

سامانه ثبت پیشنهادات و انتقادات

این سؤال تنها با زبان‌های *Go* ، *Python* ، *PHP* و *JS (Node.js)* قابل حل است.

دیجی‌کالا قصد دارد برای بخش تحویل محصولات خود یک سامانه‌ی ساده‌ی ثبت پیشنهادات و انتقادات راه‌اندازی کند. از شما می‌خواهیم یک *API* برای این سامانه طراحی کنید.

جزئیات پروژه

پروژه‌ی اولیه را از این لینک دانلود کنید.

در این سؤال، یک *REST API* شامل *endpoint* های زیر باید پیاده‌سازی شود:

عنوان	آدرس
بودن سرویس <i>up</i> بررسی	GET /
ثبت‌نام	POST /signup
ورود به حساب کاربری	POST /login
درج پیشنهاد یا انتقاد	POST /suggestions
دریافت لیست پیشنهادات و انتقادات	GET /suggestions

در این *API* هر کاربر باید یک توکن داشته باشد. این توکن برای هر کاربر ثابت است.

endpoint های موردنیاز

در همه‌ی *endpoint* ها، پاسخ باید به‌صورت *JSON* باشد.

اطلاعات ورودی به‌صورت `application/x-www-form-urlencoded` به *endpoint* ها ارسال می‌شوند.

بررسی *up* بودن سرویس

پاسخ این *endpoint* باید به‌صورت زیر باشد:

- کد وضعیت: 200
- بدنه: {"ok":true}

ثبت نام

دو پارامتر username و password باید به این endpoint ارسال شوند. در صورتی که حداقل یکی از این پارامترها ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400
- بدنه: {"ok":false,"error":"no username or password provided"}

اگر کاربری با نام کاربری وارد شده از قبل موجود باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400
- بدنه: {"ok":false,"error":"user already exists"}

در غیر این صورت، کاربر باید ساخته شود، یک توکن یکتا برایش تولید شود و پاسخ به صورت زیر باشد:

- کد وضعیت: 201
- بدنه: {"ok":true,"token":"USER_TOKEN"}

ورود به حساب کاربری

دو پارامتر username و password باید به این endpoint ارسال شوند. در صورتی که حداقل یکی از این پارامترها ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400
- بدنه: {"ok":false,"error":"no username or password provided"}

اگر نام کاربری یا رمز عبور نادرست باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400
- بدنه: {"ok":false,"error":"invalid username or password"}

در غیر این صورت، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 200

- بدنه: {"ok":true,"token":"USER_TOKEN"}

درج پیشنهاد یا انتقاد

این *endpoint* نیازمند *authentication* است. در ریکوئست ارسالی مقدار *Authorization* باید برابر با توکن کاربر باشد (بدون **Bearer** یا موارد مشابه).

پارامتر *text* (متن پیشنهاد یا انتقاد) باید به این *endpoint* ارسال شود. در صورتی که این پارامتر ارسال نشده باشد یا برابر با رشته‌ی خالی باشد، پاسخ باید به صورت زیر باشد:

- کد وضعیت: 400

- بدنه: {"ok":false,"error":"no text provided"}

در غیر این صورت، پیشنهاد یا انتقاد باید ثبت شود و پاسخ به صورت زیر باشد:

- کد وضعیت: 201

- بدنه: {"ok":true}

دریافت لیست پیشنهادات و انتقادات

این *endpoint* باید لیست پیشنهادات و انتقادات ثبت شده در سامانه را در قالب یک لیست برگرداند. این لیست شامل پیشنهادات و انتقادات ثبت شده توسط همه‌ی کاربران است.

- کد وضعیت: 200

- مثالی از پاسخ:

```

1  [
2    {
3      "user": "username1",
4      "text": "sample suggestion 1"
5    },
6    {
7      "user": "username2",
8      "text": "sample suggestion 2"
9    }
10 ]
```

نکات تکمیلی

▼ نصب نیازمندی‌ها و اجرا

برای حل این سؤال می‌توانید از هر زبان و هر تکنولوژی‌ای که می‌خواهید استفاده کنید. به‌صورتی که در یک پوشه به نام `api` کد برنامه را نوشته و در فایل به نام `runner.sh` که توسط `sh` اجرا می‌شود، باید برنامه‌ی خود را اجرا کنید. توجه کنید که حتماً باید `Dockerfile` مربوط به پروژه‌ی خود را برای ما ارسال کنید.

در پروژه‌ی اولیه، ۴ داکر فایل برای `php`، `python`، `golang` و `node` قرار دادیم که می‌توانید از آن‌ها مستقیماً استفاده کنید. در صورتی که از یکی از این زبان‌ها برای حل سؤال استفاده می‌کنید، کافیهست که `Dockerfile` مربوط به آن را در پوشه‌ی `api` کپی کنید و طبق توضیحات داده شده، سؤال را حل کنید. برای نصب نیازمندی‌های پایتون از `requirements.txt`، برای پی‌اچ‌پی از `composer.json`، برای گولنگ از `go.mod` و برای نودجی‌اس از `package.json` استفاده کنید.

در صورتی که زبان مورد استفاده‌ی شما، چیزی به جز این ۴ مورد است، باید خودتان داکر فایلی در پوشه‌ی `api` به‌شکلی بنویسید که بتواند نیازمندی‌های پروژه‌ی شما را نصب کرده و برنامه‌ی شما را مانند داکر فایل‌های موجود اجرا کند.

- نیازی به *persistent* بودن داده‌ها نیست!
- سیستم داوری `docker-compose.yml` زیر را خارج از فولدر `api` پاسخ شما قرار می‌دهد و با دستور `docker-compose up --build` آن را اجرا می‌کند.

docker-compose.yml

```

1 | version: "3"
2 |
3 | services:
4 |   api:
5 |     build: "./api"
6 |     container_name: "api"
7 |     ports:
8 |       - "80:80"
```

- شما مجاز به تغییر یا ارسال `docker-compose.yml` دلخواه نیستید.

- سرویس شما باید روی پورت 80 آدرس localhost قابل دسترسی باشد.
- توصیه می‌کنیم در runner.sh خود API تان را روی 0.0.0.0:80 اجرا کنید.

▼ تغییر Dockerfile

امکان تغییر فایل Dockerfile وجود ندارد، اما در اسکریپت runner.sh می‌توانید هر دستوری را اجرا کنید.

نحوه ارسال پاسخ

شما می‌توانید تمامی محتوای موجود در پوشه‌ی api را تغییر دهید و هر فایلی که می‌خواهید اضافه یا کم کنید.

```

1 | api
2 | └─ api.py # or main.go somefile.js anyfile.php name.any ...
3 | └─ Dockerfile
4 | └─ requirements.txt # or go.mod package.json composer.json
5 | └─ runner.sh

```

توجه کنید که نام فایل کد شما برای سیستم داوری اهمیتی ندارد و این خود شما هستید که در runner.sh از نام آن برای اجرای پروژه استفاده می‌کنید.

در نهایت این پوشه را zip کرده و ارسال کنید. توجه کنید که پس از extract کردن فایل zip شما، باید پوشه‌ی api را ببینیم که درون آن Dockerfile وجود دارد.