Documentación api de prueba

Jhon Edinson Mantilla Ruiz

Aspirante

Contenido

[Documentación api rest prueba. 3](#_Toc115119013)

[Configuración inicial – base de datos: 3](#_Toc115119014)

[Configuración Base de datos: 3](#_Toc115119015)

[Tabla Autenticación 3](#_Toc115119016)

[METODO POST 3](#_Toc115119017)

[Consumo token para: 3](#_Toc115119018)

[Crear Usuarios: 5](#_Toc115119019)

[INSERTAR REGISTROS TABLA FILES 6](#_Toc115119020)

[METODO PUT 7](#_Toc115119021)

[METODOS GET 8](#_Toc115119022)

[Buscar por un campo en específico: 8](#_Toc115119023)

[Buscar coincidencias por el nombre devolver porcentaje 9](#_Toc115119024)

[ERRORES DE CONSUMO TOKEN 11](#_Toc115119025)

# Documentación api rest prueba.

# Configuración inicial – base de datos:

Ruta: modelos/connection.php en el método estático infoDatabase()

'database' => 'bd\_prueba',

            'user' => 'root',

            'pass' => ''

# Configuración Base de datos:

Las tablas deben estar escritas en plural y los nombres de las columnas deben terminar con el nombre de la tabla en singular (\_sufijo) ejemplo:

Tabla: files

Columnas:

* + id\_file
  + name\_file

## Tabla Autenticación

Para la tabla de verificación columnas obligatorias.

* + Email\_sufijo
  + Password\_sufijo
  + Token\_sufijo
  + Token\_exp\_sufijo

# METODO POST

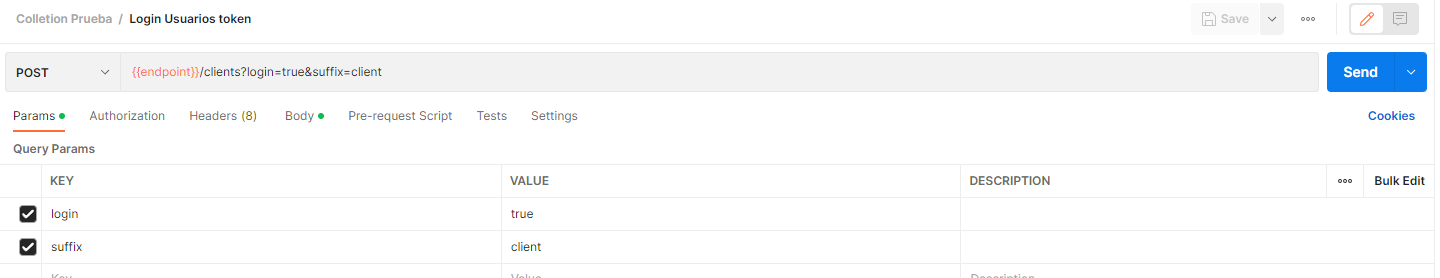
## Consumo token para:

Url: apirest.com/clients?login=true&suffix=client

Params: pasar los siguientes valores:

Login = true

Suffix = \_sufijo

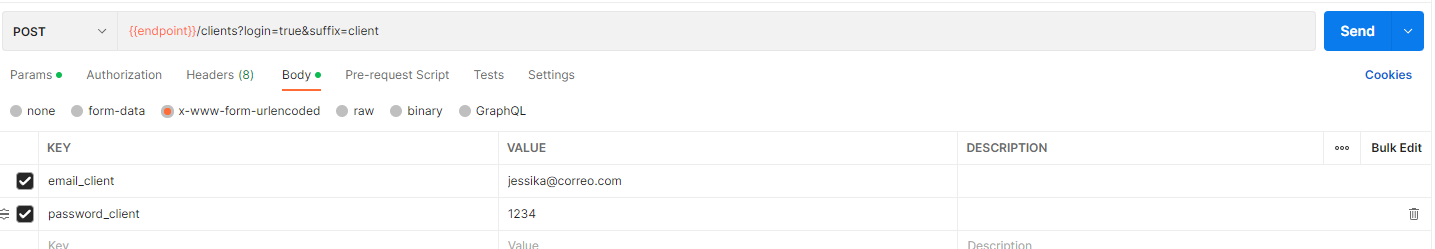


Body: pasar por parámetro:

Ejemplo:

email\_client = email\_cliente

password\_client = contraseña cliente



Response:

{

    "status": 200,

    "result": [

        {

            "id\_client": "14",

            "email\_client": "jessika@correo.com",

            "token\_client": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyMzcwMDAsImV4cCI6MTY2NDMyMzQwMCwiZGF0YSI6eyJpZCI6IjE0IiwiZW1haWwiOiJqZXNzaWthQGNvcnJlby5jb20ifX0.nFWg9HdAweTkbZTaoPV8-9n22lqLIBxwyyKAge2TX8Q",

            "token\_exp\_client": 1664323400,

            "date\_created\_client": "2022-09-26",

            "date\_updated\_client": "2022-09-26 19:01:24"

        }

    ]

}

Respuesta email incorrecto:

{

    "status": 400,

    "result": "Wrong Email"

}

Respuesta contraseña incorrecto:

{

    "status": 400,

    "result": "Wrong Password"

}

Usuario pruebas:

Usuario: [jessika@correo.com](mailto:jessika@correo.com)

Contraseña: 1234

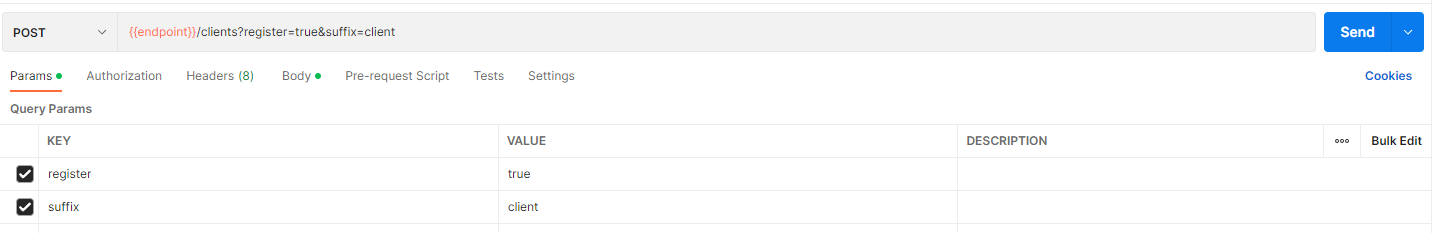
# Crear Usuarios:

URL: apirest.com/clients?register=true&suffix=client

Params

register = true

suffix = client



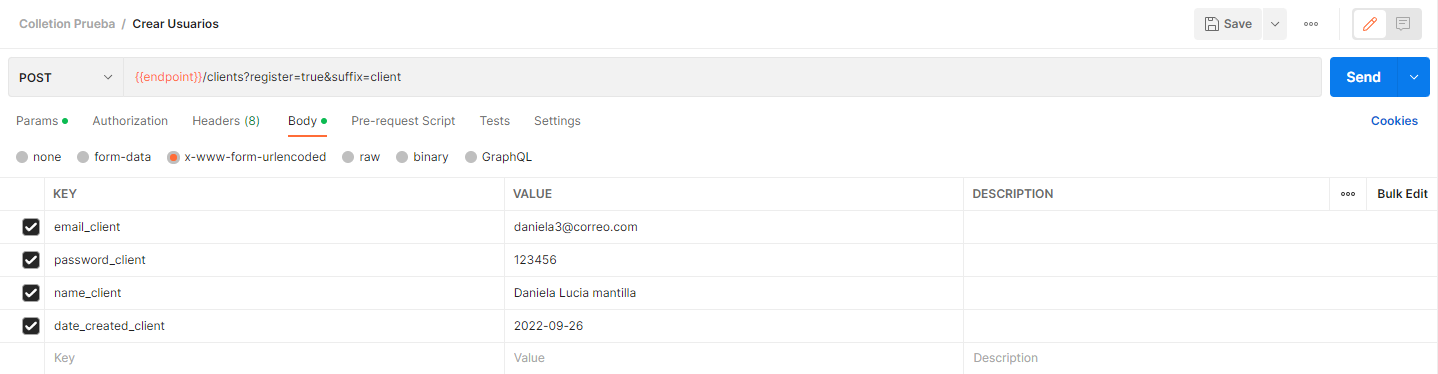
Body:

email\_client = daniela3@correo.com

password\_client = 123456

name\_client = Daniela Lucia mantilla

date\_created\_client = 2022-09-26



Response:

{

    "status": 200,

    "result": {

        "lastId": "3",

        "Results": "El process was successfull"

    }

}

Validación de campos de la tabla :

Error:

{

    "status": 400,

    "Results": "Error Fields in the form do not match the database"

}

# INSERTAR REGISTROS TABLA FILES

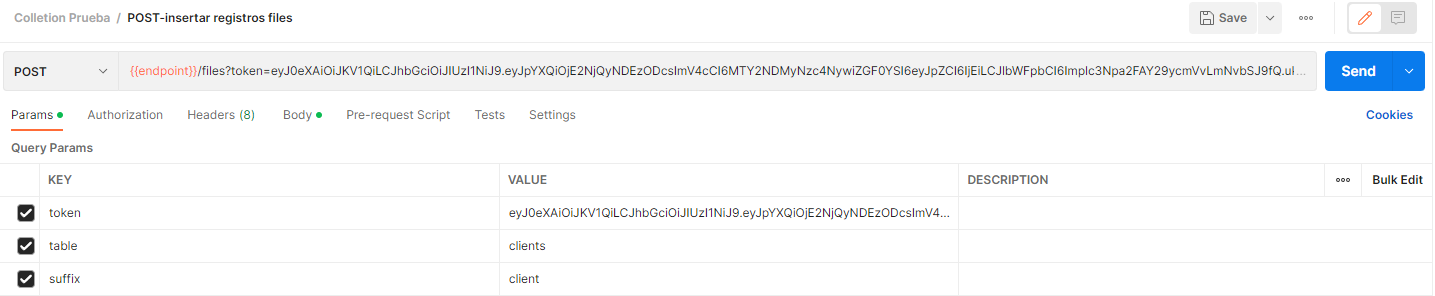
URL: apirest.com/files?token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyNDEzODcsImV4cCI6MTY2NDMyNzc4NywiZGF0YSI6eyJpZCI6IjEiLCJlbWFpbCI6Implc3Npa2FAY29ycmVvLmNvbSJ9fQ.uklnYYNRRxYoOisDXDIcwuTgvlGA-p-IUSdVqiikEBY&table=clients&suffix=client

Params

Token = eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyNDEzODcsImV4cCI6MTY2NDMyNzc4NywiZGF0YSI6eyJpZCI6IjEiLCJlbWFpbCI6Implc3Npa2FAY29ycmVvLmNvbSJ9fQ.uklnYYNRRxYoOisDXDIcwuTgvlGA-p-IUSdVqiikEBY

Table = clients

Suffix = client



Body:

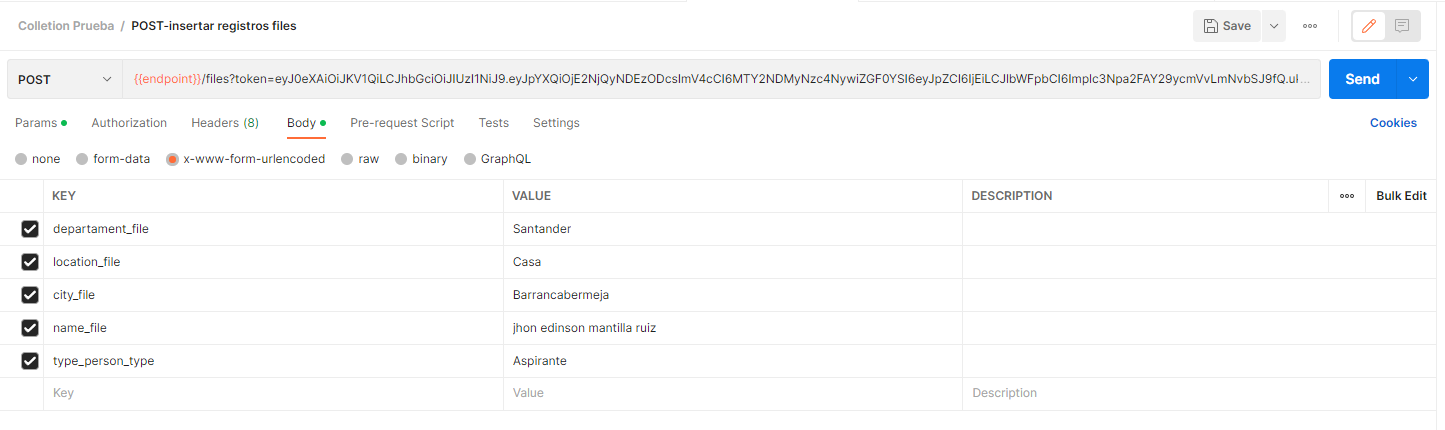
departament\_file = Santander

location\_file = casa

city\_file = Barrancabermeja

name\_file = jhon edinson mantilla ruiz

type\_person\_type = Aspirante



Response:

{

    "status": 200,

    "result": {

        "lastId": "5001",

        "Results": "El process was successfull"

    }

}

Errores de Campos.

Validación de campos de la tabla :

Error:

{

    "status": 400,

    "Results": "Error Fields in the form do not match the database"

}

# METODO PUT

URL: {{endpoint}}/files?id=1&nameId=id\_file&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyNDEzODcsImV4cCI6MTY2NDMyNzc4NywiZGF0YSI6eyJpZCI6IjEiLCJlbWFpbCI6Implc3Npa2FAY29ycmVvLmNvbSJ9fQ.uklnYYNRRxYoOisDXDIcwuTgvlGA-p-IUSdVqiikEBY&table=clients&suffix=client

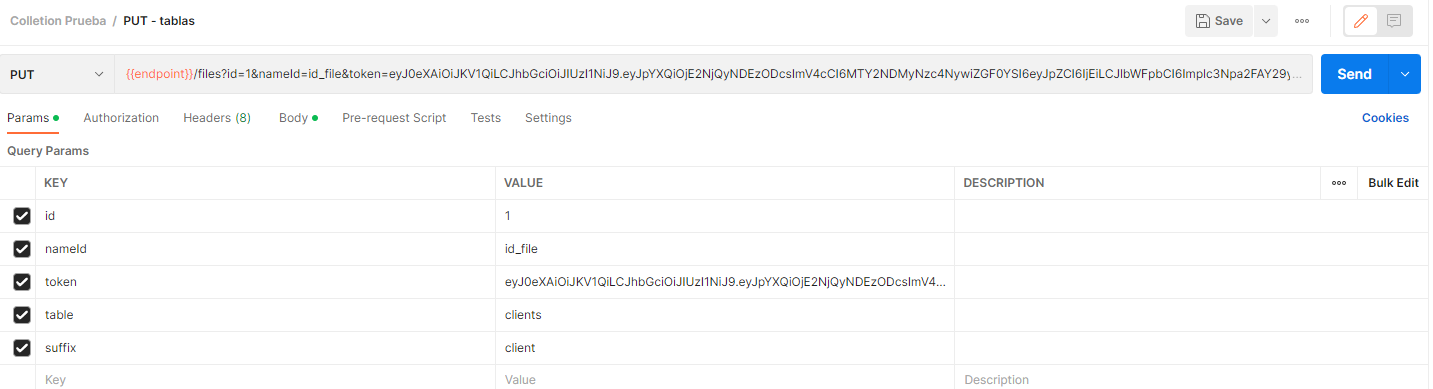
Params

Id = id

nameId = id\_file // campo a modificar

token= eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyNDEzODcsImV4cCI6MTY2NDMyNzc4NywiZGF0YSI6eyJpZCI6IjEiLCJlbWFpbCI6Implc3Npa2FAY29ycmVvLmNvbSJ9fQ.uklnYYNRRxYoOisDXDIcwuTgvlGA-p-IUSdVqiikEBY

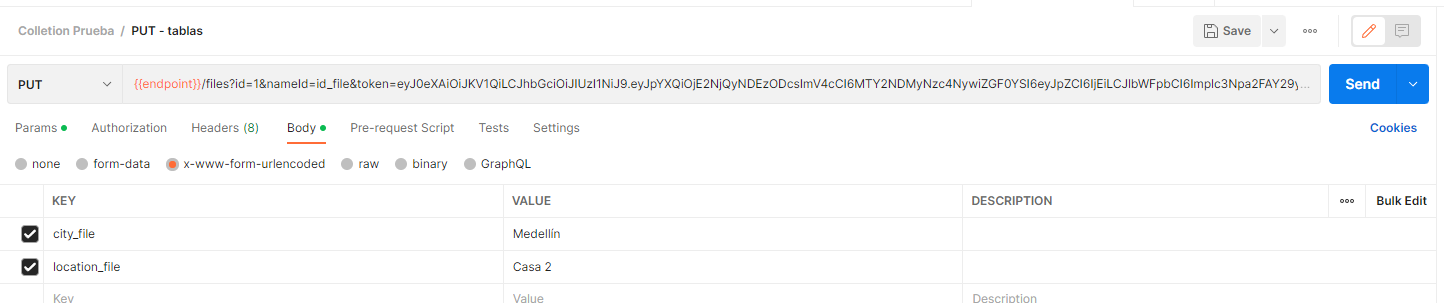
table = clients

suffix = client

Body:

city\_file = Medellín

location\_file = Casa 2



Response

{

    "status": 200,

    "result": {

        "id": "1",

        "Results": "The proccess was successful"

    }

}

Error:

{

    "status": 400,

    "Results": "Error Fields in the form do not match the database"

}

Error en caso de no encontrar un id.

{

    "status": 200,

    "result": {

        "Results": "No existe el 5010"

    }

}

# METODOS GET

## Buscar por un campo en específico:

En el campo select se puede colocar \* para traer todos los campos o seleccionar los campos específicos que quiere traer.

LinkTo puede colocar los campos concatenados con coma(,) y en la consulta se añadirá con AND si quiere una consulta más especifica.

equalTo los valores de los campos de linkTo.

Param

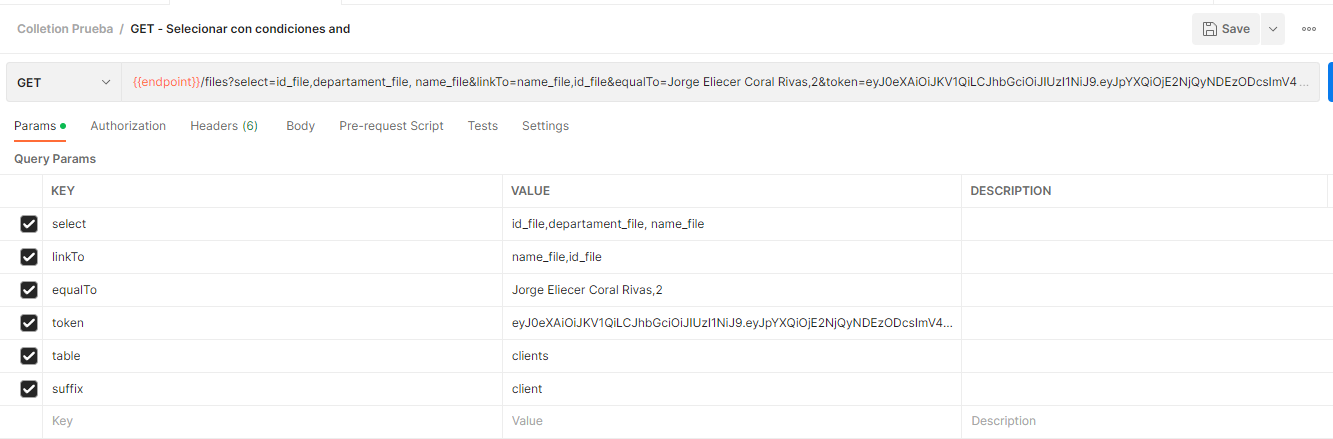
Select = id\_file,departament\_file, name\_file

linkTo = name\_file,id\_file

equalTo = Jorge Eliecer Coral Rivas,2

token = valor del token generado

table = clients

suffix = client

Response:

{

    "status": 200,

    "total": 1,

    "result": [

        {

            "id\_file": "2",

            "departament\_file": "PUTUMAYO",

            "name\_file": "Jorge Eliecer Coral Rivas"

        }

    ]

}

En caso de no encontrar registros

{

    "status": 404,

    "total": 0,

    "result": "No se encontraron resultados",

    "method": "get"

}

## Buscar coincidencias por el nombre devolver porcentaje

URL: {{endpoint}}/files?select=name\_file,type\_person\_type,type\_file,departament\_file,city\_file&name=Oneida Rayeth Pinto Perez&porcentaje=55&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NjQyNDEzODcsImV4cCI6MTY2NDMyNzc4NywiZGF0YSI6eyJpZCI6IjEiLCJlbWFpbCI6Implc3Npa2FAY29ycmVvLmNvbSJ9fQ.uklnYYNRRxYoOisDXDIcwuTgvlGA-p-IUSdVqiikEBY&table=clients&suffix=client

Params

Select = name\_file,type\_person\_type,type\_file,departament\_file,city\_file

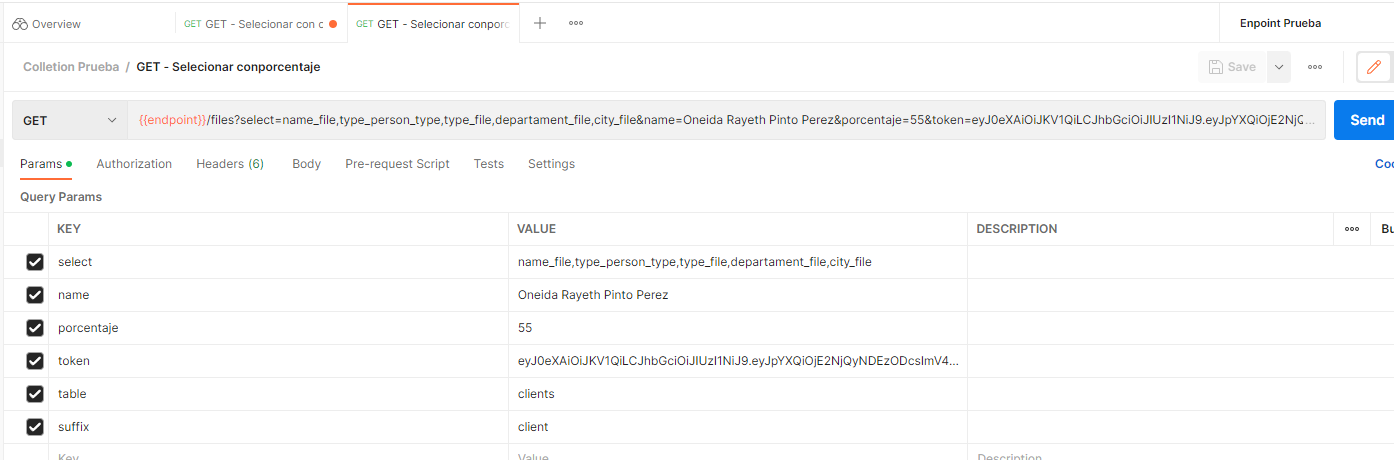
Name= Oneida Rayeth Pinto Perez

Porcentaje = 55

token = valor del token generado

table = clients

suffix = client



Response:

{

    "status": 200,

    "total": 5,

    "result": [

        {

            "nombre": "Oneida Rayeth Pinto Perez",

            "tipo\_persona": "NO APLICA",

            "tipo\_cargo": "POLITICO",

            "departamento": "LA\_GUAJIRA",

            "municipio": "Medellín",

            "porcentaje": 100

        },

        {

            "nombre": "Seonedis Leonor Perez",

            "tipo\_persona": "PREFERENTE",

            "tipo\_cargo": "POLITICO",

            "departamento": "LA\_GUAJIRA",

            "municipio": "ALBANIA",

            "porcentaje": 56.52173913043478

        },

        {

            "nombre": "Angela Janeth Giron Neira",

            "tipo\_persona": "PREFERENTE",

            "tipo\_cargo": "POLITICO",

            "departamento": "BOGOTA D.C.",

            "municipio": "BOGOTA.\_D.C.",

            "porcentaje": 56

        },

        {

            "nombre": "Heder Ladino Perez",

            "tipo\_persona": "PREFERENTE",

            "tipo\_cargo": "OTRO",

            "departamento": "HUILA",

            "municipio": "TERUEL",

            "porcentaje": 55.81395348837209

        },

        {

            "nombre": "Jose Trinidad Rios Perez",

            "tipo\_persona": "PREFERENTE",

            "tipo\_cargo": "OTRO",

            "departamento": "ARAUCA",

            "municipio": "SARAVENA",

            "porcentaje": 57.142857142857146

        }

    ]

}

Error en caso de no encontrar registros

{

    "status": 404,

    "total": 0,

    "result": "No se encontraron resultados",

    "method": "get"

}

# ERRORES DE CONSUMO TOKEN

No Suministra Token

{

    "status": 400,

    "Results": "Error: auth required"

}

Token Equivocado

{

    "status": 400,

    "Results": "Error: the token not auth"

}

Token expiro

El token tiene 24 horas después de su creación.

{

    "status": 400,

    "Results": "Error: the token has expired"

}