

Servicios REST, Códigos de Estado HTTP y Verbos (Métodos) HTTP:

1. Servicios REST:

Definición: REST (Representational State Transfer) es un estilo arquitectónico que establece restricciones para el diseño de servicios web. Opera sobre el protocolo HTTP y sigue el principio cliente-servidor.

2. Creación de Servicios REST:

2.1. Recursos:

En REST, los recursos son entidades identificables (usuarios, productos) con URIs únicas.

2.2. Operaciones CRUD:

Se utilizan las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para manipular los recursos.

2.3. URIs:

Cada recurso tiene una URI única que lo identifica, como /usuarios/1 para el usuario con ID 1.

3. Manejo de Servicios REST:

3.1. Implementación del Servidor:

Utiliza un lenguaje y un framework web para gestionar solicitudes HTTP.

3.2. Métodos HTTP:

GET para obtener información, POST para crear, PUT/PATCH para actualizar, DELETE para eliminar.

3.3. Respuestas:

El servidor responde con códigos de estado HTTP (200 OK, 201 Created) y datos (generalmente JSON o XML).

3.4. Autenticación y Autorización:

Implementa mecanismos para verificar identidad y controlar acceso.

3.5. Documentación API:

Proporciona documentación clara sobre recursos, métodos y cómo usar la API.

4. Códigos de Estado HTTP y Verbos (Métodos) HTTP:

4.1. Códigos de Estado HTTP

1xx (Respuestas Informativas):

- 100 Continue
- 101 Switching Protocols
- 102 Processing
- 103 Early Hints

2xx (Respuestas Exitosas):

- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authoritative Information
- 204 No Content
- 205 Reset Content
- 206 Partial Content
- 207 Multi-Status
- 208 Already Reported
- 226 IM Used

3xx (Redirecciones):

- 300 Multiple Choices
- 301 Moved Permanently
- 302 Found
- 303 See Other
- 304 Not Modified
- 305 Use Proxy
- 307 Temporary Redirect
- 308 Permanent Redirect

4xx (Errores del Cliente):

- 400 Bad Request
- 401 Unauthorized
- 402 Payment Required
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 406 Not Acceptable
- 407 Proxy Authentication Required
- 408 Request Timeout
- 409 Conflict
- 410 Gone
- 411 Length Required
- 412 Precondition Failed
- 413 Payload Too Large
- 414 URI Too Long
- 415 Unsupported Media Type
- 416 Range Not Satisfiable
- 417 Expectation Failed
- 418 I'm a teapot
- 421 Misdirected Request
- 422 Unprocessable Entity
- 423 Locked
- 424 Failed Dependency
- 425 Too Early

- 426 Upgrade Required
- 428 Precondition Required
- 429 Too Many Requests
- 431 Request Header Fields Too Large
- 451 Unavailable For Legal Reasons

5xx (Errores del Servidor):

- 500 Internal Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout
- 505 HTTP Version Not Supported
- 506 Variant Also Negotiates
- 507 Insufficient Storage
- 508 Loop Detected
- 510 Not Extended
- 511 Network Authentication Required

4.2. Verbos (Métodos) HTTP:

GET: Obtener información

Descripción: El método GET se utiliza para solicitar la información de un recurso específico o una colección de recursos desde el servidor. La solicitud no debe tener un impacto significativo en el estado del servidor, es decir, debe ser un proceso seguro e idempotente.

Ejemplo de Uso:

1. Solicitud: GET /usuarios/1
2. Respuesta Esperada: 200 OK con los detalles del usuario con ID 1.

POST: Crear un nuevo recurso

Descripción: POST se utiliza para enviar datos al servidor con el propósito de crear un nuevo recurso. Los datos enviados suelen estar en el cuerpo de la solicitud y el servidor devuelve la ubicación del recurso creado.

Ejemplo de Uso:

1. Solicitud: POST /usuarios
2. Cuerpo de la Solicitud: Datos del nuevo usuario.
3. Respuesta Esperada: 201 Created con la ubicación del nuevo usuario.

PUT: Actualizar un recurso completo

Descripción: PUT se utiliza para actualizar completamente un recurso en el servidor. La solicitud incluye la representación completa del recurso, y cualquier dato no proporcionado se considera nulo o vacío.

Ejemplo de Uso:

1. Solicitud: PUT /usuarios/1
2. Cuerpo de la Solicitud: Nueva información completa del usuario con ID 1.

3. Respuesta Esperada: 200 OK si la actualización es exitosa.

PATCH: Actualizar parcialmente un recurso

Descripción: PATCH se utiliza para realizar actualizaciones parciales en un recurso. Solo se envían los campos que se desean modificar, y el servidor aplica esos cambios al recurso existente.

Ejemplo de Uso:

1. Solicitud: PATCH /usuarios/1
2. Cuerpo de la Solicitud: Datos a actualizar (por ejemplo, solo el campo de correo electrónico).
3. Respuesta Esperada: 200 OK si la actualización parcial es exitosa.

DELETE: Eliminar un recurso

Descripción: DELETE se utiliza para solicitar al servidor que elimine un recurso específico.

Ejemplo de Uso:

1. Solicitud: DELETE /usuarios/1
2. Respuesta Esperada: 204 No Content si la eliminación es exitosa.

OPTIONS: Obtener información sobre métodos permitidos

Descripción: OPTIONS se utiliza para obtener información sobre los métodos HTTP permitidos en un recurso específico. Esto puede incluir información sobre la autenticación, los encabezados permitidos, etc.

Ejemplo de Uso:

1. Solicitud: OPTIONS /usuarios
2. Respuesta Esperada: 200 OK con detalles sobre los métodos permitidos.

HEAD: Similar a GET, pero solo obtiene encabezados

Descripción: HEAD es similar a GET, pero se utiliza cuando solo se necesitan los encabezados de respuesta y no el cuerpo. Es útil para verificar la existencia de un recurso o para obtener información sobre su última modificación sin descargar el cuerpo completo.

Ejemplo de Uso:

1. Solicitud: HEAD /usuarios/1
2. Respuesta Esperada: 200 OK con los encabezados correspondientes, pero sin el cuerpo de la respuesta.