



TRACKING LAB

YOUR BEST SHIPPING CONFIDANT

Manual de desarrollador

Adrián Soler Navarro

Contenido

1	Introducción	3
2	Prerrequisitos	3
3	Obtener información general de la API	3
4	Realizar una solicitud HTTP	3
5	Llamadas a endpoints específicos	3
5.1	Registro Signup	4
5.2	Inicio de sesión Login	5
5.3	Conectado LoggedIn	6
5.4	Solicitar token de acceso Get access token	7
5.5	Comprobación de acceso a la API Check Access	8
5.6	Crear tracking Create Tracking	9
4	Manejo de errores y excepciones	11
5	Utilizando herramientas de desarrollo de API	11
5.1	Postman	11
5.2	Insomnia	12
6	Conclusiones	12

1 Introducción

Este manual proporciona una guía básica sobre cómo realizar llamadas HTTP a la API de TrackingLab con la finalidad de poder realizar diversas operaciones. A lo largo de este manual, se utilizarán ejemplos para ilustrar los conceptos explicados.

2 Prerrequisitos

- Conocimiento básico del protocolo HTTP.
- Una herramienta de desarrollo y uso de APIs.
- Suscripción a un plan de TrackingLab.

3 Obtener información general de la API

Antes de comenzar, es importante familiarizarse con la nomenclatura de la API que se va a utilizar. Esta documentación proporciona información sobre la estructura de las URL, los métodos HTTP admitidos, los parámetros requeridos y opcionales, así como las respuestas esperadas.

4 Realizar una solicitud HTTP

Para interactuar con una API, es posible hacer solicitudes HTTP utilizando un lenguaje de programación compatible, pero para este manual, se utiliza la herramienta gráfica de uso de APIs llamada Insomnia (<https://insomnia.rest/download>), aunque también es posible hacer uso de otras herramientas similares, como por ejemplo, PostMan (<https://www.postman.com/downloads/>)

5 Llamadas a endpoints específicos

A continuación, se muestran ejemplos de cómo llamar a todos los endpoints públicos de la API de TrackingLab utilizando diferentes llamadas HTTP.

Nota: Asegúrese de proporcionar los parámetros necesarios según la siguiente documentación.

5.1 Registro | Signup

En el momento que alguien se suscribe a algún plan de la plataforma, se genera automáticamente una nueva cuenta de usuario en las bases de datos de TrackingLab. Para probar esta funcionalidad, puedes utilizar el método POST del endpoint signup.

PETICIÓN

URL: <http://localhost:4001/api/v1/identity/auth/signup>

Método: POST

Cuerpo del mensaje: JSON

Propiedades requeridas:

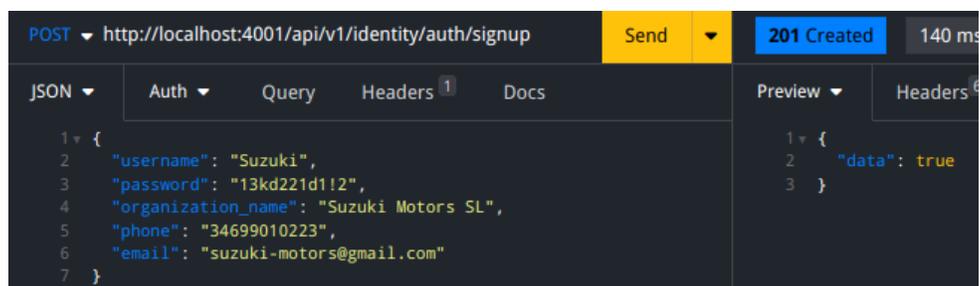
- username: string
- password: string
- organization_name: string
- phone: string
- email: string

Cabeceras:

- Content-Type: application/json

RESPUESTA

- data: boolean



The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:4001/api/v1/identity/auth/signup
- Status:** 201 Created
- Response Time:** 140 ms
- Request Body (JSON):**

```
1 {
2   "username": "Suzuki",
3   "password": "13kd221d1!2",
4   "organization_name": "Suzuki Motors SL",
5   "phone": "34699010223",
6   "email": "suzuki-motors@gmail.com"
7 }
```
- Response Body (JSON):**

```
1 {
2   "data": true
3 }
```

5.2. Inicio de sesión | Login

Este endpoint recibe el nombre de usuario y la contraseña usados en el registro.

PETICIÓN

URL: <http://localhost:4001/api/v1/identity/auth/signin>

Método: POST

Cuerpo del mensaje: JSON

Propiedades requeridas:

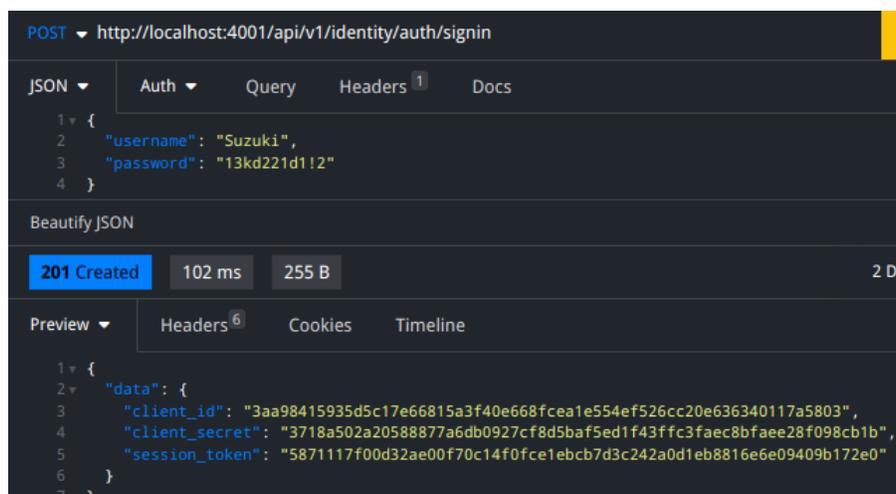
- username: string
- password: string

Cabeceras:

- Content-Type: application/json

RESPUESTA

- client_id: string
- client_secret: string
- session_token: string



The screenshot shows a REST client interface for the endpoint `POST http://localhost:4001/api/v1/identity/auth/signin`. The request body is a JSON object with `username: "Suzuki"` and `password: "13kd221d1!2"`. The response status is `201 Created` with a response time of `102 ms` and a body size of `255 B`. The response body is a JSON object containing `data` with `client_id`, `client_secret`, and `session_token` fields.

```
POST http://localhost:4001/api/v1/identity/auth/signin
JSON Auth Query Headers 1 Docs
1 {
2   "username": "Suzuki",
3   "password": "13kd221d1!2"
4 }
Beautify JSON
201 Created 102 ms 255 B 2 Da
Preview Headers 6 Cookies Timeline
1 {
2   "data": {
3     "client_id": "3aa98415935d5c17e66815a3f40e668fcea1e554ef526cc20e636340117a5803",
4     "client_secret": "3718a502a20588877a6db0927cf8d5baf5ed1f43ffc3faec8bfae28f098cb1b",
5     "session_token": "5871117f00d32ae00f70c14f0fce1ebcb7d3c242a0d1eb8816e6e09409b172e0"
6   }
7 }
```

5.3. Conectado | LoggedIn

En futuros inicios de sesión y mientras haya un “session_token” disponible, se llamará a este endpoint. A diferencia del inicio de sesión convencional, el parámetro de entrada de la contraseña se sustituye por el token de sesión.

PETICIÓN

URL: <http://localhost:4001/api/v1/identity/auth/loggedin>

Método: POST

Cuerpo del mensaje: JSON

Propiedades requeridas:

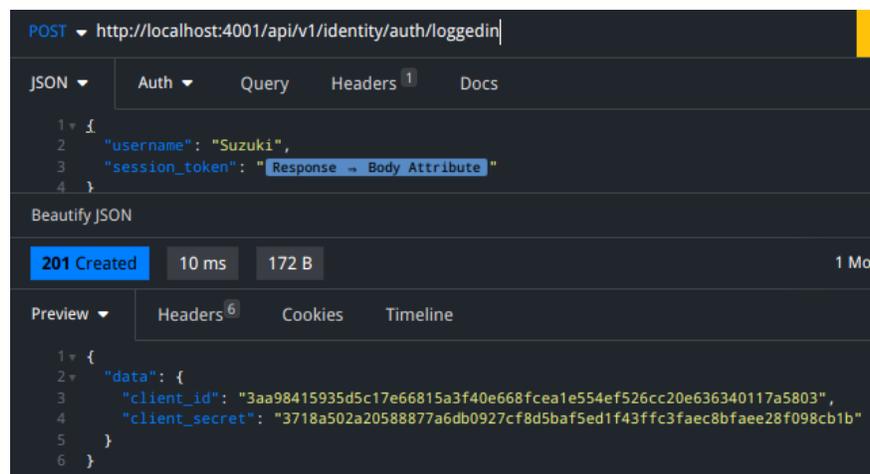
- username: string
- session_token: string

Cabeceras:

- Content-Type: application/json

RESPUESTA

- client_id: string
- client_secret: string



The screenshot shows a REST client interface with the following details:

- Request:** Method: POST, URL: `http://localhost:4001/api/v1/identity/auth/loggedin`. The body is a JSON object: `{ "username": "Suzuki", "session_token": "Response -> Body Attribute" }`.
- Response:** Status: 201 Created, Time: 10 ms, Size: 172 B. The response body is a JSON object: `{ "data": { "client_id": "3aa98415935d5c17e66815a3f40e668fcea1e554ef526cc20e636340117a5803", "client_secret": "3718a502a20588877a6db0927cf8d5baf5ed1f43ffc3faec8bfaee28f098cb1b" } }`.

5.5. Comprobación de acceso a la API | Check Access

En algunos casos, es posible que el desarrollador necesite verificar rápidamente si tiene acceso a un cierto recurso, para ello, se expone el siguiente endpoint que tan solo recibe el token de acceso recuperado en el paso 3.4.

PETICIÓN

URL: <http://localhost:4001/api/v1/identity/auth/checkAccess>

Método: POST

Cabeceras:

- Authorization: Bearer {access_token}

RESPUESTA

- OK | Unauthorized

The screenshot shows a REST client interface with the following details:

- Method: POST
- URL: <http://localhost:4001/api/v1/identity/auth/checkAccess>
- Authentication: Bearer
- Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IiN1enVraSIsInN1YiI6ImZAsImhdCI6I
- Response: 201 Created (3.21 ms, 2 B)
- Status: OK

5.6. Crear tracking | Create Tracking

Por último, el endpoint de creación de trackings recibe un conjunto de propiedades, algunas obligatorias y otras opcionales, así como el token de acceso recibido en el apartado 3.3.

PETICIÓN

URL: <http://localhost:4000/api/v1/tracking/core/create>

Método: POST

Cuerpo del mensaje: JSON

Propiedades requeridas:

- courier: string
- tracking_number: string,
- service: string
- zip_code: string
- phone: string (identificador de Telegram)

Propiedades opcionales:

- orderNo?: string
- otification_platform?: string[]
- recipient_notification?: string
- recipient?: string
- email?: string
- street?: string
- city?: string

Cabeceras:

- Content-Type: application/json
- Authorization: Bearer {access_token}

RESPUESTA

- data: string | Error with code

The screenshot shows a REST client interface for a POST request to `http://localhost:4000/api/v1/tracking/core/create`. The request body is a JSON object with the following fields: `courier` (dhl), `tracking_number` (1111-11111), `service` (default), `zip_code` (46701), `orderNo` (1093X23D), `notification_platform` (telegram), `recipient_notification` (Adrian Soler), `recipient` (Adrian), `email` (adrian@gmail.com), `street` (c/lazaro 2), `city` (Gandia), and `phone` (2072745433). The response is a JSON object with a single field `data` containing the string "Tracking has been created". The status bar indicates a 201 Created response with a 38.8 ms response time and 36 B of data.

```
POST http://localhost:4000/api/v1/tracking/core/create Send
JSON Bearer Query Headers 1 Docs
1 {
2   "courier": "dhl",
3   "tracking_number": "1111-11111",
4   "service": "default",
5   "zip_code": "46701",
6   "orderNo": "1093X23D",
7   "notification_platform": ["telegram"],
8   "recipient_notification": "Adrian Soler",
9   "recipient": "Adrian",
10  "email": "adrian@gmail.com",
11  "street": "c/lazaro 2",
12  "city": "Gandia",
13  "phone": "2072745433"
14 }
Beautify JSON
201 Created 38.8 ms 36 B 2 Days Ago
Preview Headers 6 Cookies Timeline
1 {
2   "data": "Tracking has been created"
3 }
```

Nota: El token es obligatorio para la creación de trackings.

The screenshot shows a REST client interface for a POST request to `http://localhost:4000/api/v1/tracking/core/create`. The request is configured with a Bearer token and the "ENABLED" checkbox is checked. The token value is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IiN1enVr...`. The response is a JSON object with a single field `data` containing the string "Tracking has been created". The status bar indicates a 201 Created response with a 38.8 ms response time and 36 B of data.

```
POST http://localhost:4000/api/v1/tracking/core/create Send
JSON Bearer Query Headers 1 Docs
ENABLED 
TOKEN eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IiN1enVr...
201 Created 38.8 ms 36 B 2 Days Ago
Preview Headers 6 Cookies Timeline
1 {
2   "data": "Tracking has been created"
3 }
```

4. Manejo de errores y excepciones

Es importante tener en cuenta los posibles errores y excepciones al realizar llamadas a la API. Esto puede incluir errores de red, respuestas inesperadas de la API o problemas en la autenticación. Asegúrate de manejar adecuadamente estos casos en tu código.

5. Utilizando herramientas de desarrollo de API

Además de realizar llamadas a una API utilizando código, existen herramientas populares que facilitan el proceso de interactuar con endpoints y probar la funcionalidad de la API. Dos de las herramientas más comunes son Postman e Insomnia. A continuación, se proporciona información básica sobre estas aplicaciones.

5.1. Postman

Postman es una herramienta de desarrollo de API que permite enviar solicitudes HTTP de manera fácil y rápida. Proporciona una interfaz gráfica intuitiva que facilita la construcción, prueba y documentación de API. A continuación, se muestra un ejemplo de cómo utilizar Postman para realizar una solicitud GET a un endpoint de una API:

1. Descarga e instala Postman desde el sitio web oficial: <https://www.postman.com>.
2. Abre Postman y crea una nueva solicitud.
3. Ingresa la URL del endpoint al que deseas realizar la solicitud.
4. Selecciona el método HTTP adecuado, como GET, POST, PUT o DELETE.
5. Agrega cualquier parámetro requerido en la pestaña "Params" si es necesario.
6. Haz clic en el botón "Send" para enviar la solicitud a la API.
7. Postman mostrará la respuesta recibida, incluyendo el código de estado, los encabezados y el cuerpo de la respuesta.

Postman ofrece muchas características adicionales, como guardar solicitudes, automatizar flujos de trabajo y generar documentación de API. Explora la documentación oficial de Postman para obtener más detalles sobre sus funcionalidades.

5.2. Insomnia

Insomnia es otra herramienta popular para el desarrollo de API. Proporciona una interfaz amigable para enviar solicitudes HTTP y realizar pruebas exhaustivas. A continuación, se muestra un ejemplo de cómo utilizar Insomnia para realizar una solicitud GET a un endpoint de una API:

1. Descarga e instala Insomnia desde el sitio web oficial: <https://insomnia.rest>.
2. Abre Insomnia y crea un nuevo proyecto.
3. Crea una nueva solicitud dentro del proyecto.
4. Ingresa la URL del endpoint al que deseas realizar la solicitud.
5. Selecciona el método HTTP adecuado en el menú desplegable.
6. Agrega cualquier parámetro requerido en la sección "Query" si es necesario.
7. Haz clic en el botón "Send" para enviar la solicitud a la API.
8. Insomnia mostrará la respuesta recibida, incluyendo el código de estado, los encabezados y el cuerpo de la respuesta.

Insomnia ofrece características como guardar y organizar solicitudes, gestionar entornos, y automatizar flujos de trabajo complejos. Consulta la documentación oficial de Insomnia para obtener más información sobre sus capacidades.

Ambas herramientas, Postman e Insomnia, son ampliamente utilizadas en el desarrollo de API debido a su facilidad de uso y capacidad para simplificar el proceso de prueba y depuración.

6. Conclusiones

En este manual básico de desarrollador, se ha mostrado cómo realizar llamadas a la API de TrackingLab utilizando el protocolo HTTP. Además, se han presentado herramientas populares como Postman e Insomnia, que ayudan en la interacción con los endpoints de la API. Para mejorar tu flujo de trabajo de desarrollo con nuestra API, se recomienda explorar todas las capacidades de estas herramientas.