

## ТЕСТОВОЕ ЗАДАНИЕ ДЛЯ ВАКАНСИИ TRAINEE BACKEND DEVELOPER (JAVA)



*Привет!*

*Предлагаем тебе выполнить тестовое задание, чтобы мы могли оценить твои скиллы, полученные во время обучения.*

*Срок выполнения: 3 дня.*

*Результат пришли, пожалуйста, в почту [hr@digitalchief.ru](mailto:hr@digitalchief.ru) либо в телеграм [https://t.me/hiring\\_it](https://t.me/hiring_it) вместе со своим резюме, контактными данными и временем, за которое ты выполнил тестовое задание.*

*Удачи в выполнении!*

### **Задание:**

1. Выбрать предметную область. На основе выбранной области, придумать композицию, состоящей из не менее 2 объектов, отношения между которыми "один ко многим". Пример композиций: город - достопримечательность, университет - факультет, студент - оценка по предмету. Необходимо в произвольной форме сделать описание предметной области, выбранной композиции и используемые в ней объекты. Для каждого объекта описать не менее 5 свойств.
2. Создать Git репозиторий в котором будут храниться все файлы необходимые для сборки и запуска будущего приложения.
3. Установить реляционную базу данных (БД) на выбор: MySQL или PostgreSQL. Создать схему БД для хранения и управления данными для смоделированных объектов. Добавить файлы инициализации схемы данных в Git репозиторий. Дополнительным, совсем необязательным, плюсом будет использование Docker контейнера для БД.
4. Создать REST API приложение в архитектурном стиле MVC на основе фреймворка Spring Boot. Требования:
  - Java, версия от 8+
  - приложение содержит пул подключений к БД
  - имплементированы классы-модели в Java для отображения смоделированных объектов
  - решена задача объектно-реляционного отображения
  - создан интерфейс API (Application Programming Interface) на основе архитектурного стиля REST для проведения CRUD (Create, Read, Update, Delete) операций над объектами. Взаимодействие с API реализовано в формате JSON
  - в корне проекта содержится README.md файл. README.md содержит описание всех зависимостей подключенных в проект, а также пошаговые инструкции для сборки и запуска приложения

- подготовлены по одному примеру запроса для каждого эндпоинта в одном из следующих форматов: cURL, HTTP, wget. Примечание: можно сгенерировать результат с использованием инструмента тестирования Postman или инструмента автоматической генерации документации Swagger. При использовании Postman добавить файлы коллекции Postman в структуру проекта

**Ожидаемый результат:**

1. Описана предметная область и композиция, формулировки четкие и не допускают двусмысленности
2. Код приложения можно скачать с Git-хранилища
3. Приложение запускается по инструкции. Все зависимости описаны
4. Подготовленные запросы проходят успешно, соответствующие данные в БД обновляются