

مقدمه

هدف از این فاز پروژه، آشنایی با مفاهیم سمت سرور با پیاده سازی API برای سایت فیلم، مشابه آنچه در قسمت‌های قبل مورد استفاده قرار داده بودید هست.

هیچ محدودیتی در انتخاب زبان، تکنولوژی و Framework برای این بخش وجود ندارد و استفاده از یکی از گزینه های زیر پیشنهاد می‌شود:

Language	Frameworks	Tutorial / Docs
PHP	Laravel - Lumen - Slim - [pure php]	https://laravel.com/docs https://lumen.laravel.com/docs
Node.js	Express - Adonis - Hapi - Loopback	https://expressjs.com http://yon.ir/sBYhN
Python	Flask - Django	http://flask.pocoo.org http://yon.ir/9WW9o
ASP.NET	MVC 5	https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started

نکته: به دانشجویان کلاس مهندسی اینترنت 2 (مدیری) توصیه می‌گردد با توجه به ضروری بودن پیاده‌سازی به صورت MVC و استفاده از ORM از ASP.NET استفاده نمایند در این صورت استفاده از Razor نیز برای این دسته از دانشجویان بلامانع خواهد بود.

نکات کلی در مورد پروژه

وقتی که وبسایتی API نسبتاً مفصلی دارد (مانند همین پروژه که APIهای متعددی برای لیست فیلمها، جزییات هر فیلم، کامنتها و ... دارد) پیاده سازی همه آنها در قالب یک فایل در سمت سرور امکان پذیر نیست. بنابراین باید از یک روش سیستماتیک و مدیریت پذیری استفاده کرد که APIها در قالب فایلهای مختلف پیاده سازی شوند. در روش دستی، روال رایج به این صورت است که یک (یا چند البته خیلی محدود) API در یک فایل پیاده سازی شده در یک مسیر مشخصی در سرور قرار گرفته و در سمت کلاینت بر اساس این مسیر فراخوانی انجام می گیرد. برای مثال برای گرفتن لیستی از آخرین فیلمها یک API در فایل به اسم recent.php پیاده سازی می شود که این فایل در سمت سرور در آدرس /movies/recent.php قرار می گیرد و برای فراخوانی آن باید در سمت کلاینت درخواست GET /movies/recent.php انجام گیرد. این فرایند که در آن یک درخواست (فراخوانی یک API) به یک فایل مسیریابی می شود را تعریف route می گویند. وقتی که از فریم ورکهای سمت سرور استفاده می شود، در عمل نیاز نیست این فرایند مسیریابی به صورت دستی انجام شود. بلکه همه این فریم ورکها امکاناتی را فراهم می کنند که شما می توانید مشخص کنید درخواست به یک route منجر به فراخوانی چه تابعی شود. برای مثال تعریف میکنیم که درخواست GET /movies/recent (توجه کنید که پسوندی وجود ندارد) معادل اجرای تابع get_recent_movies است (که میتواند در یک فایل با هر اسم دلخواهی باشد).

وظیفه اصلی شما در این پروژه، پیاده سازی routeهایی که است در ادامه تعریف می شود. برای هر route دو روش پیشنهاد شده که روش دوم، روش دستی بوده و در حالتی که از هیچ فریم ورکی استفاده نمی کنید مناسب تر است. در صورت نیاز می توانید از آدرس route دلخواه دیگری نیز استفاده نمایید. پیاده سازی route های مشخص شده در ادامه به غیر از بخش هویت سنجی به همراه فیلدهای مشخص شده ضروری هست.

در تعریف، پارامترهای هر route به شکل param: مشخص شده اند. برای مثال در صورتی که وب سرور شما بر روی پورت ۳۰۰۰ در حال اجرا باشد، درخواست های HTTP با متد GET به آدرس

<http://localhost:3000/movies/123> توسط روت `/movies/:id` پاسخ داده می شود. معمولا Framework ها مکانیزم های کمکی برای تعریف route را در اختیار می گذارند.

برای راحتی خود توصیه می شود که api بک اند خود را مشابه تمرین قبل ایجاد کنید که نیازی به تغییر در پروژه خود نداشته باشید. همچنین بهتر است فقط قسمت های مورد نیاز را در جواب برگردانید.

نکته : از آنجایی که ASP.NET MVC دارای موتور مسیریابی می باشد نیازی به پیاده سازی مجدد این بخش وجود ندارد و صرفا کدها و تنظیمات مورد نیاز خود را به یکی از روش بیان شده برای دسترسی به Controller و Action Method ها انجام دهید.

پایگاه داده و ساختار اطلاعات

برای نگهداری اطلاعات فیلم ها نیاز استفاده از یک پایگاه داده مناسب دارید. استفاده از پایگاه داده MySQL برای PHP و Microsoft Sql Server برای ASP.NET و MongoDB برای Node.js و Python پیشنهاد می شود. در صورت استفاده از پایگاه داده های SQL، بررسی امنیتی ساده در برابر حملات SQL Injection بررسی می شود. نکته : استفاده از ORM برای دانشجویان کلاس مهندسی اینترنت 2 (مدرس مدیری) اجباری است و افرادی که از ASP.NET MVC استفاده می کنند حتما از Entity Framework به صورت Code First یا Database First استفاده نمایند. به اینصورت نیازی به تست های Sql Injection نخواهد بود.

مدل Movie

- id int (_id for mongo)
- created_at: DateTime
- title: String - VARCHAR(255)
- original_title: String - VARCHAR(255)
- rate: Number [0-5]

- year: Number
- length: String - VARCHAR(255)
- language: String - VARCHAR(255)
- country: String - VARCHAR(255)
- description: String - TEXT
- director: String - VARCHAR(255)

مدل Comment

- id int (_id for mongo)
- movie_id: int (movie for mongo)
- created_at: DateTime
- author: String - VARCHAR(255)
- comment: String - TEXT
- [rate]: Number [0-5]

نکته: دانشجویان کلاس مهندسی اینترنت 2 (مدیری) تلاش کنند به دلایل امنیتی اشاره شده در کلاس درس از

uniqueidentifier در سمت بانک اطلاعاتی و GUID در سمت Application به عنوان کلید مدل به جای int

استفاده نمایند

ذخیره سازی عکس کاور فیلم‌ها

برای نگه داری کاور فیلم‌ها پیشنهاد می‌شود اسم فایل‌ها با کلید id فیلم یکسان بوده و در یک پوشه نگهداری شود. برای نمونه: (فریم ورک Laravel)

- public/posters/123456.jpg
- <http://localhost/posters/123456.jpg>

route های مد نظر در پروژه

GET /movies/recent/[:number]

GET /movies/recent.php[?number=]

کاربرد:

نمایش در اسلایدار صفحه نخست.

توضیحات:

چند فیلم اضافه شده اخیر به دیتابیس برحسب زمان. تعداد فیلم ها با توجه به پارامتر number قابل تعیین هست. (این عدد را به صورت پیش فرض ۱۰ قرار دهید)

پاسخ JSON باید شامل فیلد های زیر باشد:

- id
- title
- original_title
- year

GET /movies/:id/details

GET /movies/details.php?id=

کاربرد:

استفاده در صفحه نمایش جزئیات فیلم.

توضیحات:

شامل تمام فیلدهای مدل Movie مورد نظر.

امتیازی: در خروجی یک فیلد امتیاز کلی با توجه به میانگین امتیاز تمامی نظرات داده شده افزوده شود.

GET /movies/:id/comments

GET /movies/comments.php?id=

کاربرد:

دریافت لیستی از دیدگاه‌های قرارداده شده برای یک فیلم.

توضیحات:

شامل آرایه ای مرتب شده بر اساس زمان از تمامی مدل های Comment مرتبط با فیلم با کلید id:

POST /movies/:id/comments

POST /movies/comments.php?id=

کاربرد:

ارسال دیدگاه به همراه نام ارسال کننده و امتیاز برای یک فیلم.

توضیحات:

بدنه درخواست شامل فیلدهای author - comment و rate در صورت پیاده سازی بخش امتیازی است. بعد از ارسال درخواست یک نمونه از مدل Comment با توجه به فیلدهای داده شده در دیتابیس ذخیره می شود.

بدنه درخواست شامل فیلدهای زیر باید باشد:

- comment: String
- author: String
- rate: Number

GET /search?q=

GET /search.php?q=

کاربرد:

جست و جو بین فیلم ها بر اساس عنوان

توضیحات:

برای جست و جوی فیلم از پارامتر query به اسم q بر اساس عنوان (و یا توضیحات) فیلم. خروجی شامل آرایه ای از فیلدهای زیر می باشد:

- id
- title
- cover

POST /submit

POST /submit.php

کاربرد:

ارسال یک فیلم برای نمایش در وب سایت

توضیحات:

در این قسمت باید یک فیلم را ذخیره کنید برای این کار در ابتدا لازم است که یک صفحه با جزئیات حداقلی مورد

نیاز ایجاد کنید (لازم به ذکر است که کار اضافی بر روی GUI این قسمت نمره اضافه ای در بر ندارد)

اطلاعاتی که برای آپلود یک فیلم ضروری است عبارت است از:

- یک فیلد برای شناسه فیلم
- یک فیلد برای نام فیلم
- یک فیلد برای زمان فیلم
- یک فیلد برای سال انتشار
- یک فیلد برای کشورها
- یک فیلد برای توضیحات
- یک فیلد برای کارگردان
- یک فیلد برای نویسندگان
- یک فیلد برای ستارگان
- یک فیلد برای دسته ی فیلم
- یک انتخاب گر فایل برای آپلود عکس کاور فیلم

امتیازی: در صورت پشتیبانی از فیلم trailer به صورت کامل یک انتخابگر فایل برای ارسال trailer نیز اضافه نمایید.

بخش امتیازی: هویت سنجی و پروفایل کاربری

در صورت تمایل به پیاده سازی این بخش، به مستندات Framework مورد استفاده مراجعه کنید. معمولاً روش های استاندارد برای تعریف مدل User و فرآیند هویت سنجی وجود دارد.

ورود کاربر:

POST /login

POST /login.php

خروج کاربر:

GET /logout

GET /logout.php

دریافت اطلاعات پروفایل:

GET /user

GET /user.php

تغییر در اطلاعات پروفایل:

POST /user

POST /user.php

نکاتی در مورد تحویل پروژه



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

مهندسی اینترنت

فاز پایانی - پیاده سازی سمت سرور

- مهلت آپلود پروژه تا ساعت ۲۲ روز جمعه ۶ بهمن می باشد.
- تحویل آخرین بخش سمت فرانت اند به همراه بخش سمت سرور به صورت حضوری می باشد که در روز شنبه ۷ بهمن انجام می شود.
- در صورت هرگونه مشکل در مورد پیاده سازی در گروه تلگرام و یا از طریق ایمیل با تدریس یاران مربوطه مطرح فرمایید.

موفق باشید ☺