# Комментарий к выполненному тестовому заданию.

Согласно требованию в результате решения тестового задания был создан блок тест-кейсов для проверки функционала сервера, а также написана система для автоматизированного тестирования сервера.

Созданная система для автоматизированного тестирования реализована ЯП Python версии 3.7.

Базируется на фреймворках:

* PyTest. Фреймворк для создания удобной тестовой среды.
* Requests. Библиотека для работы с API.

## Общие комментарии системе и написанным автотестам:

* Реализовано 7 позитивных автотестов, базирующихся на 6 тест-кейсах, описанных отдельно. Путь к автотестам: tests. Короткие названия автотестов:
  + Создание медведей всех типов.
  + Создание медведей с лишними полями.
  + Удаление всех медведей.
  + Удаление определенных медведей.
  + Чтение определенного медведя.
  + Чтение всех медведей.
  + Изменение медведя.
* Большинство автотестов будут “красными”. Сервер работает с ошибками:
* Content-type ответов сервера декларируется как “text/html;charset=utf-8”, однако данные в теле ответа возвращаются в формате JSON.
* На часть негативным тестов сервер отвечает статус кодом 500. Данный код сообщает, что сервер не может обработать данную ситуацию. Нет смысла автотестам ожидать ответ 500, так как причиной данной ошибки может быть все что угодно.

Как следствие в данному случае автотесты будут ожидать код ответа 200 (или 3xx/4xx), с текстом ошибки.

* Создание медведя с типом GUMMY работает не корректно. В результате создается медведь с телом: {"bear\_id":22,"bear\_type":"UNKNOWN","bear\_name":"EMPTY\_NAME","bear\_age":0.0}. При чтении медведя получаю ответ: null.
* При редактировании медведя игнорируются поля bear\_type, bear\_age. Ошибки в данных полях или их отсутствие также игнорируются.
* При редактировании имени медведя оно сохраняется в том регистре, в котором было указано, однако при создании медведь создается только в верхнем регистре.
* Класс в bear\_server/bear\_body.py описывает структуры JSON тела медведя. Аналог POJO из Java.
* API запросы заранее подготовлены в файле bear\_server/prepared\_requests.py.
* Система логирования отсутствует в связи с экономией времени.
* Конфигурирование системы производится bear\_server/server\_settings.py.
* Кроме того, возможно добавить эндпоинты в bear\_server/end\_points.py.