

Привет! Предлагаем тебе тестовое задание, которое поможет нам быстрее понять твой уровень знаний и навыков, а также узнать ход твоих мыслей.

Необходимо выполнить одно из тестовых заданий на выбор. Все задания рассчитаны на выполнение в течении не более 16 часов рабочего времени. Желательно выполнить на Scala но можно и на одном из языков (C#, Java, ...).

**Внимание!** Тестовые задания на Python, в одном файле, в процедурном стиле - не принимаются т.к. задания выполненные в подобном стиле не могут показать нам ваши навыки построения приложения.

Исходный код необходимо предоставить в виде репозитория на github или аналогичном сервисе. Также необходимо написать небольшой документ с описанием архитектуры приложения. По умолчанию предполагается API - REST.

Общие дополнительные (необязательные, но будут плюсом) условия:

- Сохранение запросов к сервису и вывод лога в api;
- Статистика запросов к сервису (сколько, время обработки и прочее);
- Кэширование запросов;
- Docker файл для развертывания сервиса;
- Фронтенд для сервиса;
- Unit тесты;
- GraphQL для API.

Решение мы принимаем на почту: [scala\\_job@embedika.ru](mailto:scala_job@embedika.ru)

Что нужно указать в письме:

1. ФИО;
2. ваше резюме в PDF или отправьте ссылку на HH.ru;
3. контакты (номер телефона или tg).

### Важно:

По времени выполнения мы не ограничиваем, ты сам(а) устанавливаешь для себя срок, сообщи нам, пожалуйста, сколько времени тебе понадобится.

## ТЕСТОВОЕ ЗАДАНИЕ “ПАРСИНГ ДАННЫХ И API”

Необходимо распарсить открытые данные о цене на сырую нефть марки “Юралс” и предоставить к ним доступ по API.

### Данные

#### Описание API:

Метод	Параметры	Возвращаемый результат
Цена на заданную дату	Дата	Значение цены
Средняя цена за промежуток времени	Диапазон дат	Значение цены
Максимальная и минимальная цены за промежуток времени	Диапазон дат	Json с полями min и max
Статистика по загруженным данным	Нет параметров	Json с полями всего записей

Дополнительные необязательные условия (будут плюсом):

- Любая дополнительная статистика на ваш выбор;
- Другие источники данных.

## ТЕСТОВОЕ ЗАДАНИЕ “СПРАВОЧНИК АВТОМОБИЛЕЙ”

Необходимо реализовать сервис справочника автомобилей с хранением данных в базе или файле.

Минимальная информация по объекту:

- Номер (регистрационный знак, например AA999A96);
- Марка;
- Цвет;
- Год выпуска.

Описание минимально возможного API:

Метод	Параметры	Возвращаемый результат
Вывод списка	Нет параметров или параметры для фильтрации (один или несколько из возможных атрибутов объектов)	Список всех объектов
Добавление автомобиля	Json с параметрами объекта	Результат операции (успех, ошибка, объект уже существует)
Удаление автомобиля	Идентификатор объекта	Результат операции (успех, ошибка, объект не найден)
Статистика базы	Нет параметров	Статистика по базе (например количество записей, дата добавления первой записи, дата добавления последней записи и прочие на ваше усмотрение)

Дополнительные необязательные условия (будут плюсом):

- Сортировка по атрибутам при запросе списка;
- Дополнительные атрибуты объектов;
- API можно усложнять и менять, при условии соответствия минимальным требованиям.

## ТЕСТОВОЕ ЗАДАНИЕ “АГРЕГАТОР ПОГОДНЫХ СЕРВИСОВ”

Необходимо написать агрегатор информации о погоде из открытых сервисов.

**ПРИМЕР СЕРВИСА**, альтернативы нагуглить не сложно.

### Описание минимально возможного API:

Метод	Параметры	Возвращаемый результат
Вывести информацию о погоде в городе	Город или название местности	Список погоды из сервисов с которыми умеет работать приложение
Список сервисов, с которыми работает приложение	Нет параметров	Список сервисов с которыми умеет работать приложение и их статус доступности (работает в текущий момент или нет)

Дополнительные необязательные условия (будут плюсом):

- Написать текстом как бы вы обрабатывали ситуация с разными названиями городов в каждом сервисе;
- Унификация вывода информации от каждого из сервисов;
- Можно добавлять другие методы API.