

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.

/auth المصادقة والحسابات (2)

مع تحقق المدخلات JSON body تستخدم معظم نقاط النهاية التالية

(تسجيل حساب (بالبريد

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>" }
```

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

(OTP) تفعيل البريد

POST /auth/verify-otp

Body

json

CopyEdit

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

200 OK

json

CopyEdit

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

(Email/Password) تسجيل الدخول

POST /auth/login

Body

json

CopyEdit

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

200 OK

json

CopyEdit

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

`{ error: جديد ويُعاد خطأ 401 مع بيانات OTP إذا كان الحساب غير مُفعّل، يتم إرسال كاستثناء). قد controller يعالج في الـ ("unverified", userId, resentOtp: true }`
`/auth/refresh` للاستخدام في `HttpOnly Cookie` ← `Refresh Token` يضبط الخادم 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)

- **No Content.** الخاص بالتجديد، وتُعاد **204 Cookie** يُمسح سجل الجلسات ويُزال

(Refresh) تجديد الوصول

`POST /auth/refresh`
200 OK

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

`Refresh.` عند فشل التحقق من **401**

التسجيل بالهاتف

`POST /auth/register-phone`
Body

json
CopyEdit

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

201 Created

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

(Admin) تسجيل دخول المشرف

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

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

(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 تنازليًا 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

(QBank) بنوك الأسئلة

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

/courses الدورات 4)

عرض الدورات

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

(دروس الدورة (محمية

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

(إنشاء/تعديل دورة (محمية

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

json

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

201 Created → كائن الدورة

- **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** (فقط admin)
Body

json

CopyEdit

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

/content (المحتوى (فيديوهات الدروس 5)

(رفع فيديو درس (مدرس/مشرف

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

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

(الحصول على رابط تشغيل فيديو (محمية

GET /content/lessons/:lessonId
200 OK

json

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

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

/enrollments (طالب) الالتحاق بالدورات 6)

student. جميع النقاط التالية محمية وتتطلب دور

- **POST /enrollments/:courseId** → يخضم السعر من محفظة الطالب ويُوزّع الحصة على مُدرّسي
{ **message: "Enrollment successful", courseId** } الدورة، ثم يُسجّل الالتحاق. **201** مع
عند رصيد غير كافٍ / حالات تحقق **402/400**.
- **GET /enrollments** → قائمة دورات الطالب الملتحق بها

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

/instructors /المدرّسون (7)

جميع النقاط التالية محمية

- **GET** `/instructors/dashboard` → إحصائيات المبيعات والأرباح وعدد الدورات.
 - **GET** `/instructors/courses` → دورات المدرّس.
 - **GET** `/instructors/sales` → جميع المبيعات والأرباح لكل دورة.
 - **GET** `/instructors/withdrawals` → سجلات طلبات السحب (type: payout).
 - **POST** `/instructors/withdrawals/request` → طلب سحب رصيد. **201** عند النجاح.
 - **GET** `/instructors/:id` → ملف المدرّب (عام أو محمي بحسب تطبيقك). **404** إن لم يوجد.
 - **PUT** `/instructors/:id` (Admin المالك أو) → تحديث الاسم/السيرة/الصورة.
 - **GET** `/instructors/:id/wallet` → محفظة المدرّس وحركاتها. **404** إذا لم توجد.
-

/payments المدفوعات والمحفظات (8)

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** يُحدّث رصيد المحفظة ويُسجّل حركة

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

9) الإشعارات /notifications (محمية)

- **GET** /notifications → (كل إشعارات المستخدم (الأحدث أولاً
- **POST** /notifications/:id/read → تعليم إشعار كُؤقِرئ
- **POST** /notifications/read-all → تعليم جميع إشعارات المستخدم كُؤقِرئت. (الخدمة تُنشئ إشعارات لأحداث مثل الالتحاق أو ربح المدرّس

10) **/users** (محمية) المستخدمين

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

11) **/qbank** (بنك الأسئلة) للطالب

student. تتطلب دور

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