



КАЗУС 2



# API - Документация

## Описание

Това е Flask API приложение за управление на увеселителния парк, което използва Snowflake като база данни. API-то предоставя възможност за извличане на информация за атракции, служители, билети, събития, посетители и финансови транзакции.

## Конфигурация

API-то използва следната конфигурация за свързване със Snowflake:

```
SNOWFLAKE_CONFIG = {  
    "user": "GERGANADIAVOLSKA",  
    "password": "SugaJhopeRm777",  
    "account": "ATLSZRI-IV85483",  
    "warehouse": "COMPUTE_WH",  
    "database": "LUNAPARKMADHOUSE",  
    "schema": "PUBLIC",  
    "role": "ACCOUNTADMIN"  
}
```

**Важно:** Паролите и чувствителната информация трябва да се съхраняват безопасно и да не бъдат включвани в кода.

## Стартиране на приложението

За да стартирате API-то, изпълнете следната команда:

```
python app.py
```

Приложението ще бъде достъпно на <http://127.0.0.1:5000/>.

## Крайни точки (Endpoints)

### Главна страница

- **GET /**
  - Връща съобщение "Welcome to the Amusement Park API!"

## Атракции (Attractions)

- **GET /attractions**
  - Връща списък с всички атракции.
- **GET /attractions/<int:id>**
  - Връща информация за атракция по ID.

## Служители (Employees)

- **GET /employees**
  - Връща списък с всички служители.
- **GET /employees/<int:id>**
  - Връща информация за служител по ID.

## Билети (Tickets)

- **GET /tickets**
  - Връща списък с всички билети.
- **GET /tickets/<int:id>**
  - Връща информация за билет по ID.

## Събития (Events)

- **GET /events**
  - Връща списък с всички събития.
- **GET /events/<int:id>**
  - Връща информация за събитие по ID.

## Посетители (Visitors)

- **GET /visitors**
  - Връща списък с всички посетители.
- **GET /visitors/<int:id>**
  - Връща информация за посетител по ID.

## Финансови транзакции (Financial Transactions)

- **GET /financial**
  - Връща списък с всички финансови транзакции.
- **GET /financial/<int:id>**
  - Връща информация за финансова транзакция по ID.

## Обработка на грешки

- **404 Not Found** – ако заявеният ресурс не съществува.
- **500 Internal Server Error** – ако възникне вътрешна грешка на сървъра.

Примери за грешки:

```
{
  "error": "Not found"
}
{
  "error": "Internal server error"
}
```

## Бележки

- **Свързването с базата данни Snowflake** използва функцията `get_snowflake_session()`.
- **Извличането на данни** става чрез `fetch_results(query, params=None)`.
- **Заявките използват SQL**, което изисква внимание към SQL инжектиране и сигурност.
- **Приложението е в debug режим** – не се препоръчва за продукционна среда.

## Разширяване на функционалността

За разширяване на API-то, можете да добавите следните функции:

- **Добавяне, редактиране и изтриване** на записи в базата данни.
- **Филтриране и сортиране** на резултатите по различни критерии.
- **Автентикация и авторизация** на потребителите.
- **Логване и мониторинг** на заявките.

## Заклучение

Това API осигурява основна функционалност за извличане на информация от базата данни на увеселителния парк. За разширение на системата могат да бъдат добавени нови крайни точки и механизми за обработка на данни.