**Testing Manual for FHIR APIs**

Contents

[**1.** **Introduction** 2](#_Toc110534018)

[**2.** **Accessing Arcadia FHIR APIs for Testing** 2](#_Toc110534019)

[Development Sandbox FHIR APIs 2](#_Toc110534020)

[3. **Testing Manual for UI of FHIR API** 8](#_Toc110533688)

1. **Introduction**

Arcadia as part of its product evolution roadmap identified HL7 FHIR APIs that need to be supported to offer new data access mechanisms sought by customers. The primary scope of the FHIR API project is to build out the necessary FHIR APIs that can be used for compliance to the ONC 21st Century Cures Act regulation and the [USCDI v1](https://www.healthit.gov/isa/sites/isa/files/2020-10/USCDI-Version-1-July-2020-Errata-Final_0.pdf) data elements.

FHIR APIs are RESTful endpoints, that support access to healthcare data with appropriate security and privacy protocols. For introduction to FHIR, please refer to [FHIR Background](http://hl7.org/fhir/documentation.html) especially the background section. Arcadia FHIR APIs are implemented in conformance to the [US Core FHIR Implementation Guide version 3.1.1](http://hl7.org/fhir/us/core/STU3.1.1/).

1. **Accessing Arcadia FHIR APIs for Testing**

Development Sandbox FHIR APIs

The FHIR APIs deployed in the development sandbox are for integration, testing, and customer experimentation. The data contained in the sandbox is fictitious and does not contain any PHI/PII data of any customer. The data and the APIs published here is to get customers, internal stakeholders, development, testing, and integration teams familiar with FHIR.

The following are the steps to understand how to access the FHIR APIs

1. Choose a FHIR Client App that can be used for experimentation. If you do not have a FHIR Client, you can use [POSTMAN](https://www.postman.com/downloads/). The rest of the document assumes the usage of Postman application.
2. Import the Arcadia provided Postman collection and Environment
3. Import Collection

Text

Description automatically generatedGo To Collection, Click on Import, Click on Upload Files and upload the given collection

Fig-1: Import Collection

1. Import Environment

A screenshot of a computer

Description automatically generatedGo to Environments, Click on Import, Click on Upload Files and upload the given environment file.

Fig-2: Import Environment

1. Generate JWT using Client Id and Private Key
   1. Get Client Id and Private Key from your Arcadia Customer Representative to be used for accessing the FHIR APIs in the Development Sandbox.
   2. Create JWT using **client Id** and **private key** by following below steps
   3. Go to <https://jwt.io/#debugger-io>
   4. Set **Header** as below

{

"alg": "RS384",

"typ": "JWT"

}

* 1. Set **Payload** as below

{

"aud": "arcadia",

"iss": Client Id provided by Arcadia,

"sub": Client Id provided by Arcadia,

"jti": Unique UUID,

"iat": Current timestamp in [Unix epoch time](https://www.epochconverter.com/clock).

"exp": Current timestamp + 5 minutes(300 seconds)

}

Graphical user interface, text, application, chat or text message

Description automatically generated

Fig-3: Header and Payload

* 1. Open the provided private key file using some text editor and copy the content and paste it in Private Key block under ‘Verify Signature’ and make sure to erase everything from public key block. In the Encoded block JWT token will be generated.

Graphical user interface, text, application

Description automatically generated

Fig-4: Adding Private Key

1. Get access token from Auth Server using signed JWT token.
   1. Go to Arcadia provided Postman collection
   2. Go to Token EndPoint request under Get Access Token folder
   3. Go to Body
      1. Replace **client\_assertion** with the generated JWT token
      2. Replace **client\_id** with Arcadia provided client id.
      3. Set **scope** as per your requirement. The allowed scopes are system/AllergyIntolerance.read,system/CarePlan.read,system/CareTeam.read,system/Claim.read,system/ClaimResponse.read,system/Condition.read,system/Coverage.read,system/Device.read,system/DiagnosticReport.read,system/DocumentReference.read,system/Encounter.read,system/Endpoint.read,system/ExplanationOfBenefit.read,system/FamilyMemberHistory.read,system/Goal.read,system/Group.read,system/HealthcareService.read,system/Immunization.read,system/InsurancePlan.read,system/ListResource.read,system/Location.read,system/MedicationAdministration.read,system/MedicationDispense.read,system/MedicationKnowledge.read,system/MedicationRequest.read,system/Medication.read,system/MedicationStatement.read,system/Observation.read,system/OrganizationAffiliation.read,system/Organization.read,system/Patient.read,system/Practitioner.read,system/PractitionerRole.read,system/Procedure.read,system/Provenance.read,system/RelatedPerson.read,system/ServiceRequest.read,system/\*.read,system/\*.\*
      4. For example, to access Patient APIs we can set scope as system/Patient.read and we can include multiple scopes as comma separated string.
   4. A screenshot of a computer

      Description automatically generatedHit Send & you will be receiving the access token along with scopes and expiry time.

Fig-5: Access token generation

1. Go to the imported environment file.
   1. Update **authorizationToken**“Bearer <access\_token> received in step# 4.d

Graphical user interface, text

Description automatically generated

Fig-6: Authorization Token of FHIR API Environment

1. Access the FHIR API.
2. Select the environment

Text

Description automatically generated

Fig-7: To select the Environment

1. Access the API

Text

Description automatically generated

Fig-8: Access the API

1. Repeat step# 6 for each FHIR API that you wish to experiment with.
2. Repeat from step# 3, if token is expired.

1. **Testing Manual for UI of FHIR API**
2. Click on the link of <https://authserver.development.arcadiaanalytics.com/ix-auth-server/#/> the login page should be navigated.

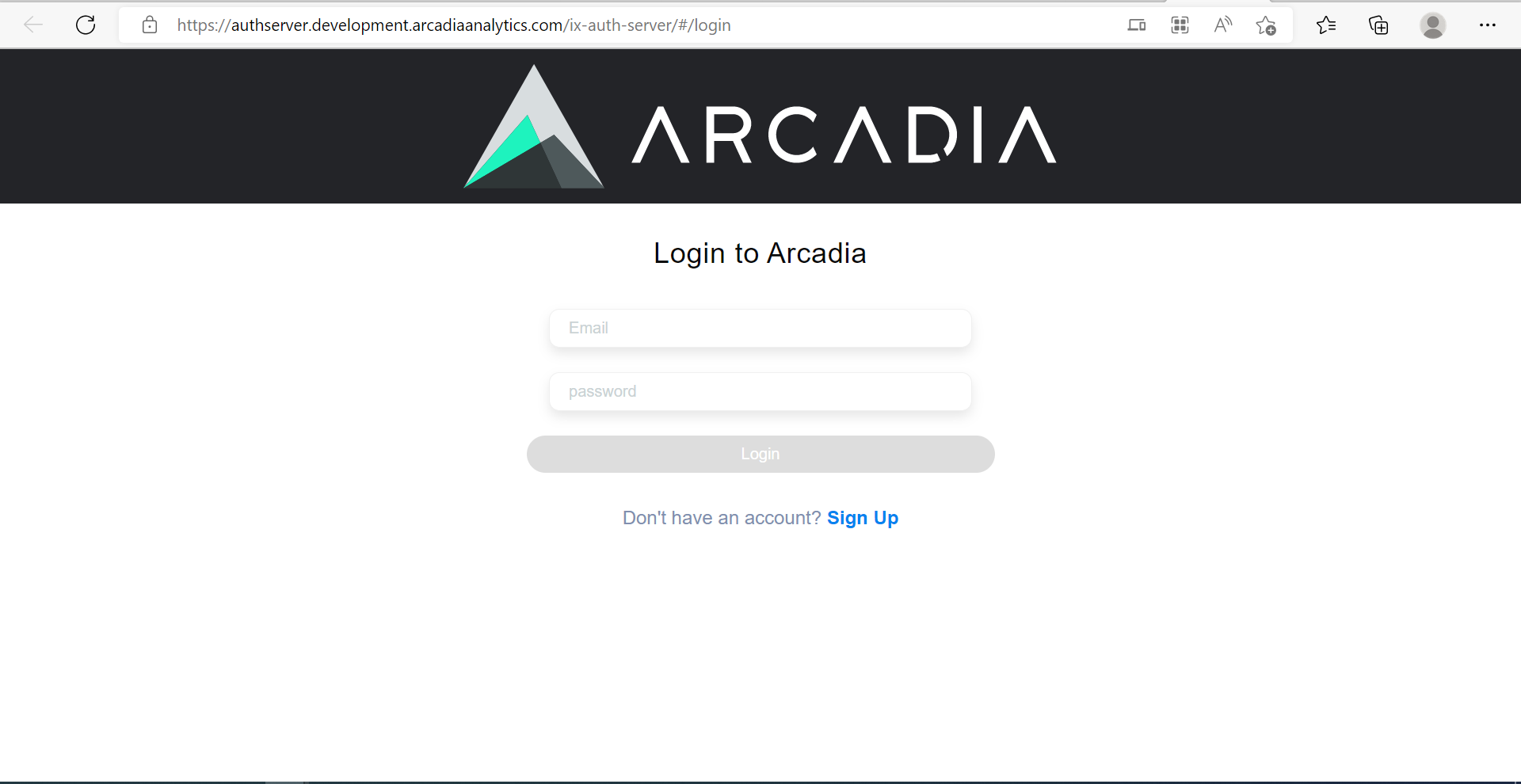


Fig-1: Login page

1. Sign up to the Arcadia account by entering the following details.

* In the signup page enter Email, First Name, Middle Name, Last Name, Mobile Number, Password and Confirm Password.
* Except Middle Name all fields are mandatory.

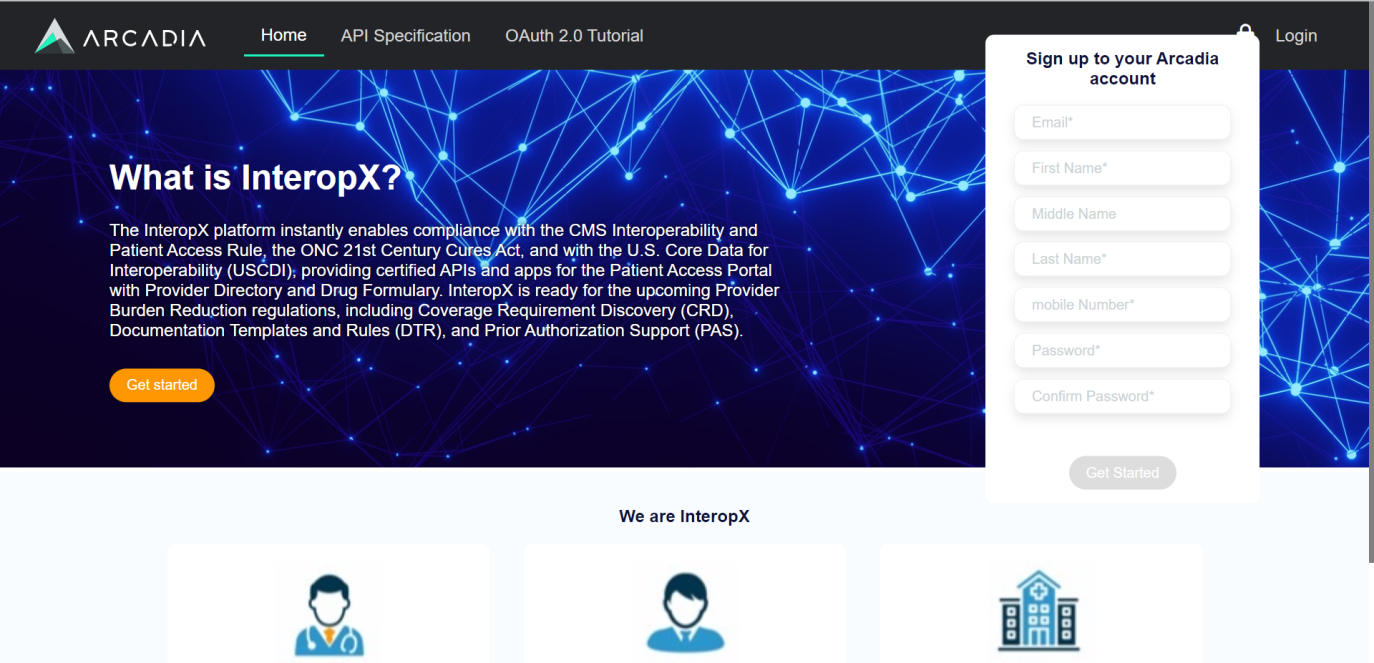


Fig-2: Sign up page

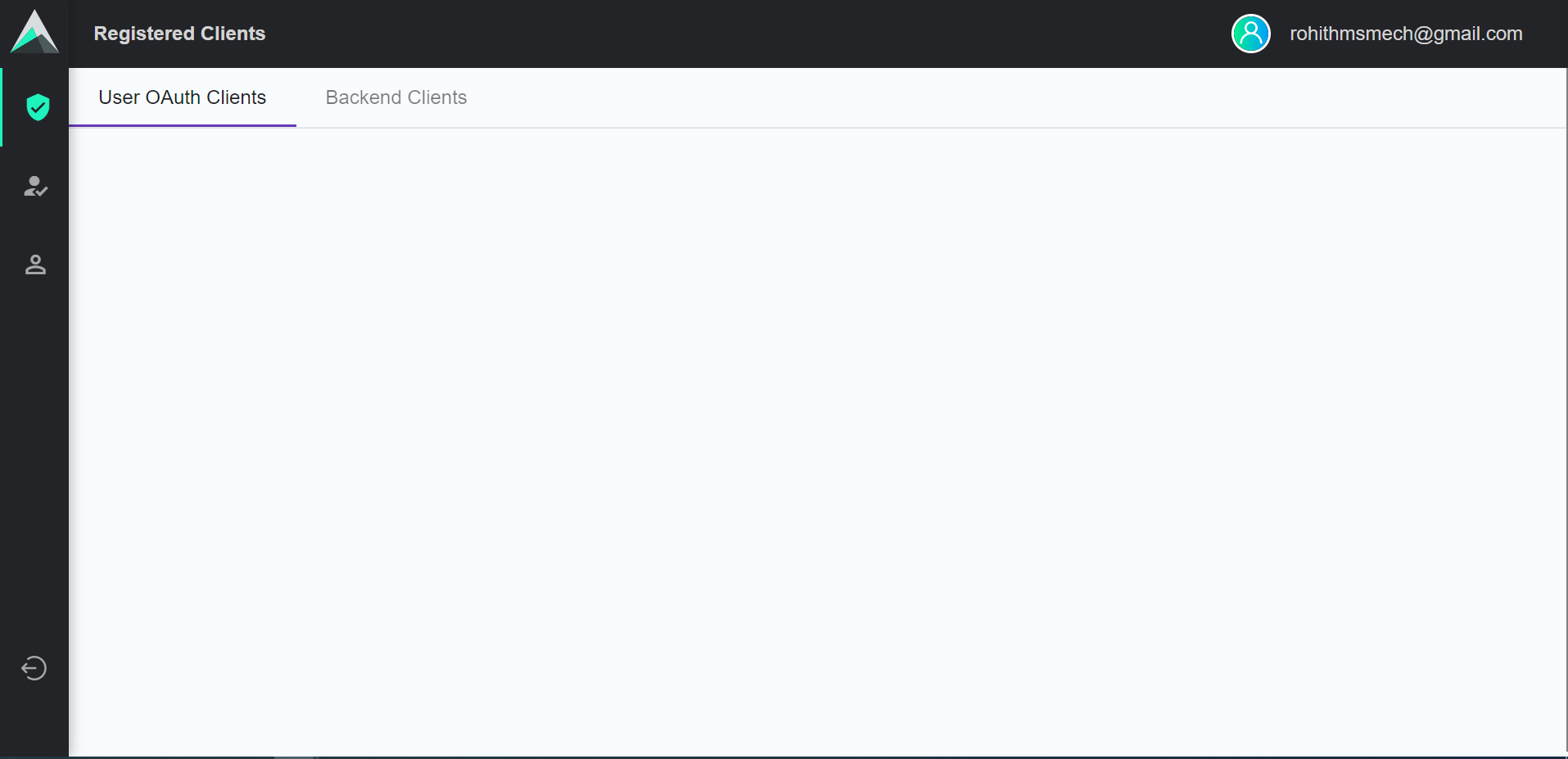
1. Once the Sign up is done it will navigate to the login page. Enter valid Email and Password it will navigate to the Registered Clients page.

Fig-3: Registered Clients page

1. Mouse over and select Backend Client Registration then enter the following details.

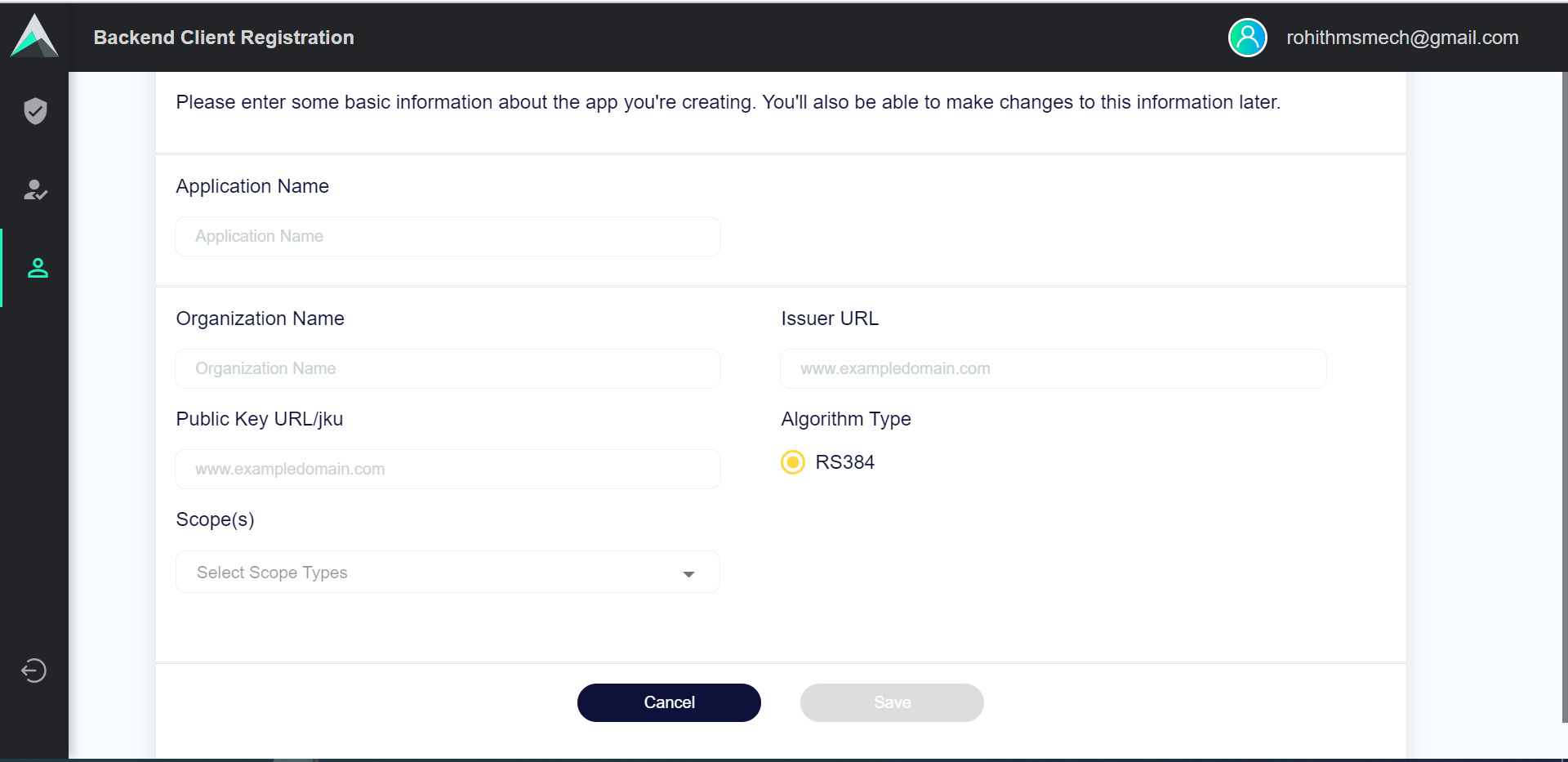
* In Backend Client Registration page enter Application Name, Organization Name and select Algorithm Type as RS384.
* Enter Public Key URL and Issuer URL as <https://dev.interopx.com/docs/RS384.public.json> also select Scopes as All and Save the details for Backend Client Registration.

Fig-4: Backend Client Registration page

1. Once the Backend Client Registration is done mouse over to the Registered Clients and select Backend Clients. Once the approval is done by admin it should display as Approved with the unique Client ID.

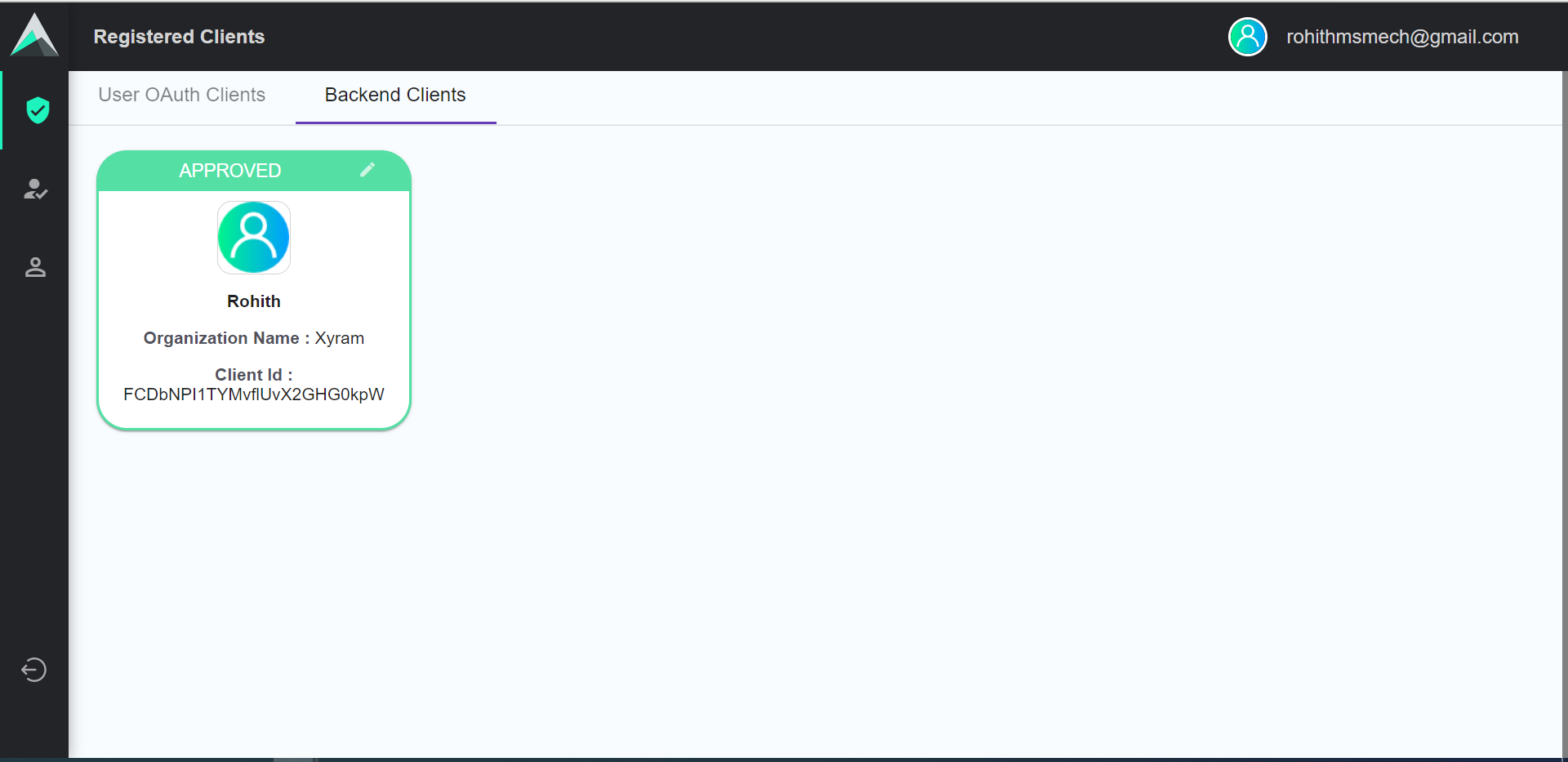


Fig-5: Approved Backend Clients page