

**TALENT LAND 2023 HACKATHON TRACK: ASTRAZENECA**

Proyecto: Software de Monitoreo de servicios BTG

Nombre del software: Service Tracker

Integrantes del equipo:

* Raymundo Paz Valverde: Jesús Manuel Fierro Hernández
* Derek Leonardo Terrazas Caro:
* Luis Alberto Barrera Sandoval:
* Axel Javier Reynosa Acosta:

Fecha: 13 de abril de 2023

Contenido

[SOLICITUDES API 1](#_Toc132247208)

[Solicitud de inicio de sesión. (POST) 1](#_Toc132247209)

[Estructura de solicitud de inicio de sesión. 1](#_Toc132247210)

[Respuesta ante usuario valido 1](#_Toc132247211)

[Respuesta ante solicitud no valida 2](#_Toc132247212)

[Solicitud de registro de usuario. (POST) 3](#_Toc132247213)

[Estructura de solicitud de registro de usuario 3](#_Toc132247214)

[Respuesta del servidor ante operación correcta 3](#_Toc132247215)

[Respuesta del servidor ante un error en la base de datos: 3](#_Toc132247216)

[Respuesta del servidor ante una solicitud incorrecta 4](#_Toc132247217)

[Solicitud de validación de token. (POST) 5](#_Toc132247218)

[Estructura de solicitud de validación de token 5](#_Toc132247219)

[Respuesta de validación de token (token valido) 5](#_Toc132247220)

[Respuesta de validación de token (token invalido) 5](#_Toc132247221)

[Solicitud de registro de eventos. (GET) 6](#_Toc132247222)

[Estructura de solicitud de registro de eventos 6](#_Toc132247223)

[Respuesta correcta del servidor web 6](#_Toc132247224)

[Respuesta ante error de la base de datos 7](#_Toc132247225)

[Solicitud de estados de los servicios (GET) 8](#_Toc132247226)

[Estructura de solicitud de estado de los servicios 8](#_Toc132247227)

[Respuesta correcta del servidor web 8](#_Toc132247228)

[Respuesta ante error de la base de datos 9](#_Toc132247229)

# SOLICITUDES API

## Solicitud de inicio de sesión. (POST)

Solicitud HTTP donde se autentica la cuenta de usuario, devolviendo un token que permite la autorización a los servicios de dicha cuenta.

### Estructura de solicitud de inicio de sesión.

POST http://[ServerIP]:3000/login

body {

“identifier”: “example@example.com”,

“Password”: “example1234”

}

### Respuesta ante usuario valido

POST RESPONSE http://[ServerIP]:3000/login

body {

“requestStatus”: “SUCCESS”,

“user”: {

“idUser”: “1”,

“username”: “user”,

“email”: “example@example.com”,

“name”: “user example”,

“role”: “1”,

“state”: “0”

},

“token”: “Ij0g0o2B8gfjf/WrbHhaIsLcxOgV4?QrI8sAsBOLfDsDmduneONQrpqQ?PuKBFxOummmSK3HQMzr6I0FISRMyaCMA8VcQzrMfWfVfnJlV?XdYIzBYJRJtvDnNHgi8Xq5kte6G7FW4w/UDfmWOgnREQ6m8pbXsmAwmCHWAdvsKv5wNKQP8MIR4OW-yochoahon4Nw=/J5HFlaf0uTRl2sk0SQF8yMUqAOMPX=-071aV1Gg6eZJ-RH=INdJ=BGc5y5”

}

### Respuesta ante solicitud no valida

POST RESPONSE http://[ServerIP]:3000/login

body {

“requestStatus”: “ERROR”,

“loginStatusCode”: 1,

“error”: {

“message”: “Internal server error”

}

}

## Solicitud de registro de usuario. (POST)

Solicitud HTTP donde se envía la información de los usuarios y esta crea un registro desde de dicho usuario:

### Estructura de solicitud de registro de usuario

POST http://[ServerIP]:3000/users/register

body {

“username”: “user”,

“email”: “example@example.com”,

“password”: “example1234”,

“name”: “user example”

}

### Respuesta del servidor ante operación correcta

POST RESPONSE http://[ServerIP]:3000/users/register

body {

“requestStatus”: “SUCCESS”,

“registerStatusCode”: 0

}

### Respuesta del servidor ante un error en la base de datos:

POST RESPONSE http://[ServerIP]:3000/users/register

body {

“requestStatus”: “ERROR”,

“registerStatusCode”: 1

“error": {

“message”: “Internal server error”

}

}

### Respuesta del servidor ante una solicitud incorrecta

POST RESPONSE http://[ServerIP]:3000/users/register

body {

“requestStatus”: “ERROR”,

“registerStatusCode”: 1

“error": {

“message”: “Bad query request”

}

}

## Solicitud de validación de token. (POST)

En esta solicitud, se validará el token de un usuario necesario para realizar acceder a los servicios de la API.

### Estructura de solicitud de validación de token

POST http://[ServerIP]:3000/users/validate\_token

body {

“token”: “Ij0g0o2B8gfjf/WrbHhaIsLcxOgV4?QrI8sAsBOLfDsDmduneONQrpqQ?PuKBFxOummmSK3HQMzr6I0FISRMyaCMA8VcQzrMfWfVfnJlV?XdYIzBYJRJtvDnNHgi8Xq5kte6G7FW4w/UDfmWOgnREQ6m8pbXsmAwmCHWAdvsKv5wNKQP8MIR4OW-yochoahon4Nw=/J5HFlaf0uTRl2sk0SQF8yMUqAOMPX=-071aV1Gg6eZJ-RH=INdJ=BGc5y5”

}

### Respuesta de validación de token (token valido)

POST RESPONSE http://[ServerIP]:3000/users/validate\_token

body {

“requestStatus”: “SUCCESS”,

“isTokenValid”: “true”,

}

### Respuesta de validación de token (token invalido)

POST RESPONSE http://[ServerIP]:3000/users/validate\_token

body {

“requestStatus”: “SUCCESS”,

“isTokenValid”: “false”,

“error”: {

“message”: “Invalid token”

}

}

## Solicitud de registro de eventos. (GET)

Responde a la petición con los registros de eventos de los servicios y componentes.

### Estructura de solicitud de registro de eventos

GET http://[ServerIP]:3000/record

### Respuesta correcta del servidor web

GET RESPONSE http://[ServerIP]:3000/record

body{

“requestStatus”: “SUCCESS”,

“queryStatusCode”: “0”,

“result”:{

{

“idRecord”: 1,

“idService”: “1-1”,

“message”: “”,

“timestamp”: “2023-04-12T06:30:41.471Z”

“state”: 2

},

{

…

}

}

}

### Respuesta ante error de la base de datos

GET RESPONSE http://[ServerIP]:3000/record

body{

“requestStatus”: “ERROR”,

“queryStatusCode”: “1”,

“error”: {

“message”: “Internal server error”

}

}

## Solicitud de estados de los servicios (GET)

Obtiene el estado en tiempo real de los servicios, por medio de HTTP.

### Estructura de solicitud de estado de los servicios

GET http://[ServerIP]:3000/services

### Respuesta correcta del servidor web

GET RESPONSE http://[ServerIP]:3000/services

body{

“requestStatus”: “SUCCESS”,

“queryStatusCode”: “0”,

“result”:{

{

“id”: 5,

“name”: “HR Critical Service 1”,

“description”: “Description”,

“state”: 0

},

{

…

}

}

}

### Respuesta ante error de la base de datos

GET RESPONSE http://[ServerIP]:3000/services

body{

“requestStatus”: “ERROR”,

“queryStatusCode”: “1”,

“error”: {

“message”: “Internal server error”

}

}

Contacto de linkedin:

* Raymundo Paz Valverde: <https://www.linkedin.com/in/raymundo-paz-valverde-a2703926b/>
* Jesús Manuel Fierro Hernández: <https://www.linkedin.com/in/jesus-fierro-0116b1215/>
* Derek Leonardo Terrazas Caro: <https://www.linkedin.com/in/derek-leonardo-terrazas-caro-762b0a23a/>
* Luis Alberto Barrera Sandoval: <https://www.linkedin.com/in/luis-alberto-barrera-sandoval-185113218/>
* Axel Javier Reynosa Acosta: <https://www.linkedin.com/in/samaaxl/>