Metodos Petición HTTP

Presentado por: Julian Felipe Latorre

24 de julio de 2024

Contenido

Desarrollo Backend con Node.js y Express.js

- Node.js: Entorno de ejecución para JavaScript
- Express.js: Framework web para Node.js
- Importancia de los métodos HTTP en APIs RESTful

GET

- Usado para recuperar información
- Idempotente y seguro
- No modifica el estado del servidor
- Altamente cacheable

POST

- Usado para crear nuevos recursos
- No idempotente
- Permite enviar datos complejos en el cuerpo de la solicitud
- Consideraciones de seguridad importantes

PUT

- Usado para actualizar recursos existentes
- Idempotente
- Reemplaza completamente el recurso
- Requiere enviar todos los campos del recurso

PATCH

- Usado para actualizaciones parciales de recursos
- No necesariamente idempotente
- Más eficiente que PUT para actualizaciones pequeñas
- Requiere manejo cuidadoso para mantener la integridad de los datos

DELETE

- Usado para eliminar recursos
- Idempotente
- Consideraciones importantes de seguridad y auditoría
- Posibilidad de implementar "soft delete"

Implementación en Express.js

- Uso de app.get(), app.post(), etc.
- Manejo de rutas y controladores
- Middleware para procesamiento de datos
- Consideraciones de seguridad y rendimiento

Consideraciones Generales

- Seguridad: autenticación, autorización, validación de entrada
- Rendimiento: optimización de consultas, caching
- Diseño de API: consistencia, versionado, manejo de errores
- Cumplimiento normativo: GDPR, auditoría

Conclusión

- Importancia crítica de los métodos HTTP en el desarrollo backend
- Node.js y Express.js como plataforma potente para APIs RESTful
- Necesidad de comprender y aplicar correctamente cada método
- Continua relevancia en el desarrollo web moderno