

ما هو ال API؟

ال API هو اختصار لـ "Application Programming Interface"، أي واجهة برمجة التطبيقات، وهو وسيلة تُستخدم لتبادل البيانات بين تطبيقك والخوادم (Servers).

كيف يعمل ال API؟

يمكن تشبيهه بالمطعم:

المستخدم (أنت): مثل الزبون في المطعم.

ال API: مثل الجرسون (Waiter)، ينقل طلبك إلى المطبخ ويعيد لك الطعام.

الخادم (Server): مثل الطباخ (الشيف)، يستقبل الطلب ويجهزه ويرسله للجرسون

(Waiter) ليقدّمه لك.

ما هو REST API؟

هو نوع من أنواع ال APIs يستخدم بروتوكول HTTP لإرسال واستقبال البيانات. REST

يعتمد على عناوين URLs وطرق HTTP.

عناصر REST API:

العميل (Client): الجهاز أو التطبيق الذي يرسل الطلب (مثل تطبيق Flutter).

طرق الطلب (HTTP Methods):

GET: جلب البيانات.

POST: لإرسال بيانات جديدة.

PUT: لتعديل بيانات موجودة.

DELETE: لحذف بيانات.

شكل ال REST API الأساسي (Basic Format)

يحتوي REST API على 3 مكونات رئيسية:

1- نقطة النهاية (URL Endpoint)

هو رابط (URL Link) يحدد المورد (Resource) الذي نريد الوصول اليه (مثل بيانات المستخدم).

مثال:

example.com/surveys ← تتفاعل مع الاستبيانات

example.com/surveys/123/responses ← تتفاعل مع الردود الخاصة بالاستبيان رقم

123.

2- ال HTTP Verb

هو ما يخبر الخادم (Server) بما نريد فعله بالبيانات، وهناك 5 HTTP Verb رئيسية:

1- GET لجلب البيانات (قراءة فقط).

2- POST لإرسال بيانات جديدة (إنشاء مورد جديد).

3- PUT لتحديث بيانات كاملة.

4- PATCH لتحديث جزء من البيانات فقط.

5- DELETE لحذف بيانات معينة.

مثال:

POST example.com/surveys/123/responses

هذا يعني أننا نريد إرسال رد جديد للاستبيان رقم 123.

3- محتوى الطلب (Body Message)

هي بيانات نرسلها مع الطلب، خاصة مع POST, PUT, أو PATCH، وغالباً تكون بصيغة

.JSON

مثال على ال body:

{

"survey_id": 123,

"nps_score": 9,

"message": "Love the service",

"respondent_id": 42

}

HTTP Response Status Codes

عند إرسال طلب من تطبيقك (Client) إلى الخادم (Server) باستخدام API، يقوم الخادم بالرد برمز رقمي يُسمى "HTTP Status Code"، وهذا الرمز يُخبرك بنتيجة الطلب: هل نجح؟ فشل؟ خطأ من الخادم؟ خطأ من العميل؟

أهم الأكواد في كل فئة:

✓ فئة النجاح (2xx):

200 OK: الطلب تم بنجاح. النتيجة تختلف حسب نوع الطلب (GET, POST, إلخ).

201 Created: تم الانشاء بنجاح (عادة بعد POST).

202 Accepted: تم استلام الطلب لكن لم يُنفذ بعد (النتيجة غير مضمونة).

✗ فئة الأخطاء من العميل (4xx):

400 Bad Request: الطلب غير صحيح (صيغته خطأ).

401 Unauthorized: المستخدم غير مصرح له (بحاجة لتسجيل الدخول).

403 Forbidden: غير مسموح لك بالوصول.

404 Not Found: لم يتم العثور على العنوان المطلوب.

🔥 فئة الأخطاء من السيرفر (5xx):

500 Internal Server Error: خطأ عام في السيرفر.

501 Not Implemented: السيرفر لا يدعم هذا النوع من الطلبات.

502 Bad Gateway: السيرفر حصل على رد غير صالح من سيرفر آخر.

ما هو Pagination؟

هو طريقة لتقسيم المحتوى الكبير إلى صفحات صغيرة ومنفصلة، بدلاً من عرض كل البيانات مرة واحدة.

مثال: عندما تتصفح متجر إلكتروني وتظهر لك 10 منتجات في كل صفحة بدلاً من عرض 1000 منتج دفعة واحدة.

لماذا يعتبر Pagination مهماً؟

- 1- تحسين تجربة المستخدم (UX):
- 2- عرض عدد محدود من العناصر في كل مرة يجعل التصفح أسهل وأسرع.
- 3- تحسين الأداء: حيث أنه يتم تحميل عدد قليل من العناصر يقلل الضغط على التطبيق ويزيد سرعته.
- 4- تحسين نتائج محركات البحث (SEO): تنظيم المحتوى على صفحات متعددة يسهل على محركات البحث أرشفته وفهمه.

أفضل الممارسات في Pagination 

1- تحديد عدد مناسب من العناصر لكل صفحة (مثلاً 10 أو 20).

2- عرض أزرار تنقل واضحة (التالي / السابق).

3- إظهار عدد الصفحات أو أرقامها بسهولة الانتقال.

4- تحميل تلقائي عند الوصول للنهاية (Infinite Scroll)