

Тестовое задание

Описание

Тестовое задание направлено на ознакомление студентов с работой API веб сервисов. На примере двух сервисов Вы должны получить данные о себе, структурировать, отобразить их в виде веб-странички и развернуть свое приложение на сервере.

Задание состоит из трех частей: работа с API, создание веб приложения, развертывание на сервере, которые делаются последовательно.

Работа с API

Описание

В первой части тестового задания Вам предстоит создать небольшое приложение на выбранном языке программирования. Эта программа будет запрашивать данные о Вас из выбранных веб сервисов и сохранять их в файлы. Загрузка работы производится в репозиторий Gitlab, перейти в который можно через SmartLMS.

В итоге у вас должен получиться репозиторий с пустой веткой master и новой веткой, соответствующей названию вашей фичи с вашей программой. Также, вы должны создать merge request из вашей ветки в главную. Не позднее чем за два дня до дедлайна вы можете попросить консультацию в виде ревью вашего merge request-a, позднее проводятся только консультации по вопросам в чате, а работа проверяется без возможности исправления.

Чтобы применять любой язык программирования, кроме Python, необходимо предупредить преподавателя в общем чате как минимум за неделю до дедлайна, иначе использование другого языка не разрешено.

Оценивание задания

Требования

- Ведение репозитория
 - git flow — фичи ведутся в отдельных ветках: фичи ведутся в ветках, для вливания новой фичи в главную ветку используются merge request-ы;

- количество коммитов — количество коммитов больше одного;
- названия коммитов
 - понятное название — по названию коммитов можно понять что в них было сделано;
 - единый стиль — коммиты названы в едином стиле.
- структура репозитория — названия папок соотносятся с общим стилем названий, валидное расположение файлов.
- Структура кода
 - проверка на качество — прохождение проверки на качество кода;
 - единый стиль — код написан в едином стиле;
 - использование библиотек — используемые библиотеки упрощают работу программы, не используется лишних библиотек.
- Работоспособность кода
 - успешный запуск — код успешно запускается при автоматической проверке, а также по инструкции из README;
 - правильный результат — код успешно запускается и успешно завершается с правильным итоговым результатом.
- Работа с API — произведена интеграция с двумя онлайн сервисами, из которых получены данные о достижениях студента и сохранены в текстовые файлы
 - сохранение в файлы — текстовые файлы в формате json должны представлять человеко читаемый текст, файлы должны иметь название, соответствующее названию сервисов;
 - интеграция с сервисами — сервисы должны предоставлять информацию о студенте, который разработал данную программу, сервисы могут быть любыми, главное, чтобы доступ к ним осуществлялся через API;
- Безопасность кода — для получения доступа к онлайн сервисам используемые credentials не находятся в репозитории, а выводятся в конфигурационный файл вида “.env” . При загрузке программы в Gitlab репозиторий необходимо указать в переменных окружения Gitlab необходимые переменные, чтобы Gitlab мог запустить автоматическую проверку кода, инструкцию как это сделать смотрите ниже;
- Документация — файл README.md репозитория содержит информацию про программу, мануал по её запуску;
- Оригинальность — применение необычных технологий и подходов к программированию;
- Самостоятельность написания — при наличии плагиата больше 20% за работу ставится оценка 0.

Оценивание требований

- Ведение репозитория = 2 балла
 - git flow = 1 балла;
 - количество коммитов = если сделан всего 1 коммит, оценка снижается до 1 балла за блок, при полном отсутствии коммитов ставится оценка 0 за всю работу;
 - названия коммитов = 0.5 балла;
 - структура репозитория = 0.5 балла;
- Структура кода = 1 балл
 - проверка на качество = 0.5 балла;
 - единый стиль = 0.5 балла;
 - использование библиотек = при использовании лишних библиотек или при нерациональном использовании библиотек оценка снижается до 0 баллов за этот блок;
- Работоспособность кода = если код не запускается, то оценка за задание снижается вплоть до 0. Если код выводит неправильный результат оценка за задание снижается вплоть до 0;
- Работа с API = 4 баллов
 - сохранение в файлы = 1 балл
 - оценка снижается за неправильный формат файлов, нечитаемый текст в файле, некорректное название файлов
 - интеграция с сервисами = 4 баллов
 - при интеграции только с одним сервисом оценка за блок снижается до 2 баллов;
 - при интеграции с сервисами, из которых не получается информация о студенте оценка снижается до 0 баллов;
- Безопасность кода = 0.5 балла
 - если токены помещены внутрь программы без сокрытия ставится 0 за этот пункт;
 - если токены вынесены в .env файл, но не введены в Gitlab - это влияет на пункт “Работоспособность кода”;
- Документация = 0.5 балла;
- Оригинальность = 2 балла;
- Самостоятельность написания — при наличии плагиата больше 20% за работу ставится оценка 0.

Как добавить переменные окружения в репозиторий

Для запуска проверки с переменными окружения необходимо во вкладке CI/CD в Gitlab вручную запустить проверку кода.

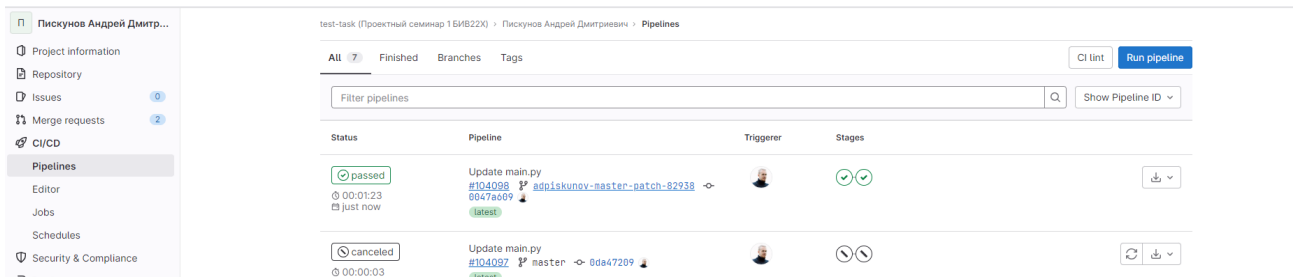


Рисунок 1 — Вкладка CI/CD в репозитории Gitlab

Нажимаем Run Pipeline, выбираем ветку, далее вводим необходимые переменные, с таким же названием, как и в .env файле.

Нажимаем Run Pipeline.

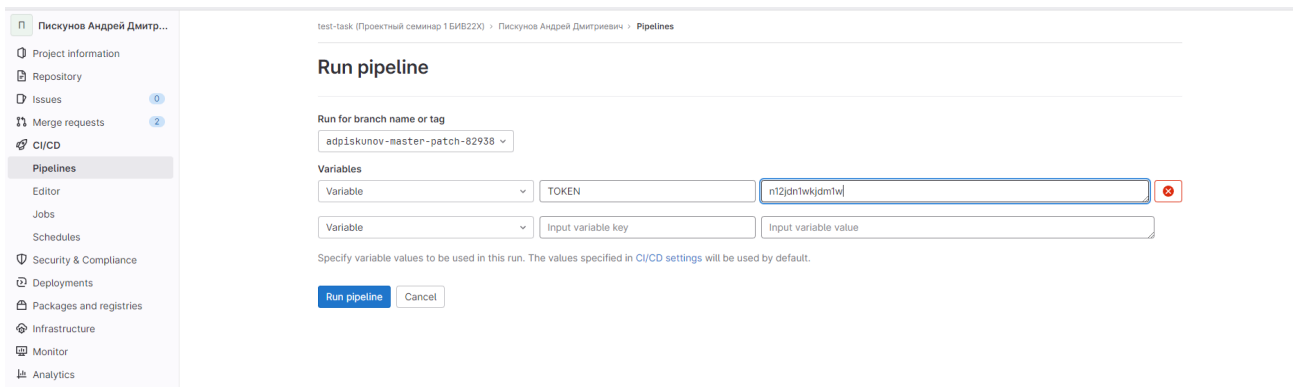


Рисунок 2 — Ручной запуск пайплайна в Gitlab CI/CD

Данные действие необходимо делать, когда код прошел первый этап проверки Linting, так как переменные окружения могут понадобится только для второго этапа Run, на котором запускается код вашей программы.