

API To Dos

DigitalHouse >
Coding School



**Certified Tech
Developer**

The Ultimate Degree

Índice

1. [Presentación](#)
2. [Autenticación](#)
3. [CRUD de tareas](#)

1 | Presentación

Para hacer que el proyecto To Do sea dinámico, utilizaremos una **API** para crear y administrar nuestras tareas.



API To Dos

Se trata de una API tipo REST que nos permite realizar las siguientes tareas:

- **Registrar un nuevo usuario.**
- **Iniciar sesión con un usuario existente.**
- **Crear una nueva tarea.**
- **Modificar una tarea existente.**
- **Obtener un listado de tareas del usuario.**
- **Acceder a una tarea específica del usuario.**
- **Eliminar una tarea.**

2 | Autenticación

REGISTRO	
Endpoint	/users
Método	POST
Headers	No requiere
Body (JSON)	{ "firstname": "string", // Requerido "lastname": "string", // Requerido "email": "string", // Requerido "password": "string" // Requerido }
Respuesta Exitosa (JSON)	{ "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InRhcnVzdEB0ZXN0YWZRzYXNzLmNvbSIsImkljoxLCJpYXQiOjE2MjU2MTU0OTV9.ClhW4omj9Lw8F29Poq1ROJEAZCnTRv785Yr6PWL3v3Q" }

LOGIN

Endpoint	/users/login
Método	POST
Headers	No requiere
Body (JSON)	{ "email": "string", // Requerido "password": "string" // Requerido }
Respuesta Exitosa (JSON)	{ "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InRhcnVzdEB0ZXN0YWRzYXNzLmNvbSIsImkljoxLCJpYXQiOjE2MjU2MTU0OTV9.ClhW4o mj9Lw8F29Poq1ROJEAZCnTRv785Yr6PwL3v3Q" }

3 | CRUD de tareas

CREAR UNA TAREA

Endpoint	/tasks
Método	POST
Headers	<code>"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"</code>
Body (JSON)	<pre>{ "description": "Aprender Javascript", // Requerido "completed": false // Opcional (default: false) }</pre>
Respuesta Exitosa (JSON)	<pre>{ "id": 1, "description": "Aprender Javascript", "completed": false, "userId": 1, "createdAt": "2021-06-30T22:53:09.549Z" }</pre>

OBTENER UN LISTADO DE TAREAS

Endpoint	/tasks
Método	GET
Headers	"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
Body	No Requiere
Respuesta Exitosa (JSON)	[{ "id": 1, "description": "Aprender Javascript", "completed": false, "userId": 1, "createdAt": "2021-06-30T22:53:09.549Z" }]

OBTENER UNA TAREA

Endpoint	/tasks/{id}
Método	GET
Headers	"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
Body	No Requiere
Respuesta Exitosa (JSON)	<pre>{ "id": 1, "description": "Aprender Javascript", "completed": false, "userId": 1, "createdAt": "2021-06-30T22:53:09.549Z" }</pre>

MODIFICAR UNA TAREA

Endpoint	/tasks/{id}
Método	PUT
Headers	"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
Body (JSON)	{ "description": "Aprender Javascript", // Opcional "completed": false // Opcional }
Respuesta Exitosa (JSON)	{ "id": 1, "description": "Aprender Javascript", "completed": false, "userId": 1, "createdAt": "2021-06-30T22:53:09.549Z" }

ELIMINAR UNA TAREA

Endpoint	/tasks/{id}
Método	DELETE
Headers	<pre>"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"</pre>
Body (JSON)	<pre>{ "description": "Aprender Javascript", // Opcional "completed": false // Opcional }</pre>
Respuesta Exitosa (JSON)	<pre>"Tarea eliminada satisfactoriamente"</pre>

A tener en cuenta

Con estos cuadros podemos destacar información que nos parezca importante.



No olvidar enviar el token (JWT) dentro del *header* para realizar cualquier petición relacionadas a tareas.



Caso contrario, la API devolverá un error.

A tener en cuenta

La API puede devolver 2 tipos de respuesta:



Respuesta exitosa: cuando la petición ha sido procesada satisfactoriamente. Devuelve un status 2XX.



Error: cuando la petición no puede procesarse debido a algún error. Devuelve un status 4XX si se trata de un error en la petición, o un status 500 si el error es del servidor.

DigitalHouse>
Coding School