

Base URL: <https://core-academy-api-production.up.railway.app/>

الصحة والتوجيه العام (1)

فحص الصحة

GET /health
200 OK

json
CopyEdit

```
{ "status": "OK", "environment": "production" }
```

:المصدر

(خريطة المسارات (أعلى المستوى

- /public, /auth, /users, /admin, /courses, /payments, /enrollments, /instructors, /notifications, /content.
-

المصادقة والحسابات (2) /auth

مع تحقق المدخلات JSON body تستخدم معظم نقاط النهاية التالية

(تسجيل حساب (باليبريد

POST /auth/register
Body

json
CopyEdit

```
{ "email": "user@example.com", "password": "Secret12", "username": "hanan" }
```

201 Created

json
CopyEdit

```
{ "message": "OTP sent to email", "userId": "<uuid>" }
```

- وإلا تُعاد رسالة خطأ "Email already registered".
في حال كان البريد مسجلًا وغير مُفعّل: يعاد إرسال `resentOtp: true`.

تفعيل البريد (OTP)

POST /auth/verify-otp
Body

json
CopyEdit

```
{ "userId": "<uuid>", "otpCode": "123456" }
```

200 OK

json
CopyEdit

```
{ "message": "Email verified successfully" }
```

تسجيل الدخول (Email/Password)

POST /auth/login
Body

json
CopyEdit

```
{ "email": "user@example.com", "password": "Secret12" }
```

200 OK

json
CopyEdit

```
{ "token": "<ACCESS_TOKEN>", "role": "student" }
```

{ **error**:
"unverified", **userId**, **resentOtp: true** } إذا كان الحساب غير مُفعل، يتم إرسال
كاستثناء. قد يعالج في الـ controller Refresh Token ـ HttpOnly Cookie للاستخدام أيضًا.

(أنا (بيانات المستخدم

GET /auth/me (محمية)
200 OK

json
CopyEdit
{ "user": {
"id": "<uuid>", "email": "...", "username": "...", "role": "student", "isVerified": true } }

تسجيل الخروج

POST /auth/logout (محمية)

- الخاص بالتجدد، وتحتاج **Cookie 204** لمسح سجل الجلسات ويزال **No Content**.

تجدد الوصول (Refresh)

POST /auth/refresh
200 OK

json
CopyEdit
{ "accessToken": "<NEW_ACCESS_TOKEN>", "user": { "id": "<uuid>", "role": "student" } }

401 عند فشل التحقق من Refresh.

التسجيل بالهاتف

POST /auth/register-phone
Body

json
CopyEdit

```
{ "phoneNumber": "+2189XXXXXXX", "password": "Secret12" }
```

201 Created

json

CopyEdit

```
{ "message": "OTP sent to phone", "userId": "<uuid>" }
```

تسجيل دخول المشرف (Admin)

POST /auth/admin/login

Body

json

CopyEdit

```
{ "username": "admin", "password": "Secret12" }
```

200 OK

json

CopyEdit

```
{ "success": true, "token": "<ACCESS_TOKEN>", "role": "admin",  
"user": { "id": "<uuid>", "username": "admin", "name": "..." } }
```

Google OAuth

- **GET /auth/google**

عند النجاح أو 401 عند الفشل → **GET /auth/google/callback**: عودة { token, user }

(بدون مصادقة) /public المسارات العامة (3)

المجموعات

GET /public/groups

200 OK

```
json  
[ { "id":<uuid>, "name":"Group A", "courseCount": 5 }, ... ]
```

(قائمة الدورات (مع عوامل تصفية

GET /public/courses?groupId=<uuid>&categorySlug=...&q=...
200 OK

```
json  
[ { "id":<uuid>, "title":"...", "price": 99.0, "thumbnail":"...",  
    "description":"..." }, ... ]
```

النتائج مرتبة تنازلياً بتاريخ. **groupId**، **categorySlug**، (بحث بالعنوان) **q**: يتم دعم عوامل التصفية
الإنشاء.

تفاصيل دورة

GET /public/courses/:id
200 OK

json

```
{  
  "id":<uuid>, "title":"...", "price":99.0, "description":"...",  
  "categories": [ {"id": "...", "name": "..."} ],  
  "rating": 4.6  
}
```

إن لم توجد الدورة **404**.

بنوك الأسئلة (QBank)

- **GET** /public/qbanks → قائمة الاسم والوصف والسعر →.
- **GET** /public/qbanks/:id → تفاصيل بنك أسئلة، 404 إن لم يوجد →.

4 الدورات /courses

عرض الدورات

- **GET /courses** جميع الدورات مع بيانات المدرسين →.
- **GET /courses/:id** دورة واحدة + الدروس →.

(دروس الدورة (محمية

- **GET /courses/:id/lessons** طالب مسجل في الدورة أو مدرسها)) . يعيد قائمة الدروس بالترتيب ومواردها، مع فرض تحقق الانضمام/الصلاحيات. **403** إذا لم يكن الطالب منضمًا .

(إنشاء/تعديل دورة (محمية

- **POST /courses** (admin instructor) **Body**

json

```
{ "title": "...", "description": "...", "price": 49.99 }
```

كائن الدورة → **201 Created**

- **PUT /courses/:id** (admin أو المدرس المكلف بها فقط) الكائن بعد التحديث. **403** إذا لم يكن المرسل هو المكلف. **404** إذا لم توجد الدورة → **200 OK**

إنشاء درس داخل دورة

- **POST /courses/:id/lessons** (admin **Body**)

json
CopyEdit

```
{ "title": "Intro", "videoUrl": "https://..." }
```

كائن الدرس → **201 Created**

5) المحتوى (فيديوهات الدروس **/content**

(رفع فيديو درس (مدرس/مشرف

POST /content/upload حقل video ملف واحد (**lessonId**)
Form-Data

- **lessonId** = UUID
- **video** = 500 MB ملء فيديو (حتى **MIME** تبدأ بـ **video/**)
201 Created في S3 يحفظ مفتاح → **lesson.videoUrl**.

(الحصول على رابط تشغيل فيديو (محمية

GET /content/lessons/:lessonId
200 OK

json

```
{ "...lesson fields...", "signedUrl": "https://s3...." } (مؤقت)
```

لمدة افتراضية ساعة. الوصول مقيد بالطالب المنضم أو المدرس المكلف S3 من **Signed URL** يتم توليد.

6) طالب (الالتحاق بالدورات **/enrollments**)

جميع النقاط التالية محمية وتنطلب دور **student**.

- **POST /enrollments/:courseId** يخصم السعر من محفظة الطالب ويُؤرِّج الحصص على مُدرسي → **201** مع الدورة، ثم يُسجَّل الالتحاق. **message: "Enrollment successful"**, **courseId** عند رصيد غير كافٍ / حالات تحقق **402/400**.

- **GET /enrollments** قائمة دورات الطالب الملتحق بها →

- **GET** /enrollments/:courseId/lessons (دورس الدورة مع الموارد (يتحقق من الاشتراك →
تعليم الدرس كمكتمل →
Idempotent). **204 No Content.**
-

(مدرس/مشرف) /instructors (7)

جميع النقاط التالية محمية.

- **GET** /instructors/dashboard (إجماليات المبيعات والأرباح وعدد الدورات →
دورات المدرس →
جميع المبيعات والأرباح لكل دورة →
• **GET** /instructors/courses (type: payout).
• **GET** /instructors/withdrawals (سجلات طلبات السحب →
طلب سحب رصيد. **201** عند النجاح →
ملف المدرس (عام أو محلي بحسب تطبيقك). **404** إن لم يوجد →
تحديث الاسم/السيرة/الصورة →
محفظة المدرس وحركاتها. **404** إذا لم توجد →
Admin) أو

8 (المدفوعات والمحفظة) /payments

محمية حسب الدور.

- **POST** /payments/vouchers (Admin)
Body

json
CopyEdit
{ "amount": 100 }

201 Created → كود قسيمة مثل { "code": "AB12CD", "credit": 100, ... }.

- **POST /payments/redeem** (Student)
Body

json

```
{ "code": "AB12CD" }
```

200 OK

json

```
{ "message": "Voucher redeemed", "balance": 250 }
```

يرفض القسائم غير الموجودة أو المستخدمة. **credit** يُحدث رصيد المحفظة ويسجل حركة.

- **GET /payments/wallet** (Student/Instructor)
رصيد المحفظة مع الحركات →
- **GET /payments/wallet/transactions** (Student/Instructor)
قائمة الحركات مرتبة تنازلياً →
- **POST /payments/wallet/redeem-voucher** (Student/Instructor)
استرداد قسيمة عبر →
مسار بديل.

9 الإشعارات /notifications (محمية)

- **GET /notifications** (الأحدث أولاً → كل إشعارات المستخدم).
 - **POST /notifications/:id/read** → تعلم إشعار كُوفرى.
 - **POST /notifications/read-all** → تعلم جميع إشعارات المستخدم كُوفرى.
(الخدمة تُنشئ إشعارات لأحداث مثل الالتحاق أو ربح المدرس).
-

10 المستخدمون /users (محمية)

- **GET** /users/:id ملف المستخدم (يتضمن → student/instructor).
• **PUT** /users/:id (تحديث الاسم، وسيرة الطالب/المدرب وصورته (إن قدمت →)).
 - **GET** /users/:id/progress/:courseId (Student) → تقدم المستخدم في دورة محددة (دروس → مكتملة).
 - **POST** /users/:id/progress/:lessonId (Student) وضع درس كمكتمل → (Idempotent).
-

11 بنك الأسئلة (للطالب) /qbank

بنوك الأسئلة تتطلب دور **student**.

- **GET** /qbank (بنوك الأسئلة المشترك فيها المستخدم →).
- **POST** /qbank/:qbankId/subscribe (الاشتراك (يخصم من المحفظة ويُضيف اشتراكاً). 201 عند → النجاح.
- **GET** /qbank/:qbankId/questions (في أسئلة بنك معين دون تسريب الحقل → isCorrect الخيارات. 403 إن لم