

Принцип работы приложения

«Система скоринга 1Т»

Разработчик: Яковлев А.Л. DO

Оглавление

Оглавление	1
Принцип работы приложения	2
Определения	2
Предположение №1 Получение данных	3
Опровержение.....	3
Предположение №2 Получение данных	4
Предлагаемая структура данных JSON.....	4
Пример получения данных и обработка их	5
Работа приложения	6

Принцип работы приложения

Определения

Исполнитель – команда разработчиков, создающая приложение

Приложение – разрабатываемая система исполнителем

ЛМС – система заказчика, реализующая процесс обучения

Пользователь – обучающийся имеющий уникальный номер `id_user`
(надо продумать о пользователях обучающихся на двух курсах
одновременно!)

Статус – показатель обучающегося на определенный день обучения

DA – группа дата-аналитиков

Датасет – набор данных, сформированный по определенным правилам,
удовлетворяющим требованиям для получения статуса обучающегося

Данные – текстовая информация, описывающая процесс обучения
пользователя

База данных(база) – база данных (PostgreSQL) на стороне исполнителя

ВЕБ интерфейс – визуальное представление разрабатываемого
приложения, доступный через браузер по адресу

API – (Application Programming Interface) набор правил, по которым
программы взаимодействуют между собой.

JSON - (JavaScript Object Notation) стандартный текстовый формат для
хранения и передачи структурированных данных.

Предположение №1 Получение данных

Приложение будет получать данные по запросу от ЛМС заказчика, для
выставления статуса.

Предполагалось, что ЛМС заказчика будет присылать нам данные по
одному пользователю в установленном формате и мы будем возвращать
результат по одному пользователю.

Для этого необходимо прислать следующие данные: (из датасета
подготовленного DA, поля проверены @TatianaGlu)

table: exercise_results_v2 user_id (select) activity_id (select, where) created_at (select) result (select) success (where)	table: activity_history_viewed_v2 user_id (select), page_id (select) activity_id (select) attestation (select) created_at (select) page_type (where)
table: schedule_v2 course_id (select) date_shown (select) activity_id (where) - по этому ключу делаем join type (where)	table: activities_guide_v2 activity_id (select, where) - выводим поле и делаем по нему join obyaz_priznak (select) att_priznak (select)
table: users_v2 user_id (select, where) course_id (select) m2_progress (select)	table: users_logs_v2 user_id (select) comment (where)

Данные могут присылаться как с начала обучения и до текущей даты, так и за указанный период.

В данном случае на нашей стороне, необходимо сохранять данные в базу данных и контролировать обновление данных.

Опровержение

Предположение не удовлетворяет требованиям заказчика о предоставлении данных в разрабатываемой системе. Заказчик хочет видеть в разрабатываемой системе больше информации, а в данном предположении сформировали минимум информации для оценивания обучающегося, и сформировать дополнительную информацию не сможем.

Предположение №2 Получение данных

Данные необходимо предоставлять, разрабатываемому приложению, в том формате, в котором были предоставлены для разработки приложения. Такой формат позволит реализовать в ВЕБ интерфейсе пожелания заказчика об обучающихся на курсах.

Предлагаемая структура данных JSON

Для передачи данных от ЛМС заказчика разрабатываемому приложению, предлагаю использовать следующий формат передачи данных.

Формат данных JSON, способ передачи: отправка данных на API приложения, адрес API для передачи: /save_data, тело запроса в формате JSON представлен ниже:

```
{
  "table": "activities_guide_v2",
  "data":[
    {
      "course_id": 3,
      "course": "Нейро.PY",
      "provider": "1T",
      "modul": 2,
      "theme_id": 178,
      "theme": "Дочерние классы ",
      "task_id": 2577,
      "exercise": "SQLite3",
      "task_position": 6,
      "att_priznak": 0,
      "activity_id": 11974,
      "activity_type": "slide",
      "activity": "Установка и подключение к базе данных SQLite3 на Python",
      "obyaz_priznak": 0,
      "visibility": "{}"
    }
  ]
}
```

table – поле отвечающее за то в какую таблицу сохранять данные

data – поле отвечающее за сами данные, возможно передавать несколько данных за раз.

Остальные данные в таком же виде передаются.

Пример получения данных и обработка их

При старте курса по обучению ЛМС присылает:

1. На первый день информацию об обучающихся,

2. Приложение сохраняет в базу
3. Приложение пытается провести анализ и выставить статус (мало данных)
4. на второй день информацию об обучающихся (или за второй день только, или за 1-2 день)
5. Приложение сохраняет в базу (если только за второй день, то добавляет в базу информацию, если за 1 и 2 день, должна проверить изменения в первом дне и обновить у себя и второй день добавить)
6. Приложение пытается провести анализ и выставить статус (мало данных)
7. так продолжается до 7 дней
8. на 7 дне у нас уже в базе есть данные об обучающихся за 7 дней и данных достаточно для выдачи статуса по обучающимся
9. Приложение анализирует данные за первую неделю по каждому обучающемуся и выдает статус обучающегося
10. Все полученные статусы сохраняем в новую таблицу, необходимо сохранить в структуру:

Имя поля	Описание	Пример
id	уникальное поле для записи	1
data_create	таймстамп создания записи	1731856047
metrika	имя сохраняемой метрики	m2_progress
model_info	имя используемой модели (в будущем планируем развиваться и улучшать модели)	project_1611
id_user	номер пользователя	19489
value	полученный результат от модели	80.85
day_num	номер дня обучения	7

Работа приложения

Приложение имеет API которое реализует:

1. Получение данных от ЛМС заказчика – при получении сохраняет данные в базу данных
2. Получение информации по обучающемуся (id_user, id_course) – статус обучающегося

API может быть расширено для реализации других методов (ручек, что не любит Ирина)

Получение данных - ЛМС присылает данные в приложение в установленное время (один раз в три дня), есть данные, которые присылаются один раз для курса (такие как темы курса, расписание, список пользователей и т.д.), есть данные, которые присылаются постоянно (действия пользователей в системе ЛМС)

Приложение в фоновом режиме производит обработку данных и результат работы сохраняет в базу данных для дальнейшего использования.

По требованию предоставляет данные по обучающимся.

Приложение имеет графический интерфейс представленный в виде ВЕБ интерфейса.

API реализовано с помощью FastAPI и Uvicorn (работающего как служба)

ВЕБ интерфейс реализован с помощью nginx, PHP, Smarty, PostgreSQL