

# OpenAPI definition v0 OAS 3.0

/v3/api-docs

## Servers

http://localhost:8080 - Generated server url ▾

## Habit Types API para gestión de tipos de hábito

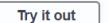


GET /api/habit-types/habit-type/{id} Obtener tipo por ID



Busca un tipo de hábito por su ID único

## Parameters



Name Description

id \* required ID del tipo (debe ser positivo)

string  
(path)

id

## Responses

Code Description

Links

200 Tipo encontrado

No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{  
  "id": 1,  
  "name": "Salud"  
}
```

400 ID inválido

No links

404 Tipo no encontrado

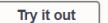
No links

PUT /api/habit-types/habit-type/{id} Actualizar tipo existente



Modifica los datos de un tipo de hábito

## Parameters



Name Description

id \* required ID del tipo a actualizar

integer(\$int64)  
(path)

id

Request body required

application/json

Datos actualizados del tipo

Example Value | Schema

```
{  
  "name": "Productividad"  
}
```

## Responses

Code Description

Links

200 Tipo actualizado exitosamente

No links

Media type

application/json

	Controls Accept header.
	<a href="#">Example Value</a>   <a href="#">Schema</a>
	<pre>{\n  "id": 1,\n  "name": "Salud"\n}</pre>
400	Datos inválidos
404	Tipo no encontrado

**DELETE /api/habit-types/habit-type/{id}** Eliminar tipo

Elimina un tipo de hábito del sistema por su ID

**Parameters**

**Name** **Description**

<b>id</b> <small>* required</small>	ID del tipo a eliminar
<small>integer(\$int64)</small>	id
<small>(path)</small>	

**Responses**

<b>Code</b>	<b>Description</b>	<b>Links</b>
204	Tipo eliminado exitosamente	<a href="#">No links</a>
400	ID inválido	<a href="#">No links</a>
404	Tipo no encontrado	<a href="#">No links</a>

**POST /api/habit-types/habit-type** Crear nuevo tipo

Registra un nuevo tipo de hábito en el sistema

**Parameters**

No parameters

**Request body** required

application/json

Datos del nuevo tipo de hábito

Example Value | Schema

```
{\n  "name": "Productividad"\n}
```

**Responses**

<b>Code</b>	<b>Description</b>	<b>Links</b>
201	Tipo creado exitosamente	<a href="#">No links</a>
400	Datos inválidos	<a href="#">No links</a>

**GET /api/habit-types/habit-types** Obtener todos los tipos de hábito

Retorna una lista completa de tipos de hábito

**Parameters**

No parameters

## Responses

Code	Description	Links
200	Lista de tipos encontrada	No links
	Media type <input checked="" type="button"/> */*	
	Controls Accept header.	
	<a href="#">Example</a> <a href="#">Value</a>   <a href="#">Schema</a>	
	<pre>{\n  "id": 1,\n  "name": "Salud"\n}</pre>	
204	No hay tipos registrados	No links

**GET** /api/habit-types/habit-type/name/{name} Buscar tipo por nombre

Busca un tipo de hábito por su nombre (sensible a mayúsculas)

## Parameters

[Try it out](#)

Name	Description
name <small>* required</small>	Nombre del tipo de hábito

string  
(path)

name

## Responses

Code	Description	Links
200	Tipo encontrado	No links
	Media type <input checked="" type="button"/> */*	
	Controls Accept header.	
	<a href="#">Example</a> <a href="#">Value</a>   <a href="#">Schema</a>	
	<pre>{\n  "id": 1,\n  "name": "Salud"\n}</pre>	
400	Nombre inválido	No links
404	Tipo no encontrado	No links

**Habits** API para gestión de hábitos

^

**GET** /api/habits/habit/{id} Obtener hábito por ID

^

Busca un hábito específico por su ID

## Parameters

[Try it out](#)

Name	Description
id <small>* required</small>	ID del hábito (debe ser positivo)

string  
(path)

id

## Responses

Code	Description	Links
200	Hábito encontrado	No links
	Media type <input checked="" type="button"/> */*	
	Controls Accept header.	
	<a href="#">Example</a> <a href="#">Value</a>   <a href="#">Schema</a>	
	<pre>{\n  "id": 1,\n  "name": "Ejercicio diario",\n  "description": "30 minutos de cardio",\n  "frequency": "Diario",\n  "habit_type": "Salud"\n}</pre>	

```
        "startDate": "2024-01-01",
        "endDate": "2024-12-31",
        "habit typeName": "Salud",
        "userId": 1
    }
```

400	ID inválido	No links
404	Hábito no encontrado	No links

## PUT /api/habits/habit/{id} Actualizar hábito

^

Modifica los datos de un hábito existente

### Parameters

Try it out

Name	Description
<b>id</b> * required	ID del hábito a actualizar

### Request body required

application/json

▼

Datos actualizados del hábito

Example Value | Schema

```
{
    "name": "Meditación matutina",
    "description": "15 minutos de meditación guiada",
    "frequency": "Diario",
    "startDate": "2024-01-01",
    "endDate": "2024-12-31",
    "userId": 1,
    "habitTypeId": 2
}
```

### Responses

Code	Description	Links
200	Hábito actualizado exitosamente	No links
400	Datos inválidos	No links
404	Hábito no encontrado	No links

## DELETE /api/habits/habit/{id} Eliminar hábito

^

Elimina un hábito del sistema por su ID

### Parameters

Try it out

Name	Description
<b>id</b> * required	ID del hábito a eliminar

### Responses

Code	Description	Links
204	Hábito eliminado exitosamente	No links
400	ID inválido	No links
404	Hábito no encontrado	No links

**POST** /api/habits/habit Create nuevo hábito

Registra un nuevo hábito en el sistema

**Parameters**

No parameters

**Request body** required

Datos del nuevo hábito

[Example Value](#) | [Schema](#)

```
{  
  "name": "Meditación matutina",  
  "description": "15 minutos de meditación guiada",  
  "frequency": "Diario",  
  "startDate": "2024-01-01",  
  "endDate": "2024-12-31",  
  "userId": 1,  
  "habitTypeId": 2  
}
```

**Responses**

Code	Description	Links
201	Hábito creado exitosamente	<a href="#">No links</a>
400	Datos inválidos	<a href="#">No links</a>

**GET** /api/habits/habits Obtener todos los hábitos

Retorna una lista completa de hábitos

**Parameters**

No parameters

**Responses**

Code	Description	Links
200	Lista de hábitos encontrada	<a href="#">No links</a>
204	No hay hábitos registrados	<a href="#">No links</a>

**GET** /api/habits/habits/user/{userId} Obtener hábitos por usuario

Retorna todos los hábitos asociados a un usuario específico

**Parameters**

Name	Description
------	-------------

<b>userId</b> * required	ID del usuario integer(\$int64) (path)	<input type="text" value="userId"/>												
<b>Responses</b>														
<table border="1"> <thead> <tr> <th>Code</th> <th>Description</th> <th>Links</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Lista de hábitos encontrada</td> <td>No links</td> </tr> <tr> <td>204</td> <td>No hay hábitos registrados</td> <td>No links</td> </tr> <tr> <td>400</td> <td>ID inválido</td> <td>No links</td> </tr> </tbody> </table>			Code	Description	Links	200	Lista de hábitos encontrada	No links	204	No hay hábitos registrados	No links	400	ID inválido	No links
Code	Description	Links												
200	Lista de hábitos encontrada	No links												
204	No hay hábitos registrados	No links												
400	ID inválido	No links												
<b>GET</b>	/api/habits/habit/by-name-and-user	Buscar hábito por nombre y usuario												

<b>GET</b>	/api/habits/habit/by-name-and-user	Buscar hábito por nombre y usuario	Try it out						
Busca un hábito por su nombre y ID de usuario asociado									
<b>Parameters</b>									
<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>name</b> * required</td> <td>Nombre del hábito string (query)</td> </tr> <tr> <td><b>userId</b> * required</td> <td>ID del usuario integer(\$int64) (query)</td> </tr> </tbody> </table>				Name	Description	<b>name</b> * required	Nombre del hábito string (query)	<b>userId</b> * required	ID del usuario integer(\$int64) (query)
Name	Description								
<b>name</b> * required	Nombre del hábito string (query)								
<b>userId</b> * required	ID del usuario integer(\$int64) (query)								
<b>Responses</b>									
<b>Code</b>	<b>Description</b>	<b>Links</b>							
200	Hábito encontrado	No links							
204	Hábito no encontrado	No links							

<b>Users</b>	API para gestión de usuarios	Try it out
<b>GET</b> /api/users/user/{id} Obtener usuario por ID		
Busca un usuario específico utilizando su ID		
<b>Parameters</b>		
<b>Name</b>	<b>Description</b>	Try it out
<b>id</b> * required	ID del usuario (debe ser positivo) string (path)	<input type="text" value="id"/>

## Responses

Code	Description	Links
200	Usuario encontrado	No links
	Media type <code>*/*</code> Controls Accept header.	
	Example Value   Schema	
	<pre>{     "id": 123,     "username": "alvaromp",     "email": "alvaro@trackup.com" }</pre>	
400	ID inválido	No links
404	Usuario no encontrado	No links

## PUT /api/users/user/{id} Actualizar usuario existente

Actualiza los datos de un usuario utilizando su ID

### Parameters

Try it out

Name	Description
<b>id</b> * required	ID del usuario a actualizar integer(\$int64) (path) id

### Request body required

application/json

Nuevos datos del usuario

#### Example Value | Schema

```
{  
    "username": "alvaromp",  
    "email": "alvaro@trackup.com",  
    "password": "Passw0rd!"  
}
```

## Responses

Code	Description	Links
200	Usuario actualizado exitosamente	No links
	Media type <code>*/*</code> Controls Accept header.	
	Example Value   Schema	
	<pre>{     "id": 123,     "username": "alvaromp",     "email": "alvaro@trackup.com" }</pre>	
400	Datos de solicitud inválidos	No links
404	Usuario no encontrado	No links

## DELETE /api/users/user/{id} Eliminar usuario

Elimina un usuario del sistema utilizando su ID

### Parameters

Try it out

Name	Description
<b>id</b> * required	ID del usuario a eliminar integer(\$int64) (path) id

## Responses

Code	Description	Links
------	-------------	-------

204	Usuario eliminado exitosamente	No links
400	ID inválido	No links
404	Usuario no encontrado	No links

**POST /api/users/user** Crear nuevo usuario Try it out

Registra un nuevo usuario en el sistema

**Parameters**

No parameters

**Request body required** application/json

Datos del nuevo usuario

Example Value | Schema

```
{
  "username": "alvaromp",
  "email": "alvaro@trackup.com",
  "password": "Passw0rd!"
}
```

**Responses**

Code	Description	Links
201	Usuario creado exitosamente	No links
400	Datos de usuario inválidos	No links

**GET /api/users/users** Obtener todos los usuarios Try it out

Retorna una lista completa de todos los usuarios registrados

**Parameters**

No parameters

**Responses**

Code	Description	Links
200	Lista de usuarios encontrada	No links
204	No hay usuarios registrados	No links

**GET /api/users/user/{id}/habits** Obtener hábitos del usuario Try it out

Retorna todos los hábitos asociados a un usuario específico

**Parameters**

Name	Description
<b>id</b> <small>* required</small>	ID del usuario

Responses

Code	Description	Links
200	Lista de hábitos encontrada	No links
	Media type <input type="text" value="*/*"/> <small>Controls Accept header.</small> <a href="#">Example Value</a>   <a href="#">Schema</a>	
404	Usuario no encontrado	No links

GET /api/users/user/{id}/goals Obtener objetivos del usuario

Retorna todos los objetivos asociados a un usuario específico

Parameters

Name	Description
<b>id</b> * required	ID del usuario integer(\$int64) (path)

Try it out

Responses

Code	Description	Links
200	Lista de objetivos encontrada	No links
	Media type <input type="text" value="*/*"/> <small>Controls Accept header.</small> <a href="#">Example Value</a>   <a href="#">Schema</a>	
404	Usuario no encontrado	No links

GET /api/users/user/username/{username} Obtener usuario por nombre de usuario

Busca un usuario por su nombre de usuario único

Parameters

Name	Description
<b>username</b> * required	Nombre de usuario a buscar string (path)

Try it out

Responses

Code	Description	Links
200	Usuario encontrado	No links
	Media type <input type="text" value="*/*"/> <small>Controls Accept header.</small> <a href="#">Example Value</a>   <a href="#">Schema</a>	

```
{
  "id": 123,
  "username": "alvaromp",
  "email": "alvaro@trackup.com"
}
```

400 Nombre de usuario inválido

No links

404 Usuario no encontrado

No links

## Goals API para gestión de objetivos

**GET** /api/goals/goal/{id} Obtener objetivo por ID

Busca un objetivo específico por su ID único

### Parameters

[Try it out](#)

Name	Description
<b>id</b> * required	ID del objetivo (debe ser positivo) string (path)

### Responses

#### Code

#### Description

#### Links

200 Objetivo encontrado

No links

Media type

[\\*/\\*](#)

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{
  "id": 0,
  "name": "string",
  "description": "string"
}
```

400 ID inválido

No links

404 Objetivo no encontrado

No links

**PUT** /api/goals/goal/{id} Actualizar objetivo

Modifica los datos de un objetivo existente

### Parameters

[Try it out](#)

#### Name

#### Description

<b>id</b> * required	ID del objetivo a actualizar
integer(\$int64)	id

### Request body required

[application/json](#)

Datos actualizados del objetivo

[Example Value](#) | [Schema](#)

```
{
  "name": "string",
  "description": "string",
  "userId": 0
}
```

### Responses

#### Code

#### Description

#### Links

200 Objetivo actualizado exitosamente

No links

Media type

[\\*/\\*](#)

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{
  "id": 0,
  "name": "string",
  "description": "string"
}
```

400	ID inválido o datos incorrectos	No links
404	Objetivo no encontrado	No links

**DELETE** /api/goals/goal/{id} Eliminar objetivo Try it out

Elimina permanentemente un objetivo del sistema

**Parameters**

Name	Description
<b>id</b> * required	ID del objetivo a eliminar integer(\$int64) (path)
id	

**Responses**

Code	Description	Links
204	Objetivo eliminado exitosamente	No links
400	ID inválido	No links
404	Objetivo no encontrado	No links

**POST** /api/goals/goal Crear nuevo objetivo Try it out

Registra un nuevo objetivo en el sistema

**Parameters**

No parameters

**Request body** required application/json

Datos del nuevo objetivo

Example Value | Schema

```
{
  "name": "string",
  "description": "string",
  "userId": 0
}
```

**Responses**

Code	Description	Links
200	Objetivo creado exitosamente	No links
	Media type	
	*/*	
	Controls Accept header.	
	Example Value   Schema	
	[REDACTED]	
400	Datos inválidos	No links

**GET** /api/goals/goals Obtener todos los objetivos Try it out

Retorna una lista completa de objetivos registrados

**Parameters**

No parameters

**Responses**

Code	Description	Links
------	-------------	-------

200	Lista de objetivos encontrada	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">*/*</div> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{   "id": 0,   "name": "string",   "description": "string" }</pre>	
204	No hay objetivos registrados	No links

**GET /api/goals/goal/by-name** Buscar objetivo por nombre

Busca objetivos por nombre, opcionalmente filtrado por usuario

**Parameters**

[Try it out](#)

Name	Description
<b>name</b> <small>* required</small>	Nombre del objetivo (no vacío)
<small>string (query)</small>	<input type="text" value="name"/>
<b>userId</b>	ID del usuario (opcional)
<small>integer(\$int64) (query)</small>	<input type="text" value="userId"/>

**Responses**

Code	Description	Links
200	Objetivo encontrado	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">*/*</div> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{   "id": 0,   "name": "string",   "description": "string" }</pre>	
400	Nombre vacío o inválido	No links
404	Objetivo no encontrado	No links

**Schemas**

**UserRequest** v {

description: Datos requeridos para crear o actualizar un usuario. Incluye campos sensibles como la contraseña.

**username\***

- ✓ string  
pattern: ^[A-Z]\$  
example: alvaroamp
- Nombre de usuario único (no espacios, mín. 4 caracteres)

**email\***

- ✓ string(\$email)  
example: alvaro@trackup.com
- Correo electrónico válido

**password\***

- ✓ string(\$password)  
pattern: ^(?=.\*[A-Z])(?=.\*\d)(?=.\*[@\$!%\*?&])[A-Za-z\d@\$!%\*?&]{8,}\$
- example: Password!
- Contraseña segura

}

**UserResponse** v {

description: Respuesta estándar con datos de un usuario. Nunca incluye información sensible.

**id**

- ✓ integer(\$int64)  
readOnly: true  
example: 123
- ID único del usuario en el sistema

**username\***

- ✓ string  
example: alvaroamp
- Nombre de usuario único

**email\***

- ✓ string(\$email)  
example: alvaro@trackup.com
- Correo electrónico válido

}

```

HabitRequest ∨ {
    description: DTO para creación/actualización de hábitos
    name*:
        ✓ string
        example: Meditación matutina
        Nombre del hábito
    description:
        ✓ string
        example: 15 minutos de meditación guiada
        Descripción detallada
    frequency*:
        ✓ string
        example: Diario
        Frecuencia de ejecución
    startDate*:
        ✓ string($date)
        example: 2024-01-01
        Fecha de inicio (formato ISO: yyyy-MM-dd)
    endDate:
        ✓ string($date)
        example: 2024-12-31
        Fecha de finalización (formato ISO: yyyy-MM-dd)
    userId*:
        ✓ integer($int64)
        example: 1
        ID del usuario propietario
    habitTypeId*:
        ✓ integer($int64)
        example: 2
        ID del tipo de hábito asociado
}

```

```

HabitResponse ∨ {
    description: DTO de respuesta con información detallada de un hábito
    id:
        ✓ integer($int64)
        readOnly: true
        example: 1
        ID único del hábito
    name*:
        ✓ string
        example: Ejercicio diario
        Nombre del hábito
    description:
        ✓ string
        example: 30 minutos de cardio
        Descripción detallada del hábito
    frequency:
        ✓ string
        example: Diario
        Frecuencia de ejecución
    Enum:
        ✓ [ Diario, Semanal, Mensual ]
    startDate:
        ✓ string($date)
        example: 2024-01-01
        Fecha de inicio (formato ISO: yyyy-MM-dd)
    endDate:
        ✓ string($date)
        example: 2024-12-31
        Fecha de finalización (formato ISO: yyyy-MM-dd)
    habitTypeName:
        ✓ string
        example: Salud
        Nombre del tipo de hábito asociado
    userId:
        ✓ integer($int64)
        nullable: true
        example: 1
        ID del usuario propietario (excluido si es nulo)
}

```

```

HabitTypeRequest ∨ {
    description: DTO para crear/actualizar tipos de hábitos
    name*:
        ✓ string
        example: Productividad
        Nombre único del tipo de hábito
}

```

```

HabitTypeResponse ∨ {
    description: DTO con información básica de un tipo de hábito
    id:
        ✓ integer($int64)
        readOnly: true
        example: 1
        ID único del tipo de hábito
    name:
        ✓ string
        example: Salud
        Nombre del tipo de hábito
}

```

```

GoalRequestDTO ∨ {
    name:
        ✓ string
    description:
        ✓ string
    userId:
        ✓ integer($int64)
}

```

```

GoalResponseDTO ∨ {
    id:
        ✓ integer($int64)
    name:
        ✓ string
    description:
        ✓ string
}

```

