

Alias de rutas en Node.js + TypeScript + Express

1. Estructura base del proyecto

- src/
 - controllers/
 - routes/
 - utils/
 - index.ts
 - dist/
 - .env
 - tsconfig.json
 - package.json
-

2. Instalar dependencias

Producción

- `npm install express`
- `npm install module-alias`

Desarrollo

- `npm install --save-dev typescript ts-node tsconfig-paths`
 - `npm install --save-dev @types/node @types/express`
-

3. tsconfig.json (configurar paths)

```
json { "compilerOptions": { "target": "ES2020", "module": "ES2020", "moduleResolution": "node", "baseUrl": "./src", "paths": { "@controllers/*": ["controllers/*"], "@routes/*": ["routes/*"], "@utils/*": ["utils/*"] }, "rootDir": "./src", "outDir": "./dist", "esModuleInterop": true, "strict": true, "skipLibCheck": true } }
```

4. Crear alias de ejemplo

Archivo: src/utils/logger.ts

```
ts export function log(msg: string) { console.log('[LOG]', msg); }
```

5. Usar alias en src/index.ts

```
```ts import express from 'express'; import dotenv from 'dotenv'; import {
log } from '@utils/logger';

dotenv.config();

const app = express(); const PORT = process.env.PORT || 3000;

app.get('/', (_, res) => res.send('Hola desde alias'));

app.listen(PORT, () => { log(Servidor en puerto ${PORT}); }); ```
```

---

## 6. Ejecutar en desarrollo

### Instalar tsconfig-paths

- `npm install --save-dev tsconfig-paths`

### Script en package.json

```
json "scripts": { "dev": "ts-node -r tsconfig-paths/register src/
index.ts" }
```

### Ejecutar

- `npm run dev`
- 

## 7. Compilar para producción

### Compilar TypeScript

- `npx tsc`

Genera carpeta dist/ con los archivos JS

---

## 8. Configurar alias en producción

### package.json

```
json "_moduleAliases": { "@utils": "dist/utils", "@routes":
"dist/routes", "@controllers": "dist/controllers" }, "type":
"module"
```

## **dist/index.js (insertar al inicio)**

```
js import 'module-alias/register';
```

---

## **9. Variables de entorno**

### **.env**

```
PORT=3000 NODE_ENV=production
```

### **Uso en código (para servir estáticos en prod)**

```
ts if (process.env.NODE_ENV === 'production') {
 app.use(express.static('dist/public')); }
}
```

---

## **10. Ejecutar en producción**

### **Scripts en package.json**

```
json "scripts": { "build": "tsc", "start": "node dist/index.js" }
```

### **Comandos**

- npm run build
  - npm start
- 

## **Resultado Final**

- Uso de alias (`@utils`, `@routes`, `@controllers`)
- Compatible con desarrollo y producción
- Lectura de `.env`
- Código más limpio y organizado