

# HTTP Status Codes

## إيه هي ال Status Codes؟

لو بتشتغل في تطوير الويب أو APIs، أكيد سمعت عن ال Status Codes. دي رموز رقمية بيبعتها الخادم (Server) للعميل (Client) زي المتصفح أو التطبيق، عشان يقوله إيه اللي حصل لطلبك. مثلاً، لو الصفحة ظهرت كويس، هتلاقي كود 200، أما لو في خطأ، ممكن تشوف 404. الرموز دي جزء أساسي من بروتوكول HTTP، وبيتم تصنيفها حسب أول رقم فيها (زي 2xx للنجاح أو 4xx لأخطاء العميل). فهمها مهم عشان تعرف تشخص المشاكل وتحسن تجربة المستخدم.

## ليه بنهتم بال Status Codes؟

الرموز دي مش بس أرقام عشوائية؛ هي سر التواصل بين الخادم والعميل. لو مش فاهمها، هتتخبط في تصحيح الأخطاء، أو هتخلي APIs اللي بتبنيها غير موثوقة. كمان، محركات البحث زي Google بتعتمد عليها عشان تحدد إزاي الصفحات صحيحة أو لا، وده بيأثر على ال SEO.

## أنواع ال Status Codes الأساسية:

الرموز مقسمة لـ 5 فئات رئيسية، كل فئة بتبدأ برقم معين. خلينا نشوفها بالتفصيل:

### 1xx: Informational ♦

- دي رموز مؤقتة، بتقول إن الطلب وصل والسيرفر بيستمر في المعالجة.
- مش شائعة كتير، بس مهمة في حالات زي الاتصالات الطويلة.
- **مثال:** 100 Continue - يعني "تابع إرسال البيانات، كل حاجة تمام".

### 2xx: Success ♦

- أحلى فئة! معناها الطلب نجح والخادم رد بنجاح.

- مثال: 200 OK - الطلب نجح، والرد جاهز (زي لما تفتح صفحة ويظهر المحتوى).
- مثال: 201 Created - استخدمها لما تخلق مورد جديد، زي إضافة مستخدم جديد في API.

### 3xx: Redirection ♦

- الخادم يقول "الطلب تمام، بس روح لمكان تاني".
- مهمة للـ SEO ولتحسين الروابط.
- مثال: 301 Moved Permanently - الصفحة انتقلت لمكان جديد بشكل دائم (مفيد للتحويل من HTTP لـ HTTPS).
- مثال: 302 Found - إعادة توجيه مؤقت، زي لما تسجل دخول ويحولك للصفحة الرئيسية.

### 4xx: Client Error ♦

- المشكلة من ناحية العميل، زي طلب غلط أو صلاحيات ناقصة.
- مثال: 400 Bad Request - الطلب مش مفهوم أو فيه خطأ في البيانات (زي إرسال فورم ناقص).
- أشهر واحد: 404 Not Found - الصفحة مش موجودة (كلنا شفناها!).
- مثال كمان: 401 Unauthorized - تحتاج تسجل دخول أو صلاحيات أعلى.

### 5xx: Server Error ♦

- المشكلة في السيرفر نفسه، مش في الطلب.
- مثال: 500 Internal Server Error - خطأ عام في السيرفر (زي مشكلة في الكود أو قاعدة البيانات).
- مثال كمان: 503 Service Unavailable - السيرفر مشغول أو في صيانة، جرب بعد شوية.

مثال توضيحي:

تخيل عايز تفتح موقع مثل `example.com/about`:

- لو كل حاجة تمام، السيرفر هيرد بـ 200 OK، وتشوف الصفحة.
- لو الصفحة انتقلت لـ `example.com/info`، هيرد بـ 301 Moved Permanently، والمتصفح هيحولك تلقائيًا.
- لو كتبت الرابط غلط (مثل `example.com/abot`)، هتشوف 404 Not Found.
- لو السيرفر في مشكلة، زي قاعدة البيانات سقطت، هتلاقي 500 Internal Server Error.

في API، لو بترسل POST لطلب إنشاء حساب جديد:

- نجاح: 201 Created مع البيانات الجديدة.
- خطأ: 400 Bad Request لو الإيميل مش صالح.

### طيب الناس بتتلخبط في إيه؟

♦ **اختيار كود غلط:** ناس كتير بتستخدم 200 لكل حاجة، حتى لو في خطأ، وده بيخلي الـ API مش واضح.

♦ **تجاهل الـ 3xx:** لو مش عامل إعادة توجيه صح، هتخسر زوار أو تؤثر على الـ SEO.

♦ **التعامل مع 5xx:** لو السيرفر بيرد دائمًا بـ 500، مش هتعرف تشخص المشكلة سريعًا.

♦ **NULL أو قيم فارغة:** في APIs، لو رديت بـ 200 بس الرد فارغ، أحسن تستخدم 204 No Content.

⚙️ **Tips مهمة للتعامل مع Status Codes:**

✓ **دائمًا حدد الكود الصحيح في ردودك:** استخدم مكتبات زي Express.js أو ASP.NET عشان تساعدك في ذلك.

✓ **أضف رسائل توضيحية:** مثش بس الكود، أرسل معاه JSON زي {"error": "Page not found"} عشان المستخدم يفهم.

✓ **راقب اللوج:** استخدم أدوات زي ELK Stack عشان تسجل الرموز وتشخص الأخطاء.

✓ **اختبر ال APIs:** استخدم Postman أو Insomnia عشان تتحقق إن الكودات بتترد صح.

✓ **تعلم الرموز الشائعة:** ركز على 200, 201, 400, 401, 403, 404, 500 - دول 80% من اللي هتقابله.