

Финальное тестовое. «Яндекс Метро»

Задание 1

Составьте тест-план и напишите чек-лист мобильного приложения «Яндекс Метро» (скачать приложение вы можете через Apple Store, Google Play). Само тестирование по этому чек-листу проводить не нужно.

Результат приложите в виде документа с тест-планом, в котором будут все необходимые ссылки.

Оценивается:

- оформление,
- составленный тест-план,
- покрытие приложения функциональным чек-листом,
- покрытие приложения нефункциональными проверками,
- покрытие приложения специфичными мобильными проверками.

Задание 2

Перед вами API GitHub issues — один из самых популярных трекеров ошибок в мире: <https://docs.github.com/en/rest/issues/issues?apiVersion=2022-11-28#about-issues>.

GitHub issues предоставляет владельцам репозитория возможность организовывать, помечать и связывать вопросы с определенными этапами разработки.

С помощью Postman протестируйте работу следующих методов:

- Создайте issue с названием Issue 1, описанием Something went wrong. Также у этой issue должен быть label — bug — и assignee — вы (текущий логин на GitHub).
- Получите список issues.
- Измените название задачи на Issue 2.
- Удалите Issue 2.

Для начала работы с методами вам потребуется создать аккаунт на GitHub (если еще не создан), создать репозиторий с произвольным содержимым (если такого еще нет) и создать API-ключ на соответствующее приложение.

Результат прикрепите в виде готовой JSON-коллекции Postman. Также приложите README-файл (документацию) с информацией об использовании (запуске) коллекции и ссылкой на свой GitHub-репозиторий.

Оценивается:

- оформление;
- правильность запросов;
- использование и передача переменных между запросами, если это необходимо;
- проверка возвращаемых данных: написаны тестовые скрипты на проверку статус-кода и других необходимых параметров, если это необходимо;
- наличие документации к созданной коллекции.

Задание 3

Составьте API-тест-кейсы по коллекции запросов, полученной в задании 2.

Результат приложите в виде тест-кейсов, созданных в любимой вами TMS или в Google Doc.

Оценивается:

оформление тест-кейсов,
полнота проверок.

Задание 4

Даны 2 таблицы:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2 (17)
COUNTRYCODE	VARCHAR2 (3)
DISTRICT	VARCHAR2 (20)
POPULATION	NUMBER

COUNTRY

Field	Type
CODE	VARCHAR2 (3)
NAME	VARCHAR2 (44)
CONTINENT	VARCHAR2 (13)
REGION	VARCHAR2 (25)
SURFACEAREA	NUMBER
INDEPYEAR	VARCHAR2 (5)
POPULATION	NUMBER
LIFEEXPECTANCY	VARCHAR2 (4)
GNP	NUMBER
GNPOLD	VARCHAR2 (9)
LOCALNAME	VARCHAR2 (44)
GOVERNMENTFORM	VARCHAR2 (44)
HEADOFSTATE	VARCHAR2 (32)
CAPITAL	VARCHAR2 (4)
CODE2	VARCHAR2 (2)

2 таблицы связаны между собой посредством Country code.

С помощью запросов SQL:

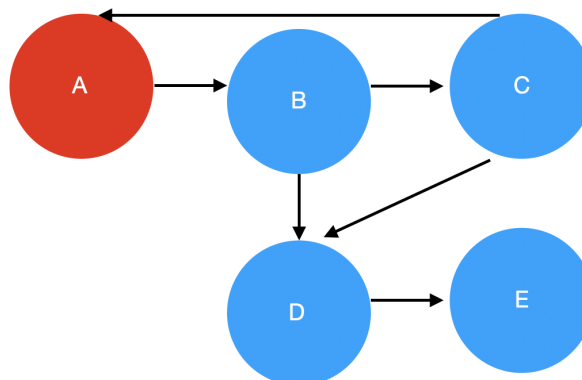
1. Выведите названия всех городов, где *CONTINENT* = *Africa*.
2. Выведите количество уникальных регионов (Country region).
3. Выведите названия всех континентов (*CONTINENT*) и их соответствующее среднее городское население (*POPULATION*).

Результат приложите в виде готовых SQL-запросов.

Оценивается: правильность SQL-запросов.

Задание 5

Есть диаграмма состояний и переходов:



Начальное состояние находится в A.

Перечислите все E2E-кейсы, которые нужно проверить.

Результат приложите в виде соответствующих переходов (укажите все возможные переходы).

Например:

A-C-E

A-D-B-D-E

...

Оценивается:

- навык определения оптимальных данных для тестирования;
- знание техники тест-дизайна: диаграмма состояний и переходов.

Задание 6

Ответьте на вопросы:

1. Разработчик не согласен с результатами тестирования. Что вы будете делать?
2. Вы единственный в новой команде тестировщик. С чего вы начнете?
3. Вы заполнили форму регистрации на сайте (email, пароль), нажимаете «Создать» и получаете ошибку 404. Ваши действия?

Результат приложите в виде текстовых ответов на вопросы.

Оценивается:

- общий кругозор и понимание будущей профессии,
- адекватность ответов.