Sprint 4

Universidad Nacional de Colombia

Docente a cargo
Brayan Steven Garcia

Integrantes
Anderson Daza
Santiago Rodiguez
Laura Patarroyo
Julian Mora

Grupo
P14

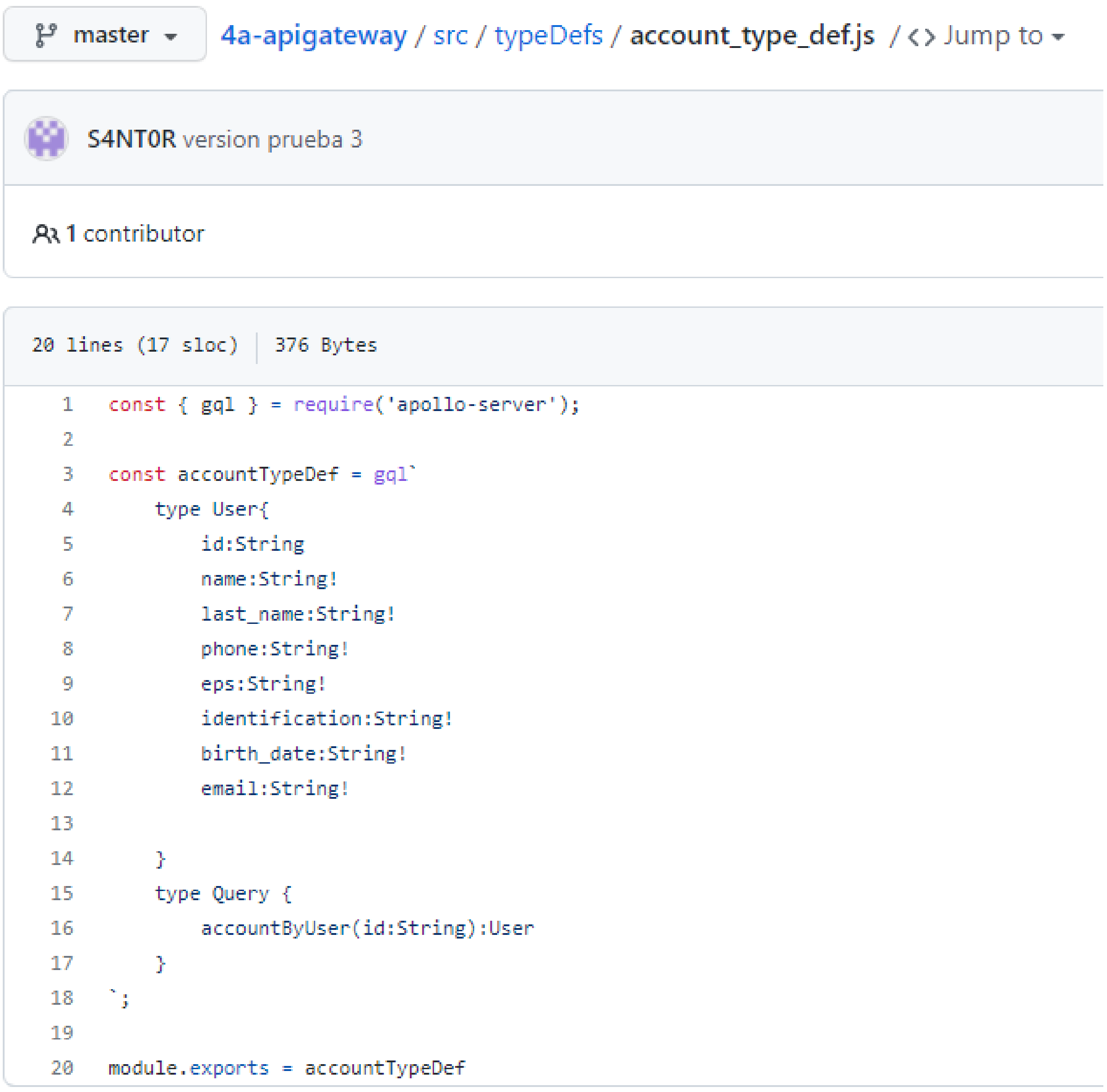
Grupo de trabajo
3

Repositorio:
https://github.com/4a-docs/4a-apigateway

Link Despliegue:
https://apigateway-healthbook.herokuapp.com

Typedefs:

account_type_def.js



master 4a-apigateway / src / typeDefs / account_type_def.js / <> Jump to

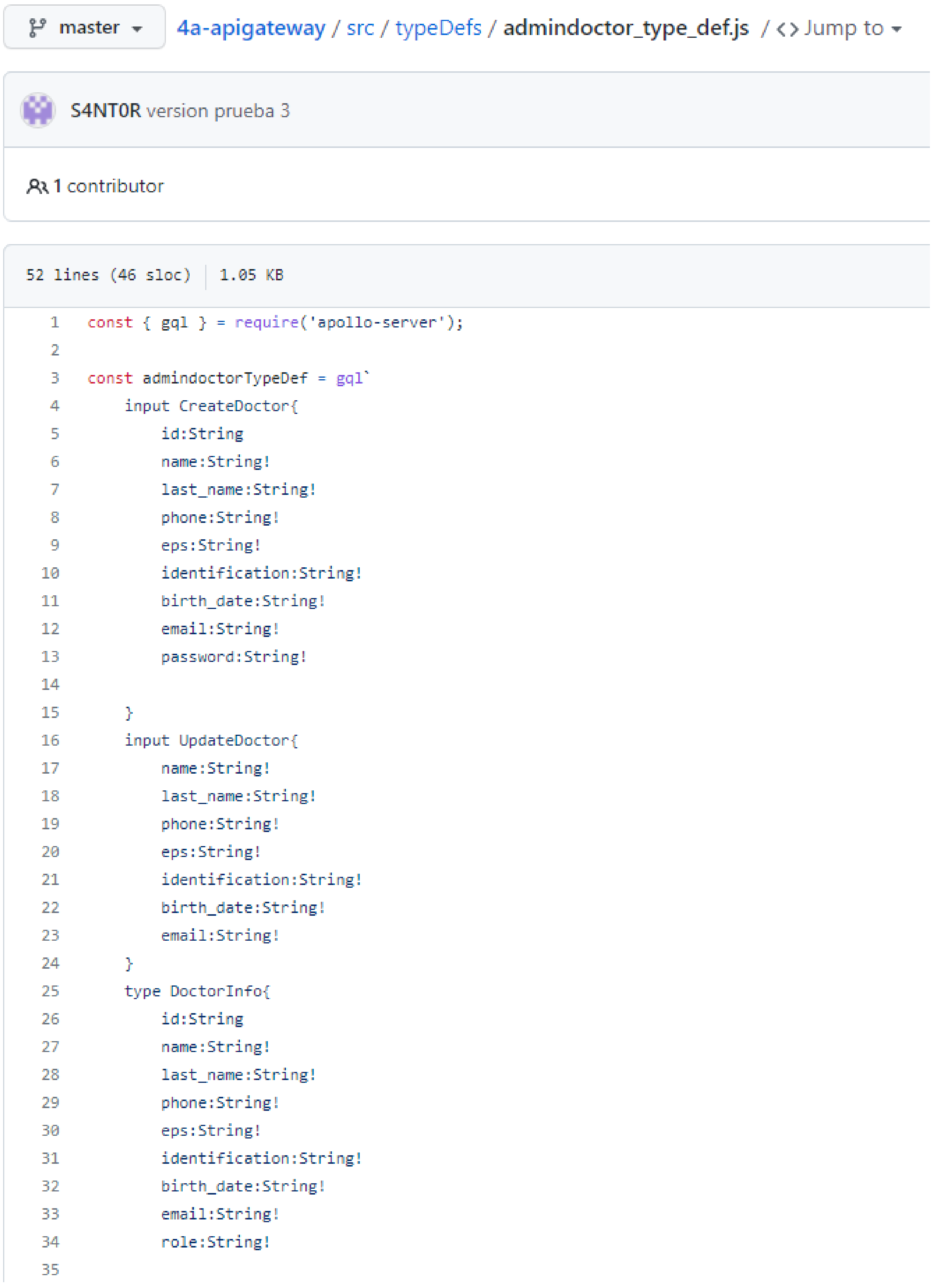
S4NT0R version prueba 3

1 contributor

20 lines (17 sloc) | 376 Bytes

```
const { gql } = require('apollo-server');

const accountTypeDef = gql`
    type User{
        id:String
        name:String!
        last_name:String!
        phone:String!
        eps:String!
        identification:String!
        birth_date:String!
        email:String!

    }
    type Query {
        accountByUser(id:String):User
    }
`;

module.exports = accountTypeDef
```

admindoctor_type_def.js



master 4a-apigateway / src / typeDefs / admindoctor_type_def.js / Jump to

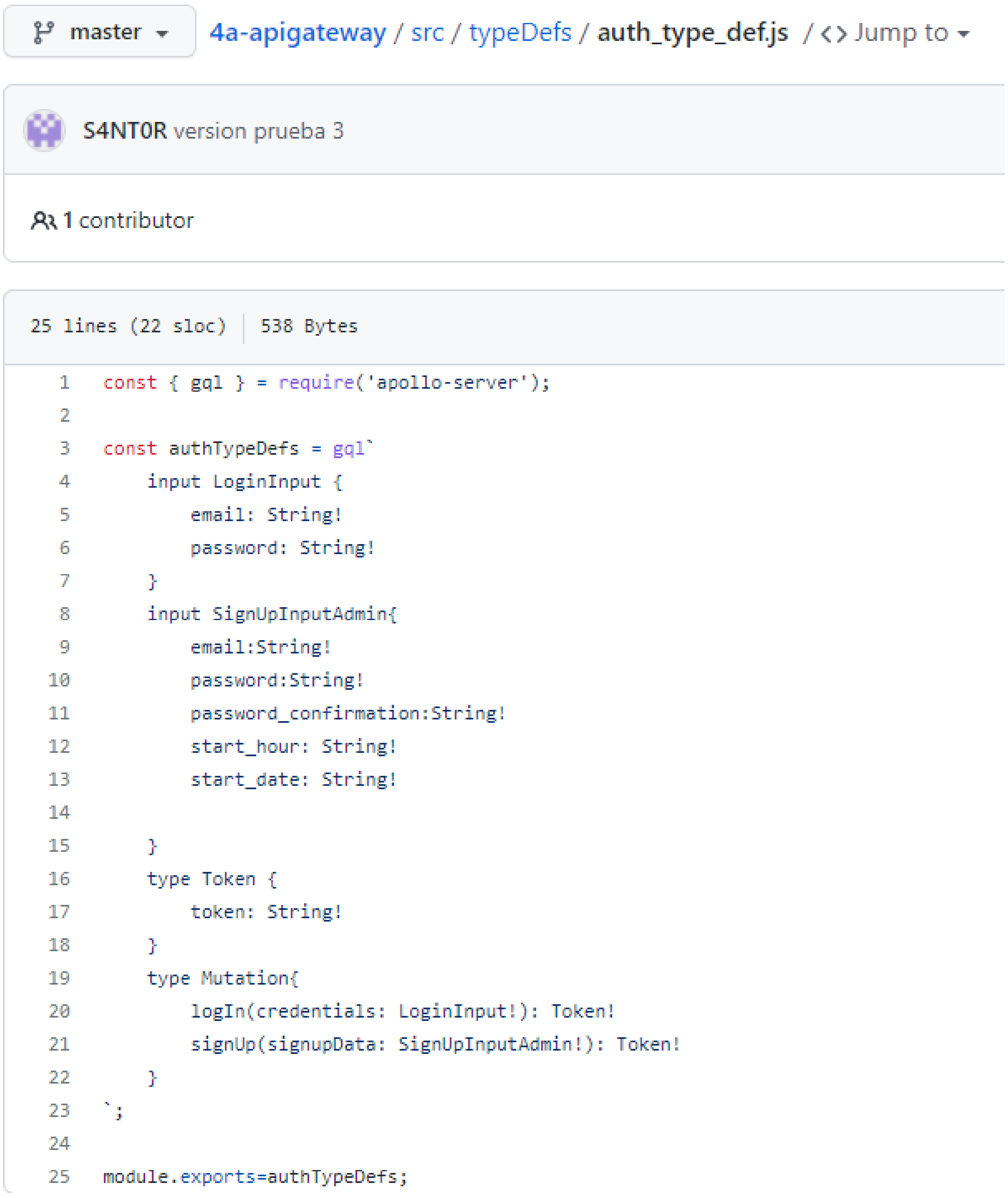
S4NT0R version prueba 3

1 contributor

52 lines (46 sloc) | 1.05 KB

```
const { gql } = require('apollo-server');

const admindoctorTypeDef = gql`
    input CreateDoctor{
        id:String
        name:String!
        last_name:String!
        phone:String!
        eps:String!
        identification:String!
        birth_date:String!
        email:String!
        password:String!

    }
    input UpdateDoctor{
        name:String!
        last_name:String!
        phone:String!
        eps:String!
        identification:String!
        birth_date:String!
        email:String!
    }
    type DoctorInfo{
        id:String
        name:String!
        last_name:String!
        phone:String!
        eps:String!
        identification:String!
        birth_date:String!
        email:String!
        role:String!

    }
    type Query {
        doctorAccount(id:String!):User
        doctorAccounts(id:String):[User]
    }
    type Token {
        key: String!
    }

    type Mutation{
        CreateDoctor(data: CreateDoctor!):DoctorInfo
        UpdateDoctor(data: UpdateDoctor!):DoctorInfo

    }
`;

module.exports = admindoctorTypeDef
```

auth_type_def.js


master

4a-apigateway / src / typeDefs / auth_type_def.js / Jump to

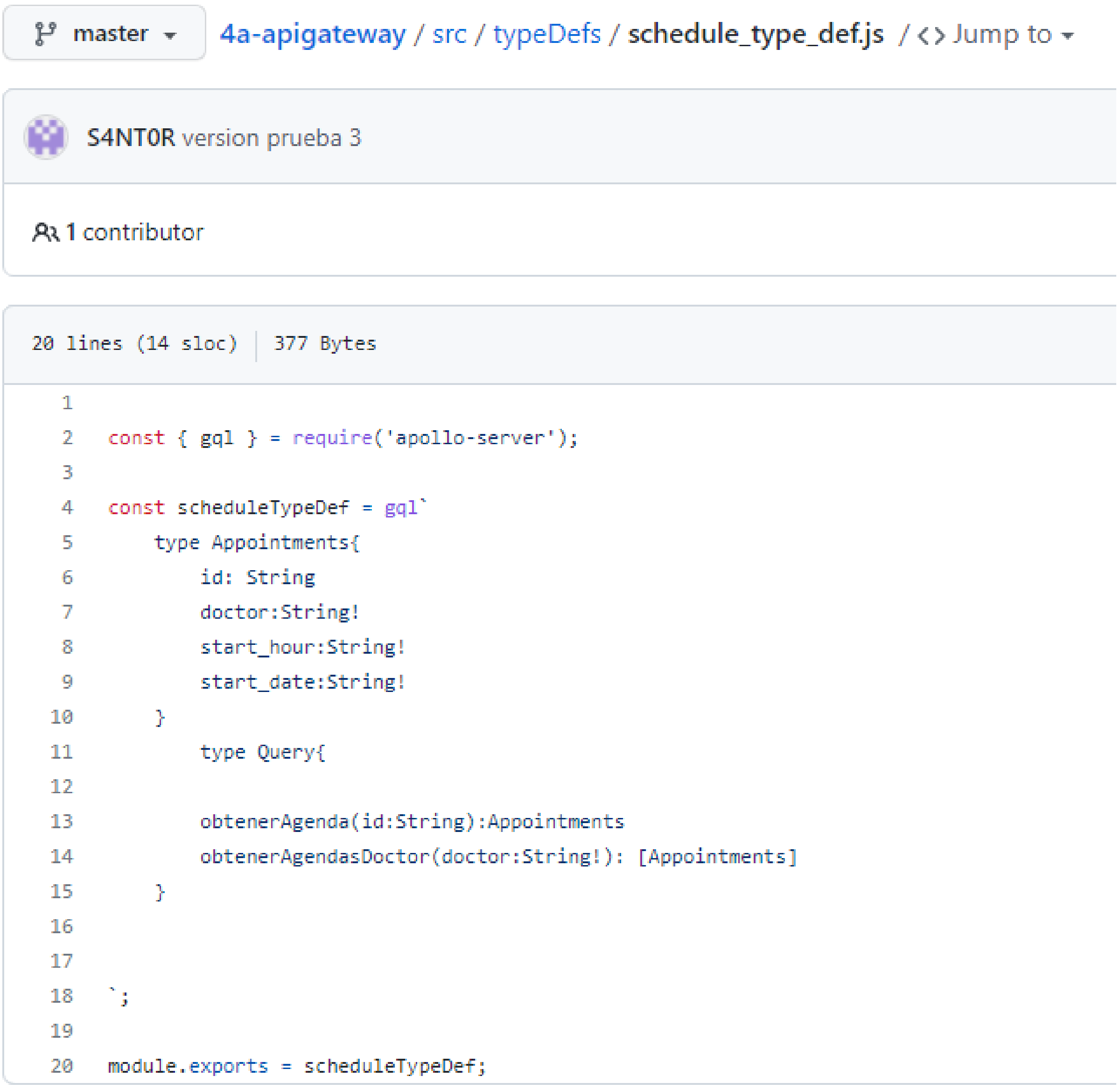
S4NT0R version prueba 3

1 contributor

25 lines (22 sloc) | 538 Bytes

```
const { gql } = require('apollo-server');

const authTypeDefs = gql`
    input LoginInput {
        email: String!
        password: String!
    }
    input SignUpInputAdmin{
        email:String!
        password:String!
        password_confirmation:String!
        start_hour: String!
        start_date: String!

    }
    type Token {
        token: String!
    }
    type Mutation{
        logIn(credentials: LoginInput!): Token!
        signUp(signupData: SignUpInputAdmin!): Token!
    }
`;

module.exports=authTypeDefs;
```

schedule_type_def.js



master 4a-apigateway / src / typeDefs / schedule_type_def.js / <> Jump to

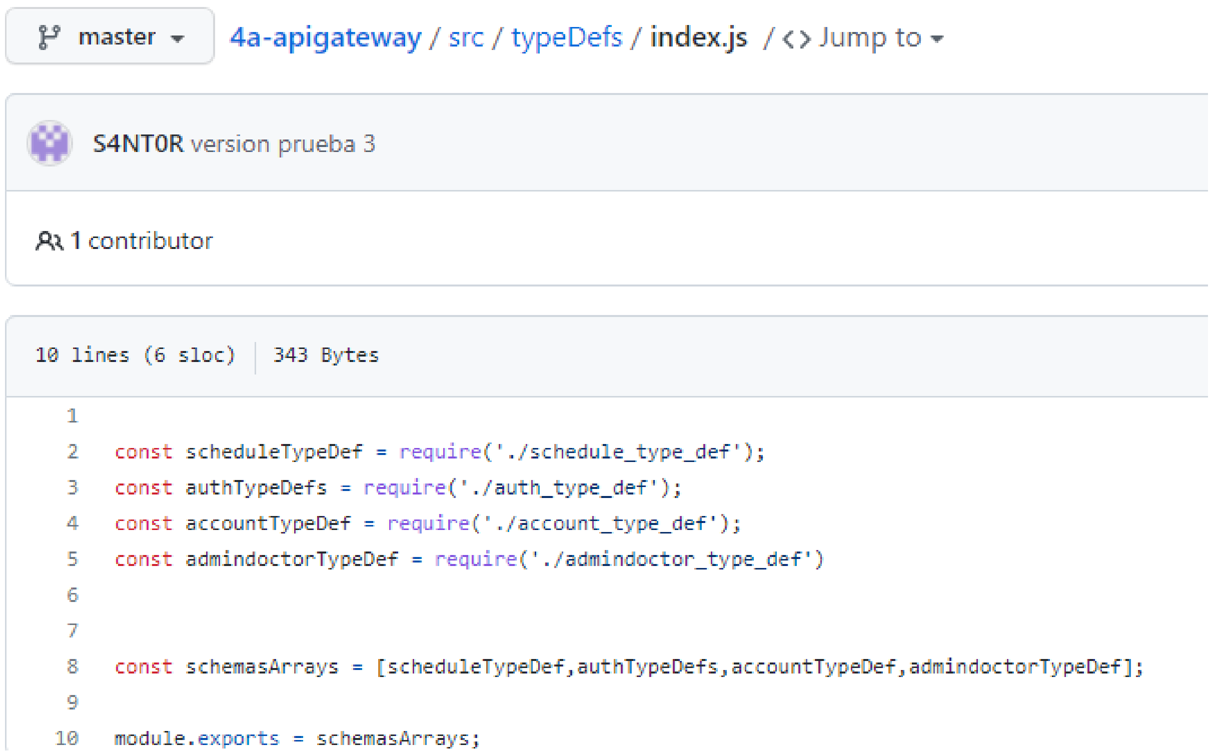
S4NT0R version prueba 3

1 contributor

20 lines (14 sloc) | 377 Bytes

```

const { gql } = require('apollo-server');

const scheduleTypeDef = gql`
    type Appointments{
        id: String
        doctor:String!
        start_hour:String!
        start_date:String!
    }
        type Query{

        obtenerAgenda(id:String):Appointments
        obtenerAgendasDoctor(doctor:String!): [Appointments]
    }


`;

module.exports = scheduleTypeDef;
```

index.js


master 4a-apigateway / src / typeDefs / index.js / <> Jump to

S4NT0R version prueba 3

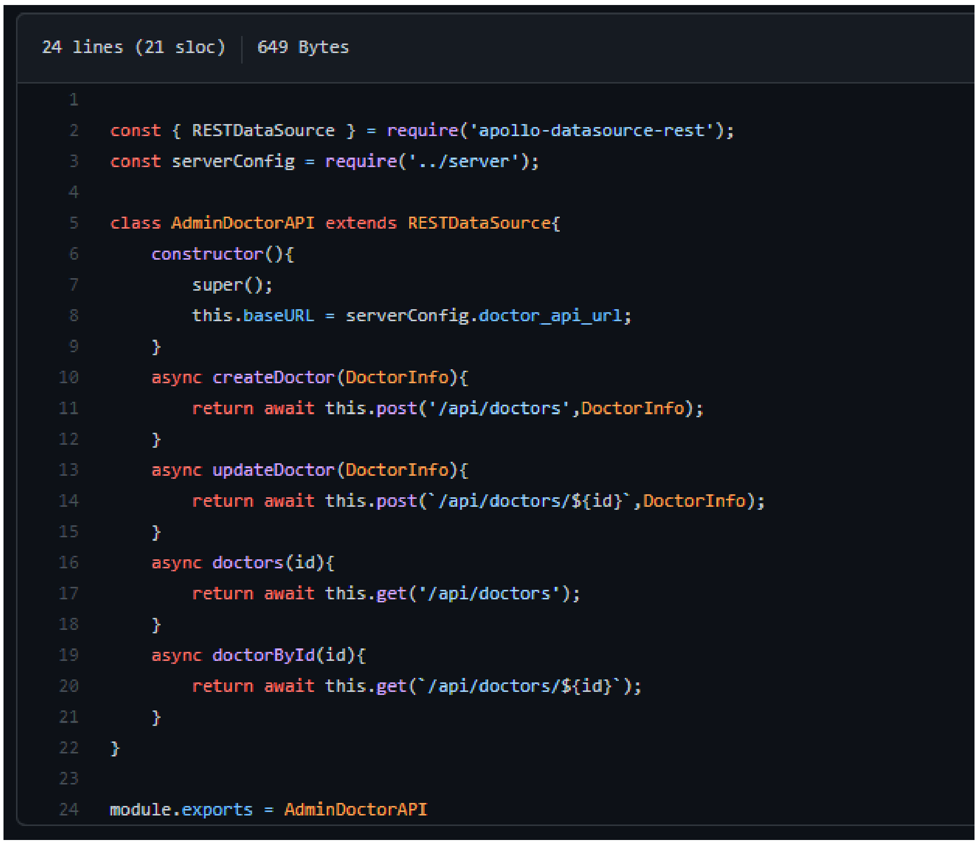
1 contributor

10 lines (6 sloc) | 343 Bytes

```javascript

const scheduleTypeDef = require('./schedule_type_def');
const authTypeDefs = require('./auth_type_def');
const accountTypeDef = require('./account_type_def');
const admindoctorTypeDef = require('./admindoctor_type_def')


const schemasArrays = [scheduleTypeDef,authTypeDefs,accountTypeDef,admindoctorTypeDef];

module.exports = schemasArrays;
```

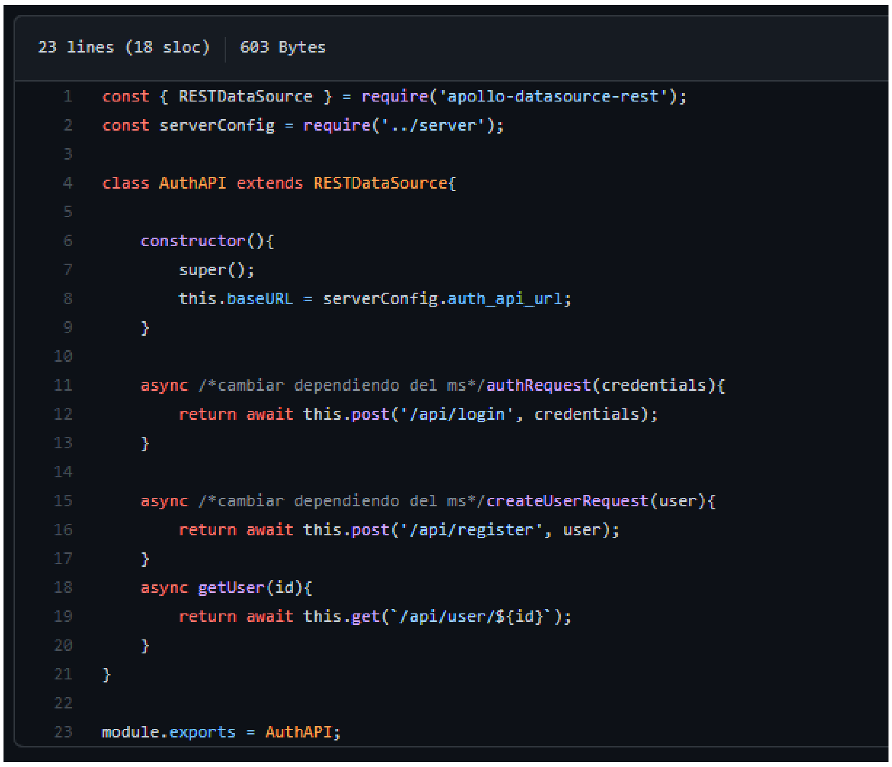
Datatypes:

- **admindoctor_api.js**



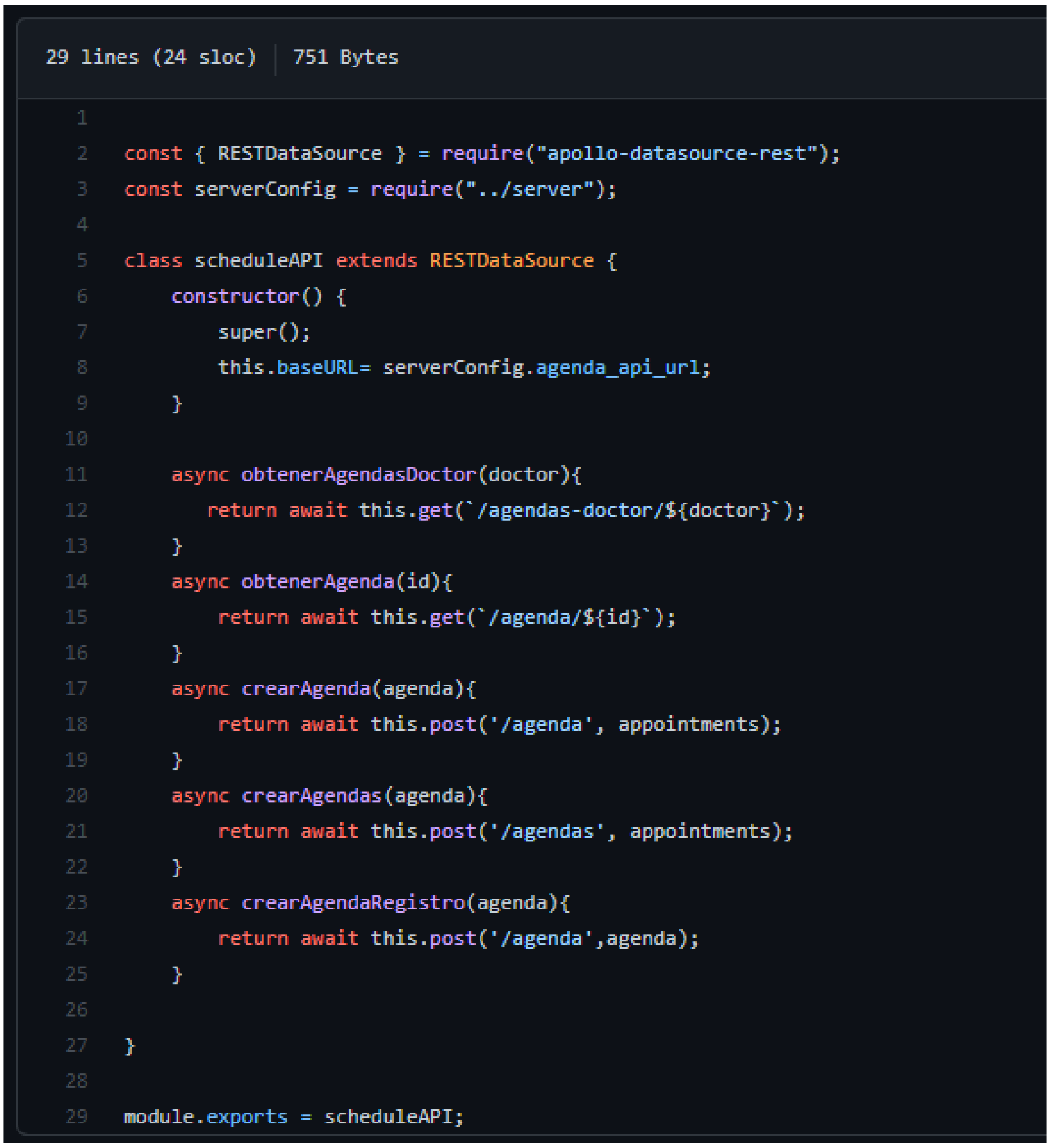
```
24 lines (21 sloc) | 649 Bytes

const { RESTDataSource } = require('apollo-datasource-rest');
const serverConfig = require('../server');

class AdminDoctorAPI extends RESTDataSource{
    constructor(){
        super();
        this.baseURL = serverConfig.doctor_api_url;
    }
    async createDoctor(DoctorInfo){
        return await this.post('/api/doctors',DoctorInfo);
    }
    async updateDoctor(DoctorInfo){
        return await this.post(`/api/doctors/${id}`,DoctorInfo);
    }
    async doctors(id){
        return await this.get('/api/doctors');
    }
    async doctorById(id){
        return await this.get(`/api/doctors/${id}`);
    }
}

module.exports = AdminDoctorAPI
```

- **auth_api.js**



```
23 lines (18 sloc) | 603 Bytes
const { RESTDataSource } = require('apollo-datasource-rest');
const serverConfig = require('../server');

class AuthAPI extends RESTDataSource{

    constructor(){
        super();
        this.baseURL = serverConfig.auth_api_url;
    }

    async /*cambiar dependiendo del ms*/authRequest(credentials){
        return await this.post('/api/login', credentials);
    }

    async /*cambiar dependiendo del ms*/createUserRequest(user){
        return await this.post('/api/register', user);
    }
    async getUser(id){
        return await this.get(`/api/user/${id}`);
    }
}

module.exports = AuthAPI;
```

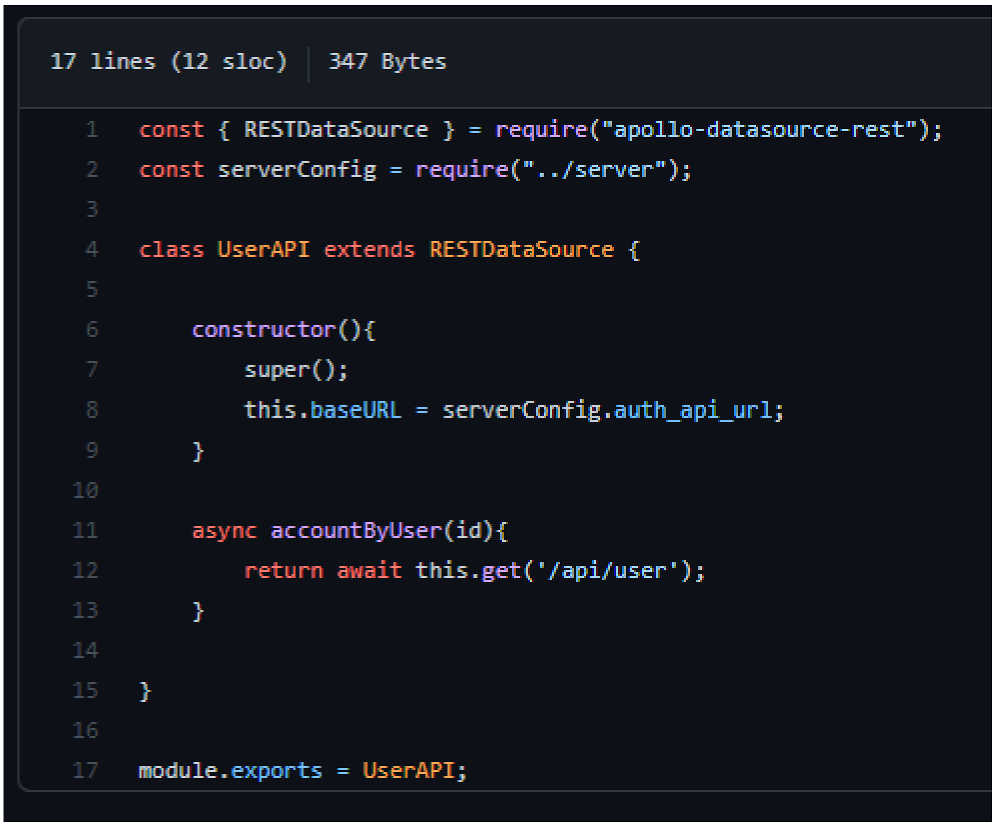
- **schedule_api.js**



29 lines (24 sloc) | 751 Bytes

```javascript

const { RESTDataSource } = require("apollo-datasource-rest");
const serverConfig = require("../server");

class scheduleAPI extends RESTDataSource {
    constructor() {
        super();
        this.baseURL= serverConfig.agenda_api_url;
    }

    async obtenerAgendasDoctor(doctor){
       return await this.get(`/agendas-doctor/${doctor}`);
    }
    async obtenerAgenda(id){
        return await this.get(`/agenda/${id}`);
    }
    async crearAgenda(agenda){
        return await this.post('/agenda', appointments);
    }
    async crearAgendas(agenda){
        return await this.post('/agendas', appointments);
    }
    async crearAgendaRegistro(agenda){
        return await this.post('/agenda',agenda);
    }

}

module.exports = scheduleAPI;
```

- user_api.js

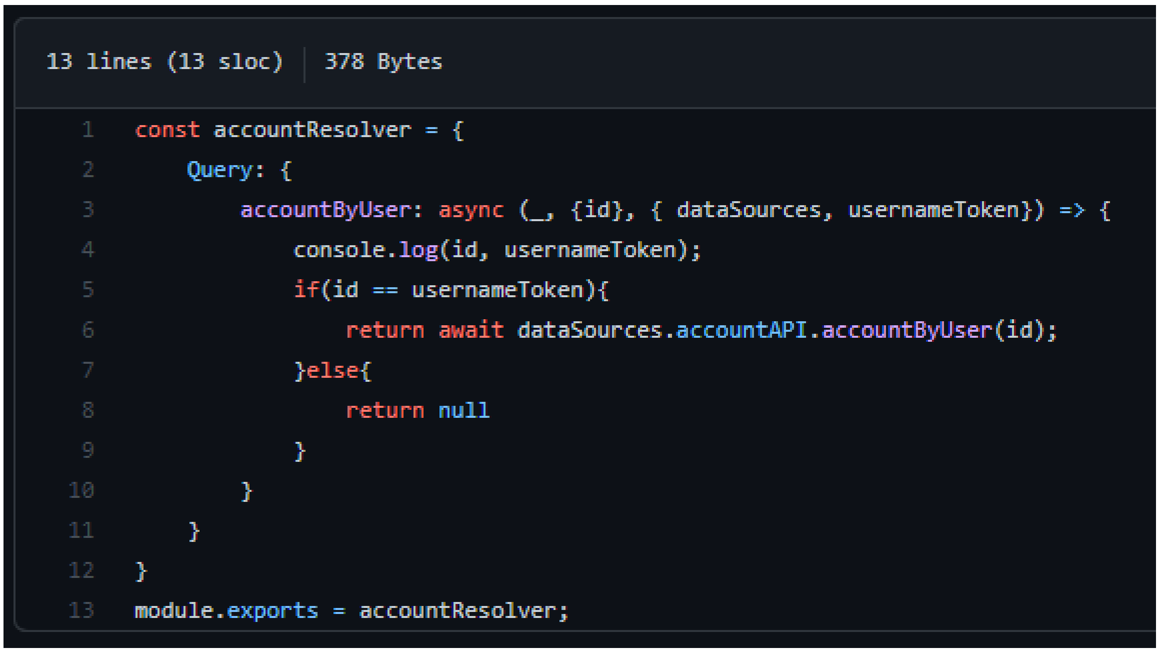


```
17 lines (12 sloc) | 347 Bytes
```

```javascript
const { RESTDataSource } = require("apollo-datasource-rest");
const serverConfig = require("../server");

class UserAPI extends RESTDataSource {

    constructor(){
        super();
        this.baseURL = serverConfig.auth_api_url;
    }

    async accountByUser(id){
        return await this.get('/api/user');
    }

}

module.exports = UserAPI;
```

Resolvers:

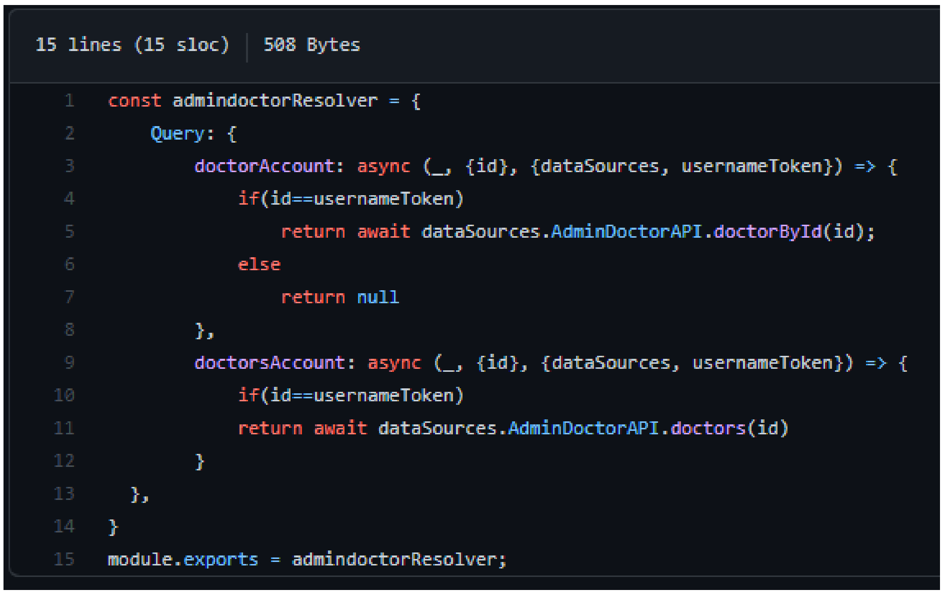
- account_resolver.js



```
13 lines (13 sloc) | 378 Bytes
```

```javascript
const accountResolver = {
    Query: {
        accountByUser: async (_, {id}, { dataSources, usernameToken}) => {
            console.log(id, usernameToken);
            if(id == usernameToken){
                return await dataSources.accountAPI.accountByUser(id);
            }else{
                return null
            }
        }
    }
}
module.exports = accountResolver;
```

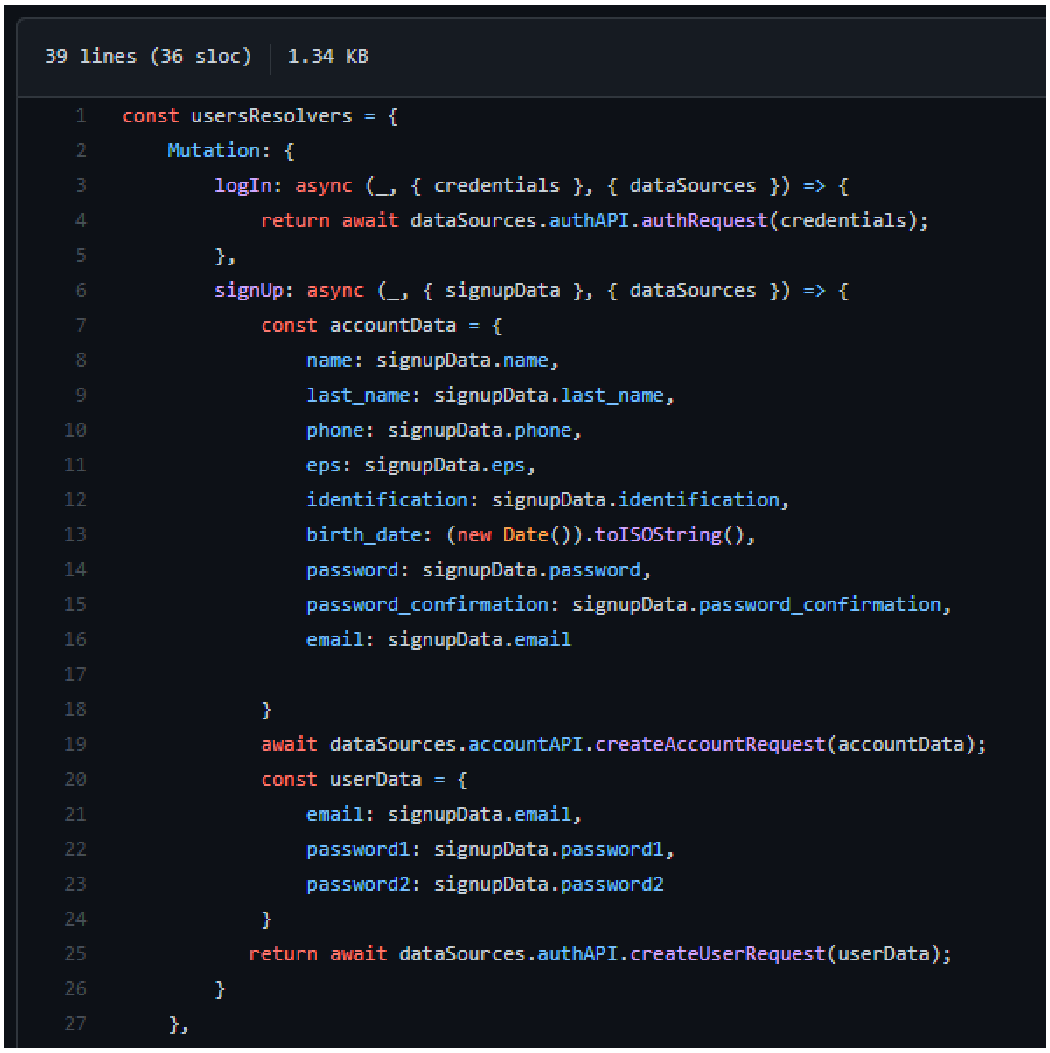
- **admin_doctor.js**



```
15 lines (15 sloc) | 508 Bytes
```

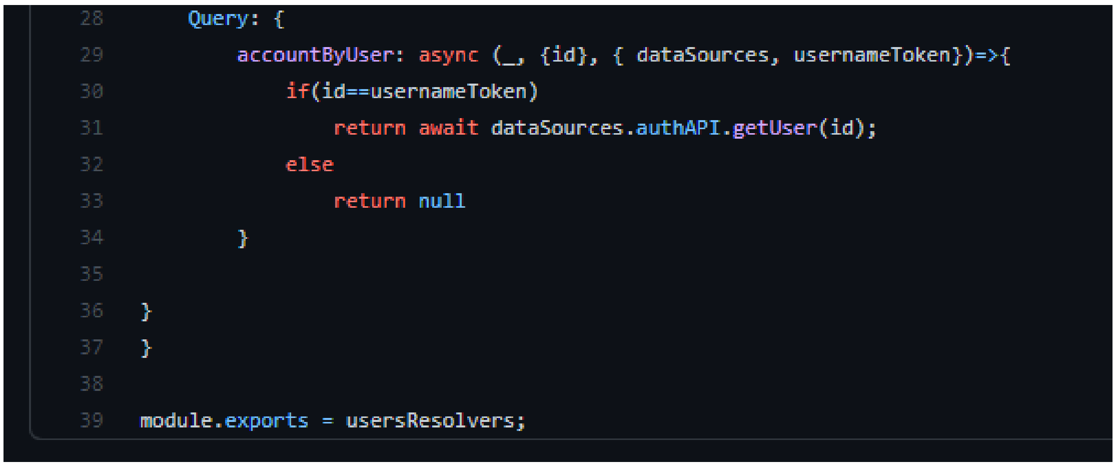
```javascript
const admindoctorResolver = {
    Query: {
        doctorAccount: async (_, {id}, {dataSources, usernameToken}) => {
            if(id==usernameToken)
                return await dataSources.AdminDoctorAPI.doctorById(id);
            else
                return null
        },
        doctorsAccount: async (_, {id}, {dataSources, usernameToken}) => {
            if(id==usernameToken)
            return await dataSources.AdminDoctorAPI.doctors(id)
        }
    },
}
module.exports = admindoctorResolver;
```

- **auth_resolver.js**

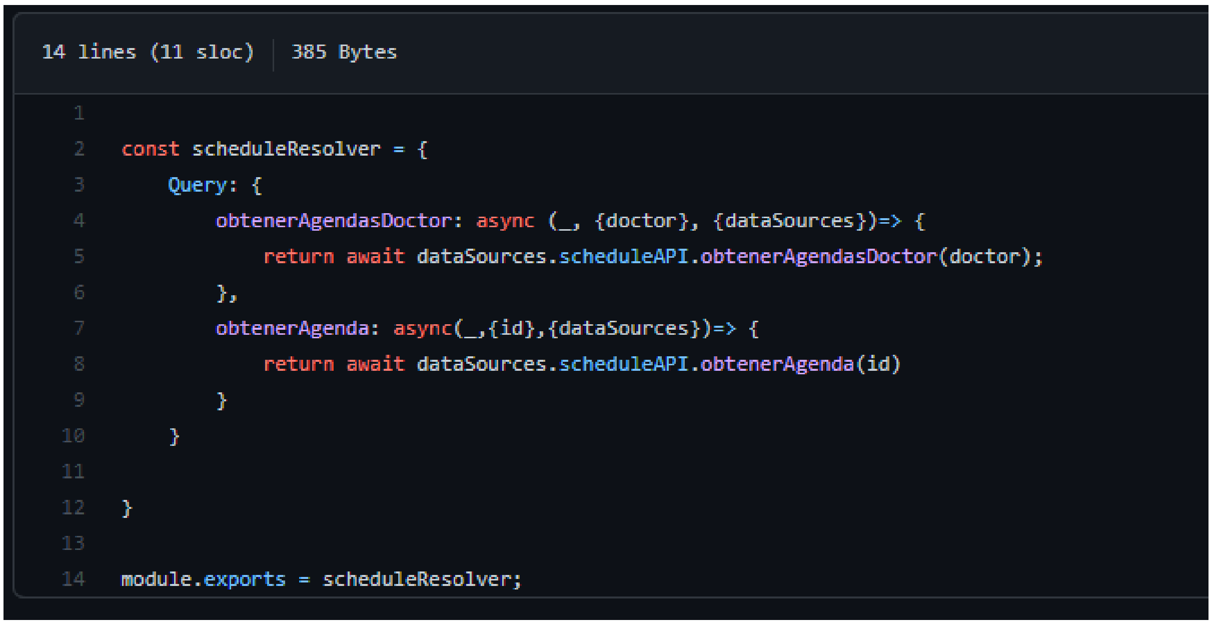


```
39 lines (36 sloc) | 1.34 KB
```

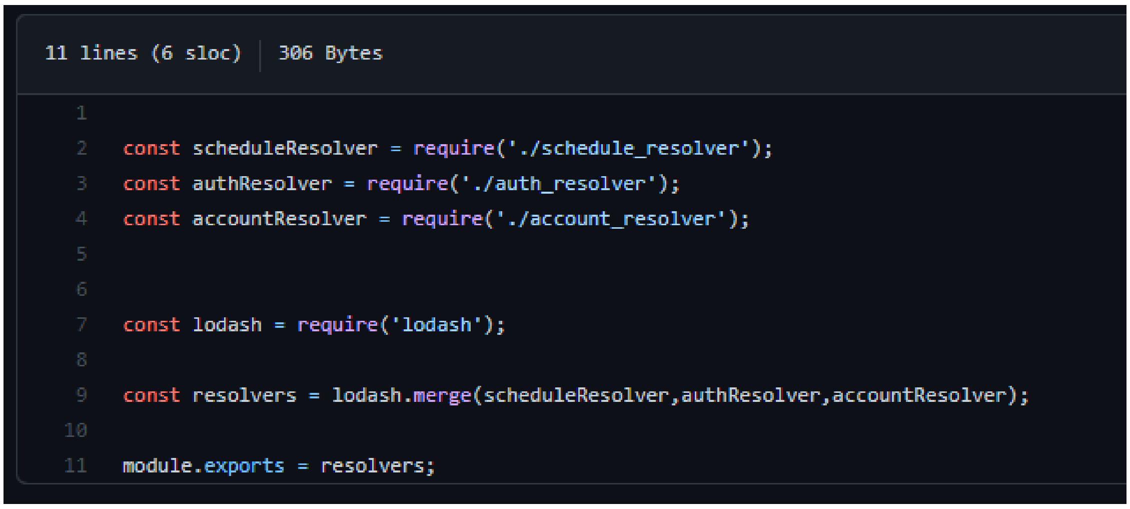
```javascript
const usersResolvers = {
    Mutation: {
        logIn: async (_, { credentials }, { dataSources }) => {
            return await dataSources.authAPI.authRequest(credentials);
        },
        signUp: async (_, { signupData }, { dataSources }) => {
            const accountData = {
                name: signupData.name,
                last_name: signupData.last_name,
                phone: signupData.phone,
                eps: signupData.eps,
                identification: signupData.identification,
                birth_date: (new Date()).toISOString(),
                password: signupData.password,
                password_confirmation: signupData.password_confirmation,
                email: signupData.email

            }
            await dataSources.accountAPI.createAccountRequest(accountData);
            const userData = {
                email: signupData.email,
                password1: signupData.password1,
                password2: signupData.password2
            }
           return await dataSources.authAPI.createUserRequest(userData);
        }
    },
```


```
    Query: {
        accountByUser: async (_, {id}, { dataSources, usernameToken})=>{
            if(id==usernameToken)
                return await dataSources.authAPI.getUser(id);
            else
                return null
        }

}
}

module.exports = usersResolvers;
```

- **schedule_resolver.js**

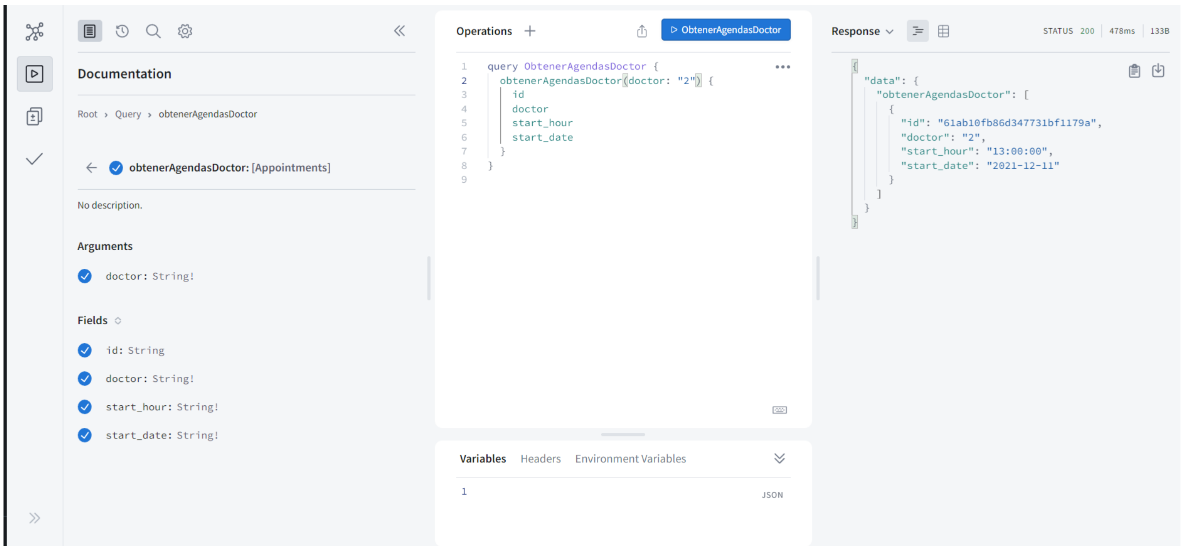

```
14 lines (11 sloc)  |  385 Bytes

const scheduleResolver = {
    Query: {
        obtenerAgendasDoctor: async (_, {doctor}, {dataSources})=> {
            return await dataSources.scheduleAPI.obtenerAgendasDoctor(doctor);
        },
        obtenerAgenda: async(_,{id},{dataSources})=> {
            return await dataSources.scheduleAPI.obtenerAgenda(id)
        }
    }

}

module.exports = scheduleResolver;
```

- **resolvers/index.js**


```
11 lines (6 sloc)  |  306 Bytes

const scheduleResolver = require('./schedule_resolver');
const authResolver = require('./auth_resolver');
const accountResolver = require('./account_resolver');


const lodash = require('lodash');

const resolvers = lodash.merge(scheduleResolver,authResolver,accountResolver);

module.exports = resolvers;
```

Prueba endpoints Apollo:

Querys:


Documentation

Root › Query › obtenerAgendasDoctor

obtenerAgendasDoctor: [Appointments]

No description.

Arguments

doctor: String!

Fields

id: String

doctor: String!

start_hour: String!

start_date: String!

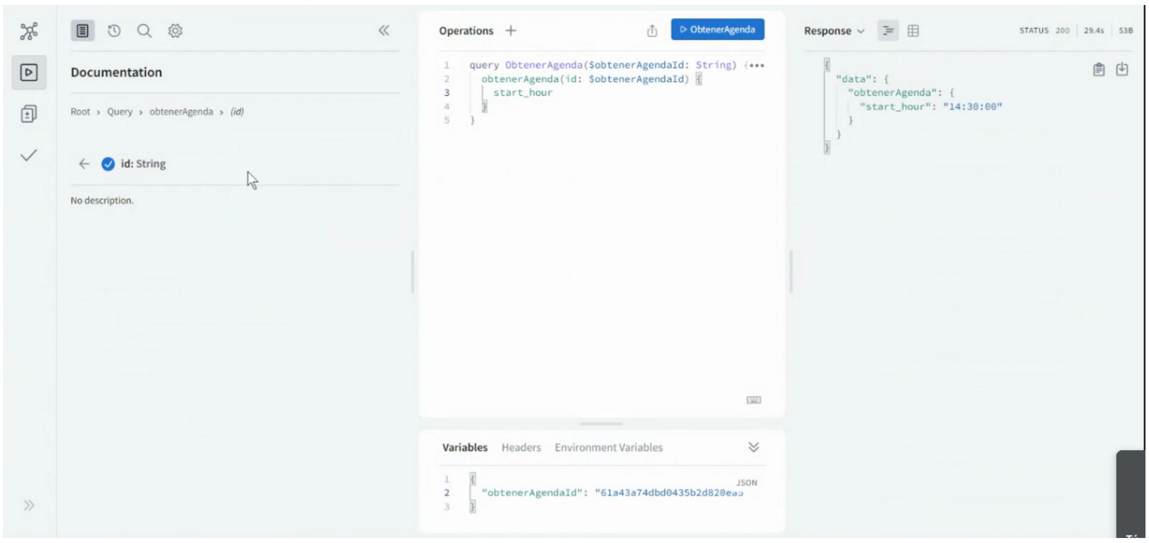
Operations — ObtenerAgendasDoctor

```
query ObtenerAgendasDoctor {
  obtenerAgendasDoctor(doctor: "2") {
    id
    doctor
    start_hour
    start_date
  }
}
```

Response — STATUS 200 | 478ms | 133B

```
{
  "data": {
    "obtenerAgendasDoctor": [
      {
        "id": "61ab10fb86d347731bf1179a",
        "doctor": "2",
        "start_hour": "13:00:00",
        "start_date": "2021-12-11"
      }
    ]
  }
}
```

Variables | Headers | Environment Variables


Documentation

Root › Query › obtenerAgenda › (id)

id: String

No description.

Operations — ObtenerAgenda

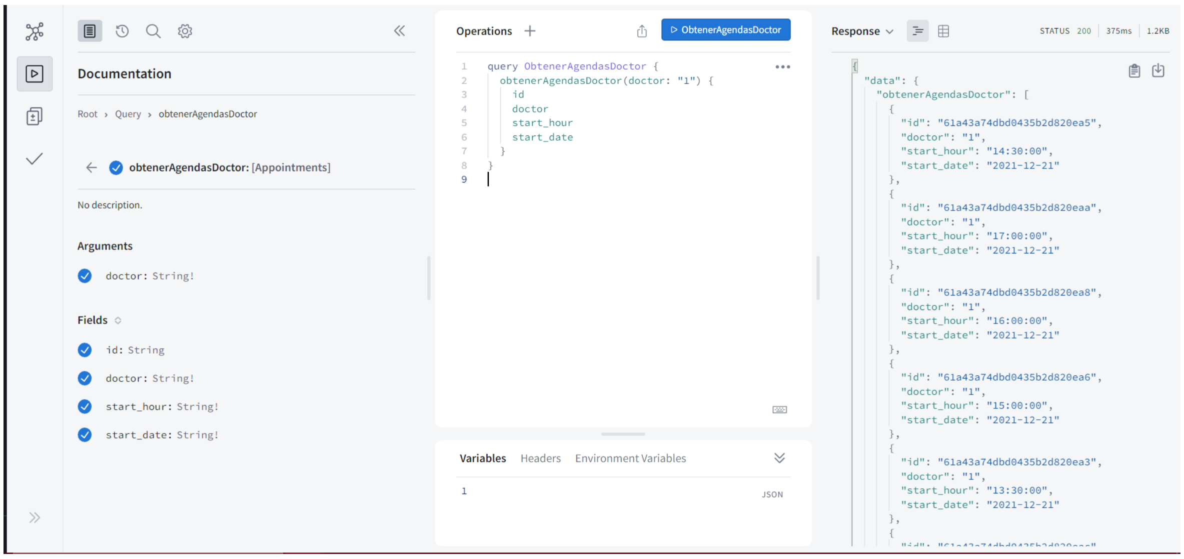
```
query ObtenerAgenda($obtenerAgendaId: String) {
  obtenerAgenda(id: $obtenerAgendaId) {
    start_hour
  }
}
```

Response — STATUS 200 | 29.4s | 53B

```
{
  "data": {
    "obtenerAgenda": {
      "start_hour": "14:30:00"
    }
  }
}
```

Variables | Headers | Environment Variables

```
{
  "obtenerAgendaId": "61a43a74dbd0435b2d820ea5
}
```


Documentation

Root › Query › obtenerAgendasDoctor

obtenerAgendasDoctor: [Appointments]

No description.

Arguments

doctor: String!

Fields

id: String

doctor: String!

start_hour: String!

start_date: String!

Operations — ObtenerAgendasDoctor

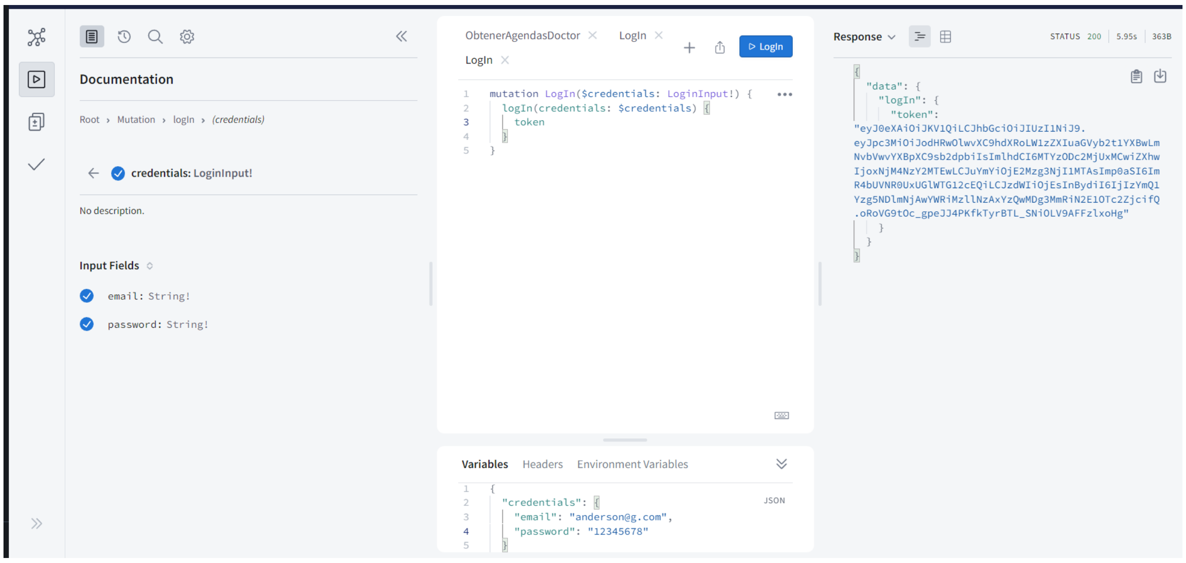
```
query ObtenerAgendasDoctor {
  obtenerAgendasDoctor(doctor: "1") {
    id
    doctor
    start_hour
    start_date
  }
}
```

Response — STATUS 200 | 375ms | 1.2KB

```
{
  "data": {
    "obtenerAgendasDoctor": [
      {
        "id": "61a43a74dbd0435b2d820ea5",
        "doctor": "1",
        "start_hour": "14:30:00",
        "start_date": "2021-12-21"
      },
      {
        "id": "61a43a74dbd0435b2d820eaa",
        "doctor": "1",
        "start_hour": "17:00:00",
        "start_date": "2021-12-21"
      },
      {
        "id": "61a43a74dbd0435b2d820ea8",
        "doctor": "1",
        "start_hour": "16:00:00",
        "start_date": "2021-12-21"
      },
      {
        "id": "61a43a74dbd0435b2d820ea6",
        "doctor": "1",
        "start_hour": "15:00:00",
        "start_date": "2021-12-21"
      },
      {
        "id": "61a43a74dbd0435b2d820ea3",
        "doctor": "1",
        "start_hour": "13:30:00",
        "start_date": "2021-12-21"
      },
      {
```

Variables | Headers | Environment Variables

Mutations:



credentials: LoginInput!

No description.

email: String!

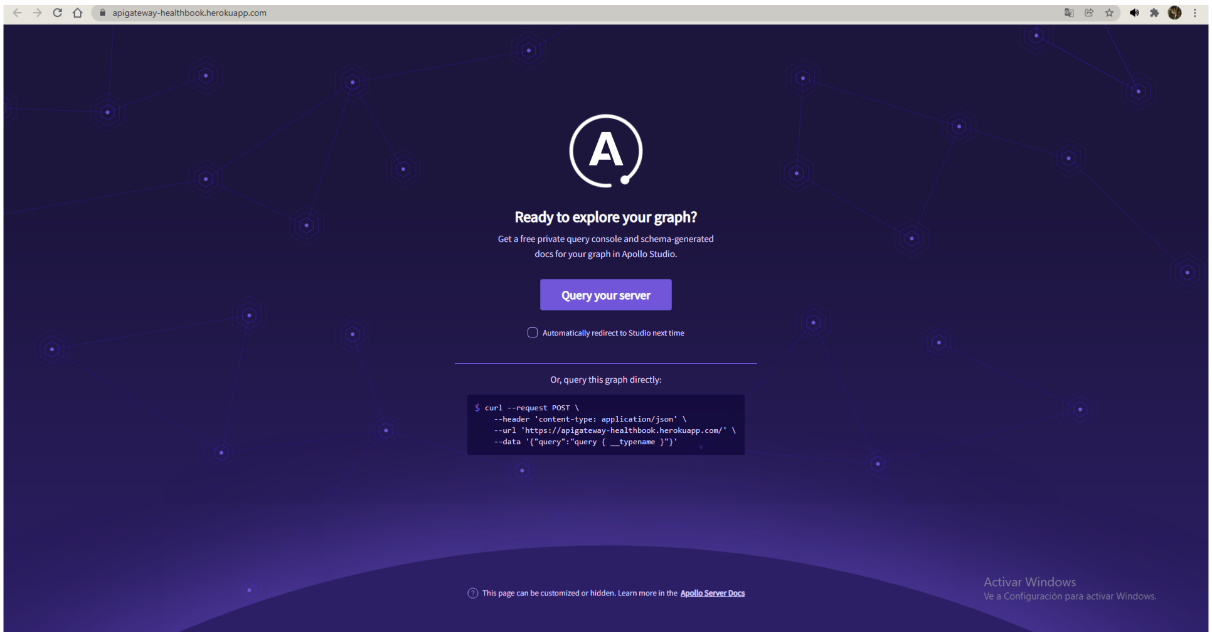
password: String!

```
mutation LogIn($credentials: LoginInput!) {
  logIn(credentials: $credentials) {
    token
  }
}
```

```
{
  "credentials": {
    "email": "anderson@g.com",
    "password": "12345678"
  }
```

```
{
  "data": {
    "logIn": {
      "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJodHRwOlwvXC9hdXRoLW1zZXIuaGVyb2t1YXBwLm
NvbVwvYXBpXC9sb2dpbiIsImlhdCI6MTYzODc2MjUxMCwiZXhw
IjoxNjM4NzY2MTEwLCJuYmYiOjE2Mzg3NjI1MTAsImp0aSI6Im
R4bUVNR0UxUGlWTG12cEQiLCJzdWIiOjEsInBydiI6IjIzYmQ1
Yzg5NDlmNjAwYWRiMzllNzAxYzQwMDg3MmRiN2E1OTc2ZjcifQ
.oRoVG9tOc_gpeJJ4PKfkTyrBTL_SNiOLV9AFFzlxoHg"
    }
  }
}
```

Despliegue Heroku:


apigateway-healthbook.herokuapp.com
Query your server
Activar Windows
Ve a Configuración para activar Windows.

Ready to explore your graph?

Get a free private query console and schema-generated docs for your graph in Apollo Studio.

Automatically redirect to Studio next time

Or, query this graph directly:

```
$ curl --request POST \
    --header 'content-type: application/json' \
    --url 'https://apigateway-healthbook.herokuapp.com/' \
    --data '{"query":"query { __typename }"}'
```

This page can be customized or hidden. Learn more in the Apollo Server Docs