

---

# Eventindex. Разработка RestAPI

Алан Газзаев

4 курс бакалаврата,  
направление подготовки “Физика”

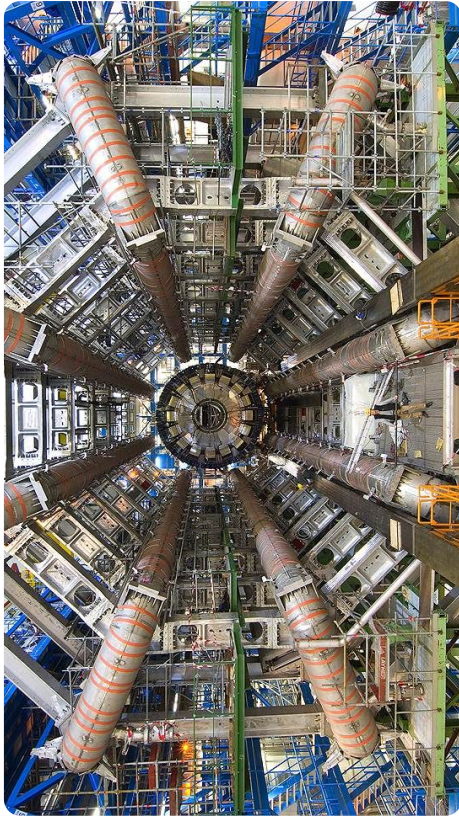
Северо- Осетинский Государственный Университет  
им. Коста Левановича Хетагурова

Научные руководители:

Прокошин Федор Валерьевич(ОИЯИ),

Тваури Инга Васильевна(СОГУ)

---



# EventIndex

# Atlas

---

# EventIndex

# SPD



При разработке eventIndex мы опирались на опыт создания аналогичной системы на установке ATLAS, с прицелом на использование в установке SPD на коллайдере NICA.

На SPD ожидается объемы данных того же порядка как на ATLAS при даже большем количестве событий. Становится ясно, что Eventindex необходимо использовать на детекторе SPD.

Несмотря на то что физические задачи этих экспериментов кардинально различны, системы обработки данных имеют сходные черты, что позволяет использовать похожие решения

Оба эксперимента имеют сходную модель данных событий и организацию хранения и обработки данных. Основное отличие: Online Filter вместо триггера.

---

# ПОЛЯ В ТАБЛИЦЕ EVENTS

```
eventindex=# SELECT * FROM events;  
run_number | event_number | olt_result | dsid_raw |                fuoid_raw
```

Для разработки системы каталога событий достаточно определить предварительный набор параметров которые мы, будем индексировать.

---

---

olt\_result - результат онлайн триггера, критерии по которым отобрано событие

dsid\_raw- id датасета, в который объединены файлы с raw-данными, внешний ключ для таблицы datasets

fuid\_raw- UUID для файла в котором хранится это событие, в файле будет храниться от тысячи до 10 тысяч событий, по uuid к файлу можно получить доступ посредством распределенной системы хранения

---

---

# ГЕНЕРАЦИЯ И ЗАПИСЬ

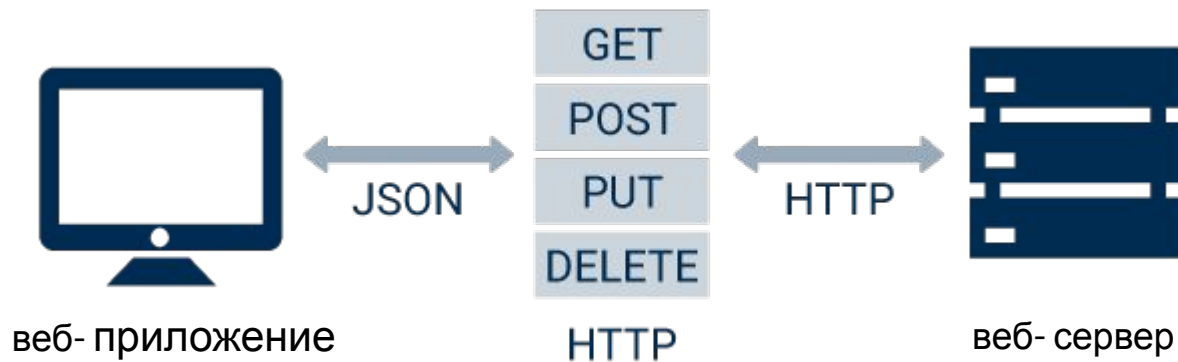
`gen_data.py` - генератор псевдоданных `event index` для последующего импорта

`store_events.py`- запись данных событий в таблицу `events` и датасета в таблицу `datasets`. Таблица датасетов имеет в качестве (`primary`) ключа `id` датасета, и содержит информацию о датасетах

---

---

# Веб- сервер(RestAPI)





---

# Структура

app.py

entry\_point.sh - основной файл для запуска приложения в двух режимах(development и production)

src:

flask\_manager.py- создает app

rest\_api.py - содержит эндпоинты

---

---

## **ЗАДАЧИ**

**Разработка RestAPI для выдачи `fuid_raw` по парам `run_number/event_number`**

## **РЕЗУЛЬТАТЫ**

**RestAPI на flask, осуществляет preprocessing данных(пар), введенных пользователем со стороны графического интерфейса, подключается к базе данных “eventindex”, осуществляет SQL- запрос к таблице “events” через интерфейс psycopg2 и возвращает список `FUID_RAW`**

---

---

# ПЛАНЫ

**Получение сертификатов безопасности от ОИЯИ**

**Разработка и внедрение системы токенов**

**Асинхронная выдача данных**

**Оптимизация загрузки данных**

---