

Задание на Middle Python developer

1. Какой опыт в коммерческой разработки на Python?

3 года

2. Приходилось ли сильно оптимизировать Python код? Какие подходы и инструменты для этого использовали?

Да, были оптимизации.

Использовал более корректные структуры данных, оптимизировал количество циклов и условий, добавлял кеширование.

3.

а) Какие из этих фреймворков Django, Flask, FastAPI вы использовали в работе?

В работе использовал Django (+DRF), FastAPI. Flask использовался в пет проекте в самом начале.

б) Какой опыт работы с ними? Дайте оценку своему уровню знаний фреймворка?

Опыт работы с Django (+DRF) - 2 года, FastAPI - 8+ месяцев (на текущем месте работы)

4. Какие брокеры сообщений использовали в работе (RabbitMQ, Kafka и тд)?

Использовал RabbitMQ.

5.

а) Работали ли ранее с SQL, (чистые запросы, ORM), какие ORM использовали?

Писал как сырые запросы SQL, так и работал с ORM - DjangoORM, Tortoise, SQLAlchemy (меньше всего)

б) Какие субд использовали в работе (Sqlite, PostgreSQL, MySQL, MongoDB, ClickHouse, и др.)?

Работал с SQLite3, PostgreSQL, Redis, MongoDB.

Также работа с s3 хранилищами.

в) Каков максимальный размер БД с которой был опыт работы?

MongoDB содержание от 20к документов в коллекциях.

PostgreSQL более 100к записей.

6. Есть ли еще какой-то интересный вам язык программирования, который используете и какие задачи предпочитаете на нем решать?

Да, хочу погрузиться в 2 языка: Go и Java.

7. Задача на подумать:

Есть веб-сервис А, принимающий запросы от пользователей, запросов может приходиться в час 200 000. В день приходит до 2 000 000 запросов

Сервис А должен отправить эти данные в сервис Б, чтоб обработать данные.

Сервис Б может обработать 100 000 запросов в час.

Запросы от пользователей поступающий в сервис А нельзя потерять.

Как бы вы организовали межсервисное взаимодействие, хранение данных, результатов, какие инструменты использовали бы для этого?

Самый первый вариант, который приходит в голову, это построить взаимодействие через брокер сообщений, например, RabbitMQ.

Сервис А принимает сообщения.

Валидирует сообщение при необходимости.

В случае корректности помещаем их в очередь. В случае некорректности, можем сразу ответить пользователю об этом.

Сервис Б обрабатывает сообщения из очереди.

Сервис Б может возвращать результат действия в очередь для результатов.

Сервис А может прочитать сообщение из очереди с результатом, если необходимо вернуть ответ пользователю.

8. Ссылка на задание <https://gitlab.zvonok.in/-/snippets/28>.