

Anexo 1: Documentación de la API

La api usa tokens JWT para autenticar y autorizar todas las llamadas siguientes. Si el token queda invalido, tambien se provee un token de refresco para renovar el token de acceso.

Las llamadas son las siguientes.

Para obtener un token valido o refrescarlo, la llamada es la siguiente.

Acción	URL	Método	Header	Body
Login	/auth	POST		<pre>{"username":<string>,"password":<string>}</pre>
Refresh	/refresh	POST	Refresh JWT "Bearer"	

Las respuesta de login sera similar a esta, en caso de no error 200:

```
{  
    "access_token":  
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjEwODAxNTIyODUsIm5iziI6MTU4MDElMjY4NSwiYWVudGl0eSI6ImhrZnVlc3RlcC0BbnBwFpbC5jb20iLCJmcmVzaCI6ZmFsc2UsInR5cGUoiOiJhY2Nlc3MifQ.tnG5P7oCiATvNgugnX9Cl2acCW0snEOmtWgm9vhYHKE",  
    "refresh_token":  
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjEwODAxNTIyODUsIm5iziI6MTU4MDElMjY4NSwiYWVudGl0eSI6ImhrZnVlc3RlcC0BbnBwFpbC5jb20iLCJ0ZXBlIjoicmVmcmVzaCJ9.9hj8v5j2KRYIA9vdus5UFqzcNGUjoh9FeJ_ayOpffcw"  
}
```

La respuesta para refresh sera simiar a esta, en caso de no error 200:

```
{  
    "access_token":  
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE0ODAxNTIzOTEsIm5iZiI6MTU4MDE1Mjc5MSwianRpIjoieZkxLWmFkbmdyLnR5c2lwZXNpdjkiLCJ0eXciOiJkaXIiLCJyb290IjoiZmlhbnQzMifQ.M89-uHvKNGOqsTjkfpwT5MU-IBOfIFdMfuJvUpwpDOY"  
}
```

Endpoints relacionados con los médicos

Las llamadas a la api para obtener informacion sobre el doctor.

Actualmente solo se usa el GET y el PUT en la app.

Acción	URL	Método	Autenticación
Crear Médico	/api/v1/doctor	POST	Private Key "PSK" in Header
Actualizar Médico	/api/v1/doctor	PUT	JWT in Header "Bearer"
Borrar Médico	/api/v1/doctor	DELETE	JWT in Header "Bearer"

Acción	URL	Método	Autenticación
Ver Medico	/api/v1/doctor	GET	JWT in Header "Bearer"

Endpoints relacionados con los pacientes

Las llamadas a la api para obtener informacion sobre el paciente.

Acción	URL	Método	Autenticación
Pacientes para el doctor del JWT	/api/v1/patients	GET	JWT in Header "Bearer"
Ver perfil del paciente <id>	/api/v1/patient/<id>	GET	JWT in Header "Bearer"
Crear Paciente	/api/v1/patient	POST	JWT in Header "Bearer"
Actualizar Paciente <id>	/api/v1/patient/<id>	PUT	JWT in Header "Bearer"

Endpoints relacionados con las medidas

Las llamadas a la api para obtener informacion sobre las medidas.

Acción	URL	Método	Autenticación
Enviar pulso para <id>	/api/v1/patient/<id>/pulse	POST	JWT in Header "Bearer"
Enviar peso para <id>	/api/v1/patient/<id>/weight	POST	JWT in Header "Bearer"
Enviar tensión para <id>	/api/v1/patient/<id>/pressure	POST	JWT in Header "Bearer"
Ver pulso para <id>	/api/v1/patient/<id>/pulse	GET	JWT in Header "Bearer"
Ver peso para <id>	/api/v1/patient/<id>/weight	GET	JWT in Header "Bearer"
Ver tensión para <id>	/api/v1/patient/<id>/pressure	GET	JWT in Header "Bearer"

Endpoints relacionados con las estadísticas

Las llamadas a la api para obtener las estadísticas, los datos de todos los pacientes, sin nombres ni informacion

Puede ser interesante dejar estos endpoints abiertos en el futuro.

Acción	URL	Método	Autenticación
Ver los pulsos	/api/v1/stats/pulse	GET	JWT in Header "Bearer"
Ver los pesos	/api/v1/stats/weight	GET	JWT in Header "Bearer"
Ver las tensiones	/api/v1/stats/pressure	GET	JWT in Header "Bearer"
Ver todas las Estadísticas	/api/v1/stats	GET	JWT in Header "Bearer"

Respuestas

La respuesta estandar va a ser de esta forma:

- En caso de no error

```
{
  "data": { ... },
  "result": "success"
}
```

- En caso de error

```
{
  "data": { ... },
  "result": "failed"
}
```

Los modelos de datos

- Doctor:

```
{
  "data": {
    "email": <email>,
    "lastName": <string>,
    "name": <string>
  },
  "result": "success"
}
```

- Paciente:

```
{
  "data": {
    "birthday": "1994-06-04",
    "gender": "male",
    "height": 175,
    "id": 1,
    "lastName": "Fuertes",
    "name": "Javier"
  },
  "result": "success"
}
```

- Pesos:

```
{
  "data": [
    {
      "timestamp": "2020-01-20 18:49:50",
      "weight": 120.6
    },
    {
      "timestamp": "2020-01-23 13:16:03",
      "weight": 120.0
    },
    ...
  ],
  "result": "success"
}
```

- Pulsos:

```
{
  "data": [
    {
      "high": null,
      "low": null,
      "pulse": 100,
      "timestamp": "2020-01-21 18:35:12"
    },
    ...
  ],
  "result": "success"
}
```

- Presión sanguínea:

```
{
  "data": [
    {
      "high": 140,
      "low": 80,
      "pulse": 101,
      "timestamp": "2020-01-24 22:57:10"
    },
    ...
  ],
  "result": "success"
}
```

- Estadísticas de pulso:

```
{
  "data": [
    {
      "age": 25,
      "gender": "male",
      "height": 175,
      "pulse": 70,
      "timestamp": "2020-01-20 18:49:50"
    },
    ...
  ],
  "result": "success"
}
```

- Estadísticas de peso:

```
{
  "data": [
    {
      "age": 25,
      "gender": "male",
      "height": 175,
      "timestamp": "2020-01-20 18:49:50",
      "weight": 120.6
    },
    ...
  ],
  "result": "success"
}
```

- Estadísticas de presión sanguínea:

```
{
  "data": [
    {
      "age": 25,
      "gender": "male",
      "height": 175,
      "high": 120,
      "low": 90,
```

```
        "pulse": 70,  
        "timestamp": "2020-01-20 18:49:50"  
    },  
    ...  
],  
    "result": "success"  
}
```