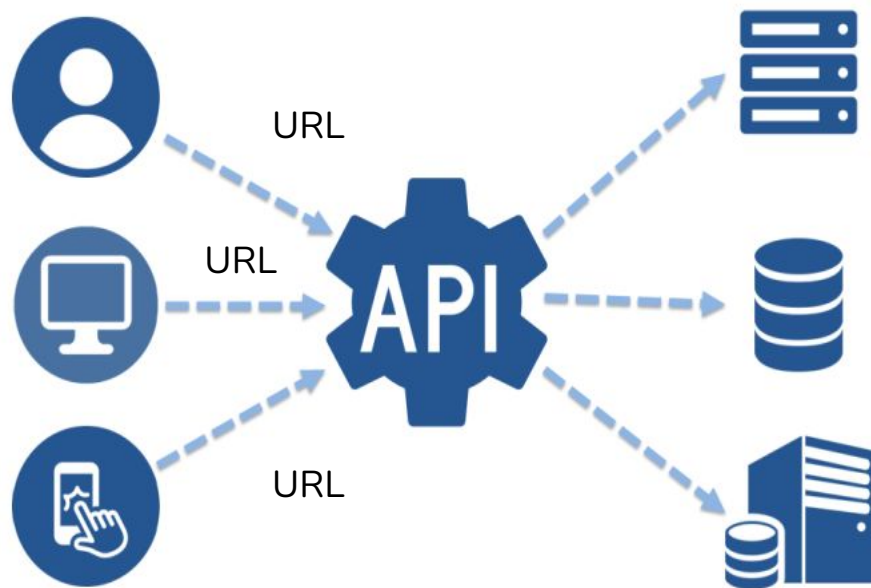


API Rest

DEV.F
DESARROLLAMOS(PERSONAS);

dev



API

(Application Programming Interface)

Una interfaz de programación de aplicaciones (API) es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones.

Es lo que permite que sus productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva.

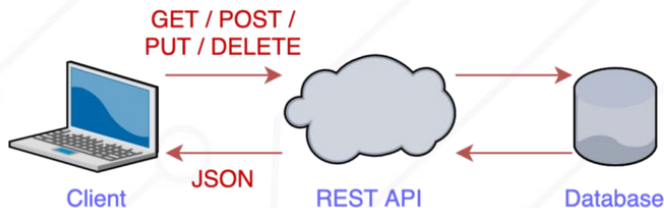
API + Rest = API Rest

API

API utiliza peticiones HTTP responsables de las operaciones básicas necesarias para la manipulación de datos (GET, POST, etc).

REST

Rest, es un conjunto de restricciones que se utilizan para que las solicitudes HTTP cumplan con las directrices definidas en la arquitectura: cliente servidor, sin estado, con cache, etc.



API REST

API Rest es el conjunto de buenas prácticas utilizadas en las peticiones HTTP realizadas por una API en una aplicación web.

Es decir, cuando se habla de API Rest, significa utilizar una API para acceder a aplicaciones back-end, de manera que esa comunicación se realice con los estándares definidos por el estilo de arquitectura Rest.

¿Qué es Rest?

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP y nos sirve para obtener, generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

- Protocolo cliente/servidor "SIN ESTADO"
- Utiliza verbos http
 - GET (leer)
 - POST (crear)
 - PUT (editar)
 - DELETE (borrar)
- Devuelve
 - JSON
 - XML



HTTP STATUS
200 – OK
201 – CREATED
403 – FORBIDDEN
404 – NOT FOUND



Ejemplos:

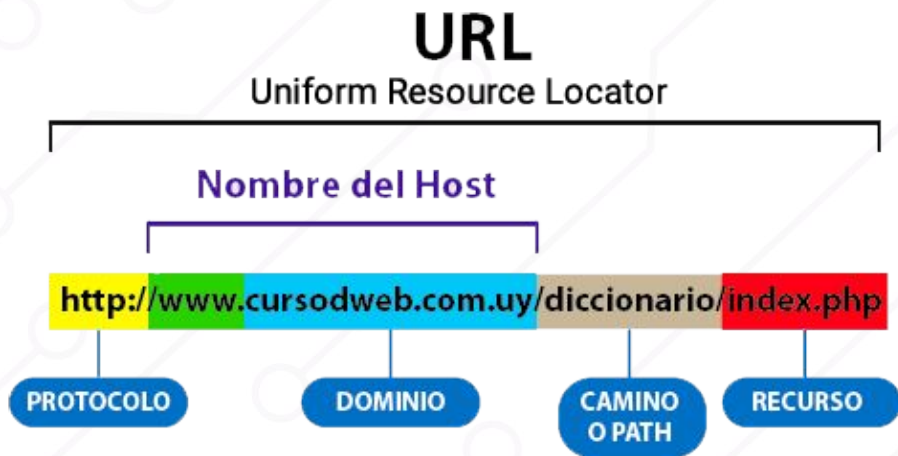
```
GET    /rest/api/pedido/{id}
DELETE /rest/api/pedido/{id}
POST   /rest/api/pedido
PUT     /rest/api/pedido/{id}/producto/{idProducto}
```



Devuelve
JSON/XML

```
{
  "nombre": "Pepito",
  "coches" : ["Ford", "Fiat", "Audi"]
}
```

URL



- Protocolo:** http, https, ftp, file...
- Subdominio**
- Dominio:** .com, .org, .edu... ccTLD, .com.uy, .com.ar, .es
- Path o camino al directorio del curso**
- Nombre de archivo:** .php, .htm, .html, .asp, .jsp...

Las URL son términos para el frontend. Si lo que devuelve es información y es para el Backend nos referimos a estos como Endpoints.

Endpoints: URI que hacen referencia a una API.



Nombrando Endpoints en API Rest

/users // lista todos los usuarios

/users/1 // GET lista a un usuario en específico //PUT

/users/123/orders // lista los pedidos de un usuario específico

/users/123/orders/0001 // lista una orden especifica de un usuario específico

Endpoints y Verbos Http

Un mismo Endpoint, hara diferentes acciones dependiendo el verbo http usado para accederlo

Recuros / Estado	POST	GET	PUT	DELETE
/casas	Crea una casa	Devuelve una lista de todos las casas	Modifica casas	Borra todas las casas
/casas/123	error	Devuelve los detalles de la casa 123	Modifica la casa	Borra la casa

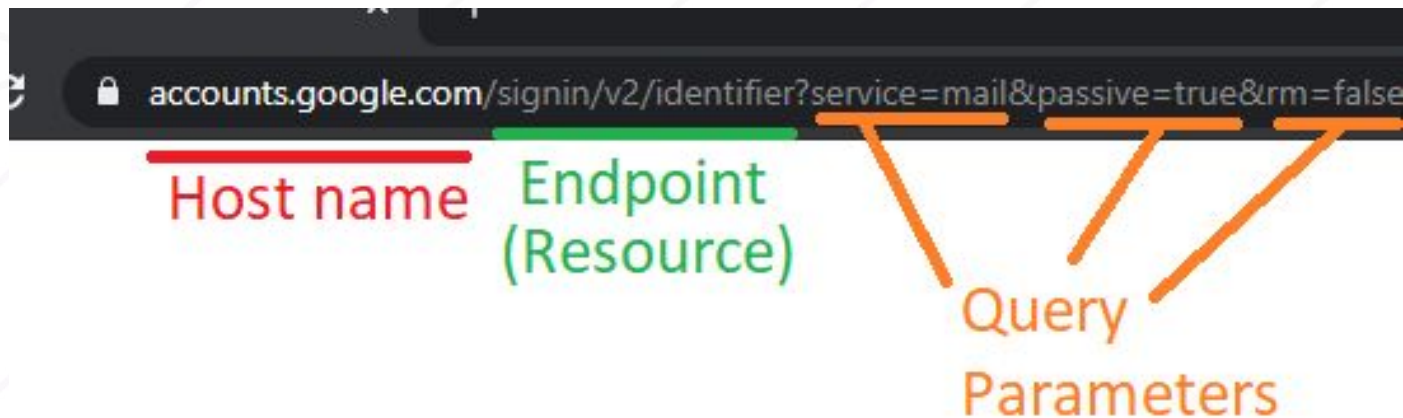
/casas/3 DELETE

/casas/3 GET

/casas POST

Query Parameters

Un query param es la clave valor (ejemplo: service=mail) que vemos al final de la URL, y como regla, siempre deberán estar después del símbolo de interrogación. Además, una URL puede tener N query params, separadas por &, cómo el siguiente ejemplo:

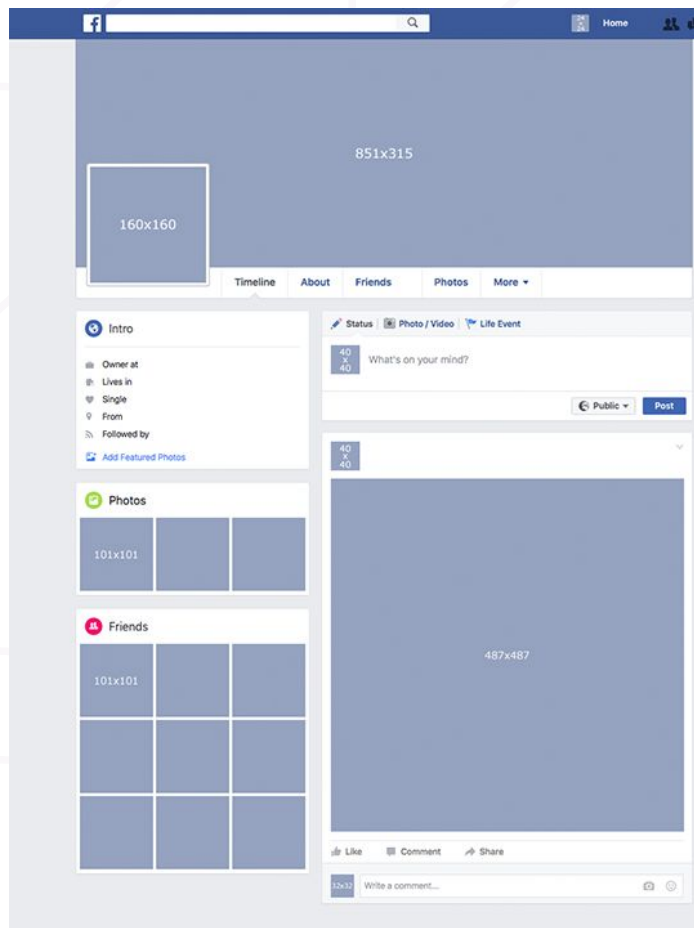


JSON

```
1 {  
2   "name": "Christopher Co",  
3   "age": 29,  
4   "level": 7,  
5   "gender": "M",  
6   "status": "good"  
7 }
```

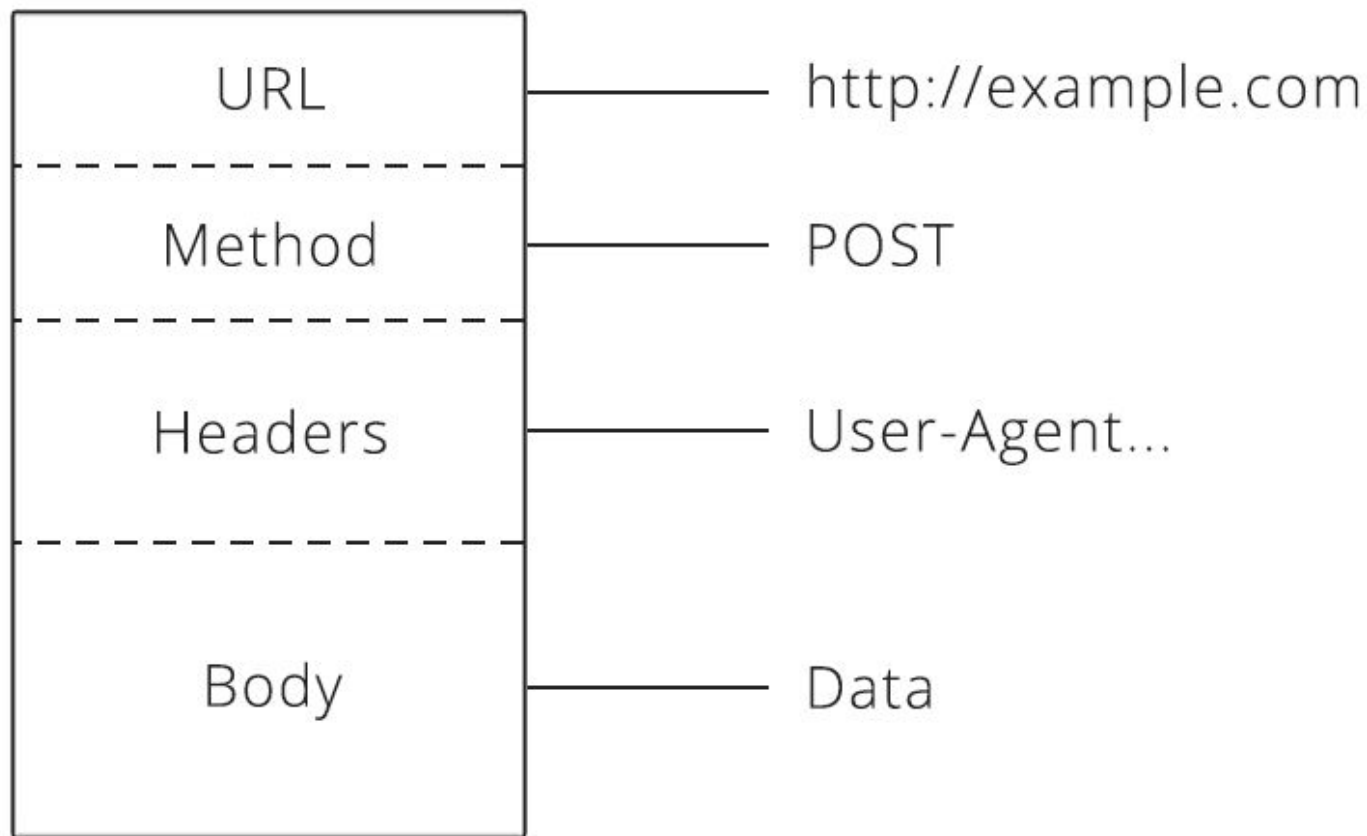
JSON es hoy en día el método estándar de comunicación para enviar/recibir datos del servidor. Tiene un formato similar a un objeto en JavaScript.

JSON = JavaScript Object Notation



```
{
  "_id": {
    "$oid": "5c3fd50c680caf5787cd11bb"
  },
  "is_important": false,
  "applied_codes": [],
  "is_active": true,
  "first_name": "Raul",
  "last_name": "Prueba Pruebita",
  "phone": "7821076070",
  "email": "dagorik@gmail.com",
  "approaches": [{}],
  "programs": [],
  "createdAt": {[]},
  "updatedAt": {
    "$date": "2019-01-28T23:28:11.470Z"
  },
  "__v": 1
}
```





Request

GET

Pedir datos al servidor

POST

Mandar datos al servidor

PUT

Actualizar datos en el servidor

PATCH

Actualizar datos en el servidor

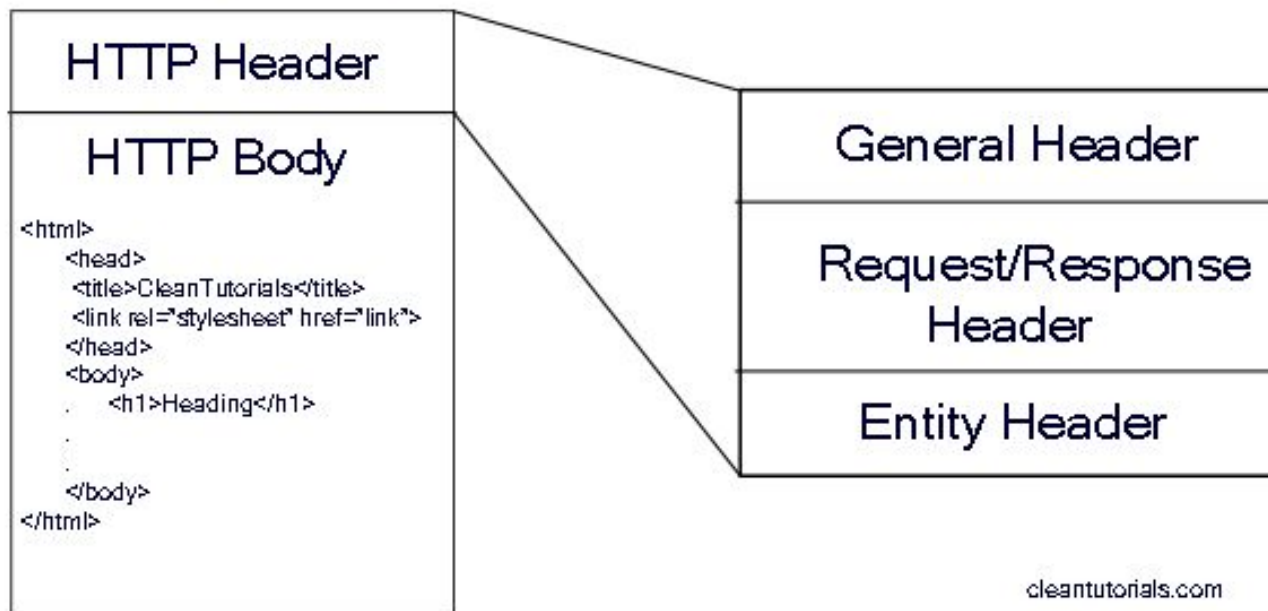
DELETE

Eliminar datos del servidor

Headers

Son meta-información relevante para que el cliente o el servidor conozca más acerca de la petición.

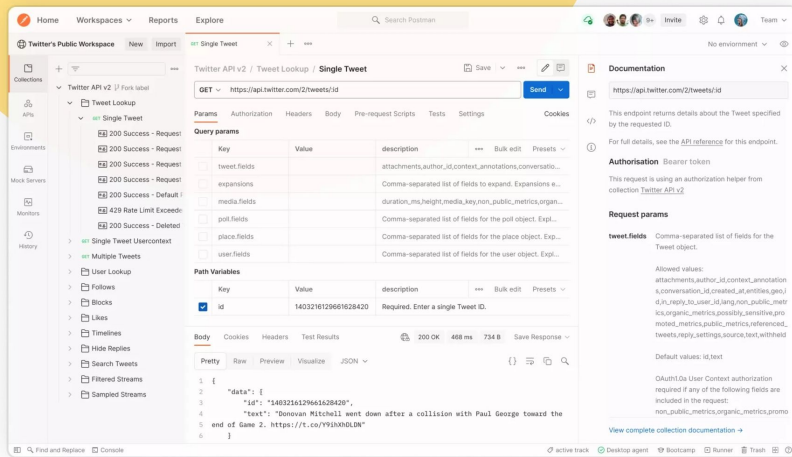
HTTP Request/response



Accediendo a API Rest

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Postman

Postman nace como una herramienta que principalmente nos permite crear peticiones sobre APIs de una forma muy sencilla y poder, de esta manera, probar las APIs.

El interés fundamental de Postman es que lo utilizemos como una herramienta para hacer peticiones a APIs y generar colecciones de peticiones que nos permitan probarlas de una manera rápida y sencilla.



<https://www.postman.com/>

DEV.F

Download Insomnia

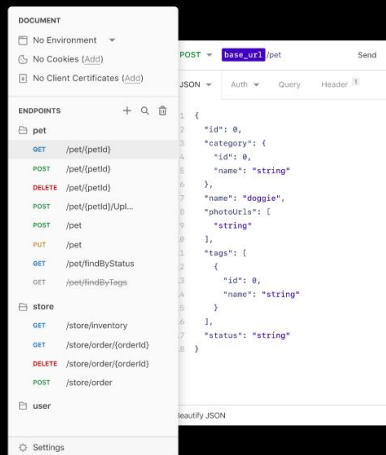
Start building, designing, testing better APIs through spec-first development driven by an APIOps CI/CD pipelines.

Download Insomnia for Windows

By downloading and using Insomnia, I agree to the Privacy Policy and Terms.

What's New? Changelog

Not your OS? Download for MacOS / Ubuntu or See all downloads.



Insomnia

Es similar a Postman, con una interfaz un poco más sencilla.



Insomnia

<https://insomnia.rest/>

APIs para probar

DEV.F
DESARROLLAMOS(PERSONAS);

dev

PokeAPI proudly announces Beta support for GraphQL. Access our free console at beta.pokeapi.co/graphql/console and take a look at the [documentation](#) ✕

Sword and Shield data might be inaccurate and lacking in various aspects due to the fact that it is not taken directly from Nintendo's Pokemon ROMs. If you spot any mistake, please report it [here](#) ✕



The RESTful Pokémon API

Serving over 60,000,000 API calls each month!

All the Pokémon data you'll ever need in one place,
easily accessible through a modern RESTful API.

[Check out the docs!](#)

Try it now!

<https://pokeapi.co/api/v2/>

Submit

Need a hint? Try [pokemon/ditto](#), [pokemon/1](#), [type/3](#), [ability/4](#), or [pokemon?limit=100&offset=200](#).

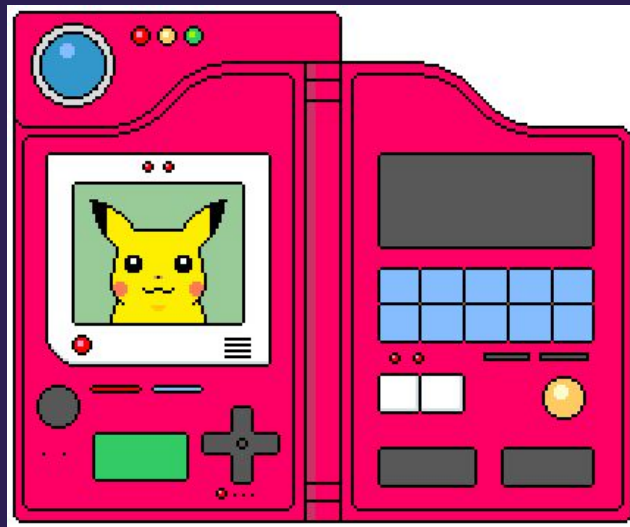
Direct link to results: <https://pokeapi.co/api/v2/pokemon/ditto>

Resource for ditto

```
▼ abilities: [] 2 items
▼ 0: {} 3 keys
  ▼ ability: {} 2 keys
    name: "lieber"
    url: "https://pokeapi.co/api/v2/ability/7/"
    is_hidden: false
    slot: 1
  ▼ 1: {} 3 keys
    ▼ ability: {} 2 keys
      name: "imposter"
      url: "https://pokeapi.co/api/v2/ability/150/"
      is_hidden: true
      slot: 3
    base_experience: 101
```

<https://pokeapi.co/>

PokéAPI



SWAPI

The Star Wars API

(what happened to swapi.co?)

All the Star Wars data you've ever wanted:

Planets, Spaceships, Vehicles, People, Films and Species

From all SEVEN Star Wars films

Now with The Force Awakens data!

Try it now!

<https://swapi.dev/api/>

request

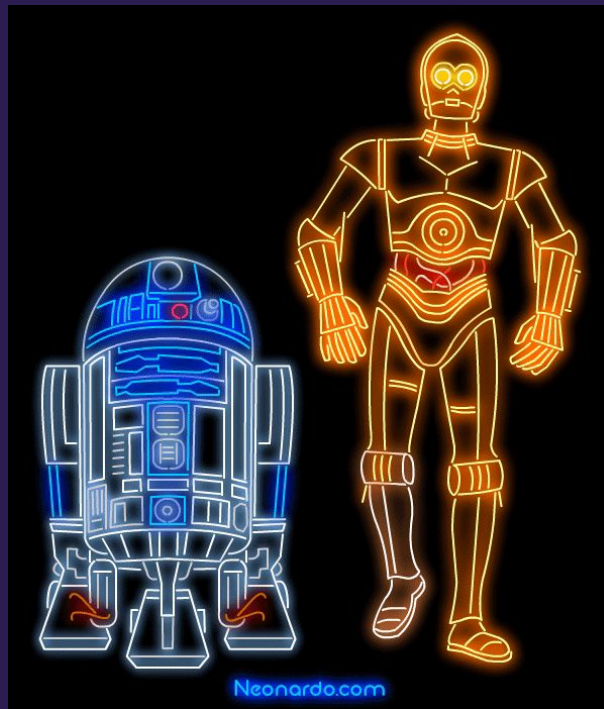
Need a hint? try [people/1](#) or [planets/3](#) or [starships/9](#)

Result:

```
{
  "name": "Luke Skywalker",
  "height": "172",
  "mass": "77",
  "hair_color": "blond",
  "skin_color": "fair",
  "eye_color": "blue",
  "birth_year": "19BBY",
  "gender": "male",
  "homeworld": "https://swapi.dev/api/planets/1/",
  "films": [
    "https://swapi.dev/api/films/2/",
    "https://swapi.dev/api/films/6/",
    "https://swapi.dev/api/films/3/",
    "https://swapi.dev/api/films/1/",
    "https://swapi.dev/api/films/7/"
  ],
  "species": [
```

<https://swapi.dev/>

SWAPI (Star Wars)



DEV.F