CURSO .GO

FUNDAMENTOS DE SERVICIOS WEB

JIMCOSTDEV

¿Qué es una API?

- API = Interfaz de Programación de Aplicaciones
- Es el "contrato" o "menú" entre dos programas.
- Define:
  - Qué métodos usar
  - Qué datos enviar
  - Qué esperar de vuelta

JIMCOSTDEV

¿Qué es un Servicio Web?

- Conjunto de reglas para que dos programas intercambien información por Internet.
- Es una API accesible por Internet.
- Todas las llamadas viajan sobre HTTP.

JIMCOSTDEV

Arquitectura Cliente-Servidor

Dos roles clave:

- **Cliente**: Genera la petición (detalles, parámetros).
- **Servidor**: Procesa la petición y devuelve un resultado o error.

JIMCOSTDEV

Arquitectura Cliente-Servidor

Intercambian:

- **Cabeceras** (Headers):
  - Etiquetas del paquete: formato, tipo de contenido, idioma, autenticación, tamaño...
- **Cuerpo** (Body):
  - El "contenido del envío": JSON, HTML, binario, datos a transferir.

JIMCOSTDEV

Estructura de una URL

Localizador Uniforme de Recursos

- La "dirección" exacta para encontrar algo en Internet.

protocolo://servidor:puerto/ruta/al/recurso

JIMCOSTDEV

Ejemplo de URL y HTTPS

- protocolo://servidor:puerto/ruta/al/recurso
- https://github.com/JimcostDev/curso-go/blob/master/main.go

Puerto implícito: 443

JIMCOSTDEV

Peticiones HTTP

✅ Una petición HTTP incluye:

- **Método**: Acción a realizar (GET, POST, PUT, DELETE...).
- **URL**: Recurso al que apuntamos.
- **Headers**: Metadatos (tipo de contenido, autenticación...).
- **Body**: Datos que enviamos (solo en algunos métodos).

JIMCOSTDEV

Peticiones HTTP

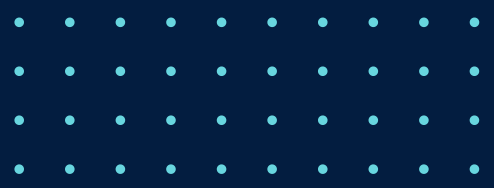
✅ El servidor responde con:

- Un código de estado (numérico).
- Opcionalmente, un cuerpo.

JIMCOSTDEV

Principales Códigos de Respuesta

- 200 OK: Todo salió bien.
- 201 Creado: Recurso creado correctamente.
- 204 Sin Contenido: Operación exitosa, nada que devolver.
- 400 Petición Errónea: Algo mal en la solicitud (JSON mal formado).
- 401 No Autorizado: Necesitas autenticarte.
- 403 Prohibido: Autenticado, pero sin permiso.
- 404 No Encontrado: Recurso solicitado no existe.
- 500 Error Interno del Servidor: Problema inesperado en el servidor (¡error en backend!).



JIMCOSTDEV

¿Qué es REST?

- Un estilo para crear servicios web.
- Usa URLs para identificar recursos (usuarios, productos).
- Opera con acciones básicas de HTTP (GET, POST, PUT, DELETE).
- Cada petición es independiente (el servidor no guarda info entre llamadas).

JIMCOSTDEV

¿Qué es REST?

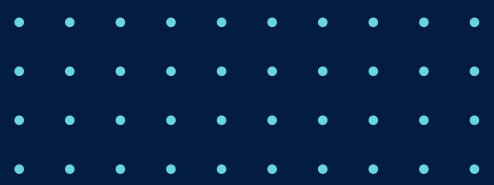
Ejemplo: https://api.com/productos/123

- **GET**: Obtener datos del producto 123.
- **PUT**: Actualizar su precio.
- **DELETE**: Borrarlo.

JIMCOSTDEV

Resumen

- Qué es un servicio web y su propósito.
- El rol del cliente y el servidor.
- Estructura de una URL y transmisión con HTTP.
- Qué es una API y cómo REST organiza peticiones.
- Los códigos de estado HTTP esenciales.



JIMCOSTDEV