

Документные базы данных

Документные СУБД

- Основной концепцией в документных базах данных является документ. База данных хранит и извлекает документы в форматах XML, JSON, BSON и т.д ..
- Эти документы представляют собой самоописываемые иерархические древовидные структуры данных, которые могут состоять из ассоциативных массивов, коллекций и скалярных значений.
- хранилища типа "ключ-значение", в которых значение допускает проверку

Назначение документных баз

- Интеграция разрозненных данных, обеспечивающая их единое представление
- Интернет вещей
- Мобильные приложения
- Аналитика в реальном времени
- Персонализация
- Управление каталогом
- Управление контентом (Сервисы блогосферы — особенно те, которые подразумевают большое количество изображений, аудио и видеоматериалов.)
- ~~Системы электронного документооборота~~

Перечень документных баз данных

- Amazon DynamoDB.(AWS SDK)
- MongoDB.
- MongoDB Atlas.
- Couchbase Server.(язык **N1QL**)
- Percona Server for MongoDB.
- Google Cloud Firestore.
- InterSystems IRIS.
- ArangoDB
- Microsoft Azure Cosmos DB
- Firebase Realtime Database

Различия в понятиях

Реляционная база	MongoDB
схема /база данных	база данных
таблица	коллекция
строка	документ
идентификатор	_id
join	DBRef

Отличия документных баз данных от реляционных

- Единица работы не таблицы, а составной объект
- У каждой СУБД свой язык для работы
- Большая часть документных баз работает с форматом json
- Документные БД разрабатывалась не как монолитный сервер, а с прицелом на горизонтальное масштабирование, распределенную систему. В проект заложена согласованность данных и устойчивость к частичной потере связности сети. Однако у сегментирования есть свои минусы: если какая-то часть коллекции потеряна, то и вся коллекция теряет ценность.

MongoDB

Документация

- <https://docs.mongodb.com/manual/reference/database-references/>

Типы данных

- **Date**
- **Timestamp**
- **ObjectId**
- **Double**
- **Int32**
- **Long (64-bit)**
- **Decimal128** (с плавающей запятой 34 значащих цифры мантиссы, порядок в диапазоне –6143 to +6144)
- **Null**
- **String**
- **Boolean**
- **Array**
- <https://www.mongodb.com/docs/manual/reference/bson-types/>

Перечисление

- db.createCollection("shipping", {
- validator: {
- \$jsonSchema: {
- bsonType: "object",
- title: "Shipping Country Validation",
- properties: {
- country: {
- enum: ["France", "United Kingdom", "United States"],
- description: "Must be either France, United Kingdom, or United States"
- }
- }
- }
- }
- }
- })

JSON/BSON

```
> printjson( db.towns.findOne({"_id" : ObjectId("4d0b6da3bb30773266f39fea")} ) )
{
  "_id" : ObjectId("4d0b6da3bb30773266f39fea"),
  "country" : {
    "$ref" : "countries",
    "$id" : ObjectId("4d0e6074deb8995216a8309e")
  },
  "famous_for" : [
    "beer",
    "food"
  ],
  "last_census" : "Thu Sep 20 2007 00:00:00 GMT-0700 (PDT)",
  "mayor" : {
    "name" : "Sam Adams",
    "party" : "D"
  },
  "name" : "Portland",
  "population" : 582000,
  "state" : "OR"
}
```

Коллекция

База данных

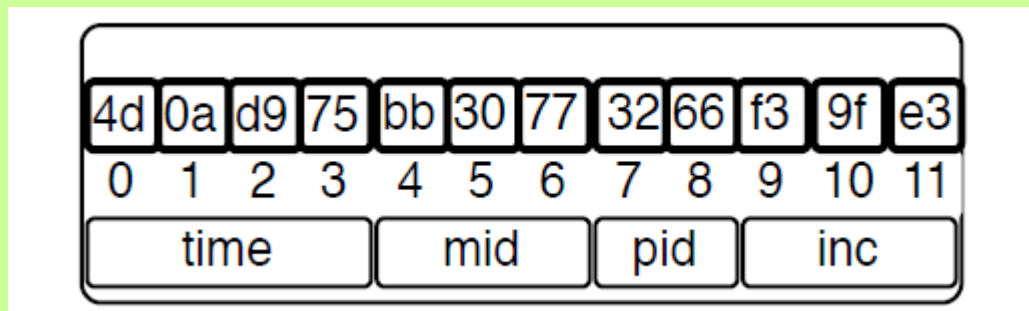
Идентификатор

Документ

Идентификатор объекта

Объект ObjectId занимает 12 байтов и состоит из

- временной метки (Time) 4 байта
- идентификатора клиентской машины (mid: machine identifier) 3 байта
- идентификатора клиентского процесса (pid process identifier) 2 байта
- инкрементируемого счетчика (inc) 3 байта



Связь между документами

- Ручные ссылки, когда вы сохраняете поле `_id` одного документа в другом документе в качестве ссылки. Затем ваше приложение может выполнить второй запрос, чтобы вернуть соответствующие данные.
- DBRefs — это ссылки из одного документа в другой, использующие значение поля `_id` первого документа, имя коллекции и, необязательно, имя его базы данных.
- Включая эти имена, DBRefs позволяет легко связать документы, расположенные в нескольких коллекциях, с документами из одной коллекции. Чтобы определить DBRefs, ваше приложение должно выполнить дополнительные запросы, чтобы вернуть документы, на которые есть ссылки. Многие драйверы имеют вспомогательные методы, которые автоматически формируют запрос для DBRef.
- <https://www.mongodb.com/docs/manual/reference/database-references/>

DBRefs

- { "\$ref" : <value>, "\$id" : <value>, "\$db" : <value> }

- \$ref

Имя коллекции, на в которой документ, на который
ссылаемся

- \$id

Значение поля `_id` документа, на который
ссылаемся

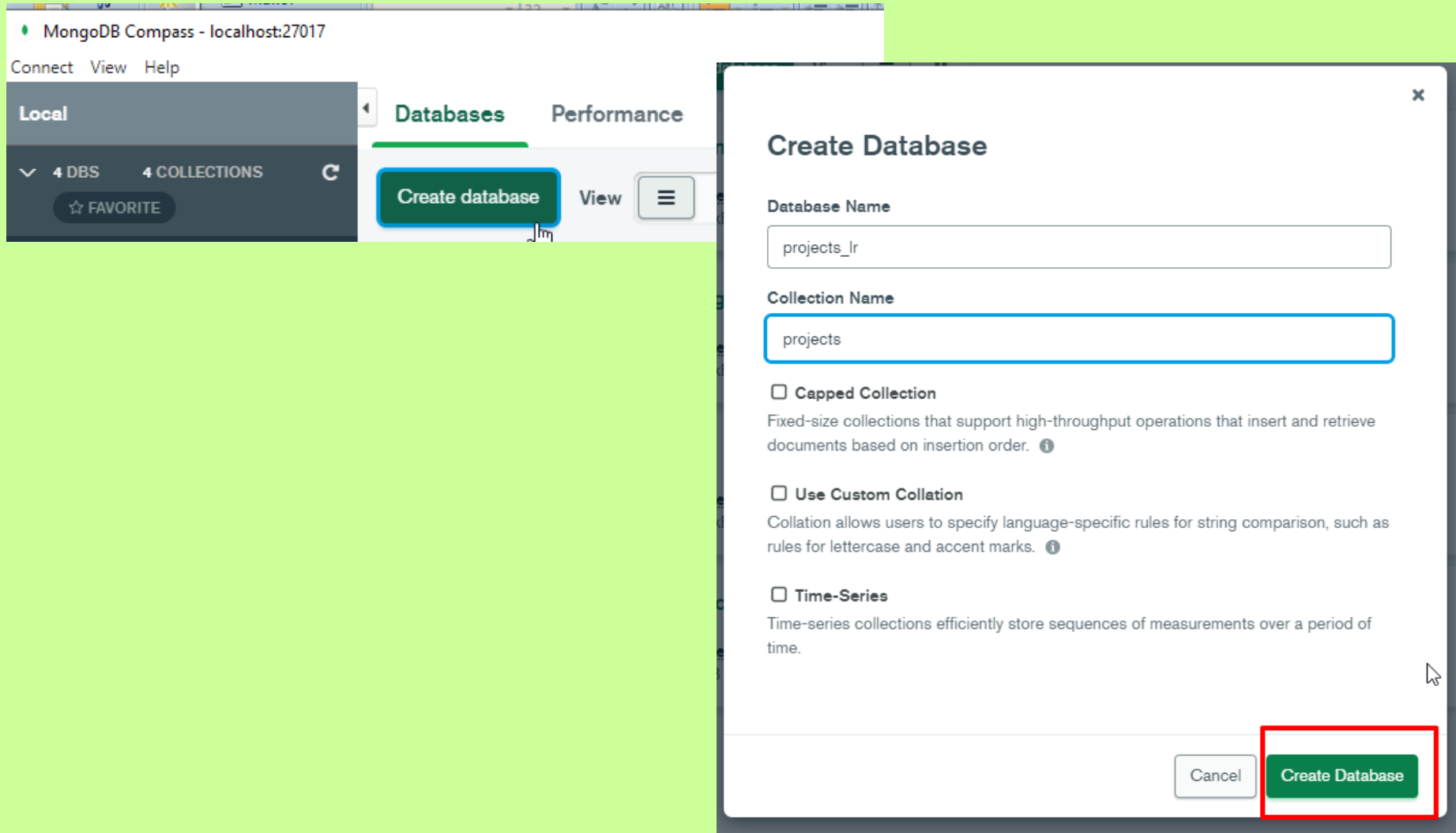
- \$db

Имя базы данных, содержащей документ, на
который ссылаемся. Необязательное

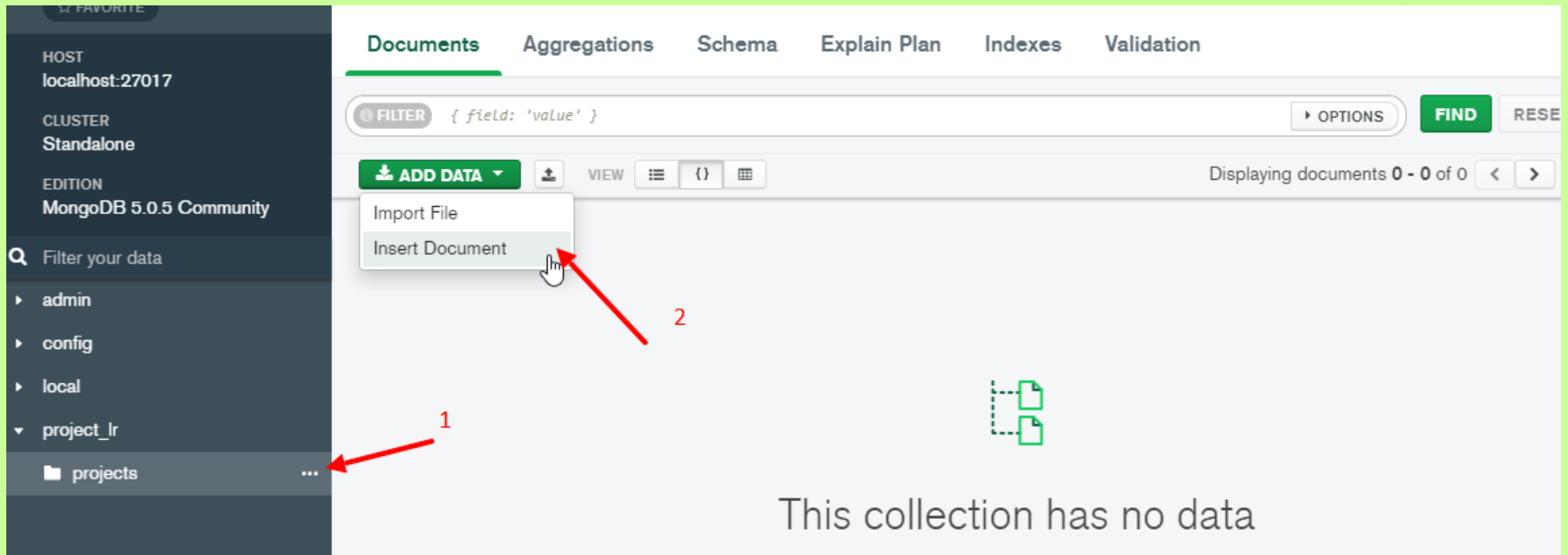
DBRefs

Driver	DBRef Support
C	Не поддерживается
C++	Не поддерживается
C#	Поддерживается
Go	Не поддерживается
Haskell	Не поддерживается
Java	Поддерживается
Node.js	Поддерживается
Perl	Поддерживается
PHP	Не поддерживается
Python	Поддерживается
Ruby	Поддерживается
Scala	Не поддерживается

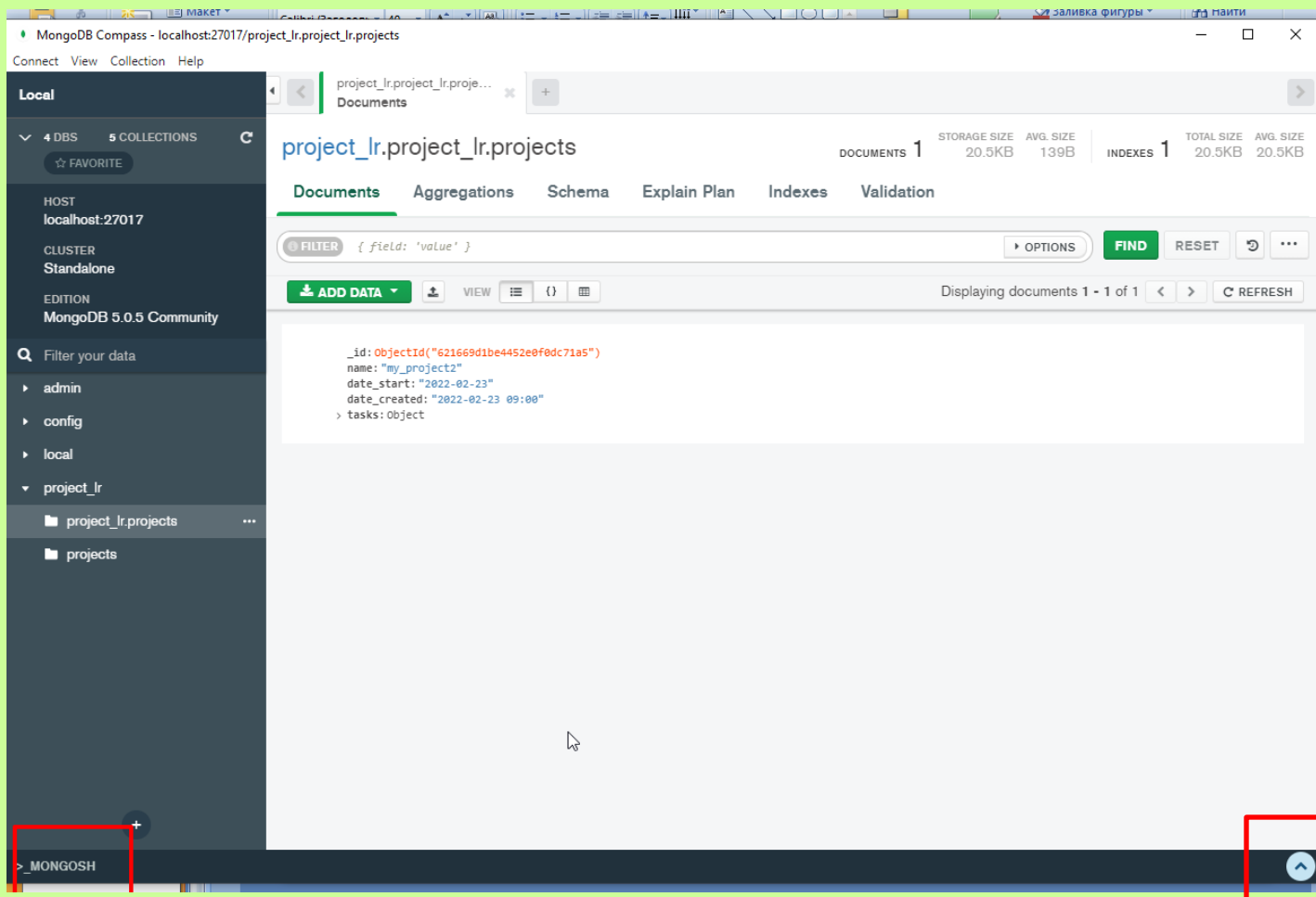
MongoDB Compass



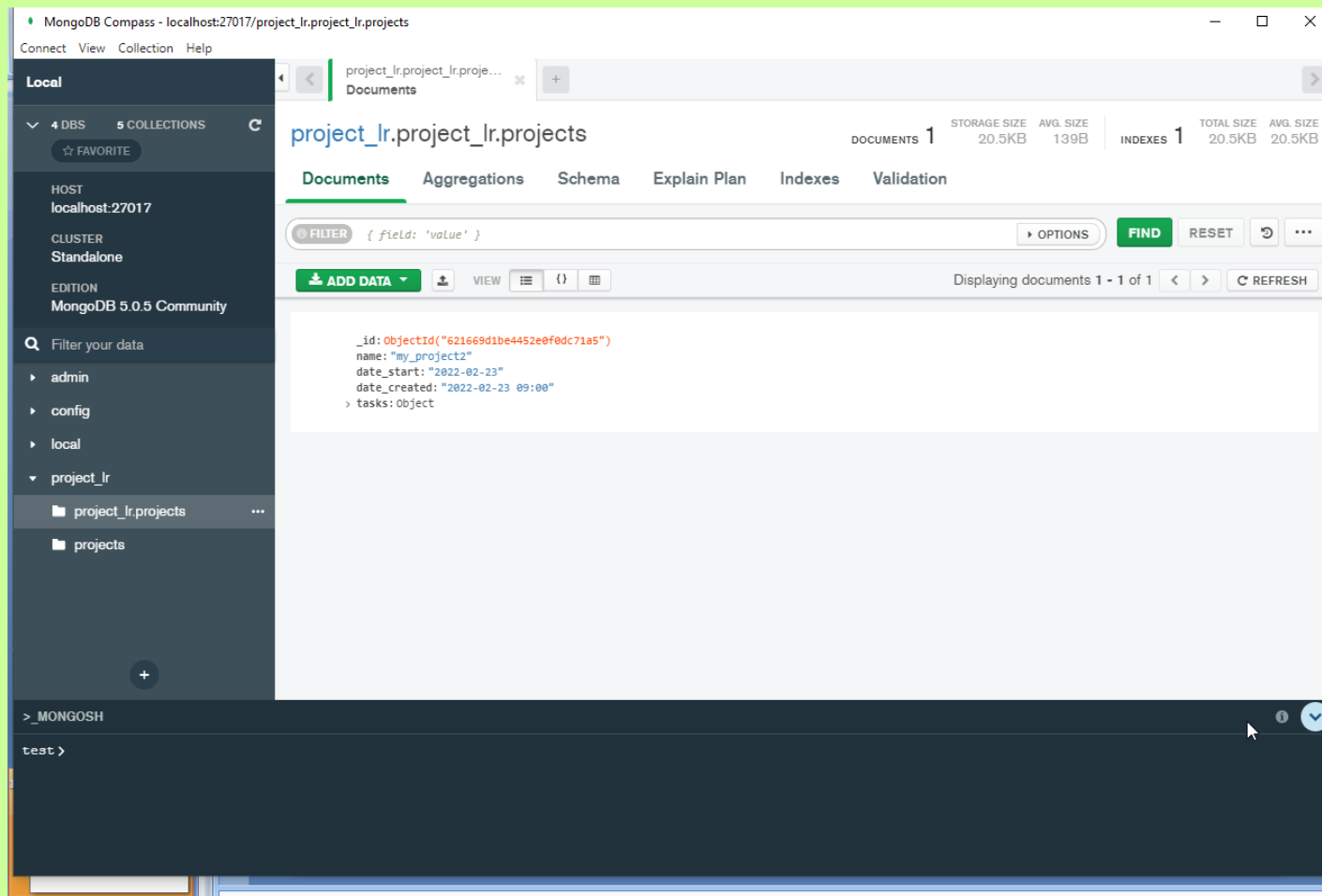
MongoDB Compass



MongoDB Compass оболочка (командная строка)



MongoDB Compass оболочка (командная строка)



Командная строка

- C:\Users\iRU> cd C:\Program Files\MongoDB\Server\5.0\bin

- C:\Program Files\MongoDB\Server\5.0\bin>Mongo

MongoDB shell version v5.0.5

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("0e1a951d-cc0b-45b6-b3ee-220aafefd3f0") }

MongoDB server version: 5.0.5

=====

Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be
removed in
an upcoming release.

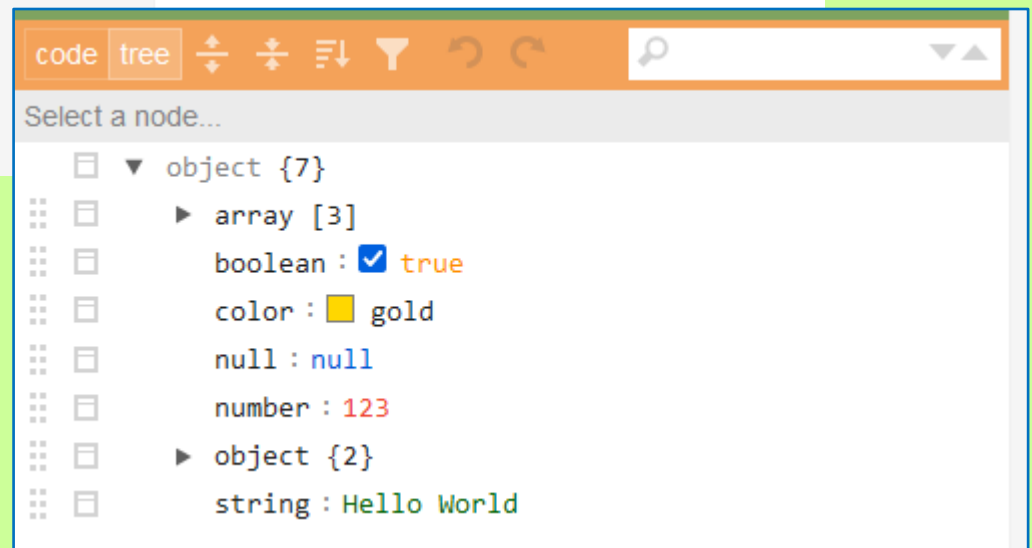
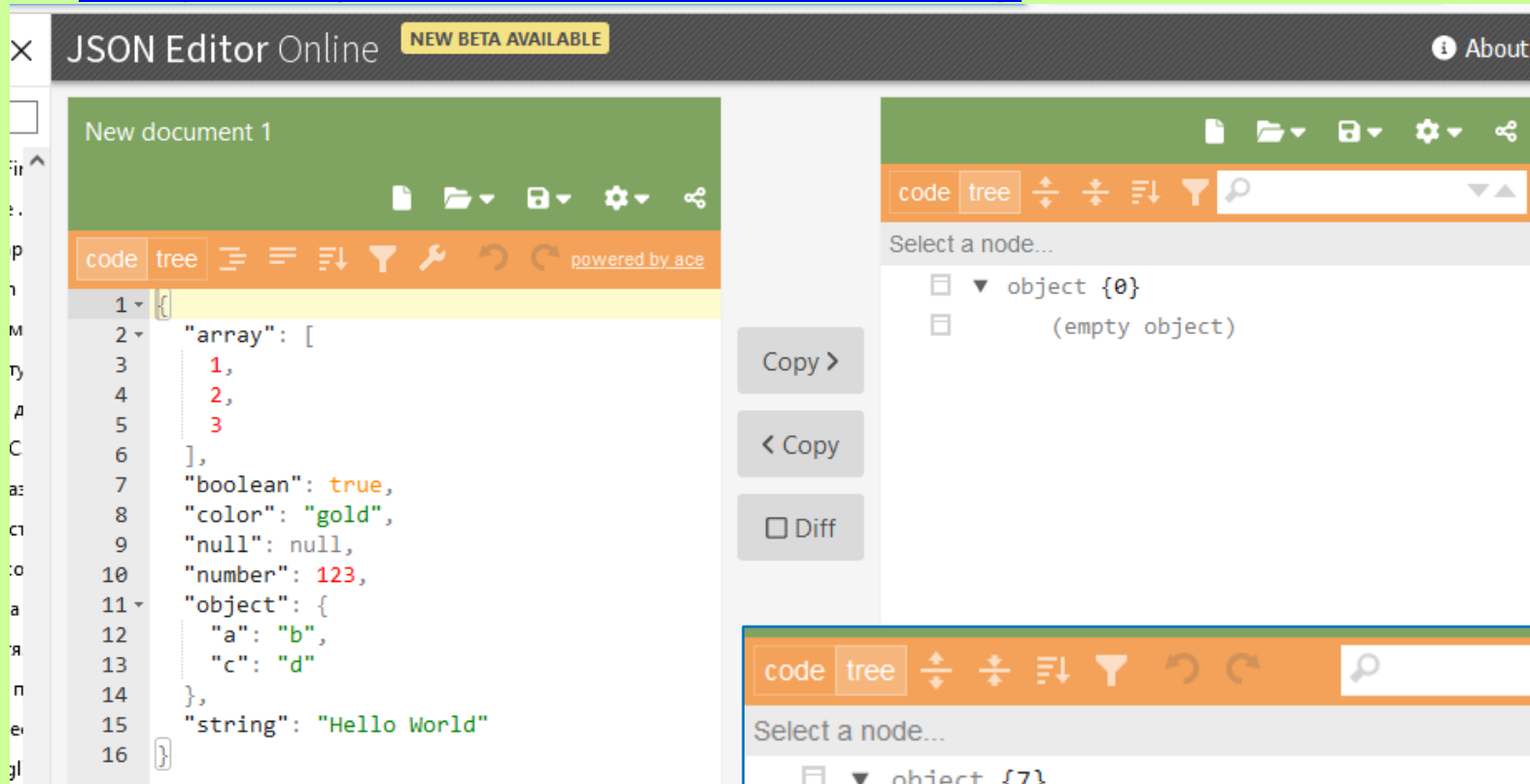
For installation instructions, see

<https://docs.mongodb.com/mongodb-shell/install/>

- Use имя_базы

Визуализация json

- <https://jsoneditoronline.org>



Вставка данных

```
db.projects.insert(  
{  
  "name": "my_project_1",  
  "created": "2022-02-24T9:00",  
  "started": "2022-02-24T9:00",  
  "ended": null,  
  "status": "создан",  
  "tasks": [  
    {  
      "_id": ObjectId(),  
      "task_name": "анализ требований",  
      "created": "2022-02-24T9:10",  
      "started": "2022-02-24T9:20",  
      "ended": "2022-03-24T18:20",  
      "is_done": false,  
      "executors": []  
    }  
  ]  
})
```

```
db.projects.find();  
{ _id: ObjectId("621796f4e04fb371ebbf032d"),  
  name: 'my_project_1',  
  created: '2022-02-24T9:00',  
  started: '2022-02-24T9:00',  
  ended: null, status: 'создан',  
  tasks: [ { _id: ObjectId("621796f4e04fb371ebbf032c"),  
    task_name: 'анализ требований',  
    created: '2022-02-24T9:10',  
    started: '2022-02-24T9:20',  
    ended: '2022-03-24T18:20',  
    is_done: false,  
    executors: [] }  
  ]  
}
```

Вставка данных

```
db.projects.insert(
{ "name": "my_project_11",
  "created": Date(),
  "started": Date(),
  "ended": null,
  "status": "создан",
  "tasks": [
{ "_id": ObjectId(),
  "task_name": "анализ требований",
  "created": Date(),
  "started": ISODate("2022-02-24"),
  "ended": new Date("2022-03-24"),
  "is_done": false,
  "executors":[]
}
]
})
```

```
db.projects.find({ "_id" : ObjectId("62184dac67d60765a42c12e1") })
{ _id: ObjectId("62184dac67d60765a42c12e1"),
  name: 'my_project_11',
  created: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
  started: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
  ended: null, status: 'создан',
  tasks: [ { _id: ObjectId("62184dac67d60765a42c12e0"),
  task_name: 'анализ требований',
  created: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
  started: 2022-02-24T00:00:00.000Z,
  ended: 2022-03-24T00:00:00.000Z,
  is_done: false, executors: [] } ]
}
```

Вставка в Массивы

- { **\$push**: { <field1>: { <modifier1>: <value1>, ... }, ... } }

\$each	Добавляет множество значений к полю массива
\$slice	Ограничивает число элементов массива, требует использование \$each
\$sort	Сортирует элементы массива, требует использование \$each
\$position	Указывает место в массиве, в которое нужно вставить новые элементы. Требуется использование модификатора \$each. Без модификатора \$position \$push добавляет элементы в конец массива.

Вставка в Массивы

- { **\$push**: { <field1>: { <modifier1>: <value1>, ... }, ... } }
- db.students.insertOne({ _id: 1, scores: [44, 78, 38, 80] })
- db.students.updateOne(
 { _id: 1 },
 { \$push: { scores: 89 } }
)
- db.students.updateOne(
 { name: "joe" },
 { \$push: { scores: { \$each: [90, 92, 85] } } }
)

Вставка задания

- `db.projects.update({_id:ObjectId("62184dac67d60765a42c12e1")},{ $push: {`
- `'tasks':{_id:ObjectId(), "task_name":`
`"пользовательский интерфейс",`
- `"created": Date(),`
- `"started": ISODate("2022-03-24"),`
- `"ended": ISODate("2022-04-24"),`
- `"is_done":false,`
- `"executors":[]}}})`

Вставка задания

```
db.projects.find({_id:ObjectId("62184dac67d60765a42c12e1")})
{ _id: ObjectId("62184dac67d60765a42c12e1"),
  name: 'my_project_11',
  created: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
  started: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
  ended: null, status: 'создан',
  tasks: [
    { _id: ObjectId("62184dac67d60765a42c12e0"),
      task_name: 'анализ требований',
      created: 'Fri Feb 25 2022 06:31:56 GMT+0300 (Москва, стандартное время)',
      started: 2022-02-24T00:00:00.000Z,
      ended: 2022-03-24T00:00:00.000Z,
      is_done: false,
      executors: [] },
    { _id: ObjectId("621859df67d60765a42c12e3"),
      task_name: 'пользовательский интерфейс',
      created: 'Fri Feb 25 2022 07:23:59 GMT+0300 (Москва, стандартное время)',
      started: 2022-03-24T00:00:00.000Z,
      ended: 2022-04-24T00:00:00.000Z,
      is_done:
false, executors: [] } ] }
```

Ручные ссылки

- `original_id = ObjectId()`
- `db.places.insertOne(`
- `{ "_id": original_id, "name": "Broadway Center",`
`"url": "bc.example.net" })`
- `db.people.insertOne(`
- `{ "name": "Erin",`
- `"places_id": original_id,`
`"url": "bc.example.net/Erin" })`

Вставка данных

```
db.employees.insert(
{"surname":"Иванов",
"name":"Иван",
"patronym":"Иванович" } )
{acknowledged: true, insertedIds: { '0': ObjectId("6218506a67d60765a42c12e2") } }
```

```
db.projects.find({ "tasks._id" : ObjectId("621890e606ae8feffc22a50a") })
```

```
db.projects.update({"tasks._id":ObjectId("621890e606ae8feffc22a50a")},{ $push: {
"tasks.$.executors":ObjectId("62179acce04fb371ebbf032e")}})
```

```
db.projects.find({ "tasks._id" : ObjectId("621890e606ae8feffc22a50a") })
{ _id: ObjectId("621796f4e04fb371ebbf032d"),
  name: 'my_project_1',
  created: '2022-02-24T9:00',
  started: '2022-02-24T9:00',
  ended: null,
  status: 'создан',
  tasks: [
    { _id: ObjectId("621796f4e04fb371ebbf032c"),
      task_name: 'анализ требований',
      created: '2022-02-24T9:10',
      started: '2022-02-24T9:20',
      ended: '2022-03-24T18:20',
      is_done: false,
      executors: [ ObjectId("62179acce04fb371ebbf032e") ]
    }, { _id:
    ObjectId("621890e606ae8feffc22a50a"),
      task_name: 'пользовательский интерфейс',
      created: 'Fri Feb 25 2022 11:18:46 GMT+0300 (Москва, стандартное время)',
      started: 2022-03-24T00:00:00.000Z,
      ended: 2022-04-24T00:00:00.000Z,
      is_done: false,
      executors: [ ObjectId("62179acce04fb371ebbf032e") ] ] ] }
```

```
db.employees.findOne({ name : /^И/ })
{ _id: ObjectId("6218506a67d60765a42c12e2"), surname: 'Иванов', name: 'Иван', patronym: 'Иванович' }
db.employees.findOne({ name : /^и/ })
null
db.employees.findOne({ name : /в/ })
{ _id: ObjectId("6218506a67d60765a42c12e2"), surname: 'Иванов', name: 'Иван', patronym: 'Иванович' }
db.employees.findOne({ name : /н^/ })
```

- db.employees.findOne({ name : /^И/ })
- { _id: ObjectId("6218506a67d60765a42c12e2"), surname: 'Иванов', name: 'Иван', patronym: 'Иванович' }
- db.employees.findOne({ name : /^и/ })
- null
- db.employees.findOne({ name : /в/ })
- { _id: ObjectId("6218506a67d60765a42c12e2"), surname: 'Иванов', name: 'Иван', patronym: 'Иванович' }
- db.employees.findOne({ name : /н^/ })