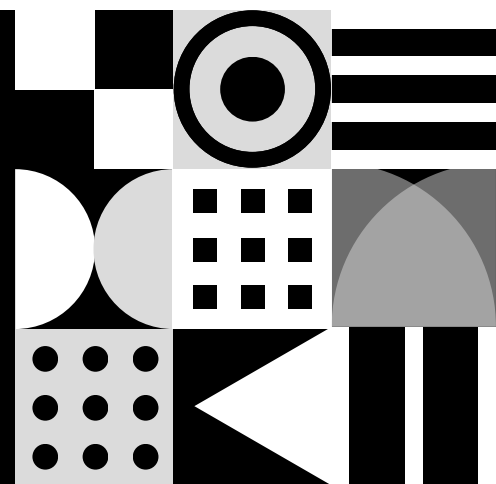


Тестовое задание

Task Management System API



v 1.0.0

Содержание

Введение

Что будет
оцениваться
дополнительно?

Технические
требования

FaQ

Дополнительные
требования

Процесс выполнения

Введение

Современные системы управления задачами играют ключевую роль в организации рабочего процесса как в корпоративной, так и в индивидуальной среде. Разработка “Task Management System API” представляет собой значимую задачу, направленную на обеспечение эффективного управления задачами в различных контекстах.

В рамках данного проекта предлагается создать универсальное решение, которое позволит клиентам выбирать наиболее подходящий способ взаимодействия с системой. Для этого

предусматривается реализация либо RESTful API, либо GRPC API, обеспечивая тем самым гибкость и универсальность в использовании.

Технические требования к разработке включают в себя создание микросервисов для различных функциональных блоков системы, реализацию CRUD операций для управления задачами, авторизация и аутентификация.

Технические требования



***Важно именно подход к
выполнению задачи, а не
100% выполнение***



В таблице приведены описания требований к разработке решения. Вы можете выполнять их не последовательно. У каждого шага есть вес, который влияет на итоговую оценку вашего ТЗ.

Вес - значимость каждого требования в процессе оценки выполнения ТЗ.

Ключевые шаги	Вес
Разработать RESTful, или GRPC API для взаимодействия с системой	50
Разработать микросервисы для различных функциональных блоков системы (например, управление задачами, управление пользователями, далее)	20 x (число микросервисов)* *увеличение числа микросервисов может стать проблемой, поэтому засчитываться будут только те, выделение в отдельный сервис которых было обоснованным
Использовать любой язык программирования для реализации API (например, Python, Node.js, Java и т.п.)	30
Выбор базы данных остается на усмотрение разработчика, но рекомендуется использовать реляционную или NoSQL БД с возможностью масштабирования.	10
Обеспечить минимальную аутентификацию и авторизацию пользователей	20
Реализовать CRUD (Create, Read, Update, Delete) операции для задач	20 x (число операций CRUD)
Предусмотреть обработку ошибок и возврат соответствующих HTTP статусов для RESTful API и кодов ошибок для GRPC API	40

Дополнительные требования

(необязательно, но будет плюсом)

Данные требования могут быть расширены вами.

Если вами было выполнено всё ТЗ и вы захотели добавить функциональность, которой не предполагалось, просьба о них написать, будет произведён перерасчёт весов.

Наши предложения:

- Обеспечить возможность прикрепления файлов к задачам (вес - 10)
- Добавить возможность присваивания тегов к задачам и фильтрации задач по тегам (вес - 30)
- Реализовать механизм оповещений (например, по электронной почте) о новых задачах или изменениях в существующих задачах (вес - 30)
- Разработать Unit и Integration тесты для проверки функциональности API (вес - 50)
- Front-end часть или документация Swagger UI при отсутствии front-end части (вес - 30)



Выполнение дополнительных требований не обязательно

Процесс выполнения

Здесь описан процесс выполнения тестового задания.

01 Разработка тестового задания

Полный цикл разработки ТЗ по предложенным требованиям в срок.

02 Отправка тестового задания

Для отправки тестового задания необходим публичный git-репозиторий

03 Получение обратной связи

В течении 1-3 дней мы предоставим обратную связь по ТЗ и сообщим о результате

Что будет оцениваться дополнительно?

Оценка данных требований будет производится отдельно.

Запускаемое решение

Включает в себя:

- Решение запускается
- Работает без вылетов и крашей
- Обработка ошибок

Оформление кода

Включает в себя:

- Понятный и логичный нейминг
- Соответствие конвенции языка

Документация к решению или front-end часть

Включает в себя:

- Swagger UI*
- Frontend

*если нет front-end части для тестирования вашего API

FaQ

Часто задаваемые вопросы по разработке:

- Можно ли использовать Docker?
 - Можно
- Если я использую БД, как передать её?
 - Создайте инициализирующий скрипт
- Оценивается ли дизайн front-end?
 - Нет, дизайн не будет критерием

Если у вас возникли дополнительные вопросы вы можете задать их в ответном письме или по контактам ниже.

Контакты

Пономарёв Артём Владимирович

RnD разработчик

г. Череповец ул. Ленина 123А

+ 7 960 294 23 55

av.ponomarev@severstal.com

