



# Project Dependencies

## Documentation

Package	Purpose (Briefly)	Detailed Usage in Project	Documentation
@types/bcrypt	TypeScript عشان bcrypt يفهم أنواع (تشفير الباسورد)	تشغيل آمن لـ hash/compare الباسورد بدون أخطاء في الـ types	<a href="#">Link</a>
@types/cors	cors أنواع لـ middleware (اللي frontend بيسمح للـ يطلب من الـ backend)	في cors() استخدام Express بـ كامل type-safety	<a href="#">Link</a>
@types/dotenv	تحميل أنواع لـ dotenv (env variables) الـ	process.env قراءة types بأمان وفهم الـ	<a href="#">Link</a>
@types/express	أهم حاجة → أنواع كله Express (Request, Response, Router, etc.)	و controllers كتابة routes بـ type و IntelliSense قوي checking	<a href="#">Link</a>
@types/express-rate-limit	أنواع لـ express-rate-limit (spam حماية من الـ)	بـ rate limiting استخدام صحيحة types	<a href="#">Link</a>
@types/helmet	helmet أنواع لـ security (إضافة headers للحماية)	بأمان helmet() تطبيق في Express	<a href="#">Link</a>

@types/jsonwebtoken	أنواع jsonwebtoken (JWT إنشاء وتأكيد الـ tokens)	sign/verify tokens مهم جدًا type-safety (auth في الـ)	<a href="#">Link</a>
@types/multer	رفع multer أنواع (الملفات / الصور)	و req.file التعامل مع req.files الـ بأمان في الـ upload routes	<a href="#">Link</a>
@types/node	أساسي جدًا → أنواع Node.js نفسه (process, path, fs, etc.)	استخدام أي حاجة من Node.js بدون أخطاء types	<a href="#">Link</a>
@types/nodemailer	nodemailer أنواع (إرسال إيميلات)	(reset إرسال إيميلات password, confirm email) types واضحة	<a href="#">Link</a>
@types/uuid	توليد uuid أنواع (عشوائية فريدة IDs)	أو uuid.v4() استخدام type-safety غيره بـ	<a href="#">Link</a>
concurrently	أداة تشغل أكثر من command في نفس الوقت	بـ dev server تشغيل الـ hot-reload TS + Node	<a href="#">Link</a>

## Quick Commands:

- To Install all DevDependencies:

Bash

```
npm install -D @types/bcrypt @types/cors @types/dotenv @types/express
@types/express-rate-limit @types/helmet @types/jsonwebtoken @types/multer
@types/node @types/nodemailer @types/uuid concurrently
```

### ملخص سريع ومهم:

- كل الحاجات اللي بتبدأ بـ @types/ دي مش بتتنفذ في السيرفر، دي بس عشان TypeScript يفهم ويساعدك وأنت بتكتب الكود (IntelliSense، منع الأخطاء، اقتراحات أوتوماتيك).
- بدونها → هتظهرلك أخطاء كتير في VS Code زي "Property 'body' does not exist on type 'Request'" أو "Cannot find name 'JwtPayload'".
- concurrently هي الوحيدة اللي فعلاً بتتنفذ (في الـ script الـ dev).

## Core Project Dependencies (Production)

Package	Why we installed it?	Main Purpose in Project	Documentation
@aws-sdk/client-s3	الرسمي AWS S3 عميل عشان نرفع ونحذف — (v3) S3 ونقرأ ملفات من	الأساسية S3 كل عمليات upload, get, delete, list	<a href="#">Link</a>
@aws-sdk/lib-storage	لرفع AWS مكتبة مساعدة من الملفات الكبيرة بسهولة (Multipart)	رفع فيديوهات/صور كبيرة بدون مشاكل في الذاكرة (Memory)	<a href="#">Link</a>
@aws-sdk/s3-request-presigner	pre-signed URLs توليد رفع/تحميل مباشر من الـ frontend)	إنشاء روابط مؤقتة آمنة للرفع والتحميل بدون مرور على السيرفر	<a href="#">Link</a>
bcrypt	تشفير كلمات السر بطريقة آمنة (Hashing + Salt) جداً	تشفير ومقارنة الباسورد signup / login في الـ	<a href="#">Link</a>
cors	يعرف frontend عشان الـ بدون يكلم backend الـ CORS مشاكل	Permissions تفعيل الـ لـ Express اللازمة في API	<a href="#">Link</a>
dotenv	قراءة المتغيرات الحساسة من بسهولة .env ملف	و AWS keys تحميل الـ والـ JWT DB URI الـ Secrets	<a href="#">Link</a>
express	Web Server إطار عمل الـ الأساسي لمشروعنا	كله: الـ API بناء الـ والـ Routes Middlewares	<a href="#">Link</a>
express-rate-limit	Spam من الـ API حماية الـ (Brute) والطلبات الكثير	تحديد عدد الطلبات في IP المسموحة لكل	<a href="#">Link</a>

	Force)	وقت معين	
<b>helmet</b>	Security إضافة Headers تلقائي لحماية السيرفر	حماية السيرفر من XSS هجمات شائعة زي Clickjacking و	<a href="#">Link</a>
<b>jsonwebtoken</b>	JWT إنشاء وتأكيد الـ tokens للـ Authentication	و الـ Sign عمل الـ Verify للـ Access/Refresh tokens	<a href="#">Link</a>
<b>mongoose</b>	اللي بيخلينا نكلم الـ ODM عن طريق الـ MongoDB Objects	للـ Schemas عمل الـ User والـ Posts وتفاعلات الداتا بيز	<a href="#">Link</a>
<b>multer</b>	متخصص في الـ Middleware (Multipart استلام الملفات Data)	req.file التعامل مع الـ الخاصة Routes في الـ برفع الصور	<a href="#">Link</a>
<b>nodemailer</b>	إرسال الإيميلات الرسمية من السيرفر	إرسال إيميلات الـ Reset Password و Confirm Email الـ	<a href="#">Link</a>
<b>uuid</b>	عشوائية فريدة جداً IDs توليد ومستحيل تتكرر	S3 تسمية الملفات في بأسماء فريدة لضمان عدم التداخل	<a href="#">Link</a>
<b>zod</b>	قوية جداً Validation مكتبة TypeScript بتدعم	التأكد من صحة الـ Body باعتها الـ المستخدم في الـ Request	<a href="#">Link</a>

## Quick Command to install them:

Bash

```
npm install @aws-sdk/client-s3 @aws-sdk/lib-storage @aws-sdk/s3-request-presigner bcrypt
cors dotenv express express-rate-limit helmet jsonwebtoken mongoose multer nodemailer
uuid zod
```