

Контекст

Кинотеатр «Искорка» показывает фильмы для детей и подростков.

У кинотеатра есть онлайн-расписание и онлайн-продажа билетов на свои сеансы через систему бронирования.

Информация о забронированных билетах вносится в базу бронирования с указанием данных покупателя. Все данные о сеансах и бронировании хранятся и обрабатываются на одном сервере.

Нужно автоматизировать решение задач:

1. Проверить наличие мест на сеанс.
2. Забронировать место.
3. Отменить бронь.
4. Изменить время сеанса.

Так как система построена на микросервисной архитектуре, для решения задачи нужно спроектировать API.

Задание

Подготовьте данные для проектирования API, отвечая на вопросы ниже. Опишите структуру хранения данных и представьте информацию в таблице.

1. Таблица сеансов “seating a sessions”

Техническое наименование	Описание	Тип данных
idsession	сеанс	int
idseat	место сеанса	int
name_film	название фильма	string
datetime_session	дата и время сеанса	string, format: datetime
status_session	статус места	string, enum: Свободно, Занято.

2. Таблица бронирования “booking”

Техническое наименование	Описание	Тип данных
idsession	сеанс	int
idseat	место сеанса	int
customer_name	имя клиента	string (100)

client_email	электронная почта клиента	string (100)
datetime	дата и время бронирования	string, format: datetime
idbooking	наличие брони	boolean
id_card	наличие бонусной карты	int

3. Таблица бонусная карта “card_report” (новая)

Техническое наименование	Описание	Тип данных
idsession	сеанс	int
id_card	наличие бонусной карты	int
id_paument	платеж	int
sum_paument	сумма платежа	number
datetime_paument	дата и время платежа	string, format: datetime
parsent_paument_ball	размер платежа баллами в %	int
count_ball	количество бонусных баллов	int
begin_date	Дата начала отчёта	string, format: datetime
end_date	Дата окончания отчёта	string, format: datetime

Вопросы для решения задания:

1. Кто будет взаимодействовать с API?
 - Система бронирования билетов;
 - клиент.
2. Какие действия будет выполнять API?

- Get получить информацию о наличии мест на сеанс из `www.serverC/seating a sessions`.
 - Post добавить запись о бронировании места `www.serverC/booking`.
 - Delete удаление брони `www.serverC/booking`.
 - Put обновление информации об изменении времени сеанса `www.serverC/seating a sessions`.
3. Какие методы HTTP-запроса будет использовать API?
 - Get, Post, Put, Delete.
 4. К каким ресурсам будет обращаться API?
 - `www.serverC/sessions`,
 - `www.serverC/booking`.
 5. Какие параметры запроса будут передаваться в каждый из методов?

5.1 Get получить информацию о наличии мест на сеанс из `www.serverC/seating a sessions`.

Запрос:

`idsession`

`status_session = 'Свободно'`

Ответ:

`idseat`

5.2 Post добавить запись о бронировании места `www.serverC/booking`.

Запрос:

`idsession`

`idseat`

`customer_name`

`client_email`

`datetime`

Ответ: '200' Успех

`idbooking`

5.3 Delete удаление брони `www.serverC/booking`.

Запрос:

`idbooking`

Ответ:

'200' Успех

5.4 Put обновление информации об изменении времени сеанса `www.serverC/seating a sessions`.

Запрос:

`idsession`

`datetime`

Ответ:

'200' Успех

'404' Не найдена запись

'500' Сервер не доступен

`datetime`

Итоговая работа:

1. В Таблицу бронирования “booking” вносим доп.информацию id_card - наличие бонусной карты.

2. Создаём новую Таблицу бонусная карта “card_report”.

3. Добавляем методы Get и Post

4. Описание добавленных методов:

4.1 Post Добавляем запись о наличии бонусной карты из booking

Запрос:

idsession

idseat

customer_name

client_email

datetime

idbooking

Ответ:

‘200’ Успех

id_card.

4.2 Get Получить список всех бронирований по бонусной карте из www.serverC/booking

Запрос:

id_card

Ответ:

‘200’ Успех

idsession

datetime

4.3 Get Получить информацию о всех платежах по бонусной карте и % оплат баллами из www.serverC/card_report.

Запрос:

id_card

begin_date

end_date

Ответ:

‘200’ Успех

id_paument

parsent_paument_ball

sum_paumen