

GET

/v2/academy/course/steps/view

Academy: View Course Steps

دنداندرگی مربی ار نآ (steps) لحرارم مامت هارمهه ب ی شزومآ هرود کی لماک تاعالطا تنیو پدنا نی دنکی مینادرگزاب و یناسرزوره ب ار (completed و current_step) ربراک تفرشی پ تیعضو نی نچمه تاعالطا و تسرا هدش انب هدش تفایرد هداد (step دروکر نی رخآ) هلحررم نی رخآ اساسا رب API نی قطنم (متسیس یلعف راتفر قبط) دوشیم جارختسا step نی رخآ نامه زآ ادمع زین هرود.

URL: `/v2/academy/course/steps/view`

Method: GET

Controller: AcademyController@viewStepsCourse

Middleware: authWithJwt

Auth: JWT Required

Query Parameters

Parameter	Type	Required	Description
course_id	integer	yes	دیهاوخی م هک یا هرود هسانش دینک هدهاشم ار نآ لحرارم
step_id	integer	no	کی طوق دوش لاسرا رگا دوشیم رتلیف صاخ هلحررم
current_step	integer	no	ه ب ربراک هک یدیج هلحررم ربارب رگا ؛ هدیسر نآ رادقم → دشاب total_steps شیزفا کی completed دبایم

Controller Logic

```
// 1. Fetch all steps of this course (optional filtering by step_id)
// 2. LEFT JOIN course + category info inside each step row
// 3. Load or create progress record in academy_course_completed
```

```

// 4. Update progress (completed / current_step)
// 5. Build steps[]
// 6. Build final course object based on the LAST step (intentional behavior)

$resultCourseSteps = DB::table('academy_course_steps')
    ->select([... course + step + category fields ...])
    ->where('academy_course_steps.course', $request->get('course_id'))
    ->where(function ($q) {
        if ($request->has('step_id')) $q->where('academy_course_steps.id', $request->get('step_id'))
    })
    ->leftJoin("academy_courses", ...)
    ->leftJoin("academy_categories", ...)
    ->orderBy('order')->orderBy('id')
    ->get();

$resultStep = DB::table('academy_course_completed')
    ->where('course', $request->get('course_id'))
    ->where('student', $request->get('operator')->id)
    ->first();

$courseCount = total number of steps in academy_course_steps;

// Progress Update Logic
if (!$resultStep) create new record with defaults;
else update based on current_step rules;

// Build steps[]
foreach (... each step ...) { ... }

// IMPORTANT: course info is taken from LAST step (as user confirmed)
$return = [
    "id" => $step->course_id,
    ...
    "steps" => $steps
];

return [...];

```

Step Object Structure

Field	Type	Description
id	integer	ه‌ل‌ح‌ر‌م ه‌س‌ا‌ن‌ش
title.fa	string	ه‌ل‌ح‌ر‌م ی‌س‌ر‌ا‌ف نا‌و‌ن‌ع
title.en	string	ه‌ل‌ح‌ر‌م ی‌س‌ی‌ل‌گ‌ن‌ا نا‌و‌ن‌ع

Field	Type	Description
subtitle.fa	string	یسراف ناونعریز
subtitle.en	string	یسیلگنا ناونعریز
content	string	هلحرم یاوتحم (HTML / Markdown)
order	integer	هلحرم شیامن بیترت
status	integer	للاعفریغ/لاعف تیعضو
created_at	datetime	تخاسخیرات
updated_at	datetime	یناسرزوره ب نیرخآ

Course Object Structure (Extracted From Last Step)

Field	Type	Description
id	integer	step نیرخآزا) هرود هسانش
slug	string	هرود گالسا
branch	integer false	ای هبعش هسانش false
price	integer "free"	هرود تمیق
necessary	array false	اهزاینشیپ
title.fa	string	هرود یسراف ناونع
title.en	string	هرود یسیلگنا ناونع
category.*	object	یدنبهتسد تاعالطا
progress.current_step	integer	یللاع هلحرم
progress.completed	integer	هدشلیمکت لهحارم دادعت
duration	integer	هرود نامز تدم
total_steps	integer	لهحارم لک دادعت
featured	integer	تسا هژیو هرود رگا 1
created_at	datetime	تخاسخیرات
updated_at	datetime	رییغت نیرخآ
steps	array	لهحارم مامت یهیارآ

Important Note

ربراک طسوت و دراد دوجو دک رد رات فر نی .دوشیم جارختسا **step** نی رخآ* زا هرود تاعالطا داد دهاوخ اطخ متسیس و هدش `step` undefined ،دشاب یلاخ steps رگا .تساهدش دیأت (اضریلع)

Example Response

Status: 200 OK

```
{
  "status": true,
  "time": 1730000000,
  "data": {
    "id": 21,
    "slug": "advanced-react",
    "branch": false,
    "price": "free",
    "necessary": false,
    "title": { "fa": "هتفرشیپ تکاا یر", "en": "Advanced React" },
    "category": {
      "id": 3,
      "flag": "frontend",
      "title": { "fa": "دناا تنارف", "en": "Frontend" }
    },
    "progress": { "current_step": 2, "completed": 1 },
    "total_steps": 12,
    "duration": 180,
    "featured": 1,
    "created_at": "2024-04-02 11:30:00",
    "updated_at": "2024-04-20 19:12:00",
    "status": 1,
    "steps": [
      {
        "id": 10,
        "title": { "fa": "یسانشآ", "en": "Introduction" },
        "subtitle": { "fa": "همدقم", "en": "Intro" },
        "content": "...",
        "order": 1,
        "status": 1,
        "created_at": "2024-04-02 11:30:00",
        "updated_at": "2024-04-02 11:30:00"
      },
      ...
    ]
  }
}
```

Flowchart

Validate JWT (operator)

↓

Fetch steps (JOIN course + category)

↓

Load or create progress record

↓

Update current_step / completed

↓

Build steps[]

↓

Extract course info from LAST step

↓

Return final structured course object

Revision #1

Created 2025-12-05 19:57:04 UTC by Alireza

Updated 2025-12-05 20:01:01 UTC by Alireza