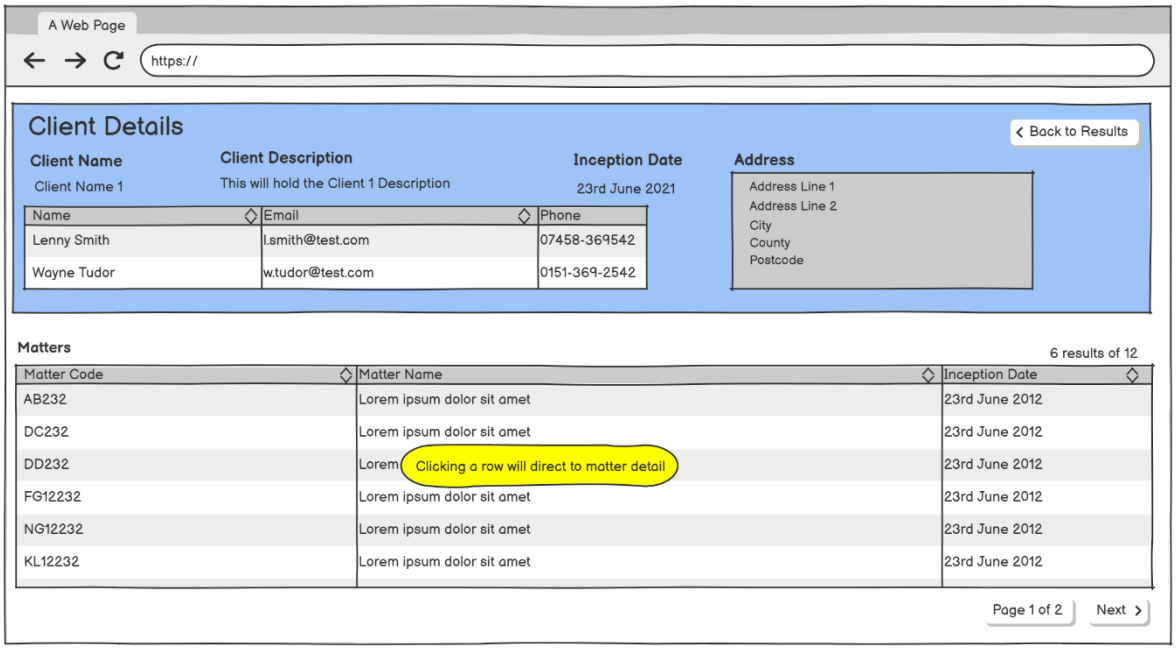
Client Name	Client Description	ClientCode	Inception Date
Client Number 1	Client Number 1 : Descriptive Text	Client1Code	23rd June 2010
Client Number 2	Client Number 2: Descriptive Text	Client2Code	2nd Dec 2016
Client Number 3	Client Number 3: Descriptive Text	Client3Code	1st June 2008
Client Number 4	Client Number 4 : Descriptive Text	Client4Code	23rd June 2010
Client Number 5	Client Number 5 : Descriptive Text	Client5Code	23rd June 2010
Client Number 6	Client Number 6 : Descriptive Text	Client6Code	23rd June 2010
Client Number 7	Client Number 7 : Descriptive Text	Client7Code	23rd June 2010
Client Number 8	Client Number 8 : Descriptive Text	Client8Code	23rd June 2010
Client Number 9	Client Number 9: Descriptive Text	Client9Code	23rd June 2010
Client Number 10	Client Number 10 : Descriptive Text	Client10Code	23rd June 2010
Client Number 11	Client Number 11 : Descriptive Text	Client11Code	23rd June 2010
Client Number 12	Client Number 12 : Descriptive Text	Client12Code	23rd June 2010

Its is expected that a row click in the results will direct the user to the Client Data screen. Again, it must be functional, and the end design is up to you as long as it adheres closely to the ask.

Client Data with Matters

This is the client data screen. It should be prepopulated with Matters that are available. The Get Client Data and Matter Search endpoints should allow this screen to be built. Again, the important

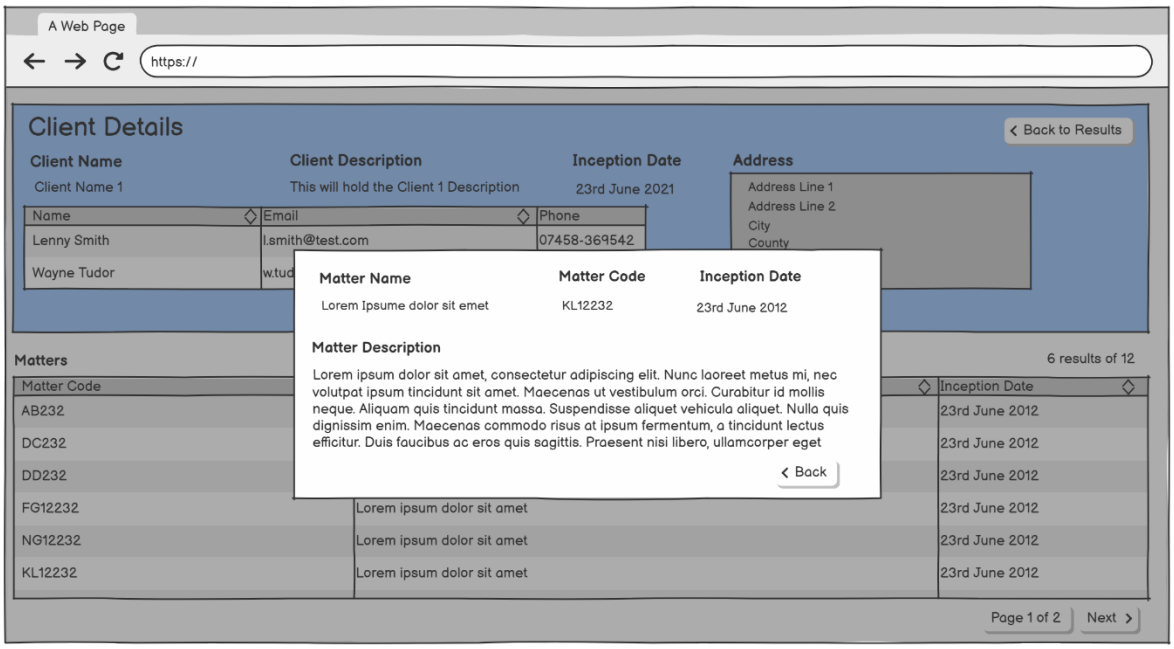
thing here is that the data outlined in the image is returned in the interface as its all available in the API. It is up to you to determine how form and function should work.



Clicking on a matter in the results set should open the matter data in a model window as depicted next. Again, it must be functional, and the end design is up to you as long as it adheres closely to the ask.

Client Data with a Matter Open

This depicts a matter open once a user has clicked on a matter in an open Client Data screen. Design is again up to you if the data outlined is returned from the API. The Get Matter endpoint should allow the modal window to be populated.



Again, it must be functional, and the end design is up to you as long as it adheres closely to the ask.

Conclusion

We require a usable application that a user can open and navigate through the process flow searching for clients and opening appropriate matters. The functional process must be as close as possible to the wire frames, function will be the most important factor for this test. Design and form are also important, but a functioning application is the priority here. You may if you decide to go beyond the wireframes to build functionality that you feel may be missing but appropriate. You will be asked to discuss your functionality and reasons in the interview.

Submission

Work should be where possible made available to us via GIT HUB repository. You should provide us access to a public GIT HUB repo for retrieval of you project.

If you have difficulty in providing us with access to a GIT HUB repository, please provide the code in a zip file attached to an email.