

Задача

Разработать RESTful API для управления списком задач с аутентификацией по токену (Sanctum). Клиент задаёт требования к данным и поведению — ваша задача реализовать это корректно, выбрав оптимальные подходы.

Требования к данным

Task (задача):

- title — строка, обязательная, макс. 255 символов;
- description — текст, optional;
- status — строка, допустимые значения: pending, in_progress, completed;
- due_date — дата в формате Y-m-d, optional;
- задача всегда принадлежит пользователю (связь с пользователем подразумевается, но структура БД — на усмотрение разработчика).

Требования к API

Аутентификация

POST /api/login

- Входные данные:

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

- Успешный ответ:

```
json
{
  "token": "abc123..."
}
```

POST /api/logout

- Требование: заголовок Authorization: Bearer <token> обязателен;
- Успешный ответ

```
json
```

```
{  
    "message": "Logged out"  
}
```

Управление задачами (требуют авторизации)

GET /api/tasks

- **Успех (200 OK):** пагинированный список задач текущего пользователя в формате:

```
json  
{  
    "data": [  
        {  
            "id": 1,  
            "title": "Купить молоко",  
            "description": "В магазине у дома",  
            "status": "pending",  
            "due_date": "2024-12-31",  
            "created_at": "2024-01-01 10:00:00",  
            "updated_at": "2024-01-01 10:00:00"  
        }  
    ],  
    "links": { ... },  
    "meta": { ... }  
}
```

GET /api/tasks/{id}

- **Успех:** JSON с задачей (структура как в списке);

POST /api/tasks

- **Входные данные:**

```
json  
{  
    "title": "Новая задача",
```

```
"description": "Детали задачи",
"status": "pending",
"due_date": "2024-12-31"
}
```

- **Успех:** JSON с созданной задачей + заголовок Location с URL задачи;
- Отправка уведомления на email пользователю об успешной задаче

PUT/PATCH /api/tasks/{id}

- **Входные данные:** аналогично POST (опционально);
- **Успех (200 OK):** JSON с обновлённой задачей;

DELETE /api/tasks/{id}

- **Успех:** пустой ответ;

Функционал проверки статуса задачи:

Если задача не была завершена в срок due_date, отправить уведомление на почту

Технические требования

1. **Фреймворк:** Laravel 10+ с Sanctum для аутентификации.
2. **БД:** PostgreSQL
3. **Миграции и модели:**
 - создать необходимые таблицы и связи (структура — на усмотрение разработчика);
 - модель Task должна учитывать принадлежность пользователю.
4. **Вывод ошибок**
 - на усмотрение разработчика
5. **Валидация:**
 - На усмотрения разработчика
6. **Тесты (минимум):**
 - Минимум 1 юнит-теста (проверка правил валидации в FormRequest);
 - Минимум 2 фича-теста, например:

- регистрация/вход/выход;
- создание задачи (с токеном и без);
- доступ к задаче другого пользователя (должен быть запрещён).

7. **Документация:** README.md с:

- инструкцией по установке (composer install, php artisan migrate, php artisan key:generate);