**Node Js**

**Se usara nodejs y express**

¿Que es una API rest?

Se comunica con distintos clientes

La nube se tiene que conectar con una base de datos

Se encuentra en medio de los clientes y la base de datos (api rest)



Verbos

Get / users trae usuarios en un arreglo

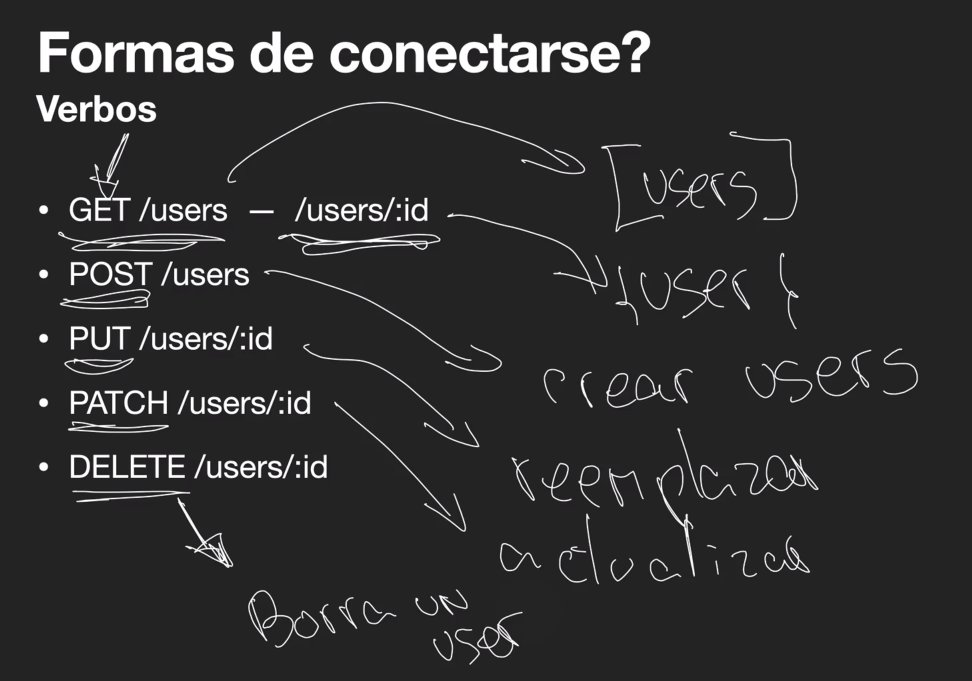
/users:id trae un usuario en especifico(objeto)

Post /users crea usuarios

Put /users/:id reemplaza

Patch /users/:id actualiza

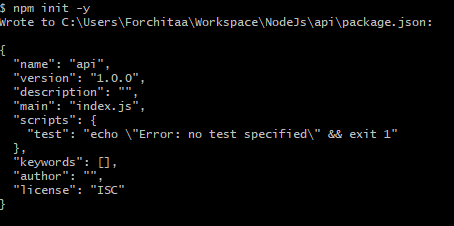
Delete/users/:id borra usuarios



Creamos la api comando

npm init -y

Instala todas las configuraciones por defecto



Instalamos express

Texto

Descripción generada automáticamente



 guarda un historial de las dependencias que se instalaron

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Se crea un nuevo archivo api.js y dentro se escribe



*// Creando Api*

*//asignacion de framework*

*//require= importar dependencia*

*//'express' es el nombre de la dependecia*

const express=require('express');

*//incializar*

const app= express();

*//puerto donde va a funcionar la aplicacion*

const port=3000;

*//ruta del navegador*

*//localhost:3000/*

app.get('/',(*req/\*request=peticion cliente\*/* ,*res/\*response= enviar cosas a user\*/*)=>{

*//envia respuesta al usuario*

*//200 es ok envia datos*

*//send sirve para poner el dato qur se dese enviar*

*res*.status(200).send('hola')

})

*//ejecutar la aplicacion*

*//port= es el puerto que usa la aplicacion*

*//funcion a ser llevada a cabo cuando se ejecute la aplicacion exitosamente*

app.listen(port,()=>{

    console.log('Iniciando ')

})

Luego en la terminal se escribe **node api.js**

****

En la web se escribe [**http://localhost:3000/**](http://localhost:3000/)

Imagen que contiene Texto

Descripción generada automáticamente

Endpoint

Es una ruta para llegar por petición por explorador web o app movil

Nota

Para detener un servicio en ejecución en la terminal se usa

**Ctrl+c**

**Endpoints**

*//primer endpoint*

*//acceso desde el explorador*

app.get('/',(*req/\*request=peticion cliente\*/* ,*res/\*response= enviar cosas a user\*/*)=>{

*//envia respuesta al usuario*

*//200 es ok envia datos*

*//send sirve para poner el dato que se desee enviar*

*res*.status(200).send('hola')

})

**En postman**

**Pantalla de computadora con letras

Descripción generada automáticamente con confianza media**

*//segundo endpoint*

*//no se puede acceder desde el explorador web*

*//se usara postman*

app.post('/',(*req*,*res*)=>{

*//status setea el estado que va a tener la peticion y luego se puede usar send*

*//201 ok creado no es necesario enviar datos al usuarios se podria enviar el id*

*res*.status(201).send('creando chanchito')

})

**En postman**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

*//tercer endpoint*

*//en la web el :id es un dato variable  /1 o /dfsdff*

app.put('/:id', (*req*,*res*)=>{

*//sendStatus solo envia el estado*

*//204 ok pero explisitamente no envia nada*

*res*.sendStatus(204)

})

**En postman**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente **

*//cuarto endpoint*

app.patch('/:id', (*req*,*res*)=>{

*res*.sendStatus(204)

})

**En postman**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente **

*//quinto endpoint*

app.delete('/:id',(*req*,*res*) =>{

*res*.sendStatus(204)

})

**En postman**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente **

*//sexto endpoint*

app.get('/:id',(*req*,*res*)=>{

*//req.params=trae el valor que se encuentra en :id*

    console.log(*req*.params)

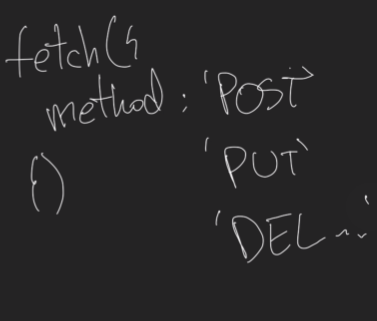
*res*.status(200).send(*req*.params)

})

**Pantalla de un video juego

Descripción generada automáticamente con confianza media**

**Fectch** se sabrá a que endpoint se esta accediendo

****

**Captura de peticiones**

//rutas no definidas

app.get('\*',(req,res) =>{

    res.status(404).send('Esta página no existe')

})

Limpieza de código **refactorización**

**Modulo personalizado**

** //** se crea un nuevo archivo

*//gestion de usuarios*

const User={

*//listar usuarios*

*//get*

    list:(*req*,*res*)=>{

*res*.status(200).send('lista de usuarios')

    },

*//create*

*//post*

    create:(*req*,*res*)=>{

*res*.status(201).send('creando usuario')

    },

*//actualizar*

*//put y patch*

    update:(*req*,*res*)=>{

*res*.status(204).send('actualizado usuario')

    },

*//eliminar*

*//delete*

    destroy:(*req*,*res*)=>{

*res*.status(204).send('eliminado usuario')

    }

get:(*req*,*res*)=>{

*res*.status(200).send('un usuario')

    }

}

*//exportar modulo personalizado*

*// User = lo que se va responder*

module.exports=User;

en api.js

*//importar modulo personalizado*

*//ubicacion de modulo*

const user= require('./user.controller')

ser reemplaza

*//primer endpoint*

*//acceso desde el explorador*

app.get('/',user.list)

Contolde rutas que no se encuentran

*//rutas no encontradasget*

    notfoundget:(*req*,*res*)=>{

*res*.status(404).send('Pagina no existe')

    },

    notfoundpost:(*req*,*res*)=>{

*res*.status(404).send('Página no encontrada')

    }

*//control de paginas que no existen get*

*//toma todas las rutas que no se han definido hasta este punto*

app.get('\*',user.notfoundget)

*//control de paginas que no existen get*

*//toma todas las rutas que no se han definido hasta este punto*

app.post('\*',user.notfoundpost)

Captura de pantalla de un celular

Descripción generada automáticamente