Setup

Prerequisits

- A tool to send HTTP-Requests like cURL or Postman

Preparations

1. Download the latest release from Github or clone the repository on your local machine.
2. Start the application. This will start a local instance that runs on Port 8081.

Checking the discovery endpoint

Open your favorite browser and go to `http://localhost:8081/`. This will open a user-interface that lets you enter the URL of the identity-provider you want to analyze. Please make sure to enter the complete path, to ensure that the program can find the correct endpoints (e.g. if your provider is hosted in a subpath).
Below this you can select which values are returned from the discovery-endpoint (IMPORTANT: If no keys are selected, no values are returned!).

If you have entered all the information hit the enter key to submit them. The results are then shown in a separate field below the input form.

It is also possible to validate the response from the discovery-endpoint against a JSON-schema. To do that, select one of the available schemas from the dropdown below the filters and the response is automatically validated against the select schema. If the response is not valid, the application returns an error message, that describes what went wrong (e.g. required key 'XY' was missing). Otherwise the response of the discovery endpoint is returned as usual and the values of the required keys are highlighted. If you want to use your own schema, you first must upload the schema via the file-upload below. After the upload you can just select your schema instead of the default ones. No other steps are necessary.

Requesting an access token

OpenID-Doctor currently supports the following OAuth2- and OpenID-Connect-flows:

- Client-Credentials

Client-Credentials

To request a token via the client-credential-flow open your terminal of choice, copy the following command into the terminal and replace the placeholders with the corresponding information:

```
curl --request POST --url
http://localhost:8081/api/token?issuer=$ISSUER_STRING --header
'content-type: application/json' --data
'{"client_id":$CLIENT_ID,"client_secret":$CLIENT_SECRET,"audience":$AUD
IENCE,"grant_type":"client_credentials"}'
```

- ISSUER_STRING: The root-URL of the identity-provider
- CLIENT_ID: The client-id of your application
- CLIENT_SECRET: The client-secret of your application
- AUDIENCE: The audience that would request the token (often a duplicate of the client-id)

Decoding and validating a token

To decode a returned access-token open your browser and go to `http://localhost:8081/api/token/decode`. This will open a user-interface, where you can enter the following information:

- Issuer: The root-URL of the identity-provider that provided the token.
- Key-Material-Endpoint: The endpoint that provides the key-material necessary to validate the signature of the access-token. (If you don't know the endpoint, have a look at the discovery-endpoint)
- Access-Token: The access-token to decode and validate

If you entered all the necessary information, click the submit-button below to start the decoding. The results of the requests are then shown below the input form:

- If the token was valid, the decoded header and payload are displayed
- If an error occurred, a short error-message with a corresponding error-code is displayed.