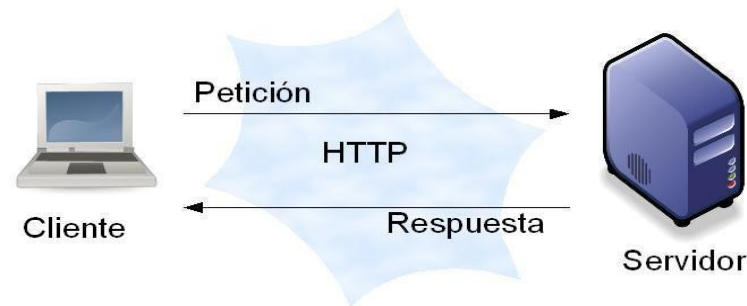


Conceptos Iniciales de Backend II



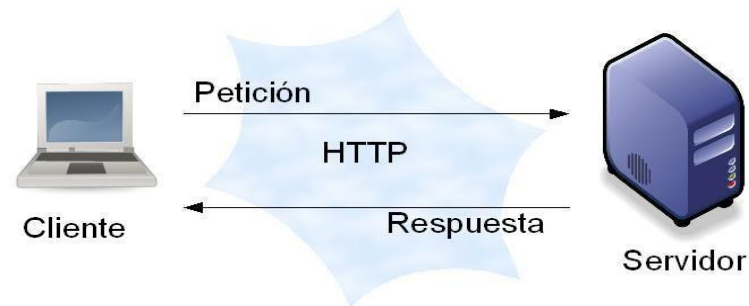
Protocolo HTTP

HTTP es el acrónimo para Hypertext Transfer Protocol (protocolo de transferencia de hipertexto, en español). HTTP es un protocolo de transferencia de la capa de aplicación basado en texto y se considera la base para la comunicación de datos entre los dispositivos de una red. Durante el proceso de solicitud-respuesta, HTTP usa reglas y estándares predefinidos para el intercambio de información. En general, HTTP es el protocolo que clientes y servidores usan para comunicarse.

TCP/IP es un modelo, pero también una pila de protocolos en la que se encuentra el HTTP. Respecto al Modelo OSI

Un dato importante sobre el protocolo HTTP es que es stateless, es decir, “sin estado”; pero, ¿qué significa esto? Significa que cada solicitud que el cliente hace es una transacción independiente, que no está relacionada a ninguna solicitud previa y cada una es diferente para el servidor. De esta forma, si se realizan diferentes solicitudes al mismo tiempo, una no sabe que la otra existe y el servidor no almacena ninguna información sobre el estado del cliente. Tan pronto como se realiza la conexión TCP, toda la información intercambiada se pierde. La reducción en el uso de la memoria en el servidor y la reducción de los problemas por una sesión expirada son las ventajas de ello.

Conceptos Iniciales de Backend II



Protocolo HTTP

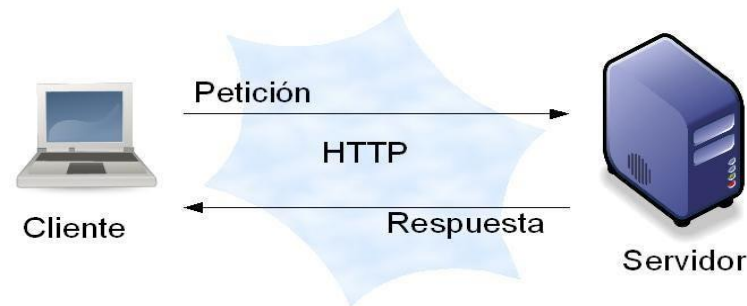
La comunicación con HTTP

Cuando un cliente quiere comunicarse con un servidor, lo primero que sucede, después que el usuario escribe la URL en el navegador o va a otra página, es que se abre una conexión TCP/IP y la solicitud HTTP es enviada al servidor. En esa solicitud, hay un mensaje con una serie de datos que describen lo que el usuario ha solicitado. Luego, el servidor envía la respuesta al cliente, que también incluye datos que pueden ser leídos. Finalmente, el proceso de solicitud-respuesta es finalizado.

¿Qué es una solicitud?

La solicitud (*request*, en inglés) es lo que el cliente necesita del servidor. Ese mensaje incluye datos específicos para describir lo que es solicitado. Los principales componentes de una solicitud son:
El método, que indica la acción que el cliente quiere realizar, como:
GET, POST, PUT, DELETE

Conceptos Iniciales de Backend II



Protocolo HTTP

¿Qué es una respuesta?

La respuesta (response) dada por el servidor contiene la información solicitada por el cliente o le informa si hay un error respecto a lo que fue solicitado. Incluye los siguientes elementos:

El encabezado de respuesta: contiene la versión del protocolo, el código de estatus de la solicitud y el tipo de contenido que está incluido en el cuerpo.

El código de estatus: indica si la solicitud fue exitosa o no. La respuesta proporciona códigos específicos, como:

- 200: la solicitud fue respondida de forma exitosa;
- 301: la solicitud fue movida permanentemente;
- 401: la solicitud no fue autorizada por el servidor;
- 404: la solicitud no fue encontrada por el servidor, y
- 500: error interno del servidor.

versión del protocolo código de estatus mensaje de estatus

HTTP/2

200

OK

Date: Wed, 31 Mar 2021 14:28:12 GMT
Server: Apache
Last-Modified: Wed, 31 Mar 14:28:12 GMT
ETag: "215af34563dc-eh13"
Accept-Ranges: bytes
Content-Length: 31257
Content-Type: text/html

encabezados

Conceptos Iniciales de Backend II



API

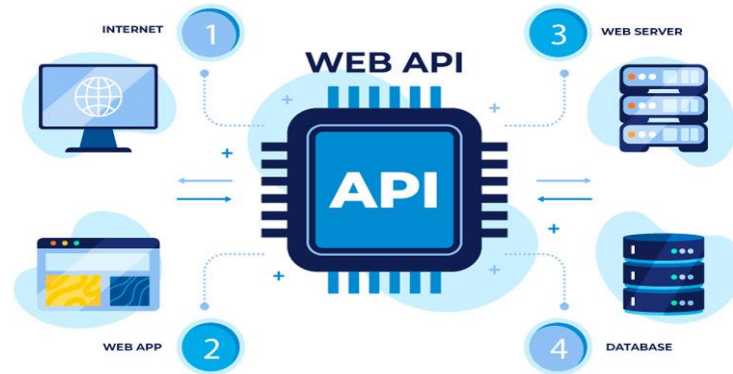
Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.

¿Cómo funciona una API?

Las API permiten que tus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API te otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones; y ofrecen oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

A veces, las API se consideran como contratos, con documentación que representa un acuerdo entre las partes: si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte.

Conceptos Iniciales de Backend II



API

¿Cómo funcionan las API?

Es útil pensar en la comunicación de la API en términos de una solicitud y respuesta entre un cliente y un servidor. La aplicación que envía la solicitud es el cliente y el servidor proporciona la respuesta. La API es el puente que establece la conexión entre ellos.

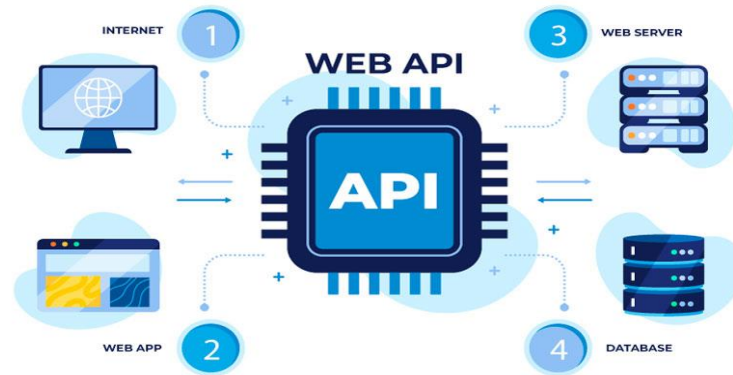
Una forma sencilla de entender cómo funcionan las API es con un ejemplo común: el procesamiento de pagos de terceros. Cuando un usuario compra un producto en un sitio de comercio electrónico, el sitio podría solicitar al usuario "Pagar con PayPal" u otro tipo de sistema externo. Esta función depende de las API para realizar la conexión. Cuando el comprador hace clic en el botón de pago, se envía una llamada a la API para recuperar la información. Esta es la solicitud. Esta solicitud se procesa desde una aplicación al servidor web a través del identificador uniforme de recursos (URI) de la API e incluye un verbo de solicitud, una cabecera y, a veces, un cuerpo de solicitud.

Tras recibir una solicitud válida desde la página web del producto, la API llama al programa externo o al servidor web, en este caso, al sistema de pago externo.

El servidor envía una respuesta a la API con la información solicitada.

La API transfiere los datos a la aplicación solicitante inicial, en este caso, el sitio web del producto.

Conceptos Iniciales de Backend II



API REST

¿Cómo funcionan las API?

REST es un conjunto de principios de arquitectura de API web. Las API de REST, también conocidas como API RESTful, son API que cumplen con ciertas restricciones de la arquitectura de REST. Las API de REST utilizan solicitudes HTTP como GET, PUT, HEAD y DELETE para interactuar con los recursos. REST hace que los datos estén disponibles como recursos, y cada recurso está representado por un URI único. Los clientes solicitan un recurso proporcionando su URI.

Características de las API REST

Son un conjunto de directrices que las aplicaciones usan para comunicarse.

El servidor identifica cada recurso con un identificador único de recursos, como un URL.

Los verbos del protocolo HTTP se usan para definir operaciones sobre los recursos de la API.

Las URL de los endpoint de la API están orientadas a recursos.

HTTP proporciona esquemas de autenticación para implementar API REST.

Ejemplos de API REST API de Power BI, API REST de AWS, API REST de Flickr, API REST de Facebook, API REST de Twitter.

Conceptos Iniciales de Backend II



Spring Boot

Java Spring Framework (Spring Framework) es una popular estructura empresarial de código abierto para crear aplicaciones independientes de nivel de producción que se ejecutan en la **máquina virtual Java (JVM)**.

Java Spring Boot (Spring Boot) es una herramienta que hace que el desarrollo de **aplicaciones web y microservicios** con Spring Framework sea **más rápido y fácil**

Características

- Crear aplicaciones Spring independientes
- Incruste Tomcat, Jetty o Undertow directamente (sin necesidad de implementar archivos WAR)
- Proporcionar dependencias de "inicio" con opiniones para simplificar la configuración de la compilación
- Configurar automáticamente Spring y bibliotecas de terceros siempre que sea posible
- Proporcionar funciones listas para producción, como métricas, controles de estado y configuración externalizada.
- No requiere generación de código y no requiere configuración XML

Conceptos Iniciales de Backend II



Spring Boot

¿Qué aporta Spring Boot a Spring Framework?

Configuración automática

Significa que las aplicaciones se inicializan con **dependencias preestablecidas** que no tienen que configurarse manualmente. Como Java Spring Boot viene con funciones de **configuración automática integradas**, configura automáticamente tanto el Spring Framework subyacente como los paquetes de terceros según su configuración (y según las mejores prácticas, lo que ayuda a evitar errores).

Aunque puede anular estos valores predeterminados una vez que se completa la inicialización, la función de configuración automática de Java Spring Boot le permite comenzar a desarrollar sus aplicaciones basadas en Spring **rápidamente y reduce la posibilidad de errores humanos**.

Enfoque obstinado

Spring Boot utiliza un enfoque obstinado para agregar y configurar dependencias de inicialización, según las necesidades de su proyecto. Siguiendo su propio criterio, Spring Boot elige qué paquetes instalar y qué **valores predeterminados** usar, en lugar de pedirle que tome todas esas decisiones usted mismo y configure todo manualmente.

Aplicaciones independientes

Spring Boot ayuda a los desarrolladores a crear aplicaciones que *simplemente se ejecutan*. Específicamente, le permite crear aplicaciones independientes que se ejecutan por sí mismas, sin depender de un servidor web externo, al integrar un servidor web como **Tomcat o Netty** en su aplicación durante el proceso de inicialización. Como resultado, puede iniciar su aplicación en cualquier plataforma con el comando **Ejecutar** (puede renunciar esta función para crear aplicaciones sin un servidor web integrado).

Gracias

CESDE | comfama

cesde.edu.co

