

Шаг 2: определение структуры документов и связей

Немного теории

1. Структура документа

- **Определение полей:** Каждый документ в MongoDB должен иметь определенные поля, которые описывают его атрибуты. Например, документ `Клиент` может иметь поля `ID клиента`, `имя`, `номер телефона`, `email`, `адрес`.
- **Типы данных:** Важно определить тип каждого поля (например, строка, число, дата) для правильной обработки и хранения данных.

2. Вложенные документы и массивы

- **Использование массивов:** Массивы используются для хранения списка элементов в одном поле документа. Например, можно хранить историю бронирований клиента в виде массива вложенных документов непосредственно в документе `Клиент`. Пример: `"бронирования": [{ "ID брони": 123, "дата заезда": "2021-07-01", "дата выезда": "2021-07-05", "статус": "подтверждено" }, { "ID брони": 124, "дата заезда": "2021-08-01", "дата выезда": "2021-08-05", "статус": "отменено" }]`
- **Вложенные документы:** Вложенные документы позволяют хранить сложные структуры данных. Например, каждый элемент в массиве `бронирования` выше является вложенным документом.

3. Ссылки на другие документы

- **Использование ссылок:** Вместо вложения всех данных, документы могут содержать ссылки на другие документы. Например, в документе `Бронь`, поле `ID номера` может быть ссылкой на документ в коллекции `Номера`.

- Преимущества ссылок: Это позволяет отделить и независимо обновлять данные, которые могут часто меняться или не всегда требуются при каждом запросе. Например, информация о номере отеля может храниться отдельно, и при необходимости ее можно извлекать, используя ID номера из документа `Бронь`.

Структура документов JSON для MongoDB

***внимание! про типы данных мы уже здесь говорить не будем, так как подробно обсуждали это в разделе про реляционные базы данных. тут вы используете те же принципы, чтобы выбрать типы данных для каждого поля, но используете типы данных не PostgreSQL, а JSON(BSON). При определении структуры данных для MongoDB, вы обычно начинаете с определения формата данных в виде JSON, а затем MongoDB автоматически преобразует эти структуры в BSON при сохранении в базе данных. <https://www.mongodb.com/basics/bson>

Коллекция `Клиенты`. Документ Клиент: Пример структуры документа для клиента в JSON-формате

```
{
  "ID клиента": "C123456",
  "имя": "Иван Иванов",
  "номер телефона": "123-456-7890",
  "email": "ivanov@example.com",
  "адрес": "г. Москва, ул. Пушкина, д. 1",
  "бронирования": [
    {
      "ID брони": "B123",
      "дата заезда": "2023-01-01",
      "дата выезда": "2023-01-05",
      "статус": "подтверждено",
      "ID номера": "N101"
    },
    {
      "ID брони": "B124",
      "дата заезда": "2023-02-01",
      "дата выезда": "2023-02-05",
      "статус": "отменено",
      "ID номера": "N102"
    }
  ]
}
```

Коллекция **Номера**. Документ Номер: Пример структуры документа для номера в JSON-формате

```
{
  "ID номера": "N101",
  "ID гостиницы": "H100",
  "тип номера": "люкс",
  "стоимость": 5000,
  "статус доступности": "доступен"
}
```

Взаимосвязь между документами и коллекциями

- В коллекции **Клиенты** , каждый элемент массива **бронирования** содержит **ID номера** , который является ссылкой на соответствующий документ в коллекции **Номера** .

- Это позволяет при необходимости извлекать детальную информацию о номере, используя его ID, сохраненный в бронировании клиента.