



WELCOME

MongoDB for VSCode

Be more productive when you build applications



Alena Khineika

Senior Software Engineer

alenakhineika

MongoDB + VSCode + Berlin

```
db.collection.aggregate([ ... ])
```

vscode (Git)

Commit Pull Push Fetch Branch Merge

WORKSPACE File status History Search BRANCHES TAGS REMOTES STASHES SUBMODULES SUBTREES

New Connection ★ Favorites

New Connection

Preview README.md — color-of-berlin-palette

mongosh mongoDB://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Last login: Sat Sep 25 19:08:33 on ttys003
alenakhineika@Alena-MacBook-Pro ~ % mongosh
Current Mongosh Log ID: 615085e9f9af4c2d24367e9
Connecting to: mongoDB://127.0.0.1:27017/?directCon...
Using MongoDB: 5.0.2
Using Mongosh: 1.0.1

For mongosh info see: <https://docs.mongodb.com/mongodb-shell>

The server generated these startup warnings when booting:
2021-09-24T14:54:20.065+02:00: Access control is not enabled.
The server is bound to localhost. Start the server with --bind_ip <address> to specify which interface to bind to all interfaces. If this behavior is desired, set the --allowInvalidOptions flag.

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/mongosh Enterprise test>

MongoDB Compass Dev - Connect

Fill in connection fields individually

Save a cluster? After you save a cluster, you can create flags.

Preview README.md — color-of-berlin-palette

Welcome to the MongoDB Documentation

Find the guides, samples, and references you need to use the database, visualize data, and build applications on the MongoDB data platform.





MongoDB for VS Code - Visual Studio Marketplace

marketplace.visualstudio.com/items?itemName=mongodb.mongodb-vsco...

MongoDB for VS Code Preview

MongoDB | 133,266 installs | ★★★★★ (17) | Free

Connect to MongoDB and Atlas directly from your VS Code environment; navigate your databases and collections, inspect your schema and use playgrounds to prototype queries and aggregations.

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

MongoDB for VS Code PREVIEW

Azure Pipelines succeeded

MongoDB for VS Code makes it easy to work with MongoDB, whether your own instance or in [MongoDB Atlas](#).

Features

Navigate your MongoDB Data

- Navigate your databases, collections, and read-only views
- See the documents in your collections
- Edit documents and save changes to the database
- Get a quick overview of your schema and your indexes

Categories

Programming Languages Snippets

Tags

Databases json keybindings MongoDB MongoDB Atlas MQL snippet terraform

Resources

Issues Repository Homepage License Changelog Download Extension

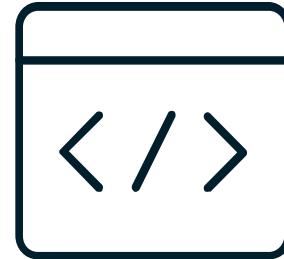


Development



MangoDB

The most natural way
to store data



VSCode

The most popular
development environment



Color of Berlin (@colorofberlin) x +

twitter.com/colorofberlin

Color of Berlin
14.7K Tweets

Following

Color of Berlin
@colorofberlin

The color of the sky in Berlin. Die Farbe des Himmels in Berlin. CURRENTLY UNDER CONSTRUCTION. 🚧🚧👷

② Berlin, Deutschland ⌂ github.com/laurendorman/c... Joined April 2020

1 Following 1,969 Followers

Followed by ira. and Lauren Dorman

Tweets Tweets & replies Media Likes

Pinned Tweet

Color of Berlin @colorofberlin · Mar 9

...



Webcam vom Wetterturm in Be ✕ +

Not Secure | met.fu-berlin.de/de/wetter/webcam/

Institut für Meteorologie

veritas iustitia libertas

Home | Lageplan | Kontakt | E-Mail/Tel. | a-z Index | Intern

Freie Universität Berlin, Institut für Meteorologie Ort: Schmidt-Ott-Str.13 (pi2) Zeit (GMT) : 2021-05-04 20:56:00

Studium

Veranstaltungen

Bibliothek

Wetterinfos

Wetterpate

Wetterdaten

Satellitendaten

Jobs

Wetterkarte

Prognoseturnier

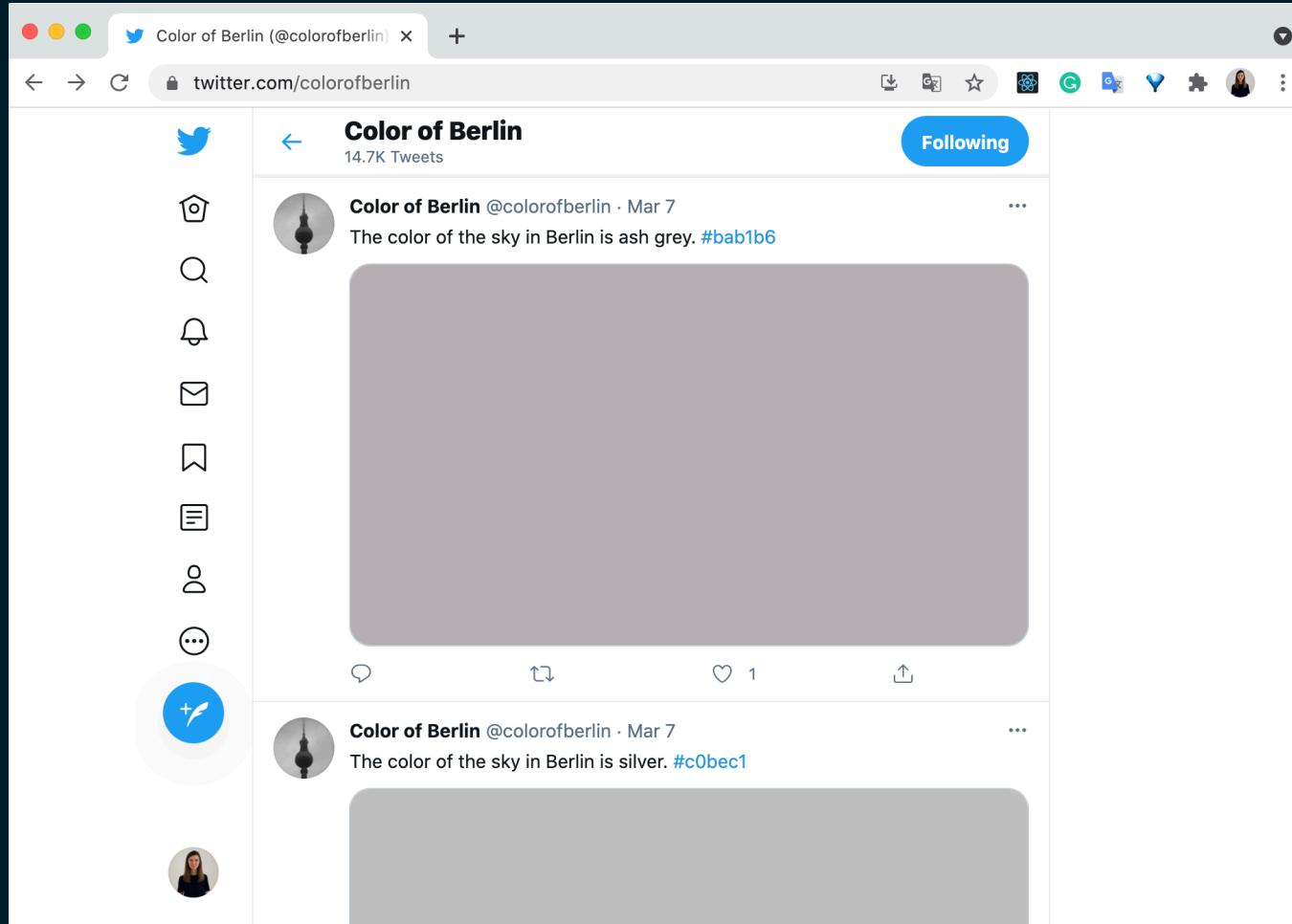
Luftgüte

DMG

Institut für Meteorologie der Freien Universität Berlin
Blick vom Wetterturm des Instituts auf dem Fichtenberg (Steglitz)
in Richtung Nord-West mit dem Funkturm im Mittelbereich

04.05.2021 22:56 MESZ
www.wind-berlin.de

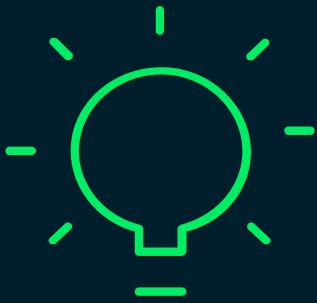
AKTUELLE Pollensaison
Hier erhalten Sie





Berlin





Features

WORK WITH MONGODB DIRECTLY WITHIN YOUR CODING ENVIRONMENT



MongoDB — color-of-berlin-palette

MONGODB ... MongoDB X ...

CONNECTIONS

No connections found.

Add Connection

8

Not connected.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

New to MongoDB and don't have a cluster?
Create one for free using [MongoDB Atlas](#).

Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45 ✓ ESLint ↗ ↘





mongodb://username:password@host:port/database?options

MongoDB — color-of-berlin-palette

MONGODB

>mongodb: connect with|

CONNECTIONS

MongoDB: Connect with Connection String... recently used

No connections found.

Add Connection

mongoDB.

Navigate your databases and collections, use playgrounds for exploring and transforming your data

Resources

Not connected.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

New to MongoDB and don't have a cluster?
Create one for free using [MongoDB Atlas](#).

Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45

✓ ESLint

MongoDB — color-of-berlin-palette

e.g. `mongodb+srv://username:password@cluster0.mongodb.net/admin`

Enter your connection string (SRV or standard) (Press 'Enter' to confirm or 'Escape' to cancel)

MONGODB

CONNECTIONS

No connections found.

Add Connection

mongoDB

Navigate your databases and collections, use playgrounds for exploring and transforming your data

Resources

Not connected.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

New to MongoDB and don't have a cluster?
Create one for free using [MongoDB Atlas](#).

Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45

✓ ESLint



MongoDB — color-of-berlin-palette

MONGODB

CONNECTIONS

No connections found.

Add Connection

MongoDB connection strings begin with "mongodb://" or "mongodb+srv://"

mongoDB.

Navigate your databases and collections, use playgrounds for exploring and transforming your data

Resources

Not connected.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

New to MongoDB and don't have a cluster?
Create one for free using [MongoDB Atlas](#).

Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45

✓ ESLint

MongoDB — color-of-berlin-palette

MONGODB

CONNECTIONS

No connections found.

Add Connection

mongodb://localhost

Enter your connection string (SRV or standard) (Press 'Enter' to confirm or 'Escape' to cancel)

mongoDB

Navigate your databases and collections, use playgrounds for exploring and transforming your data

Resources

Not connected.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

New to MongoDB and don't have a cluster?
Create one for free using [MongoDB Atlas](#).

Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45

✓ ESLint

MongoDB — color-of-berlin-palette

MongoDB X

mongoDB.

Navigate your databases and collections, use playgrounds for exploring and transforming your data

Connected to: localhost:27017

All set. Ready to start?
Create a playground.

Connect with Connection String

Advanced Connection Settings

Connect

Open form

Cmd + Shift + P for all MongoDB Command Palette options

MONGODB

CONNECTIONS

localhost:27017 connected

- > admin
- > archive
- > colorofberlin
- > tweets
 - > Documents 5K
 - > Schema
 - _id
 - contributors
 - coordinates
 - created_at
 - entities
 - extended_entities
 - favorite_count
 - favorited
 - geo
 - id
 - id_str
 - in_reply_to_screen_name
 - in_reply_to_status_id
 - in_reply_to_status_id_str
 - in_reply_to_user_id
 - Show more fields...
- > Indexes
 - > _id_

PLAYGROUNDS

HELP AND FEEDBACK

master* 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45

✓ ESLint

The screenshot shows the MongoDB Playground interface. On the left, there's a sidebar with various icons: a file icon (1), a magnifying glass, a user icon (8), a document icon, a leaf icon, a person icon, and a gear icon. The main area has tabs for 'MONGODB' and 'MongoDB'. The 'MongoDB' tab is active, displaying a code editor with the following content:

```
// MongoDB Playground • Untitled-1 — color-of-berlin-palette
// MongoDB Playground Untitled-1 1 ●

Currently connected to localhost:27017. Click here to change connection.

1 // MongoDB Playground
2 // Use Ctrl+Space inside a snippet or a string literal to trigger completions.
3
4 // The current database to use.
5 use('colorofberlin');

6
7 // Create a new index in the collection.
8 db.getCollection('tweets')
9   .createIndex(
10     {
11       /*
12       * Keys
13       *
14       * Normal index
15       * fieldA: 1, //ascending
16       * fieldB: -1 //descending
17       *
18       * Wildcard index
19       * '$**': 1, //wildcard index on all fields and subfields in a document
20       * 'path.to.field.$**': 1 //wildcard index on a specific field and its subpath
21       *
22       * Text index
23       * fieldA: 'text',
24       * fieldB: 'text'
25       *
26       * Geospatial Index
27       * locationField: '2dsphere'
28       *
29       * Hashed Index
30       * fieldA: 'hashed'
31       */
32     }, {
33   }
34   /*
```

colorofberlin.tweets:{"\$oid":"6035711519ad1107e53d8ee6"}.json — color-of-berlin-palette

MONGODB

CONNECTIONS

- localhost:27017 connected
 - admin
 - archive
 - colorofberlin
 - tweets
 - "6035711519ad1107e53d8ee6"
 - "6035711519ad1107e53d8ee7"
 - "6035711519ad1107e53d8ee8"
 - "6035711519ad1107e53d8ee9"
 - "6035711519ad1107e53d8eea"
 - "6035711519ad1107e53d8eeb"
 - "6035711519ad1107e53d8eec"
 - "6035711519ad1107e53d8eed"
 - "6035711519ad1107e53d8eee"
 - "6035711519ad1107e53d8eef"
 - Show more...
 - Schema
 - _id
 - contributors
 - coordinates
 - created_at
 - entities
 - extended_entities
 - favorite_count

PLAYGROUNDS

HELP AND FEEDBACK

```
// MongoDB Playground Untitled-1 1  {} colorofberlin.tweets:{"$oid":"6035711519ad1107e53d8ee6"}.json > ...
```

```
{  
  "_id": {  
    "$oid": "6035711519ad1107e53d8ee6"  
  },  
  "created_at": "Wed Jan 20 17:15:26 +0000 2021",  
  "id": 1351941419645214700,  
  "id_str": "1351941419645214720",  
  "text": "The color of the sky in Berlin is jet. #2d2e32 https://t.co/BMhsFbCb87",  
  "truncated": false,  
  "entities": {  
    "hashtags": [  
      {  
        "text": "2d2e32",  
        "indices": [39, 46]  
      }  
    ],  
    "symbols": [],  
    "user_mentions": [],  
    "urls": [],  
    "media": [  
      {  
        "id": 1351941418013646800,  
        "id_str": "1351941418013646851",  
        "indices": [47, 70],  
        "media_url": "http://pbs.twimg.com/media/EsMPjonXYAML3Hw.png",  
        "media_url_https": "https://pbs.twimg.com/media/EsMPjonXYAML3Hw.png",  
        "url": "https://t.co/BMhsFbCb87",  
      }  
    ]  
  }  
}
```

master* 0 1↑ 0△1 [TypeScript Importer]: Symbols: 45 Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON ✓ ESLint 🔍 ↻

// MongoDB Playground • Untitled-2 — color-of-berlin-palette

MONGODB

CONNECTIONS

- localhost:27017 connected
 - admin
 - archive
 - colorofberlin
 - tweets
 - Documents 5K
 - "6035711519ad1107e53d8ee6"
 - "6035711519ad1107e53d8ee7"
 - "6035711519ad1107e53d8ee8"
 - "6035711519ad1107e53d8ee9"
 - "6035711519ad1107e53d8eea"
 - "6035711519ad1107e53d8eeb"
 - "6035711519ad1107e53d8eec"
 - "6035711519ad1107e53d8eed"
 - "6035711519ad1107e53d8eee"
 - "6035711519ad1107e53d8eff"

Show more...

Schema

- _id
- contributors
- coordinates
- created_at
- entities
- extended_entities
- favorite_count

PLAYGROUNDS

HELP AND FEEDBACK

```
// MongoDB Playground
// Use Ctrl+Space inside a snippet or a string literal to trigger completions.

// The current database to use.
use('colorofberlin');

// Search for documents in the current collection.
db.getCollection('tweets')
  .find(
    {
      /*
       * Filter
       * fieldA: value or expression
      */
    },
    {
      /*
       * Projection
       * _id: 0, // exclude _id
       * fieldA: 1 // include field
      */
    }
  )
  .sort({
    /*
     * fieldA: 1 // ascending
     * fieldB: -1 // descending
    */
  });

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF MongoDB ✕ ESLint ✕ Prettier



MongoDB Playground - Untitled 1 - color of border palette

>mongodb create

MongoDB: Create MongoDB Playground recently used ⚙

```
// MongoDB Playground - Untitled 1 - color of border palette
>mongodb create
MongoDB: Create MongoDB Playground recently used ⚙
// Use Ctrl+Space inside a Snippet or a String literal to trigger completions.
// The current database to use.
use('colorofberlin');
// Search for documents in the current collection.
db.getCollection('tweets')
  .find()
  /*
   * Filter
   * fieldA: value or expression
   */
  },
  {
  /*
   * Projection
   * _id: 0, // exclude _id
   * fieldA: 1 // include field
   */
}
]
.sort({
  /*
   * fieldA: 1 // ascending
   * fieldB: -1 // descending
   */
});
```

Create a new playground

From the command palette

MONGODB

CONNECTIONS

- localhost:27017 connected
 - admin
 - archive
 - colorofberlin
 - tweets
 - Documents 5K
 - "6035711519ad1107e53d8ee6"
 - "6035711519ad1107e53d8ee7"
 - "6035711519ad1107e53d8ee8"
 - "6035711519ad1107e53d8ee9"
 - "6035711519ad1107e53d8eea"
 - "6035711519ad1107e53d8eeb"
 - "6035711519ad1107e53d8eec"
 - "6035711519ad1107e53d8eed"
 - "6035711519ad1107e53d8eee"
 - "6035711519ad1107e53d8eeff"

- Schema
- _id
- contributors
- coordinates
- created_at
- entities
- extended_entities
- favorite_count
- PLAYGROUNDS
- HELP AND FEEDBACK

master* 0 1 0 0 [TypeScript Importer]: Symbols: 45 Ln 9, Col 9 Spaces: 2 UTF-8 LF MongoDB ESLint Prettier

// MongoDB Playground • Untitled-1 — color-of-berlin-palette

MONGODB ...

CONNECTIONS + ...

localhost:27017 connected

> admin

> archive

colorofberlin

tweets

Documents 5K

"6035711519ad1107e53d8ee6"

"6035711519ad1107e53d8ee7"

"6035711519ad1107e53d8ee8"

"6035711519ad1107e53d8ee9"

"6035711519ad1107e53d8eea"

"6035711519ad1107e53d8eeb"

"6035711519ad1107e53d8eec"

"6035711519ad1107e53d8eed"

"6035711519ad1107e53d8eee"

"6035711519ad1107e53d8eef"

Show more...

_id

∅ contributors

∅ coordinates

A created_at

> {} entities

> {} extended_entities

favorite_count

PLAYGROUNDS

HELP AND FEEDBACK

... // MongoDB Playground Untitled-1 ●

Currently connected to localhost:27017. Click here to change connection.

```
1 // MongoDB Playground
2 // To disable this template go to Settings | MongoDB | Use Default Template For Play
3 // Make sure you are connected to enable completions and to be able to run a playgr
4 // Use Ctrl+Space inside a snippet or a string literal to trigger completions.
5
6 // Select the database to use.
7 use('mongodbVSCodePlaygroundDB');

8
9 // The drop() command destroys all data from a collection.
10 // Make sure you run it against the correct database and collection.
11 db.sales.drop();

12
13 // Insert a few documents into the sales collection.
14 db.sales.insertMany([
15   { '_id': 1, 'item': 'abc', 'price': 10, 'quantity': 2, 'date': new Date('2014-03-01') },
16   { '_id': 2, 'item': 'jkl', 'price': 20, 'quantity': 1, 'date': new Date('2014-03-01') },
17   { '_id': 3, 'item': 'xyz', 'price': 5, 'quantity': 10, 'date': new Date('2014-03-01') },
18   { '_id': 4, 'item': 'xyz', 'price': 5, 'quantity': 20, 'date': new Date('2014-04-01') },
19   { '_id': 5, 'item': 'abc', 'price': 10, 'quantity': 10, 'date': new Date('2014-04-01') },
20   { '_id': 6, 'item': 'def', 'price': 7.5, 'quantity': 5, 'date': new Date('2015-06-01') },
21   { '_id': 7, 'item': 'def', 'price': 7.5, 'quantity': 10, 'date': new Date('2015-06-01') },
22   { '_id': 8, 'item': 'abc', 'price': 10, 'quantity': 5, 'date': new Date('2016-02-01') }
23 ]);
24
25 // Run a find command to view items sold on April 4th, 2014.
26 db.sales.find({ date: { $gte: new Date('2014-04-04'), $lt: new Date('2014-04-05') } })
27
28 // Build an aggregation to view total sales for each product in 2014.
29 const aggregation = [
30   { $match: { date: { $gte: new Date('2014-01-01'), $lt: new Date('2015-01-01') } } },
31   { $group: { _id: '$item', totalSaleAmount: { $sum: { $multiply: [ '$price', '$qua
32 ] };

33 ];
```



```
import Twitter from 'twitter';
...
const tweets = await getTweets('statuses/user_timeline', { ... });
```

Prototyping

EXPLORE YOUR DATA, BUILD AGGREGATIONS, AND RUN MONGODB COMMANDS



colorofberlin.tweets:{"\$oid":"6035711519ad1107e53d8ee6"}.json — color-of-berlin-palette

MONGODB

CONNECTIONS

- localhost:27017 connected
 - admin
 - archive
 - colorofberlin
 - tweets
 - Documents 5K
 - "6035711519ad1107e53d8ee6"
 - "6035711519ad1107e53d8ee7"
 - "6035711519ad1107e53d8ee8"
 - "6035711519ad1107e53d8ee9"
 - "6035711519ad1107e53d8eea"
 - "6035711519ad1107e53d8eeb"
 - "6035711519ad1107e53d8eec"
 - "6035711519ad1107e53d8eed"
 - "6035711519ad1107e53d8eee"
 - "6035711519ad1107e53d8eef"

Show more...

Schema

- _id
- contributors
- coordinates
- created_at
- entities
- extended_entities
- favorite_count

PLAYGROUNDS

HELP AND FEEDBACK

... // MongoDB Playground Untitled-1 ● {} colorofberlin.tweets:{"\$oid":"6035711519ad1107e53d8ee6"}.json ...

```
1  [
2   "id": {
3     "$oid": "6035711519ad1107e53d8ee6"
4   },
5   "created_at": "Wed Jan 20 17:15:26 +0000 2021",
6   "id": 1351941419645214700,
7   "id_str": "1351941419645214720",
8   "text": "The color of the sky in Berlin is jet. #2d2e32 https://t.co/BMhsFbCb87",
9   "truncated": false,
10  "entities": {
11    "hashtags": [
12      {
13        "text": "2d2e32",
14        "indices": [
15          45,
16          46
17        ]
18      }
19    ],
20    "symbols": [],
21    "user_mentions": [],
22    "urls": [],
23    "media": [
24      {
25        "id": 1351941418013646800,
26        "id_str": "1351941418013646851",
27        "indices": [
28          47,
29          70
30        ],
31        "media_url": "http://pbs.twimg.com/media/EsMPjonXYAML3Hw.png",
32        "media_url_https": "https://pbs.twimg.com/media/EsMPjonXYAML3Hw.png",
33        "url": "https://t.co/BMhsFbCb87",
34      }
35    ]
36  }
37 }
```

String representation

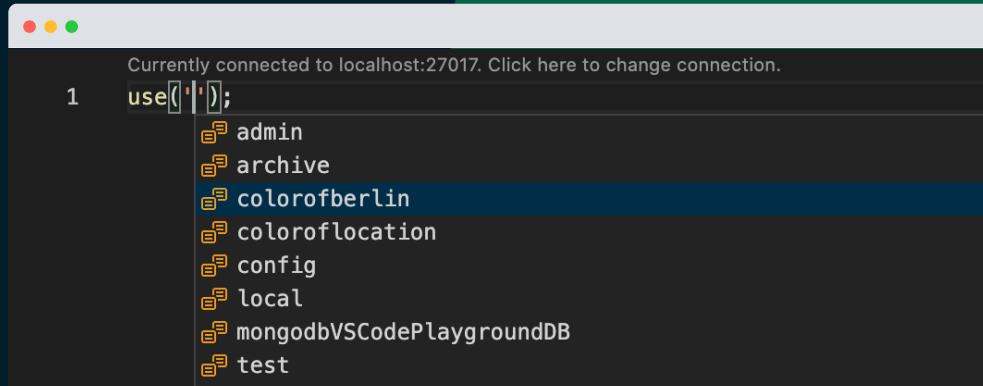
Should be converted to a date

Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON ✓ ESLint 🔍 🔍

Intelligent Autocomplete



Database names



A screenshot of a MongoDB shell window. The title bar says "MongoDB Shell". The status bar at the top right says "Currently connected to localhost:27017. Click here to change connection.". The command line shows "use(")";" followed by a dropdown menu of database names. The database "colorofberlin" is highlighted with a blue selection bar. Other databases listed include admin, archive, config, local, and test. The "coloroflocation" database is also visible.

```
use(");
admin
archive
colorofberlin
coloroflocation
config
local
mongodbVSCodePlaygroundDB
test
```

Intelligent Autocomplete



Database names



Collection names



Shell API

A screenshot of the MongoDB shell interface. The text area shows a connection message and some command history:

```
Currently connected to localhost:27017. Click here to change connection.  
1 use('colorofberlin');  
2  
3 db.t
```

The cursor is positioned at the end of "db.t". A dropdown menu is open, listing suggestions for the collection name:

- tweets
- grantRolesToRole
- grantRolesToUser
- grantPrivilegesToRole

Intelligent Autocomplete



Database names



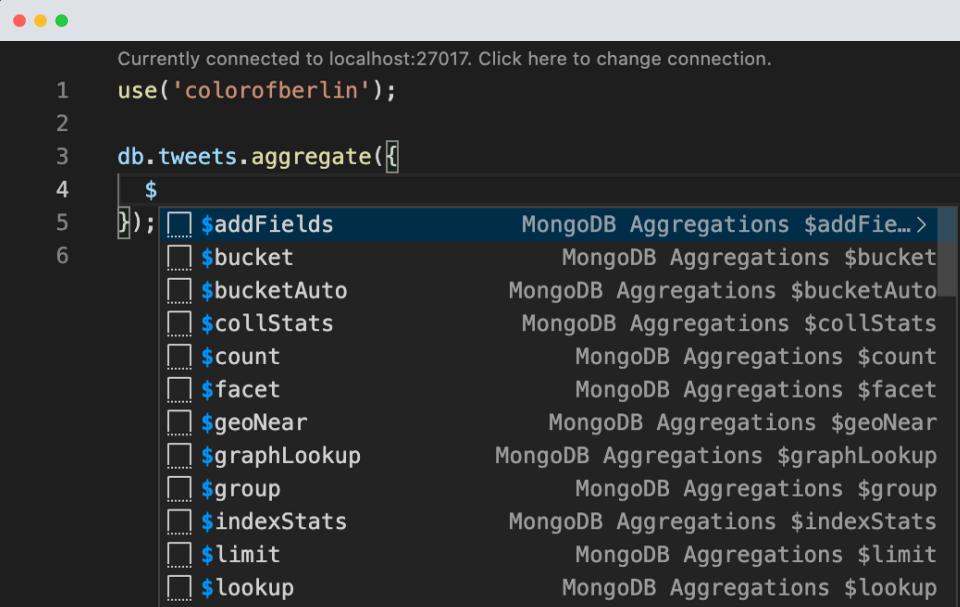
Collection names



Shell API



MongoDB operators



The screenshot shows the MongoDB shell interface. The user has typed the command:

```
use('colorofberlin');  
db.tweets.aggregate([  
  $  
]);
```

The cursor is positioned at the end of the aggregation pipeline, specifically after the first dollar sign. A dropdown menu is open, listing various MongoDB aggregation pipeline operators, each preceded by a small checkbox and followed by its full name and a tooltip:

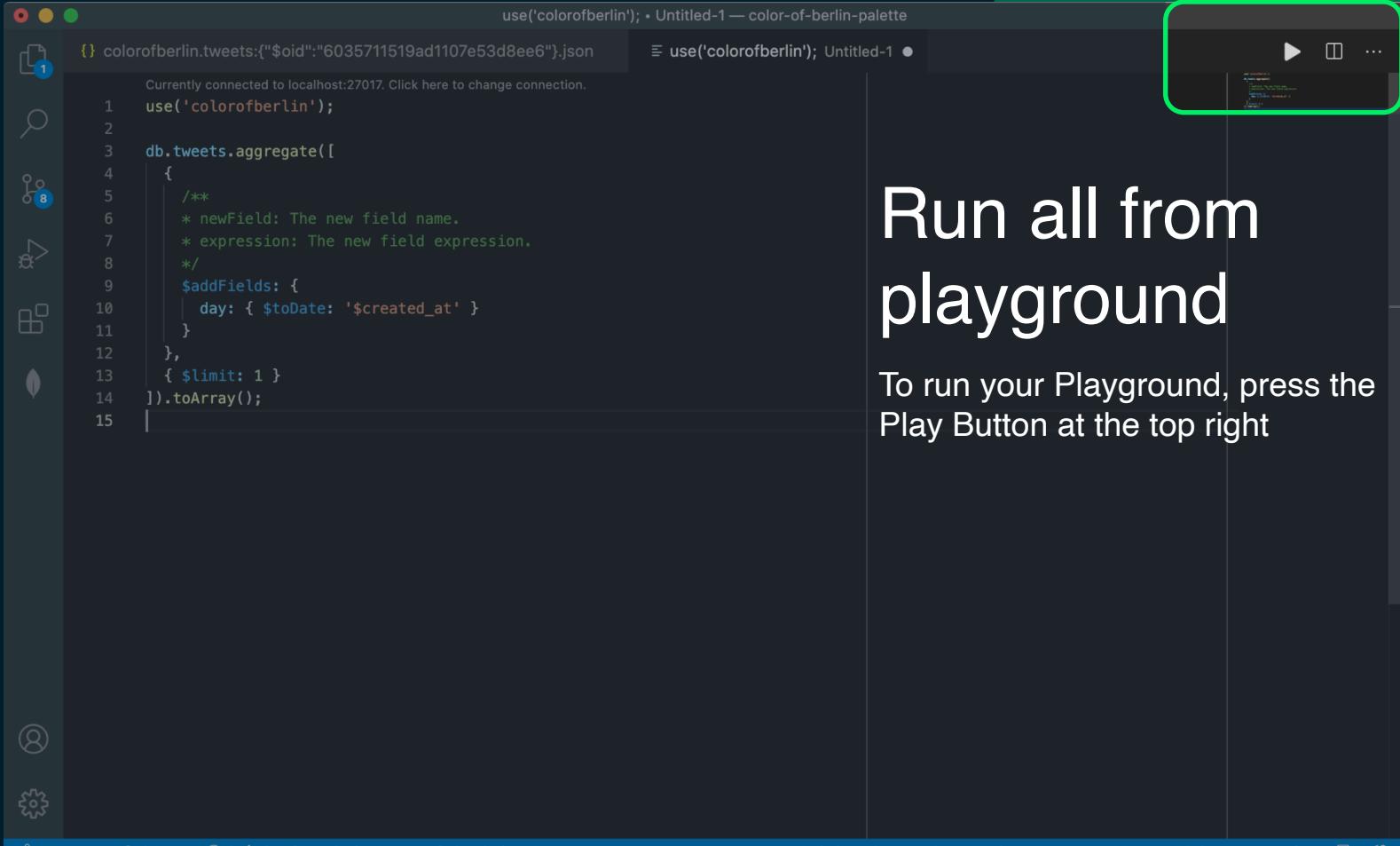
- \$addFields MongoDB Aggregations \$addFields>
- \$bucket MongoDB Aggregations \$bucket
- \$bucketAuto MongoDB Aggregations \$bucketAuto
- \$collStats MongoDB Aggregations \$collStats
- \$count MongoDB Aggregations \$count
- \$facet MongoDB Aggregations \$facet
- \$geoNear MongoDB Aggregations \$geoNear
- \$graphLookup MongoDB Aggregations \$graphLookup
- \$group MongoDB Aggregations \$group
- \$indexStats MongoDB Aggregations \$indexStats
- \$limit MongoDB Aggregations \$limit
- \$lookup MongoDB Aggregations \$lookup

The screenshot shows a MongoDB shell interface running in a dark-themed code editor. The title bar reads "use('colorofberlin'); • Untitled-1 — color-of-berlin-palette". The left sidebar contains various icons for file operations, search, users, and settings. The main pane displays the following MongoDB aggregation query:

```
use('colorofberlin');
db.tweets.aggregate([
  {
    /**
     * newField: The new field name.
     * expression: The new field expression.
    */
    $addFields: {
      day: { $toDate: '$created_at' }
    },
    { $limit: 1 }
  ].toArray());
```

The status bar at the bottom indicates the current branch is "master*", the file has 0 changes, and the commit count is 0. It also shows symbols: 45, and status: [TypeScript Importer]. Symbols: 45, Ln 15, Col 1, Spaces: 2, UTF-8, LF, MongoDB, ESLint, Prettier.





The screenshot shows the MongoDB Compass interface. On the left is a sidebar with various icons: a file (1), magnifying glass, people (8), a play button, a square, a leaf, a user, and a gear. The main area has two tabs: 'colorofberlin.tweets:{ "\$oid": "6035711519ad1107e53d8ee6" }.json' and 'use('colorofberlin'); Untitled-1 — color-of-berlin-palette'. The code in the Untitled-1 tab is:

```
use('colorofberlin');
db.tweets.aggregate([
  {
    /**
     * newField: The new field name.
     * expression: The new field expression.
    */
    $addFields: {
      day: { $toDate: '$created_at' }
    },
    { $limit: 1 }
  ].toArray());
```

At the top right of the main window, there is a toolbar with a play button, a square, and three dots. A green rounded rectangle highlights this toolbar area. At the bottom of the screen, there is a footer bar with icons for master*, 0 0 1↑, 0 0 △ 0, [TypeScript Importer]: Symbols: 45, Ln 15, Col 1, Spaces: 2, UTF-8, LF, MongoDB, ESLint, Prettier, and a search icon.

Run all from playground

To run your Playground, press the Play Button at the top right

Playground Result — color-of-berlin-palette

```
:6035711519ad1107e53d8ee6".json ⚙️ use('colorofberlin'); Untitled-1 ⚙️ ... ⚙️ { } Playground Result ×
```

Currently connected to localhost:27017. Click here to change connection.

```
1 use('colorofberlin');
2
3 db.tweets.aggregate([
4   {
5     /**
6      * newField: The new field name.
7      * expression: The new field expression.
8      */
9     $addFields: {
10       day: { $toDate: '$created_at' }
11     }
12   },
13   { $limit: 1 }
14 ].toArray());
15 }
```

```
170 "geo": null,
171 "coordinates": null,
172 "place": null,
173 "contributors": null,
174 "is_quote_status": false,
175 "retweet_count": 0,
176 "favorite_count": 0,
177 "favorited": false,
178 "retweeted": false,
179 "possibly_sensitive": false,
180 "lang": "en",
181 "day": {
182   "$date": "2021-01-20T17:15:26Z"
183 }
184 }
185 ]
```

The screenshot shows the MongoDB playground interface. On the left, the code editor contains a MongoDB query:

```
use('colorofberlin');
db.tweets.aggregate([
  {
    /**
     * newField: The new field name.
     * expression: The new field expression.
    */
    $addFields: {
      day: { $toDate: '$created_at' }
    },
    { $limit: 1 }
]).toArray();
```

The results pane on the right displays the output of the aggregation pipeline. A green box highlights the newly added 'day' field, which contains the ISODate "2021-01-20T17:15:26Z".

Playground Result — color-of-berlin-palette

Currently connected to localhost:27017. Click here to change connection.

170 "geo": null,
171 "coordinates": null,
172 "place": null,
173 "contributors": null,
174 "is_quote_status": false,
175 "retweet_count": 0,
176 "favorite_count": 0,
177 "favorited": false,
178 "retweeted": false,
179 "possibly_sensitive": false,
180 "lang": "en",
181 "day": {
182 "\$date": "2021-01-20T17:15:26Z"
183 }
184 }
185 }

New field

The result of \$addField stage

New field

The result of \$addField stage



Aggregation Stages

```
{ $addFields: { day: { $toDate: '$created_at' } } }  
{ $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }
```



Aggregation Stages

```
{ $addFields: { day: { $toDate: '$created_at' } } }  
{ $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }  
{ $sort: { day: -1 } }
```



Aggregation Stages

```
{ $addFields: { day: { $toDate: '$created_at' } } }
{ $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }
{ $sort: { day: -1 } }
{
  $group: {
    _id: { ... },
    tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
  }
}
```



Aggregation Stages

```
{ $addFields: { day: { $toDate: '$created_at' } } }
{ $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }
{ $sort: { day: -1 } }
{
  $group: {
    _id: { ... },
    tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
  }
}
{ $limit: 7 }
```



Aggregation Stages

```
{ $addFields: { day: { $toDate: '$created_at' } } }
{ $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }
{ $sort: { day: -1 } }
{
  $group: {
    _id: { ... },
    tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
  }
}
{ $limit: 7 }
{ $project: { _id: 0, day: 1, tweetsByDay: 1 }}
```

use('colorofberlin'); • Untitled-1 — color-of-berlin-palette

```
use('colorofberlin');

db.tweets.aggregate([
  { $addFields: { day: { $toDate: '$created_at' } } },
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $sort: { day: -1 } },
  {
    $group: {
      _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
      tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
    }
  },
  { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.month', day: '_id.day' } } } },
  { $sort: { day: -1 } },
  { $limit: 7 },
  { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
]).toArray();
```

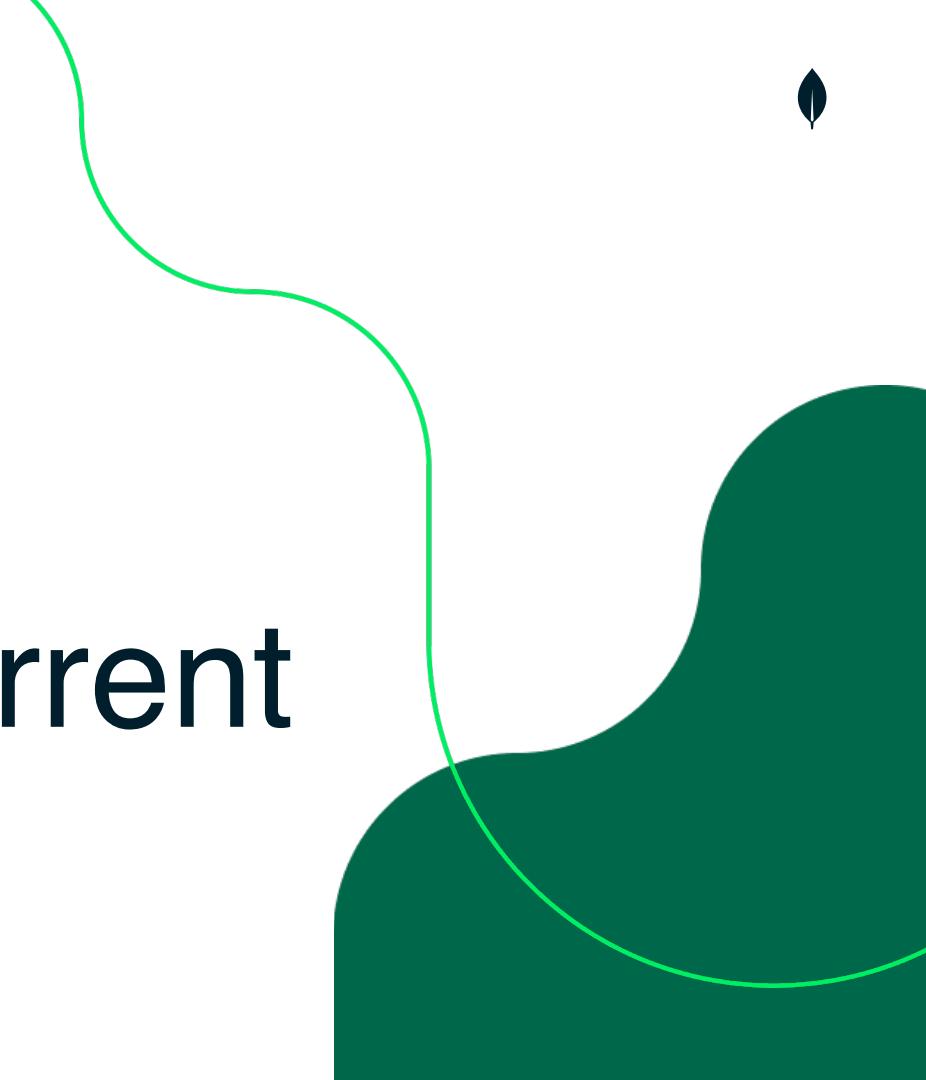
use('colorofberlin'); Untitled-1 — color-of-berlin-palette

Currently connected to localhost:27017. Click here to change connection.

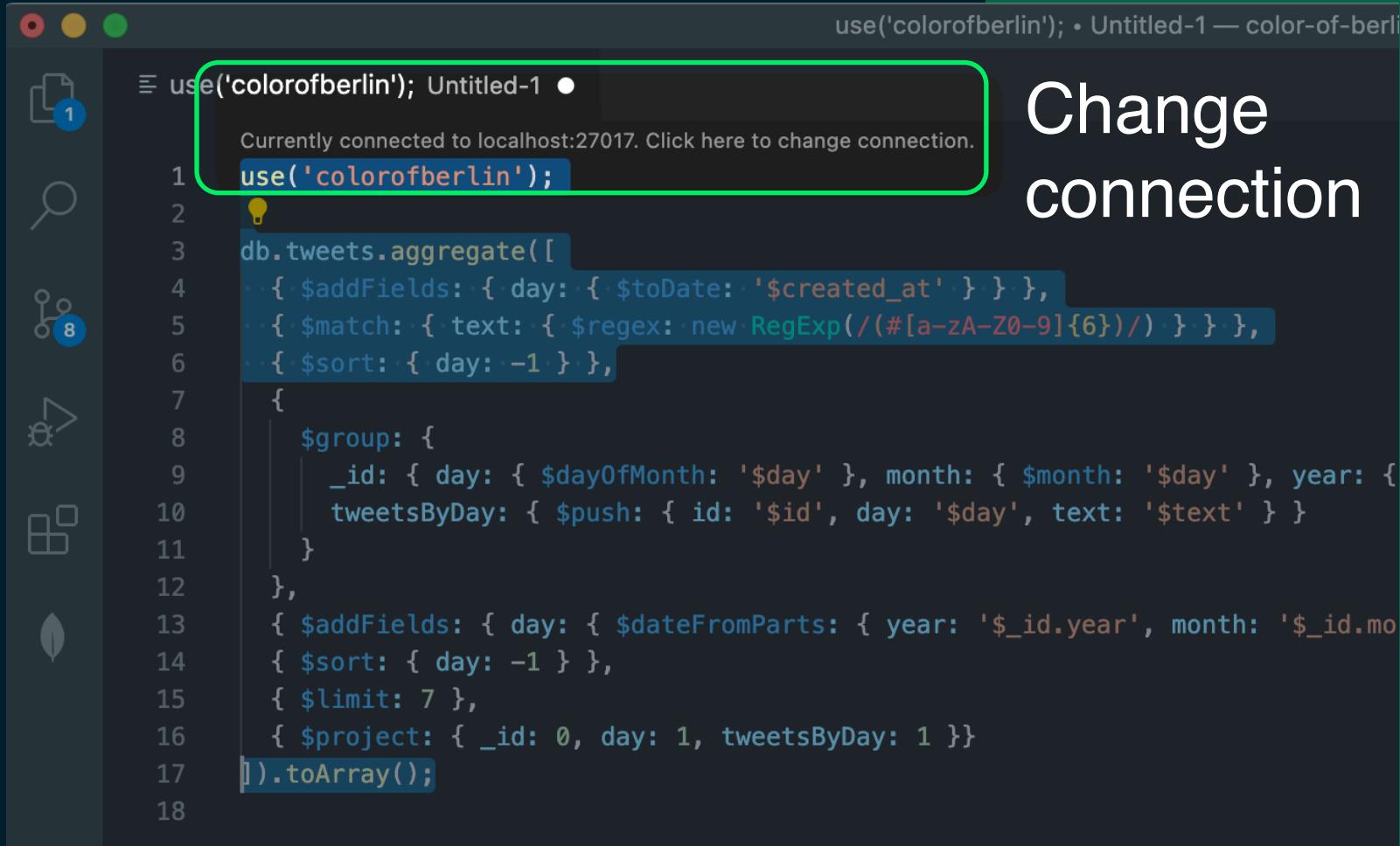
```
1 use('colorofberlin');
2
3 db.tweets.aggregate([
4   { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
5   { $sort: { day: -1 } },
6   {
7     $group: {
8       _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
9       tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
10     }
11   },
12   {
13     $addFields: { day: { $dateFromParts: { year: '$_id.year', month: '$_id.month', day: '$_id.day' } } },
14   { $sort: { day: -1 } },
15   { $limit: 7 },
16   { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
17 }).toArray();
18
```

Run selected playground blocks

master* 0 0 1↑ 0 0 △ 0 [TypeScript Importer]: Symbols: 45 2 selections (207 characters selected) Spaces: 2 UTF-8 LF MongoDB ✕ ESLint ✕ Prettier ✕



Check the current
connection



The screenshot shows the MongoDB Compass interface. At the top, there are three circular icons (red, yellow, green) and the text "use('colorofberlin'); • Untitled-1 — color-of-berlin". On the left sidebar, there are icons for file (1), search, connections (8), export, and leaf (1). The main area contains a code editor with the following MongoDB aggregation pipeline:

```
use('colorofberlin'); Untitled-1 ●  
Currently connected to localhost:27017. Click here to change connection.  
use('colorofberlin');  
  
db.tweets.aggregate([  
  { $addFields: { day: { $toDate: '$created_at' } } },  
  { $match: { text: { $regex: new RegExp(/([a-zA-Z0-9]{6})/) } } },  
  { $sort: { day: -1 } },  
  {  
    $group: {  
      _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: {  
        tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }  
      }  
    },  
    { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.mo  
    { $sort: { day: -1 } },  
    { $limit: 7 },  
    { $project: { _id: 0, day: 1, tweetsByDay: 1 } }  
  ]).toArray();  
18
```

Change connection

The screenshot shows the MongoDB Compass interface. On the left is a sidebar with icons for file (1), search (8), connections (8), and a refresh symbol. The main area has a title bar with three dots, a yellow dot, and a green dot, followed by the text "use('colorofberlin'); • Untitled-1 — color-of-berlin". A context menu is open over the code editor, with the top item "Select new connection..." highlighted in blue. The menu also includes "Add new connection" and "localhost:27017". The code editor contains a MongoDB query:

```
use('colorofberlin');
db.tweets.aggregate([
  { $addFields: { day: { $toDate: '$created_at' } } },
  { $match: { text: { $regex: new RegExp(/([a-zA-Z0-9]{6})/) } } },
  { $sort: { day: -1 } },
  {
    $group: {
      _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
      tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
    }
  },
  { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.month', day: '_id.day' } } },
    { $sort: { day: -1 } },
    { $limit: 7 },
    { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
]).toArray();
```

use('colorofberlin'); Untitled-1 — color-of-berlin-palette

```
use('colorofberlin');
db.tweets.aggregate([
  { $addFields: { day: { $toDate: '$text' } },
    { $match: { text: { $regex: new RegExp(/([d-zA-Z0-9]+[0-9]{2})\s{2}([d-zA-Z0-9]+[0-9]{2})/ ) } },
    { $sort: { day: -1 } },
    {
      $group: {
        _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
        tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
      }
    },
    { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.month', day: '_id.day' } } } },
    { $sort: { day: -1 } },
    { $limit: 7 },
    { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
  ]).toArray();
])
```

Are you sure you want to run this playground against localhost:27017? This confirmation can be disabled in the extension settings.

Cancel Yes

master* 0 0 1 0 0 ▲ 0 [TypeScript Importer]: Symbols: 45 Ln 18, Col 1 Spaces: 2 UTF-8 LF MongoDB ESLint Prettier

Playground Result — color-of-berlin-palette

```
use('colorofberlin'); Untitled-1 ●  
Currently connected to localhost:27017. Click here to change connection.  
1 use('colorofberlin');  
2  
3 db.tweets.aggregate([
4   { $addFields: { day: { $toDate: '$created_at' } } },
5   { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
6   { $sort: { day: -1 } },
7   {
8     $group: {
9       _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
10      tweetsByDay: { $push: { id: '$_id', day: '$day', text: '$text' } }
11    },
12  },
13  { $addFields: { day: { $dateFromParts: { year: '$_id.year', month: '$_id.month', day: '$_id.day' } } } },
14  { $sort: { day: -1 } },
15  { $limit: 7 },  
]}  
Playground Result ×  
1 []
2 {
3   "tweetsByDay": [
4     {
5       "id": 1369179204189819000,
6       "day": {
7         "$date": "2021-03-09T06:52:15Z"
8       },
9       "text": "The color of the sky in Berlin is slate gray. #7a788e https://t.co/rYTIRVnXRY"
10     },
11     {
12       "id": 1369168084930605000,
13       "day": {
14         "$date": "2021-03-09T06:08:04Z"
15       },
16       "text": "The color of the sky in Berlin is light slate gray. #7c7d9b https://t.co/3y1E5cgwJ8"  
]}  
master* 0 0 1↑ 0 0 0 [TypeScript Importer]: Symbols: 45 Ln 1, Col 1 Spaces: 2 JSON ✘ ESLint 🔍
```

More than prototyping

MONGODB PLAYGROUNDS ARE JAVASCRIPT ENVIRONMENTS





Debugging

```
const tweets = db.tweets.aggregate([
  ...
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $limit: 7 }
]).toArray();

console.log(`Number of tweets: ${tweets.length}`); // Number of tweets: 7

tweets;
```

use('colorofberlin'); Untitled-1 — color-of-berlin-palette

```
use('colorofberlin');

const tweets = db.tweets.aggregate([
  { $addFields: { day: { $toDate: '$created_at' } } },
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } }
  { $sort: { day: -1 } },
  {
    $group: {
      _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' } },
      tweetsByDay: { $push: { id: '$id', created_at: '$day', text: '$text' } }
    },
    { $addFields: { day: { $dateFromParts: { year: '$_id.year', month: '$_id.month', day: '$day' } } } },
    { $sort: { day: -1 } },
    { $limit: 7 },
    { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
  }).toArray();

console.log(`Number of tweets: ${tweets.length}`); // Number of tweets: 7
tweets;
```

Playground Result

```
[{"tweetsByDay": [{"id": 1363532813836947500, "created_at": {"$date": "2021-02-21T16:55:30Z"}, "text": "The color of the sky in Berlin is #e6a5a5"}, {"id": 1363512087482429400, "created_at": {"$date": "2021-02-21T15:33:09Z"}, "text": "The color of the sky in Berlin is #f0c9a7"}, {"id": 1363507001788162000, "created_at": {"$date": "2021-02-21T15:12:56Z"}, "text": "The color of the sky in Berlin is #d9e1f2"}, {"id": 1363504290480992300, "created_at": {"$date": "2021-02-21T15:02:10Z"}, "text": "The color of the sky in Berlin is #f0c9a7"}, {"id": 1363504290480992300, "created_at": {"$date": "2021-02-21T15:02:10Z"}, "text": "The color of the sky in Berlin is #f0c9a7"}, {"id": 1363504290480992300, "created_at": {"$date": "2021-02-21T15:02:10Z"}, "text": "The color of the sky in Berlin is #f0c9a7"}, {"id": 1363504290480992300, "created_at": {"$date": "2021-02-21T15:02:10Z"}, "text": "The color of the sky in Berlin is #f0c9a7"}]}
```

PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

Number of tweets: 7

Playground output



Require built-in Node.js modules

```
const util = require('util');
const tweets = db.tweets.aggregate([
  ...
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $limit: 7 }
]).toArray();

console.log(`Number of tweets: ${tweets.length}`);
console.log(tweets[0]); // [object Object]
console.log(util.inspect(tweets[0])); // { tweetsByDay: [ ... ], day: ... }
```



Logging

```
const fs = require('fs');
const tweets = db.tweets.aggregate([
  ...
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $limit: 7 }
]).toArray();

fs.appendFileSync(
  'playground.log',
  `${new Date().toISOString()} playground editor INFO ${JSON.stringify(agg)}\n`;
);
```



```
2021-04-15T14:51:32.154Z playground editor INFO [{"$addFields":{"day":{"$toDate":"$created_at"}},  
{"$match":{"text":{  
    "$regex":{  
        "}}}  
    }}]
```

Regex value is missing

```
new RegExp(/(#[a-zA-Z0-9]{6})/)
```



```
const fs = require('fs');
const tweets = db.tweets.aggregate([
  ...
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $limit: 7 }
]).toArray();

fs.appendFileSync(
  'playground.log',
  `${new Date().toISOString()} playground editor INFO ${JSON.stringify(agg)}\n`;
);
```

JSON.stringify()

Can not transform a BSON object



JSON

schema-less

key-value pairs

text

6 types

unordered

BSON

schema-less

key-value pairs

binary

21 types

ordered



Require any Node.js modules

```
const bson = require('bson');
const EJSON = bson.EJSON;
```



```
const fs = require('fs');
const tweets = db.tweets.aggregate([
  ...
  { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
  { $limit: 7 }
]).toArray();

fs.appendFileSync(
  'playground.log',
  `${new Date().toISOString()} playground editor INFO ${EJSON.stringify(agg)}\n`;
);
```

EJSON.stringify()

Transforms a BSON object to a string



```
2021-04-15T14:51:32.154Z playground editor INFO [{"$addFields": {"day": {"$toDate": "$created_at"}},  
{"$match": {"text": {
```

```
    "$regex": {  
        "$regularExpression": {  
            "pattern": "#[a-zA-Z0-9]{6}",  
            "options": ""  
        }  
    }  
}}]
```



Advanced Logging

```
const fs = require('fs');
const pinoms = require('pino-multi-stream');
const streams = [
  { stream: fs.createWriteStream('info.playground.log', { flags: 'a' }) },
  { level: 'error', stream: fs.createWriteStream('error.playground.log') }
];
const log = pinoms({ streams });

...
log.info(` ${new Date().toISOString()} playground editor INFO ${EJSON.stringify(agg)}\n`);
```



MACOS AND LINUX

\$HOME/.node_modules

\$HOME/node_modules

\$HOME/.vscode/

...

WINDOWS

C:\Users\.node_modules

C:\Users\node_modules

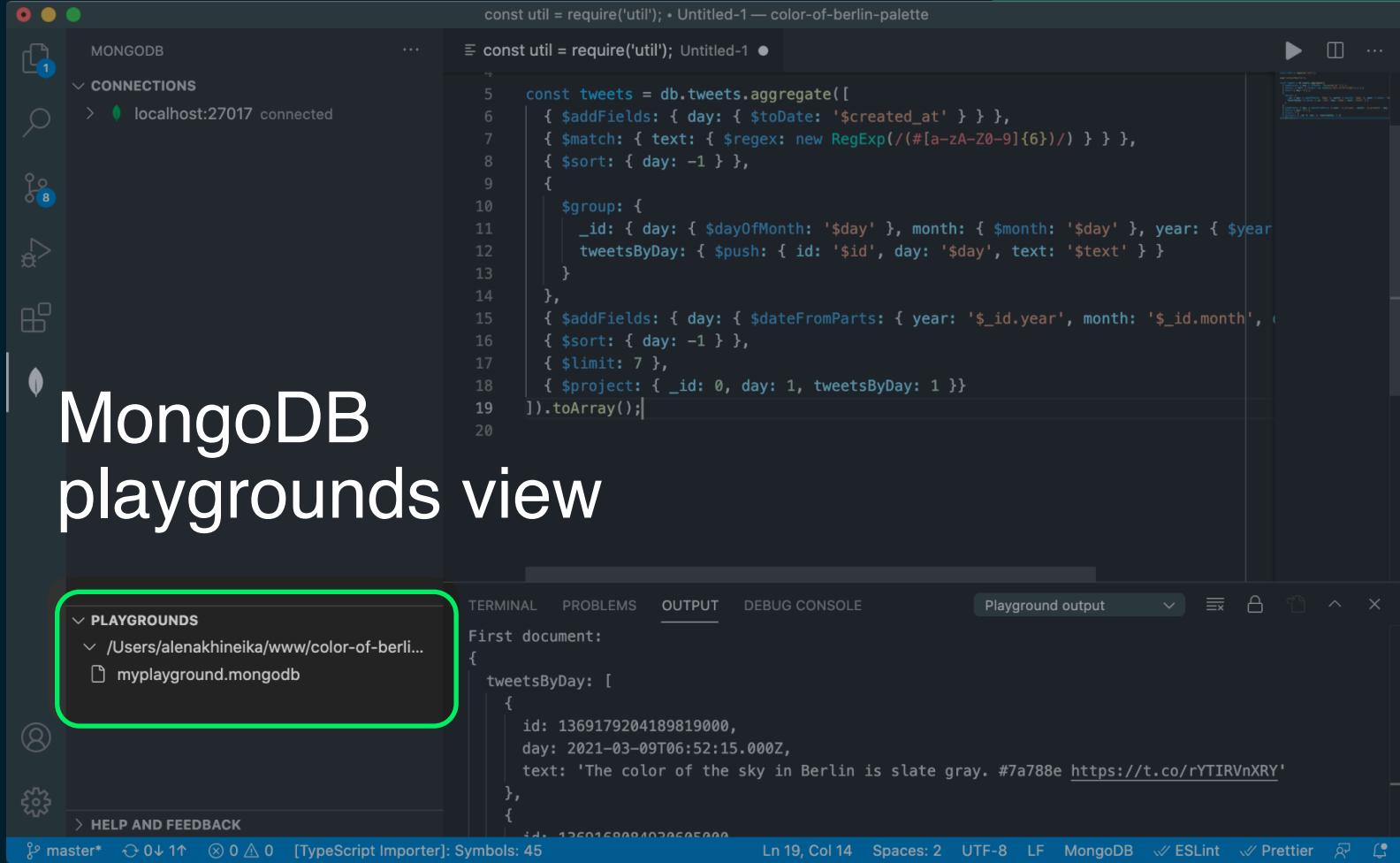
C:\Users<user>\node_modules

...

Playgrounds
are smart and easy to
use



MongoDB playgrounds view



A screenshot of the MongoDB playgrounds view in Visual Studio Code. The interface includes a sidebar with icons for file, search, connections, and playgrounds. The main area shows a code editor with a MongoDB aggregate query and a terminal showing the resulting document. A green box highlights the 'PLAYGROUNDS' section in the sidebar.

const util = require('util'); • Untitled-1 — color-of-berlin-palette

```
const util = require('util'); Untitled-1 ●
```

```
const tweets = db.tweets.aggregate([
  { $addFields: { day: { $toDate: '$created_at' } } },
  { $match: { text: { $regex: new RegExp(/([a-zA-Z0-9]{6})/) } } },
  { $sort: { day: -1 } },
  {
    $group: {
      _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
      tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
    }
  },
  { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.month', day: '_id.day' } } } },
  { $sort: { day: -1 } },
  { $limit: 7 },
  { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
]).toArray();
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

Playground output

```
First document:
{
  tweetsByDay: [
    {
      id: 1369179204189819000,
      day: 2021-03-09T06:52:15.000Z,
      text: 'The color of the sky in Berlin is slate gray. #7a788e https://t.co/rYTIRVnXRY'
    },
    {
      id: 1369179204189819001,
      day: 2021-03-09T06:52:15.001Z,
      text: 'The color of the sky in Berlin is slate gray. #7a788e https://t.co/rYTIRVnXRY'
    }
  ]
}
```

PLAYGROUNDS

- /Users/alenakhineika/www/color-of-berli...
- myplayground.mongodb

HELP AND FEEDBACK

master* 0 1 0 0 [TypeScript Importer]: Symbols: 45 Ln 19, Col 14 Spaces: 2 UTF-8 LF MongoDB ESLint Prettier

generic.ts — color-of-berlin-palette

EXPLORER OPEN EDITORS 2 UNSAVED

COLOR-OF-BERLIN-PALETTE

- > dist
- > node_modules
- > scripts
- > src
 - > client
 - > server
 - > config
 - > controllers
 - generic.ts M
 - > middleware
 - > startup
- httpException.ts
- index.ts
- > shared
- > utils
- .env
- .env.example
- .eslintignore
- .eslintrc
- .gitignore
- .prettierrc
- error.playground.log U
- info.playground.log U

OUTLINE TIMELINE NPM SCRIPTS

use('colorofberlin'); Untitled-1 ● TS generic.ts M ●

src > server > controllers > generic.ts > exports.getSevenDaysColors

```
113
114     if (!collection) {
115         return response.json({ tweets });
116     }
117
118     try {
119         tweets = await collection.aggregate([
120             /* Copy your aggregation here */
121         ]).toArray();
122
123         tweets = getColorsFromText(tweets);
124
125         response.json({ tweets });
126     } catch (error) {
127         next(new HttpException({
128             status: HttpStatus.INTERNAL_SERVER_ERROR,
129             message: error.message
130         }));
131     }
132 };
133 
```

Ln 120, Col 40 Spaces: 2 UTF-8 LF TypeScript 4.2.3 ✓ ESLint Prettier



Similar but not exactly the same



MongoDB Shell



JavaScript/TypeScript



Similar but not exactly the same



MongoDB Shell

```
use( ... );
db.tweets.aggregate([ ... ]).toArray();
```



JavaScript/TypeScript

The aggregation command is synchronous and includes the collection name.



Similar but not exactly the same



MongoDB Shell

```
use( ... );
db.tweets.aggregate([ ... ]).toArray();
```

The aggregation command is synchronous and includes the collection name.



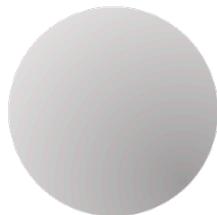
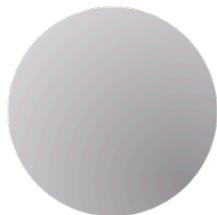
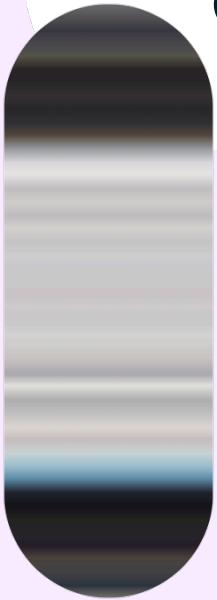
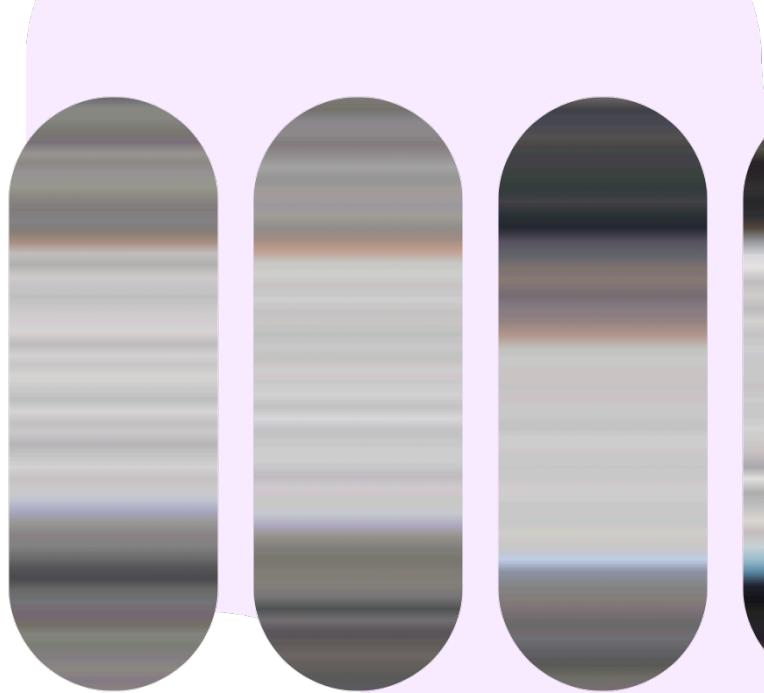
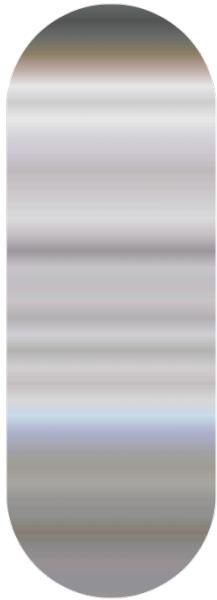
JavaScript/TypeScript

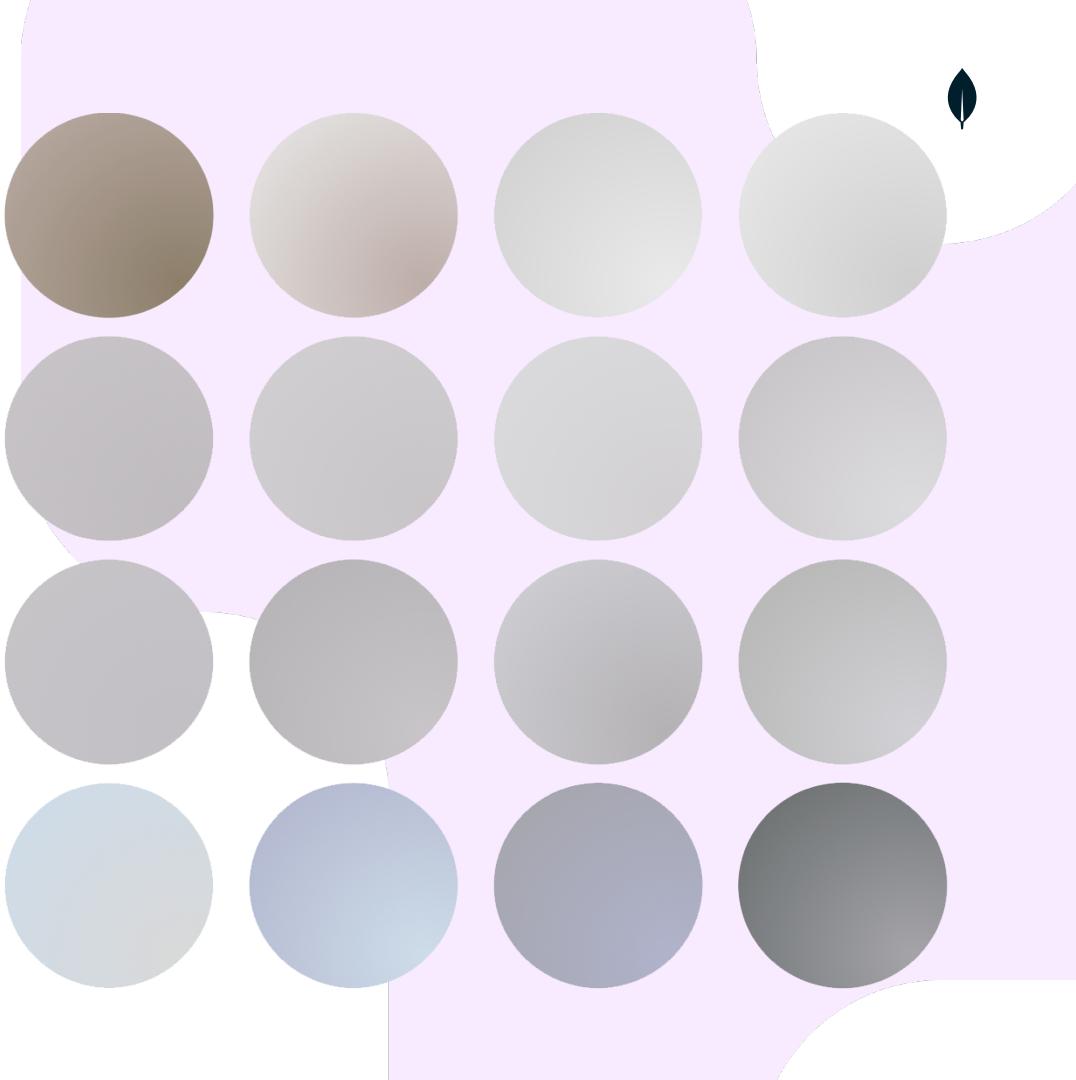
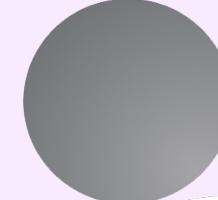
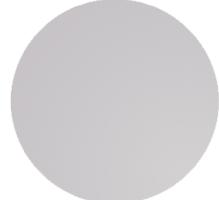
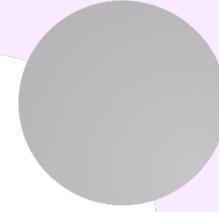
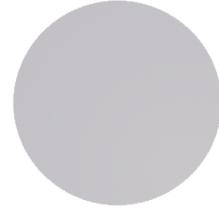
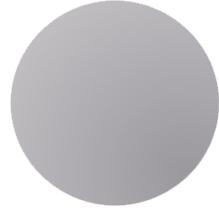
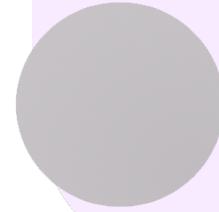
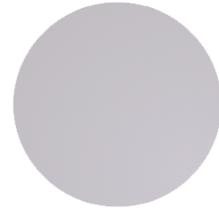
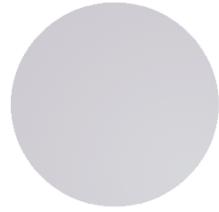
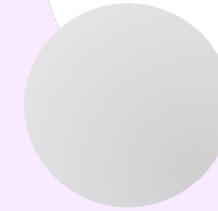
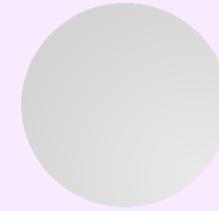
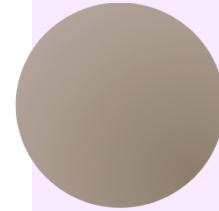
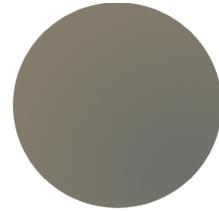
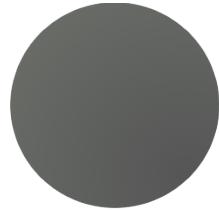
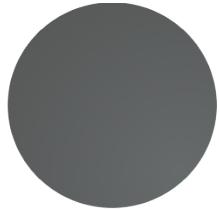
```
import { MongoClient } from 'mongodb';

const db = await mongoClient.db( ... );
const collection = await db.collection('tweets');
await collection.aggregate([ ... ]).toArray();
```

The aggregation command is asynchronous method of the mongoClient instance.









Color of Berlin

14.7K Tweets



Pinned Tweet



Color of Berlin @colorofberlin · Mar 9

...

Hi all!

Thank you for the messages — it seems that the new skycam source is not producing the best results as it is low quality and often obstructed by lights.

I have decided to pause the bot for now until I have time to dedicate to fixing the camera source and HEX codes!

7

3

90



The screenshot shows the MongoDB Atlas interface. At the top, there are navigation links for "Access Manager", "Billing", "All Clusters", "Get Help", and a user dropdown for "Alena". Below the header, there are tabs for "Atlas", "Realm", and "Charts". The "Collections" tab is currently selected. On the left sidebar under "DEPLOYMENT", the "Databases" section is active, showing "coloroflocation" with its "colors" collection selected. Other options include "Triggers", "Data Lake", "SECURITY", "Database Access", "Network Access", and "Advanced".

In the main content area, the "coloroflocation.colors" collection is displayed. It shows a collection size of 517.27KB, total documents of 4881, and index total size of 132KB. There are tabs for "Find", "Indexes", "Schema Anti-Patterns", "Aggregation", and "Search Indexes". A "FILTER" button with the query `{"field": "value"}` is present, along with "OPTIONS", "Apply", and "Reset" buttons.

The "QUERY RESULTS 1-20 OF MANY" section displays three document snippets:

```
_id: ObjectId("604149fd084d01bdf70ef743")
created_at: 2021-03-04T20:58:37.146+00:00
text: "The color of the sky in Berlin is darkgray. #979795."
colorName: "darkgray"
colorHex: "979795"
```

```
_id: ObjectId("6041580d084d01bdf70ef744")
created_at: 2021-03-04T21:58:37.672+00:00
text: "The color of the sky in Berlin is gray. #B6B6B6."
colorName: "gray"
colorHex: "B6B6B6"
```

```
_id: ObjectId("60418dc1084d01bdf70ef745")
created_at: 2021-03-05T01:47:45.849+00:00
text: "The color of the sky in Berlin is darkslategray. #3D4353."
colorName: "darkslategray"
```



January 2021

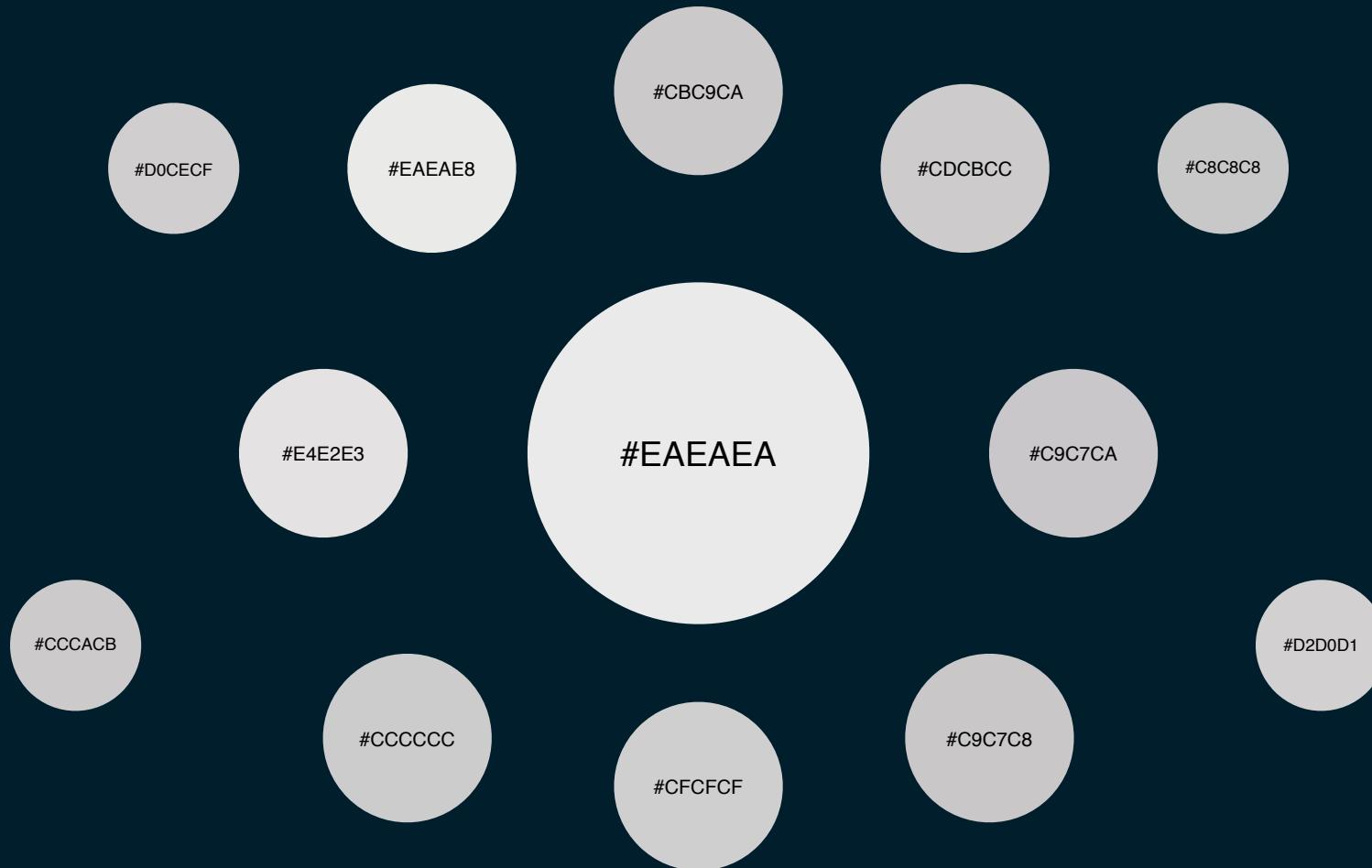
April

July



Data source issues

Switched to the lambda function



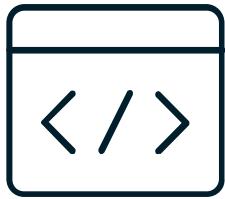
How do playgrounds work?

SHORT STORY ON HOW WE BUILT THEM





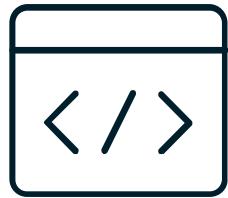
Playgrounds



VSCode Editor

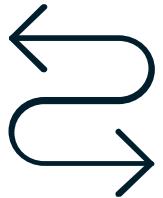
Playgrounds UI

Playgrounds



VSCode Editor

Playgrounds UI

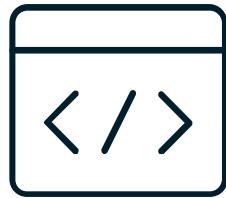


Language Server

Background Process



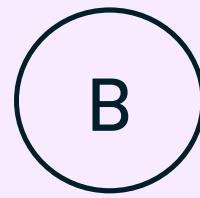
Playgrounds



VSCode Editor
Playgrounds UI



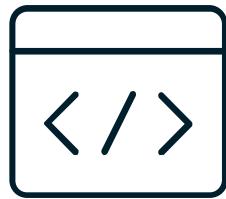
Language Server
Background Process



Babel Parser
Intelligent Autocomplete



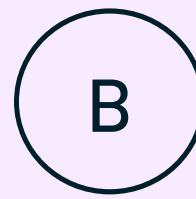
Playgrounds



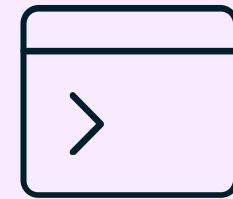
VSCode Editor
Playgrounds UI



Language Server
Background Process



Babel Parser
Intelligent Autocomplete



MongoDB Shell
Playgrounds Evaluation



The screenshot shows a dual-pane interface. On the left is a terminal window in VS Code, and on the right is the MongoDB Compass Dev application.

Terminal Session (VS Code):

```
use('colorofberlin'); Untitled-1
Currently connected to localhost:27017. Click here to switch.
1 use('colorofberlin');
2
>_MONGOSH
Enterprise test>
```

MongoDB Compass Dev:

Local

Databases Performance

CREATE DATABASE

Database Name	Storage Size	Collections	Indexes
admin	20.0KB	0	1

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Last login: Sat Sep 25 19:08:33 on ttys003
alenakhineika@Alenas-MacBook-Pro ~ % mongosh
Current Mongosh Log ID: 615085e9f9a9f4c2d24367e9
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:     5.0.2
Using Mongosh:     1.0.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting:
2021-09-24T14:54:20.065+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2021-09-24T14:54:20.065+02:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2021-09-24T14:54:20.065+02:00: Soft rlimits for open file descriptors too low
-----
Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
Enterprise test>
```



Code evaluation with mongosh

VS CODE EXTENSION

```
use('colorofberlin')
```

```
switched to db colorofberlin
```



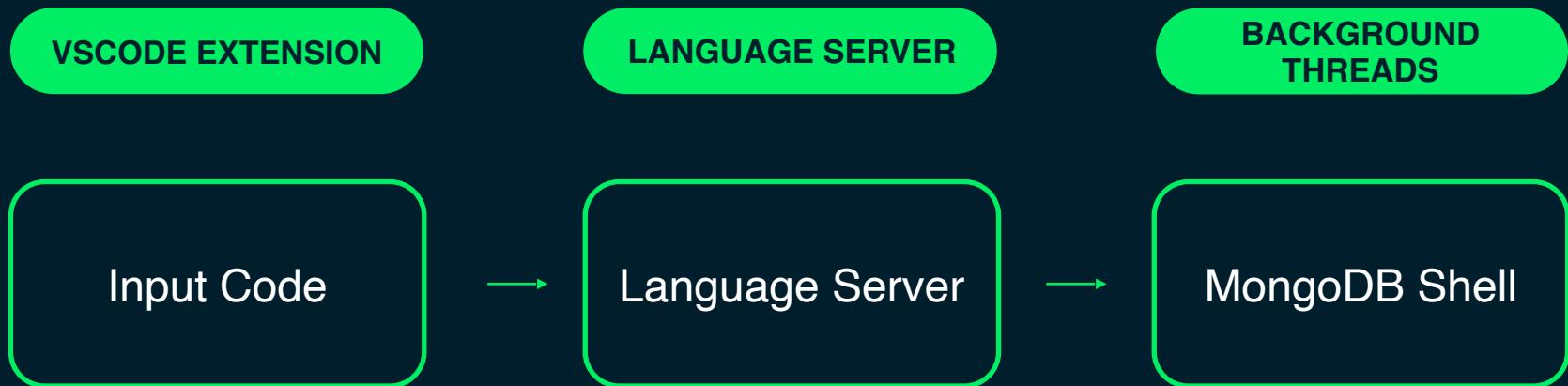
MongoDB Shell



MongoDB Node.js
Driver

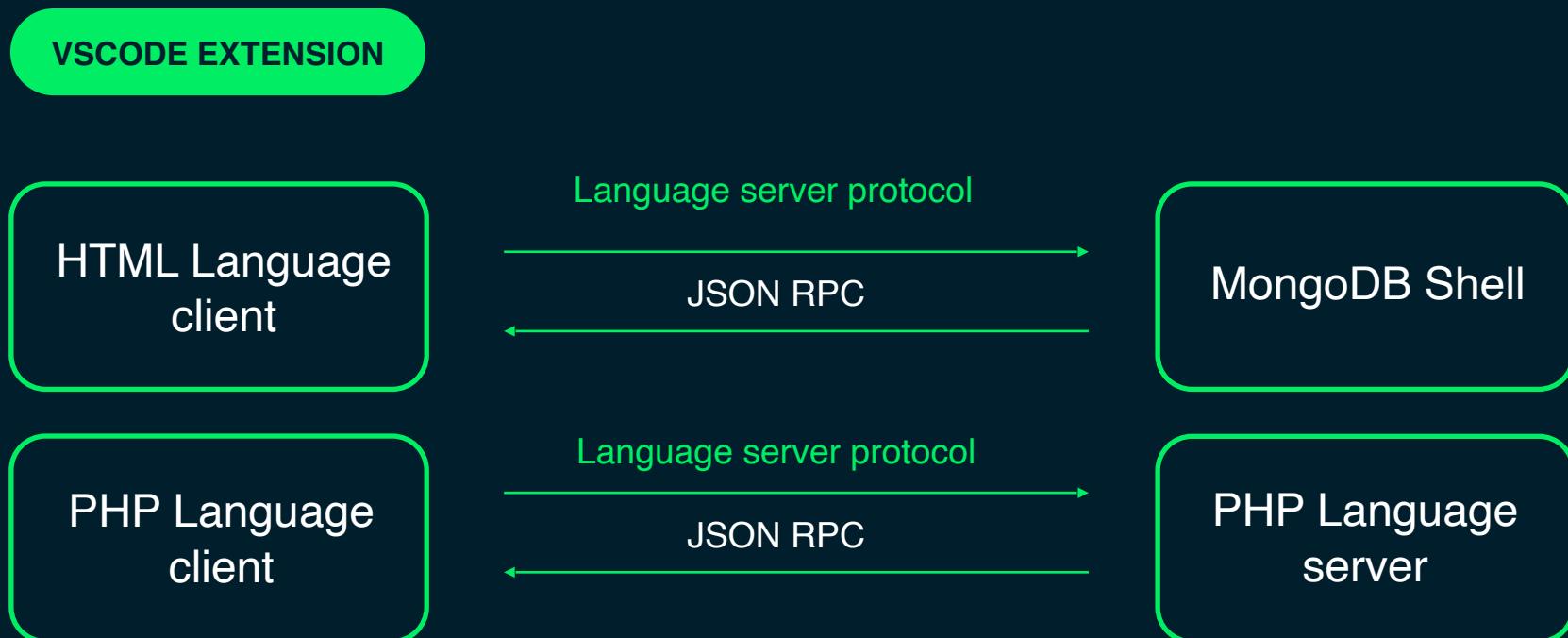


Layers of Application



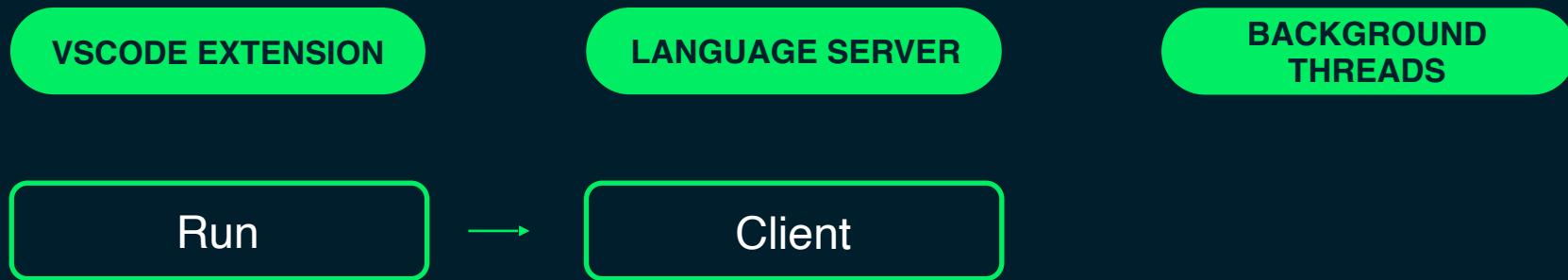


Language Server



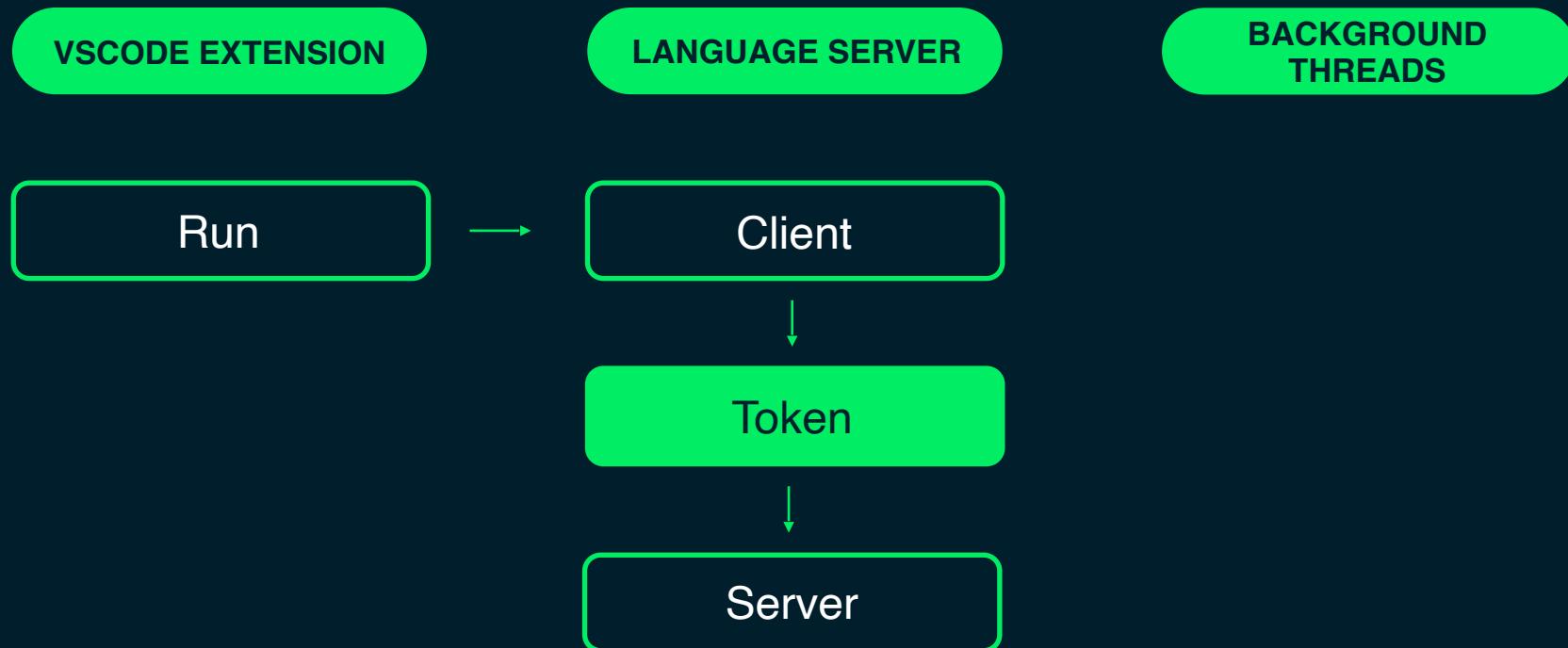


Run Request



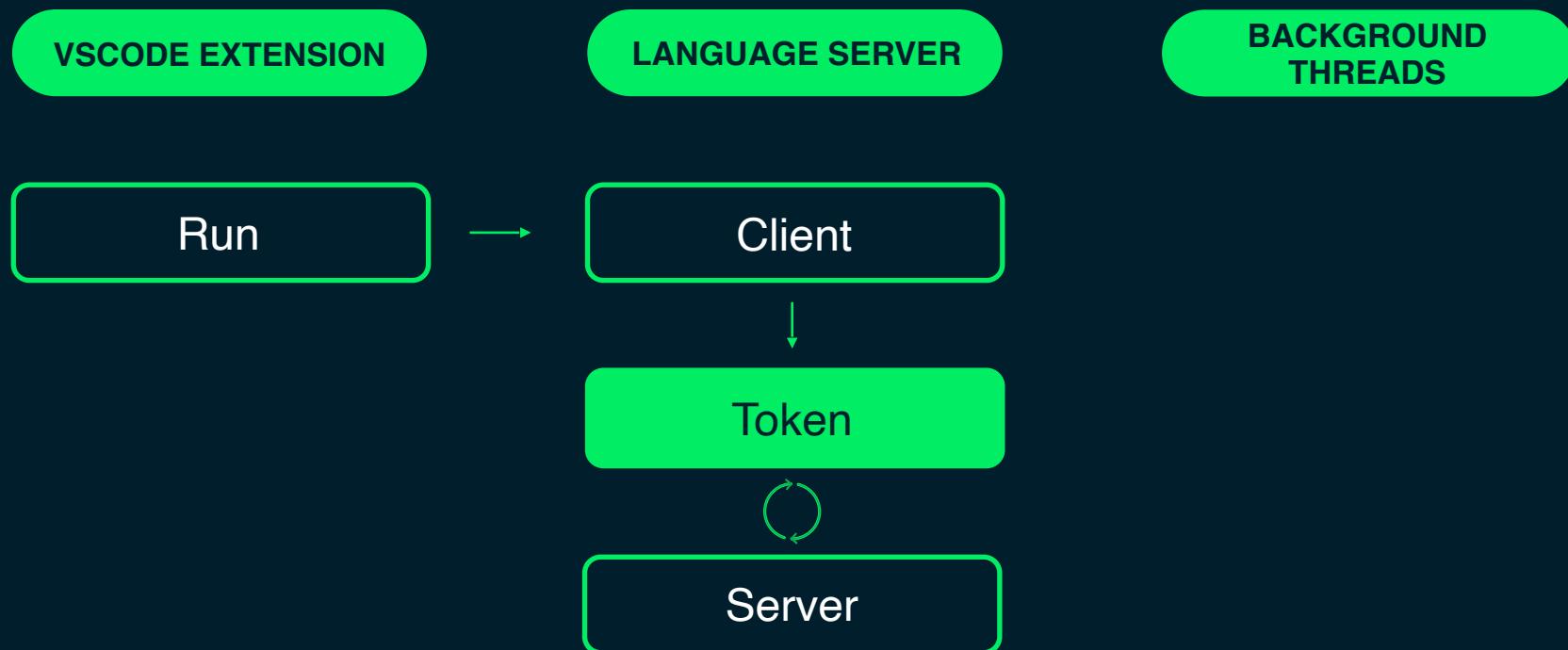


Add Cancellation Token





Listen for Cancellation Events

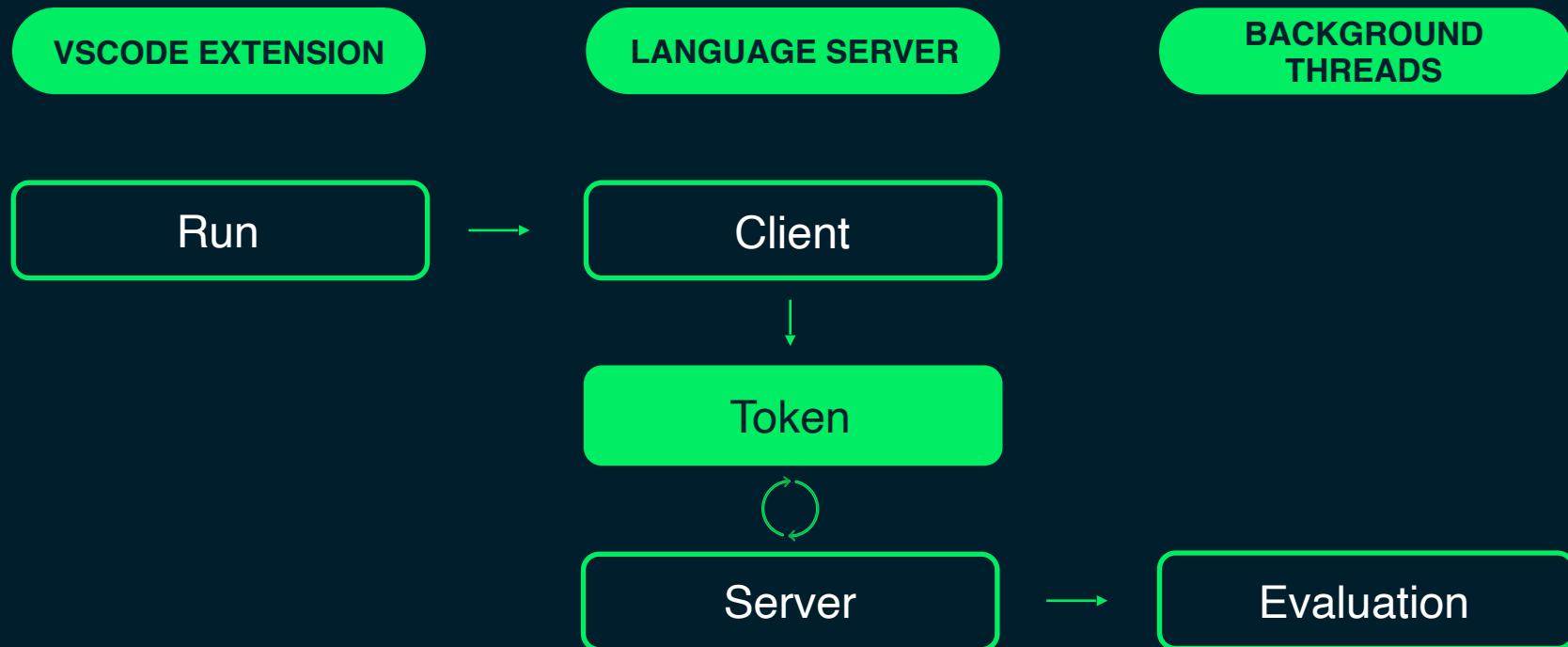




```
while (1 === 1) { }
```

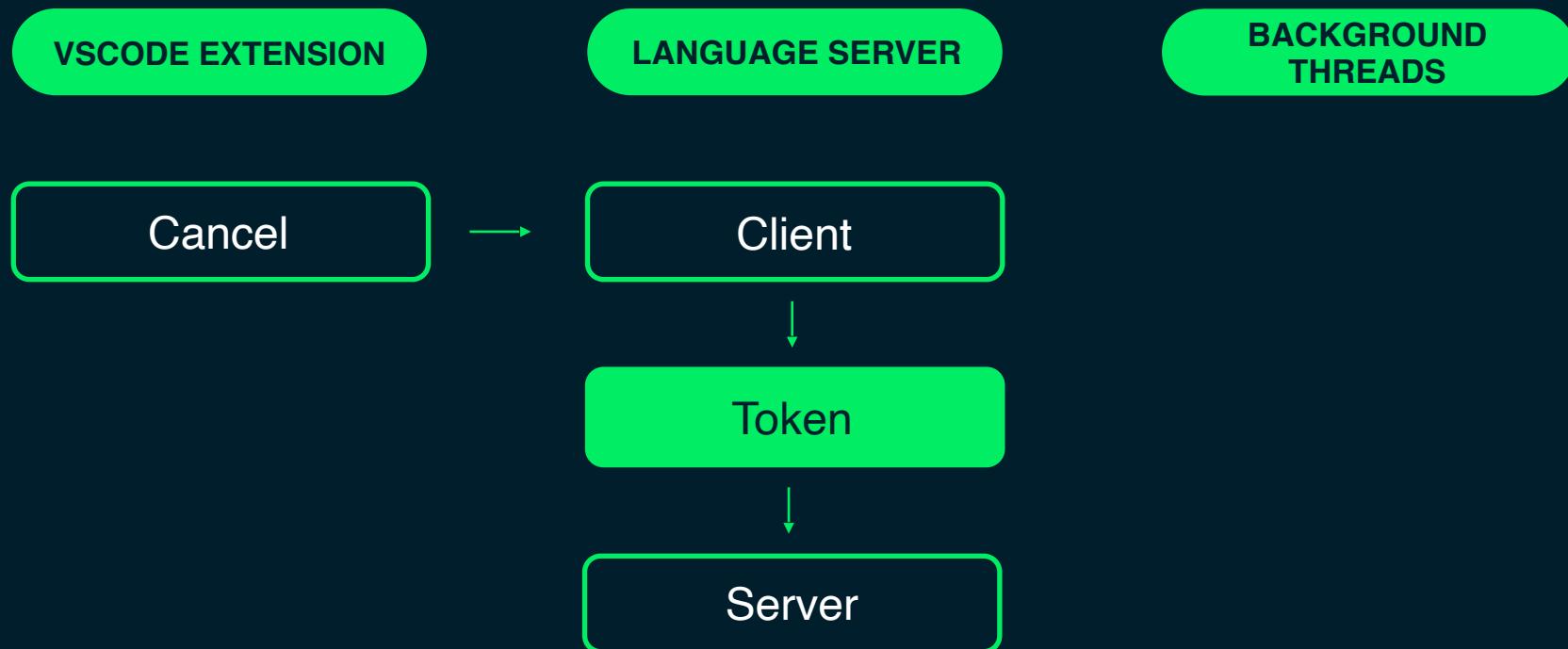


Create Background Thread





Cancel Request



while (1 === 1) {} • Untitled-1 — color-of-berlin-palette

Ξ while (1 === 1) {} Untitled-1 ●

Currently connected to localhost:27017. Click here to change connection.

```
1  while (1 === 1) {}
```

1

Cancel

Running MongoDB playground...

Source: MongoDB for VS Code (Extension)

Spaces: 2 UTF-8 LF MongoDB ✓ Prettier

master 0 △ 0 [TypeScript Importer]: Symbols: 46 Getting results...





mongodb-js/vscode: Connect t x +

github.com/mongodb-js/vscode/

Search or jump to... / Pulls Issues Marketplace Explore

Unwatch 17 Unstar 172 Fork 21

Code Issues 21 Pull requests 1 Actions Security Insights Settings

master Go to file Add file Code

About

Connect to MongoDB and Atlas and directly from your VS Code environment, navigate your databases and collections, inspect your schema and use playgrounds to prototype queries and aggregations.

marketplace.visualstudio.c...

database mongodb

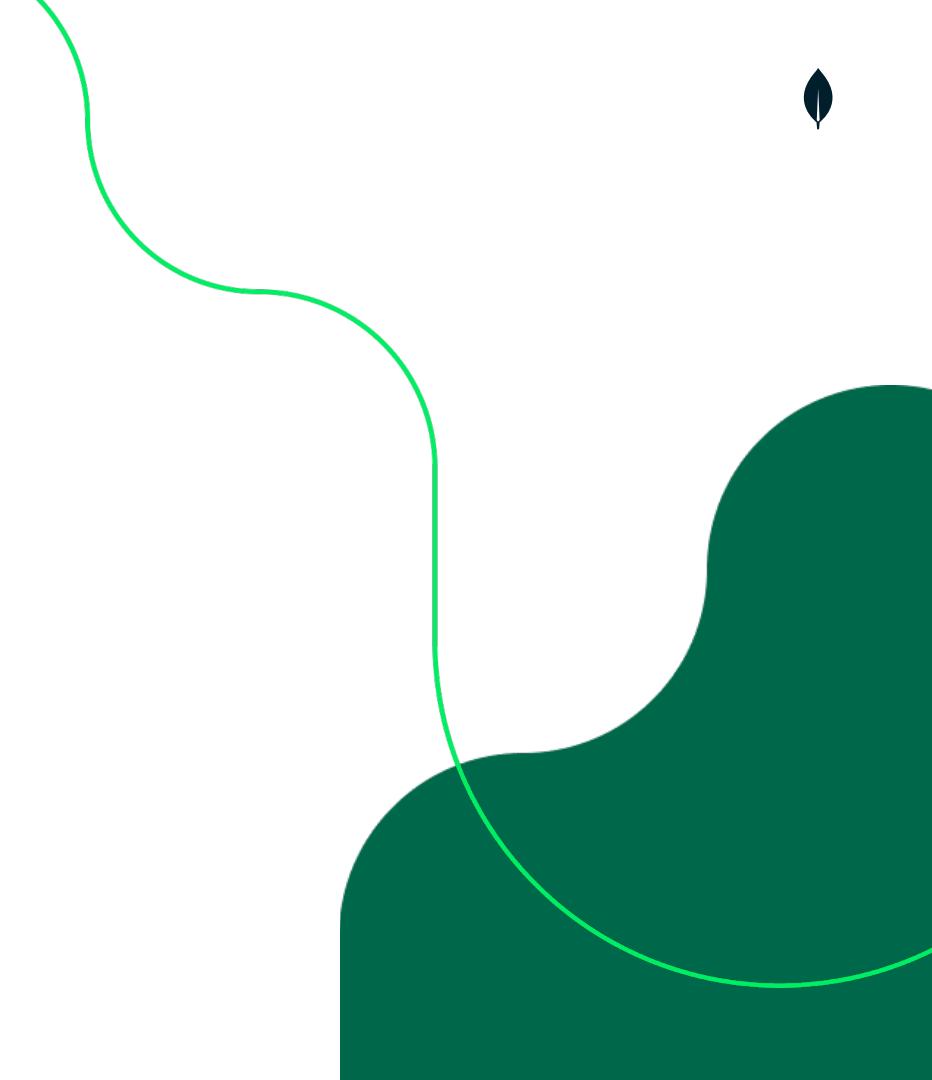
terraform vscode

vscode-extension

mongodb-atlas

File	Description	Time Ago
.github	fix: use unique names (#127)	11 months ago
.vscode	chore: Update recommended dev extension tslint...	4 months ago
images	feat(tree-view): add icons for connect and playgr...	3 months ago
languages	VS CODE-93: db and collection names completio...	12 months ago
playgrounds	VS CODE-151: Add index key type icons (#150)	10 months ago
resources/screenshots	Added copy & screenshot about document editin...	3 months ago
scripts	VS CODE-93: db and collection names completio...	12 months ago
snippets	fix(COMPASS-4492): remove sql pipeline stage o...	6 months ago

VSCode



MongoDB extension for VSCode



Work with MongoDB without switching between applications

MongoDB extension for VSCode



Work with MongoDB without switching between applications



Benefit from embedded dev tools

MongoDB extension for VSCode



Work with MongoDB without switching between applications



Benefit from embedded dev tools



Syntax highlighting, autocomplete, snippets, debugging, documentation

MongoDB extension for VSCode



Work with MongoDB without switching between applications



Benefit from embedded dev tools



Syntax highlighting, autocomplete, snippets, debugging, documentation



MongoDB Shell

MongoDB extension for VSCode



Work with MongoDB without switching between applications



Benefit from embedded dev tools



Syntax highlighting, autocomplete, snippets, debugging, documentation



MongoDB Shell



Focus on building your application



Thank you for
your time.