



# MongoDB for VSCode

Be more productive when you build applications



Alena Khineika | Senior Software Engineer | MongoDB



MongoDB.Berlin

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

vscode (Git)

Commit Pull Push Fetch Branch Merge

New Connection

WORKSPACE File status History Search BRANCHES TAGS REMOTES

Preview README.md ×

## Color of Berlin Palette DRAFT

The Color of Berlin Palette project fetches colors of the sky in Berlin from the Color of Berlin Twitter account and creates a color palette.

```
Last login: Thu Apr 29 12:31:10 on ttys001
alenakhineika@Alenas-MBP ~ % mongosh
Current Mongosh Log ID: 608ae8679309ee6c19bd45be
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&ssl=false
Using MongoDB: 4.4.4
Using Mongosh Beta: 0.7.7

For more information about mongosh, please see our docs: https://docs.mongodb.com/mongosh/
```

-----

The server generated these startup warnings when booting:

2021-04-29T19:09:18.057+02:00: Access control is not enabled for this server. Access control is unrestricted

2021-04-29T19:09:18.057+02:00: This server is bound to local interface 127.0.0.1. Start the server with --bind\_ip <address> to specify a different IP or with --bind\_ip\_all to bind to all interfaces. If this behavior is intended, add the --nojournal option or set the journal size to 0.01 to disable this warning

2021-04-29T19:09:18.057+02:00: Soft rlimits too low

-----

Enterprise >

New Connection FAVORITE

Fill in connection fields individually

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

mongosh

Welcome to the MongoDB Documentation

docs.mongodb.com

Search Documentation

string in Atlas?

Save a cluster? After you save a cluster, you can create a link to it in the Cluster list.

Color of Berlin Palette

mongosh | Documentation

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

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

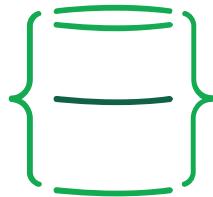
[Databases](#) [json](#) [keybindings](#) [MongoDB](#)  
[MongoDB Atlas](#) [MQL](#) [snippet](#) [terraform](#)

#### Resources

[Issues](#)  
[Repository](#)  
[Homepage](#)  
[License](#)  
[Changelog](#)  
[Download Extension](#)

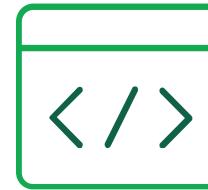
# Development

---



**MongoDB**

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

Color of Berlin  
@colorofberlin

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

📍 Berlin, Deutschland 🔗 [github.com/laurendorman/c...](https://github.com/laurendorman/c...) 📅 Joined April 2020

1 Following 1,969 Followers

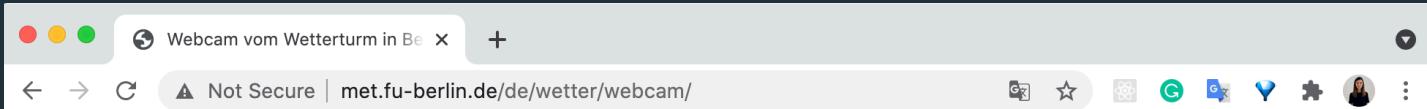
Followed by ira. and Lauren Dorman

Tweets Tweets & replies Media Likes

Pinned Tweet

Color of Berlin @colorofberlin · Mar 9

...



# Institut für Meteorologie

[Home](#) | [Lageplan](#) | [Kontakt](#) | [E-Mail/Tel.](#) | [a-z Index](#) | [Intern](#)

Studium

Veranstaltungen

Bibliothek

Wetterinfos

Wetterpate

Wetterdaten

Satellitendaten

Jobs

Wetterkarte

Prognoseturnier

Luftgüte

DMG

**AKTUELLE**  
**Pollensaison**  
Hier erhalten Sie



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

Blick vom Wetterturm (oben) Wasserturm, Höhe ca. 20 m über NN, des Instituts für Meteorologie der FU Berlin auf



← → ⌂ 🔍 star 📁 🌐 🌐 🌐 🌐 🌐 🌐

twitter.com/colorofberlin



## Color of Berlin

14.7K Tweets

Following



Color of Berlin @colorofberlin · Mar 7

The color of the sky in Berlin is ash grey. #bab1b6

...



Color of Berlin @colorofberlin · Mar 7

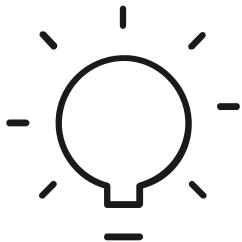
The color of the sky in Berlin is silver. #c0bec1

...



Berlin





# Features

---

Work with MongoDB and your data 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

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 [TypeScript Importer]: Symbols: 45

✓ ESLint

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

MongoDB — color-of-berlin-palette

MONGODB CONNECTIONS

No connections found.

Add Connection

>mongodb: connect with| MongoDB: Connect with Connection String... recently used

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 [TypeScript Importer]: Symbols: 45

✓ ESLint

MongoDB — color-of-berlin-palette

MONGDB

CONNECTIONS

No connections found.

Add Connection

mongodb

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](#).

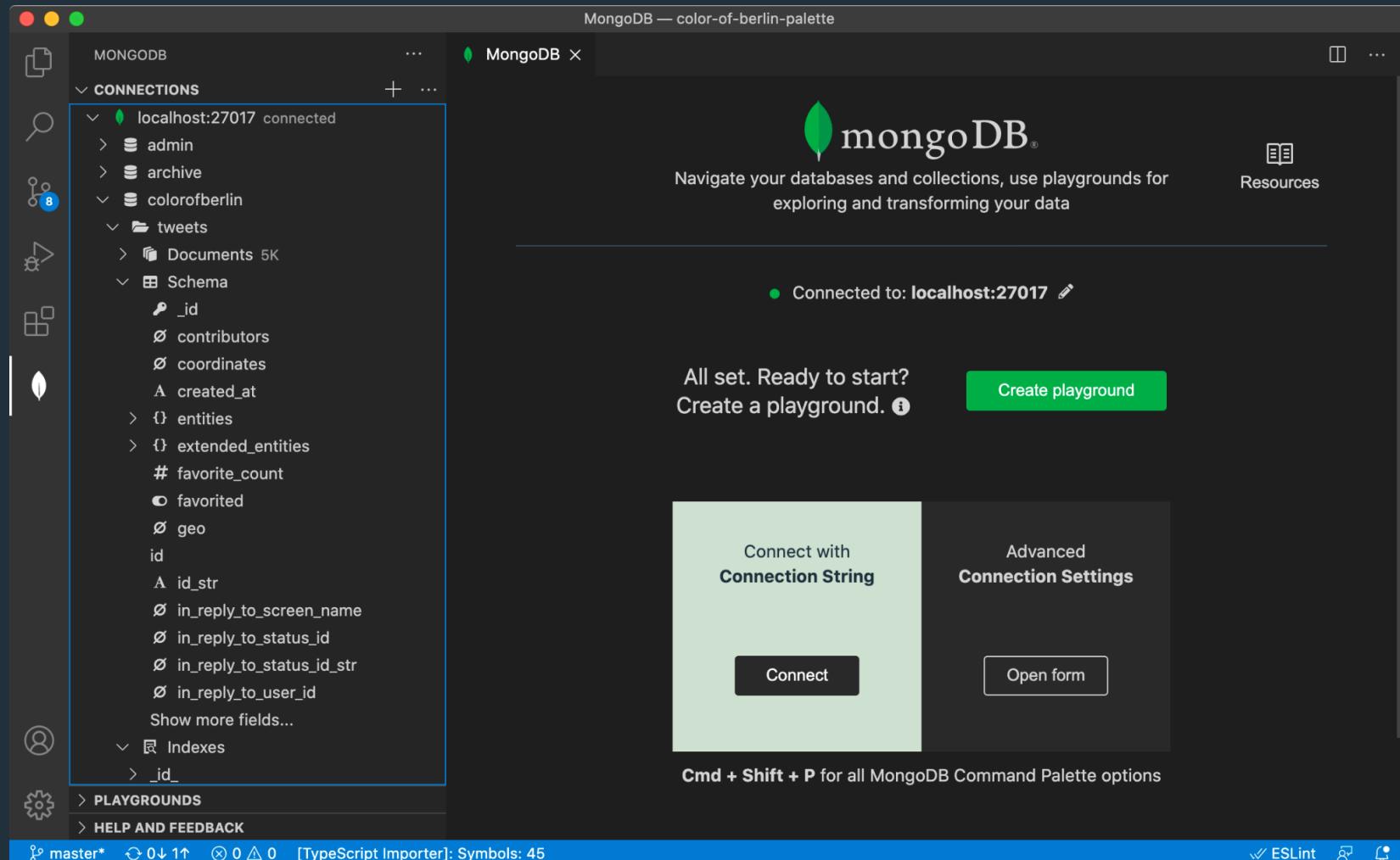
Create free cluster

PLAYGROUNDS

HELP AND FEEDBACK

master\* 0 1 0 0 [TypeScript Importer]: Symbols: 45

✓ ESLint



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

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

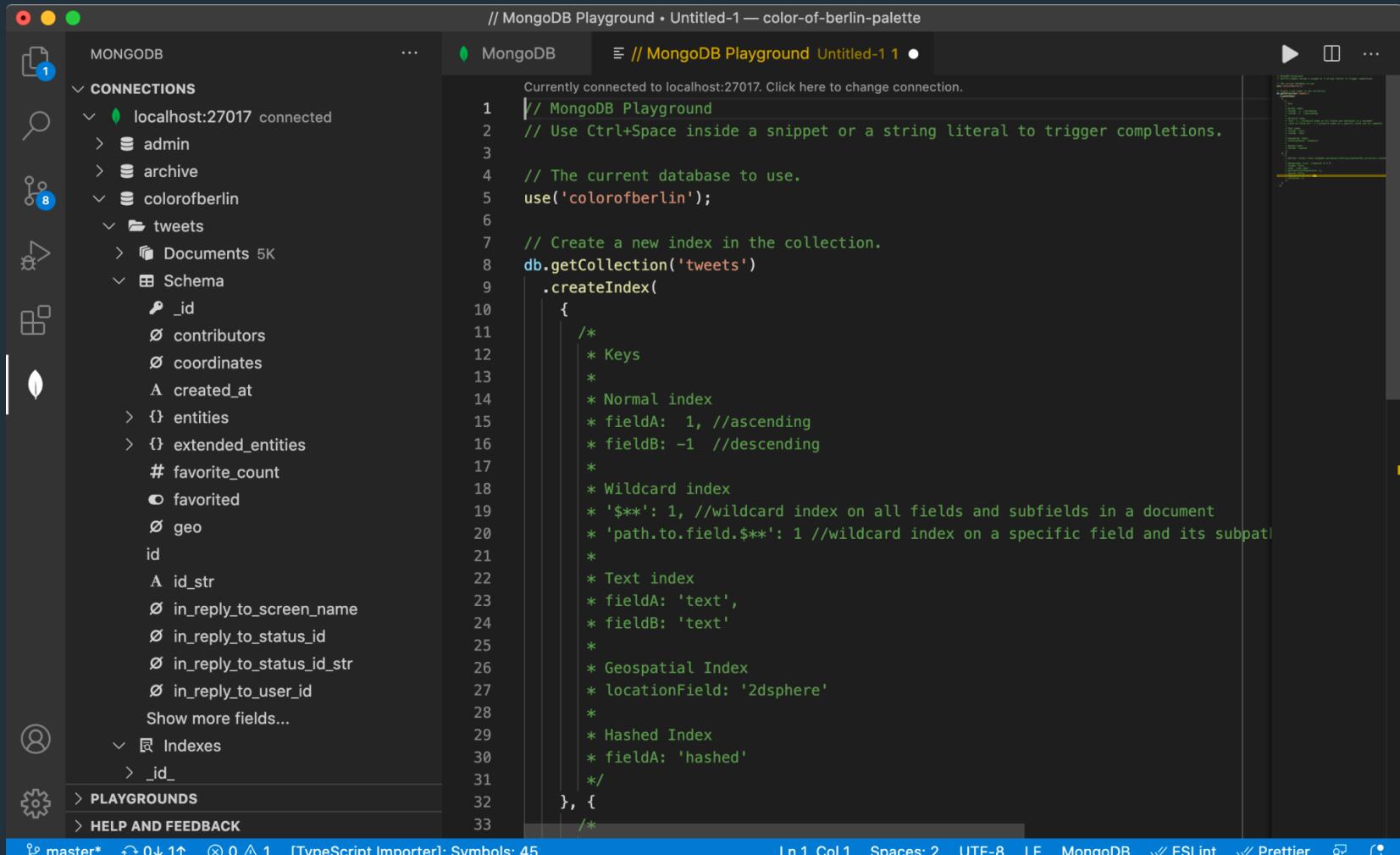
HELP AND FEEDBACK

MongoDB // MongoDB Playground Untitled-1

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

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

// Create a new index in the collection.
db.getCollection('tweets')
  .createIndex(
    {
      /*
      * Keys
      *
      * Normal index
      * fieldA: 1, //ascending
      * fieldB: -1 //descending
      *
      * Wildcard index
      * '$**': 1, //wildcard index on all fields and subfields in a document
      * 'path.to.field.$**': 1 //wildcard index on a specific field and its subpath
      *
      * Text index
      * fieldA: 'text',
      * fieldB: 'text'
      *
      * Geospatial Index
      * locationField: '2dsphere'
      *
      * Hashed Index
      * fieldA: 'hashed'
      */
    }, {
      /*
    }
```



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 × ...

```
{ 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          39,
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}
```

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"

"6035711519ad1107e53d8eef"

Show more...

Schema

- \_id
- contributors
- coordinates
- created\_at
- entities
- extended\_entities
- # favorite\_count

PLAYGROUNDS

HELP AND FEEDBACK

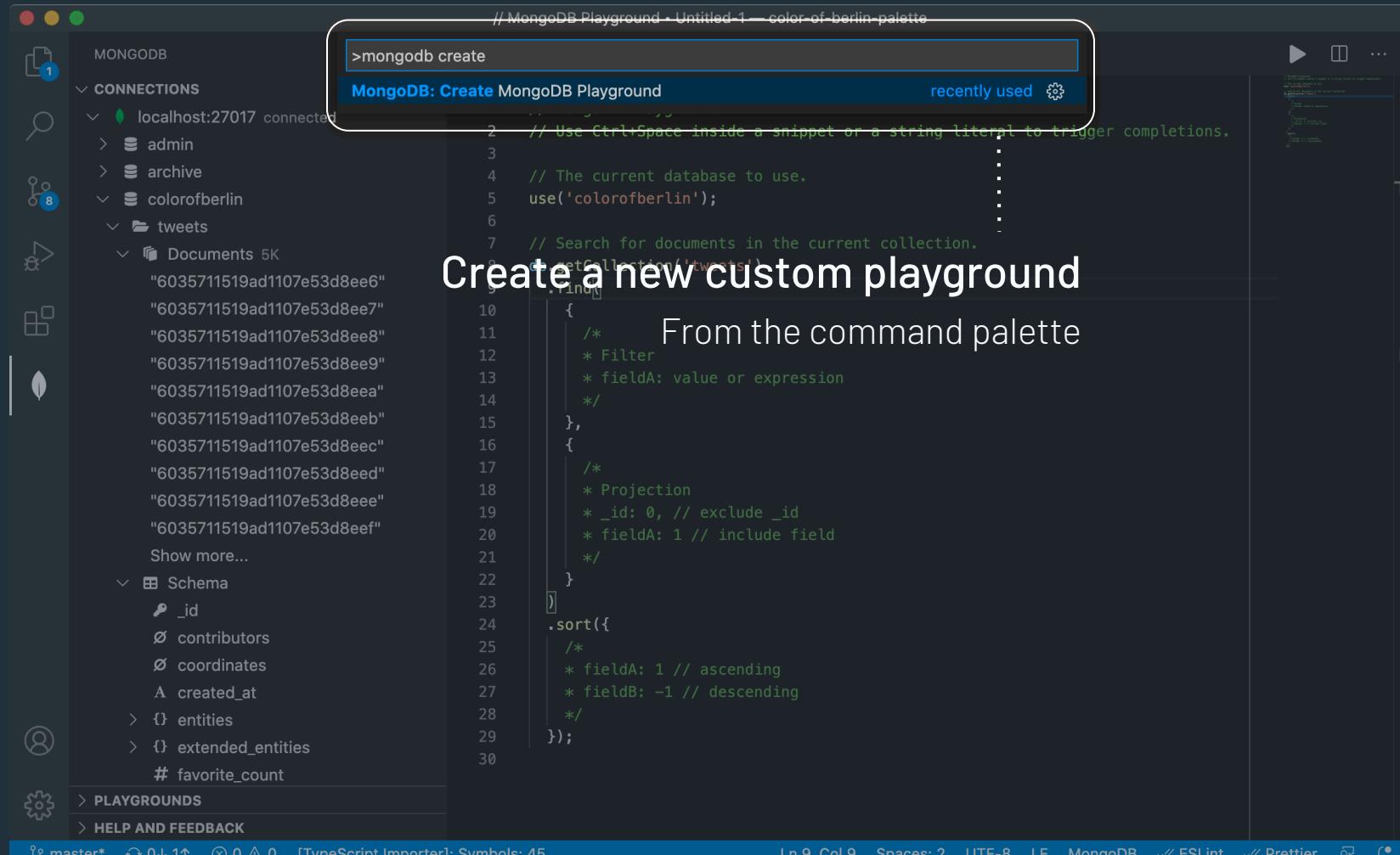
Currently connected to localhost:27017. Click here to change connection.

```
// 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



# Create a new custom playground

From the command palette

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

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 1;

33 ];

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

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

```
import Twitter from 'twitter';
```

```
...
```

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

# Prototyping

---

Explore your data, build aggregations, and run MongoDB commands

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

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

// MongoDB Playground Untitled-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" } ] } }
```

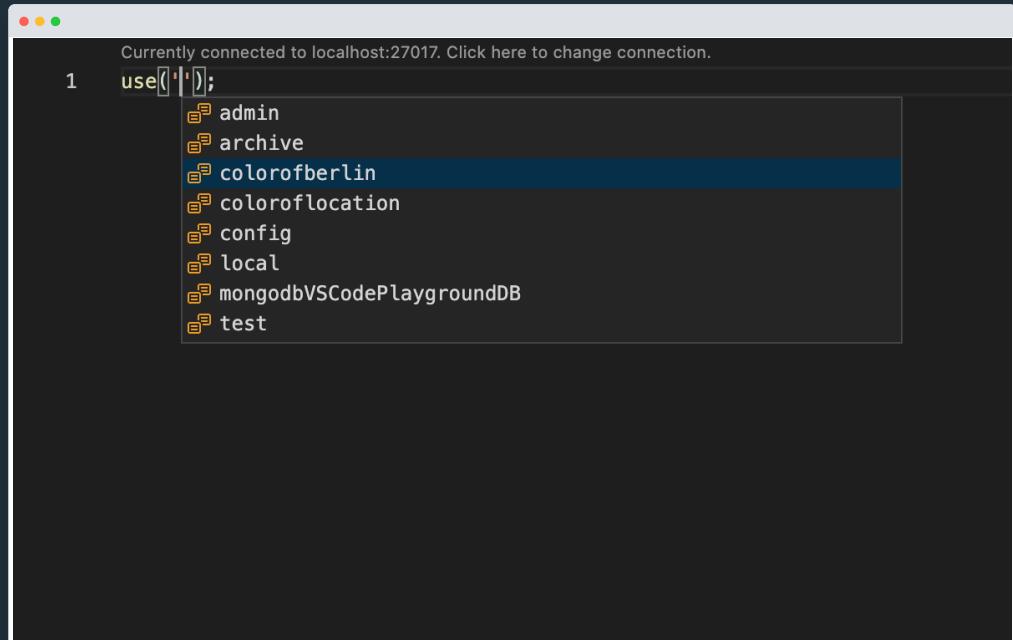
# String representation

Should be converted to a date

# Intelligent autocomplete

---

## Database names



The screenshot shows a MongoDB shell window with a dark theme. At the top, it says "Currently connected to localhost:27017. Click here to change connection." Below that, a command line starts with "1 use[" and the cursor is positioned after the opening bracket. A dropdown menu lists several database names: admin, archive, colorofberlin, coloroflocation, config, local, mongodbVSCodePlaygroundDB, and test. The "colorofberlin" database is highlighted with a blue selection bar.

```
1 use["];
  ↗ admin
  ↗ archive
  ↗ colorofberlin
  ↗ coloroflocation
  ↗ config
  ↗ local
  ↗ mongodbVSCodePlaygroundDB
  ↗ test
```

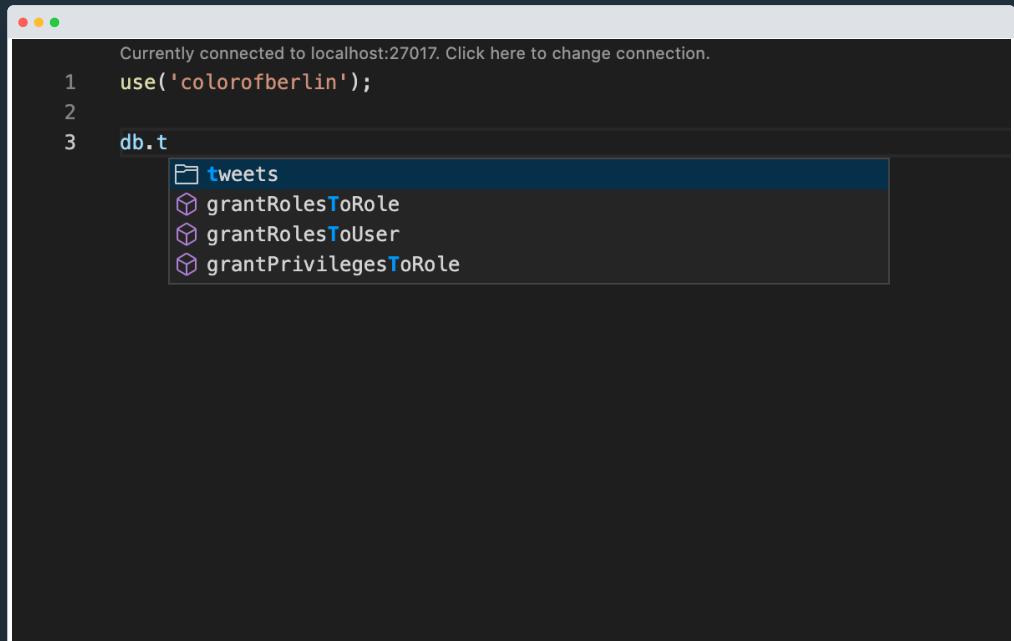
# Intelligent autocomplete

---

Database names

Collection names

Shell API



The screenshot shows a MongoDB shell window with the following content:

```
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 the command "db.t". A dropdown menu is open, listing the following items:

- tweets
- grantRolesToRole
- grantRolesToUser
- grantPrivilegesToRole

The item "tweets" is highlighted with a blue selection bar.

# Intelligent autocomplete

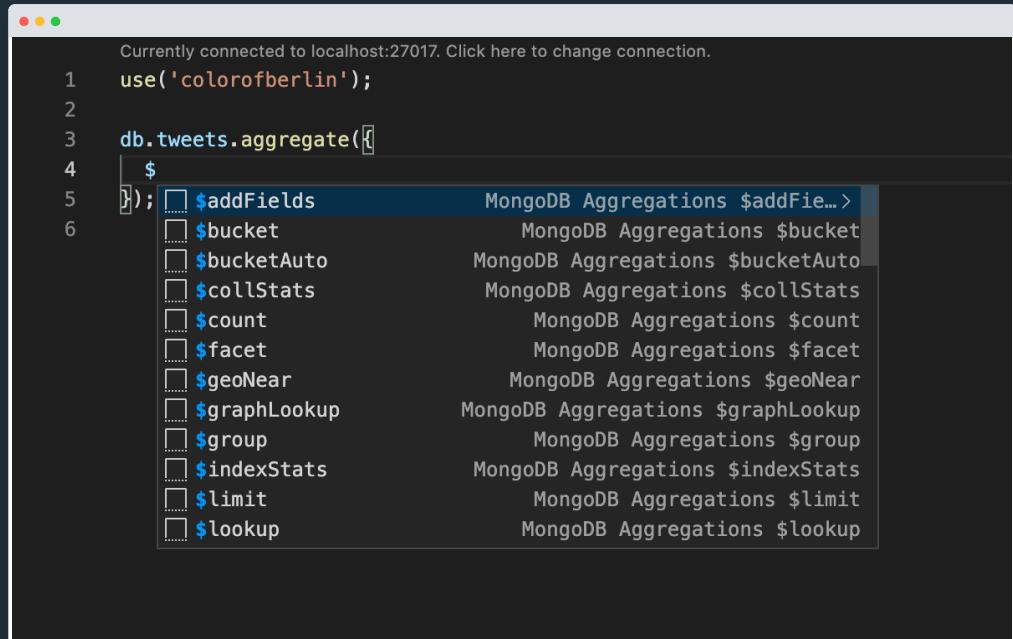
---

Database names

Collection names

Shell API

MongoDB operators



The screenshot shows the MongoDB shell interface. The command line displays:

```
1 use('colorofberlin');
2
3 db.tweets.aggregate([
4   $
5 ]);
```

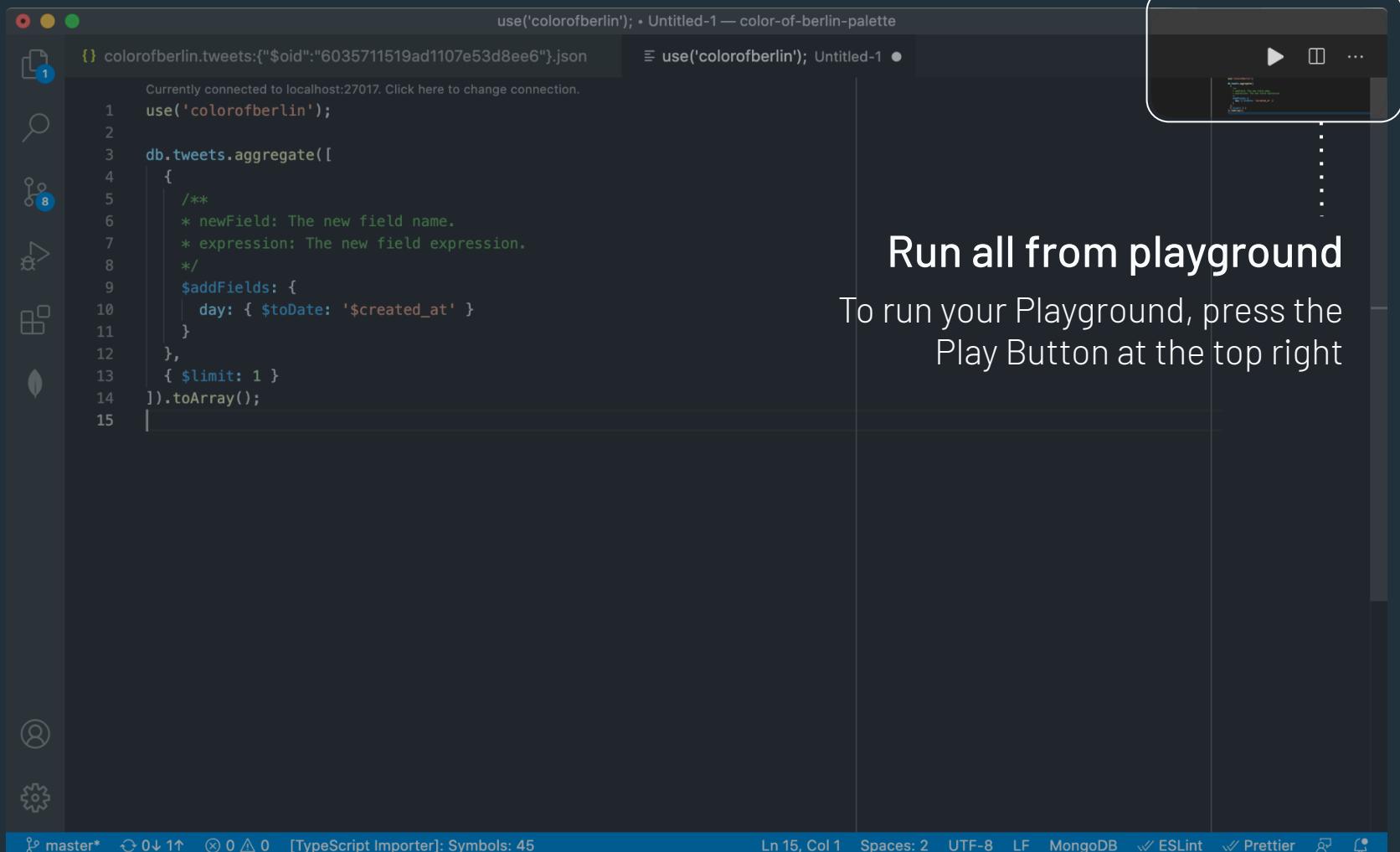
The cursor is positioned at the end of the second '\$' character in the aggregation pipeline. A dropdown menu lists various MongoDB operators, each preceded by a small checkbox icon:

- \$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 dark-themed code editor interface with a left sidebar containing various icons for file operations, search, and navigation. The main workspace displays a MongoDB aggregation pipeline:

```
use('colorofberlin'); • Untitled-1 — color-of-berlin-palette  
{} colorofberlin.tweets:{"$oid":"6035711519ad1107e53d8ee6"}.json  
use('colorofberlin'); Untitled-1 ●  
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
```

The status bar at the bottom indicates the current branch is "master\*", the file has 0 changes (0 up, 0 down), and it contains 45 symbols. It also shows the file type as "[TypeScript Importer]".



The screenshot shows a MongoDB playground interface. On the left is a sidebar with various icons: file (1), magnifying glass, wrench (B), arrow, square, leaf, user, and gear. The main area has two tabs: "colorofberlin.tweets:{\$oid":"6035711519ad1107e53d8ee6"}.json" and "use('colorofberlin'); Untitled-1". The "Untitled-1" tab contains 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();
```

Below the code editor, the status bar shows: master\* 0 0 1↑ 0 0 ▲ 0 [TypeScript Importer]: Symbols: 45 Ln 15, Col 1 Spaces: 2 UTF-8 LF MongoDB ✓ ESLint ✓ Prettier

**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.

```
use('colorofberlin');

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

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 ]

Ln 1, Col 1    Spaces: 2    JSON    ✓ ESLint    ⌂    ⌂

Playground Result — color-of-berlin-palette

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

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

```
use('colorofberlin');

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

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

master\* 0 0 1 0 0 [TypeScript Importer]: Symbols: 45    Ln 1, Col 1    Spaces: 2    JSON    ✓ ESLint

# 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: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
  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: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
  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: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: '$day' } },
  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

Run Selected Lines From Playground | 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: { $year: '$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();
```

Spaces: 2    UTF-8    LF    MongoDB    ✓ ESLint    ✓ Prettier



## ≡ use('colorofberlin'); Untitled-1 ●

▶ Run Selected Lines From Playground | Currently connected to localhost:27017. Click here to change connection.

```
1 use('colorofberlin');
2 ..... Run selected lines
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 },
16  { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
17 ]).toArray();
18
```

Check the current connection



Ξ use('colorofberlin'); Untitled-1 ●

▶ Run Selected Lines From Playground

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 },
16  { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
17 ]).toArray();
18
```

## Change active connection



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



Ξ use('colorofberlin'); Untitled-1

Select new connection...

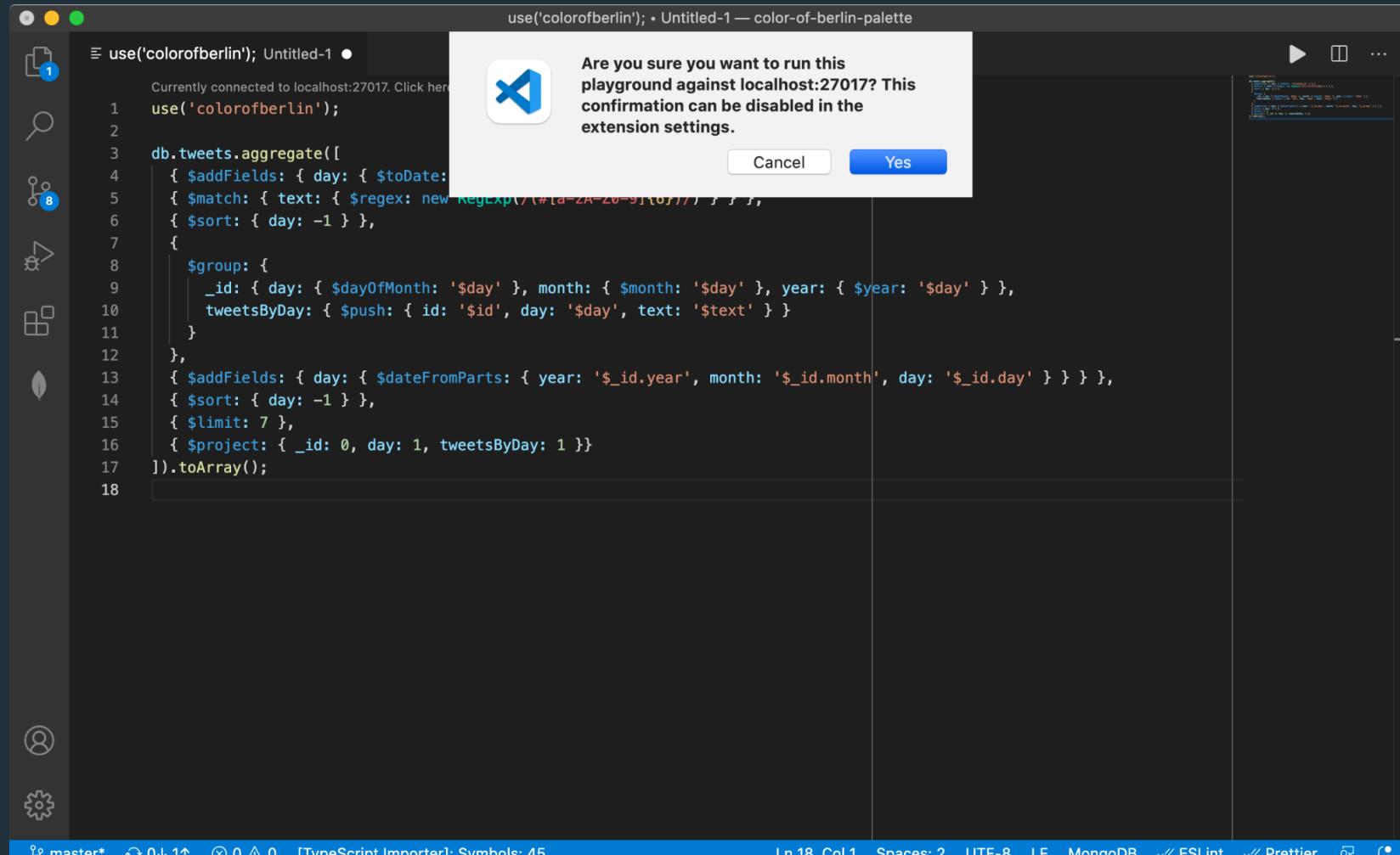
Currently connected to localhost

Add new connection

localhost:27017

```
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 },
16  { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
17 ]).toArray();
18
```





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 1 0 0 0 0 [TypeScript Importer]: Symbols: 45 Ln 1, Col 1 Spaces: 2 JSON ✓ ESLint

# More than prototyping

---

Get information on query execution, debug playgrounds, require node modules

# Debug

```
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
```

# 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: ${tweets.length}`);
console.log('First document:', `${tweets[0]}`); // First document: [object Object]
console.log('First document:', `${util.inspect(tweets[0])}`); // First document: { 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

\$regex: new RegExp(/#[a-zA-Z0-9]{6}/)

# 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');
use('colorofberlin');

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

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

## EJSON.stringify(value)

Used for serializing a BSON object to the string

```
2021-04-15T15:12:56.424Z 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

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

5 const tweets = db.tweets.aggregate([
6 { $addFields: { day: { $toDate: '$created_at' } } },
7 { $match: { text: { $regex: new RegExp(/#[a-zA-Z0-9]{6}/) } } },
8 { $sort: { day: -1 } },
9 {
10   $group: {
11     _id: { day: { $dayOfMonth: '$day' }, month: { $month: '$day' }, year: { $year: { $dateFromParts: { year: '_id.year', month: '_id.month', day: 1 } } } },
12     tweetsByDay: { $push: { id: '$id', day: '$day', text: '$text' } }
13   }
14 },
15 { $addFields: { day: { $dateFromParts: { year: '_id.year', month: '_id.month', day: 1 } } },
16 { $sort: { day: -1 } },
17 { $limit: 7 },
18 { $project: { _id: 0, day: 1, tweetsByDay: 1 } }
19 ]).toArray();
20
```

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/1234567890123456789'  
    }  
  ]  
}
```

## MongoDB playgrounds view

### PLAYGROUNDS

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

> HELP AND FEEDBACK

generic.ts — color-of-berlin-palette

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

COLOR-OF-BERLIN-PALETTE

- > dist
- > node\_modules
- > scripts
- > src
  - > client
  - > server
    - > config
    - > controllers
      - TS generic.ts M
  - > middleware
  - > startup
- TS httpException.ts
- TS index.ts
- > shared
- > utils
- ⚙ .env
- ≡ .env.example
- .eslintignore
- { } .eslintrc
- ◆ .gitignore
- { } .prettierrc
- ≡ error.playground.log U
- ≡ info.playground.log U

src > server > controllers > TS 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 
```

...    ⚙️    ⌂    ⏹

OUTLINE    TIMELINE    NPM SCRIPTS

master\*    0 0 1↑    0 0 △ 0    [TypeScript Importer]: Symbols: 45    Ln 120, Col 40    Spaces: 2    UTF-8    LF    TypeScript 4.2.3    ✘ ESLint    ⚡ Prettier

# Similar but not exactly the same

---



## MongoDB Shell

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

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.

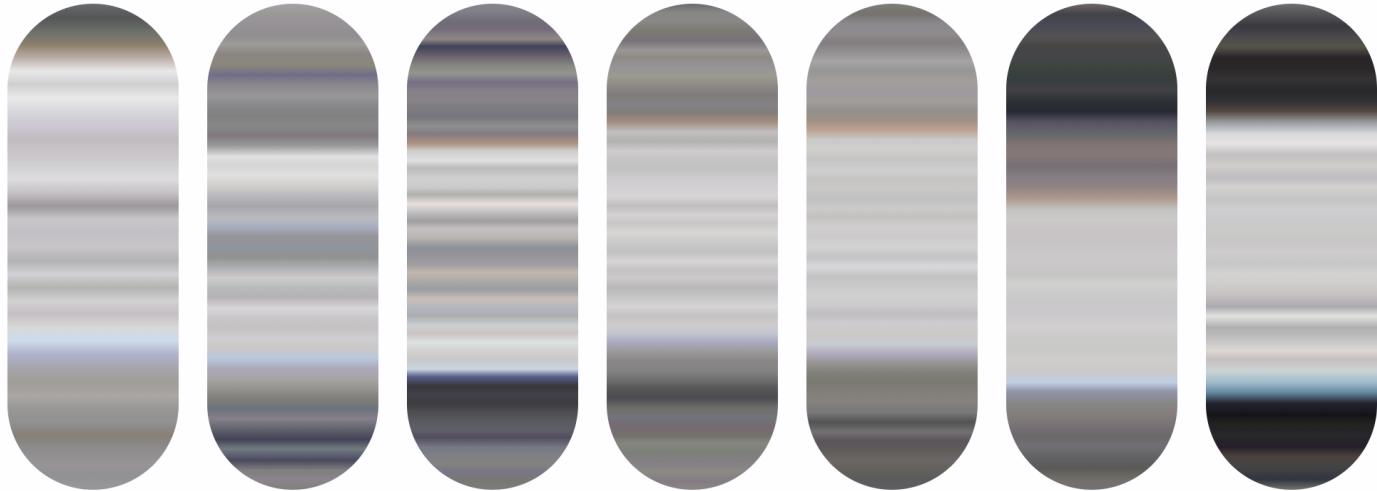


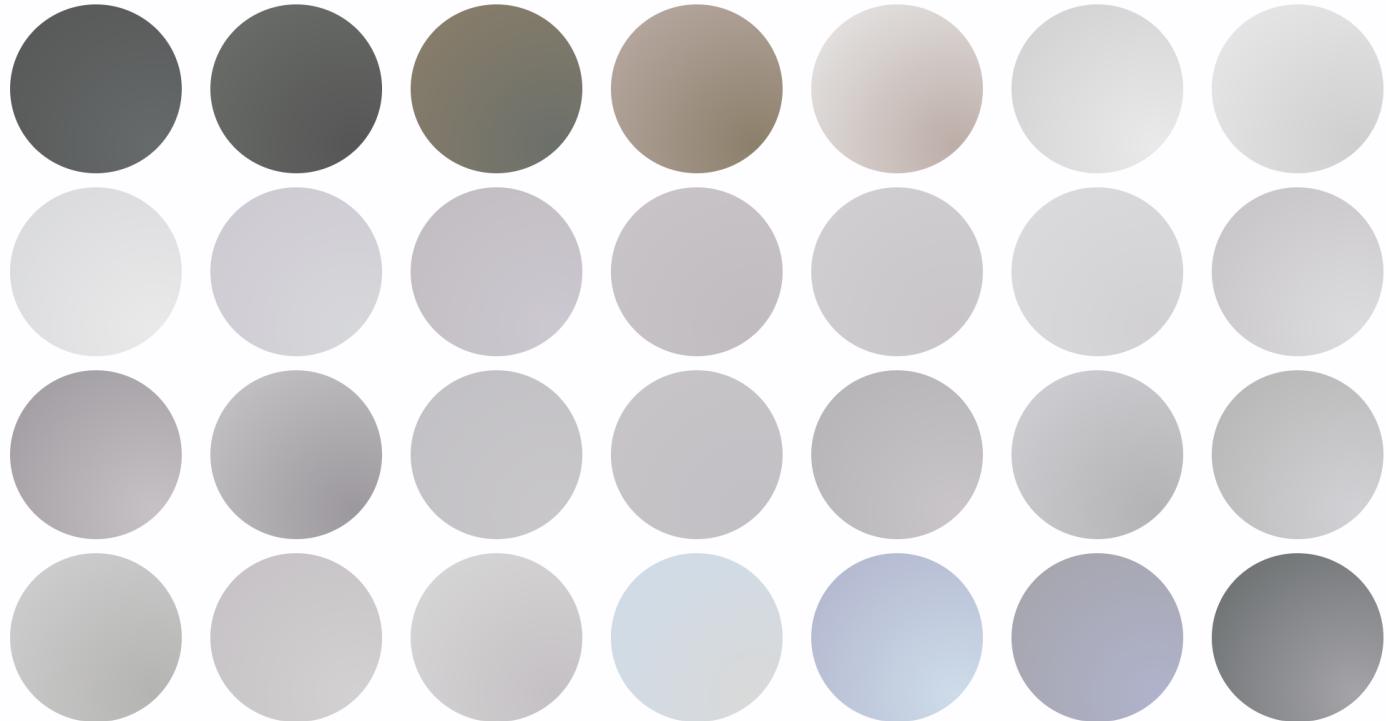
## JavaScript/TypeScript

```
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.







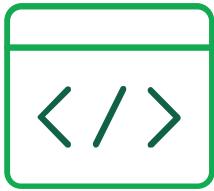
# How do playgrounds work?

---

Short story on how we built them

# Playgrounds

---

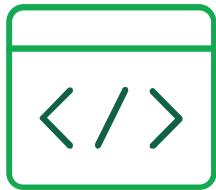


VSCode editor

Playgrounds UI

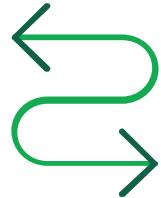
# Playgrounds

---



**VSCode editor**

Playgrounds UI

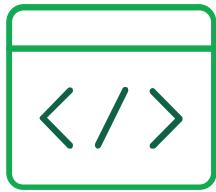


**Microsoft LSP**

Language specific features

# Playgrounds

---



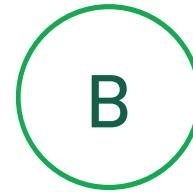
**VSCode editor**

Playgrounds UI



**Microsoft LSP**

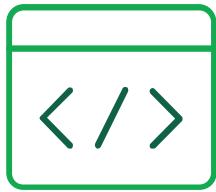
Language specific features Intelligent autocomplete



**Babel parser**

# Playgrounds

---



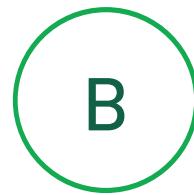
**VSCode editor**

Playgrounds UI



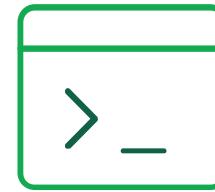
**Microsoft LSP**

Language specific features



**Babel parser**

Intelligent autocomplete



**MongoSH**

Playgrounds evaluation

The screenshot shows a desktop environment with three windows open:

- MongoDB Compass Dev - localhost:27017**: A window titled "Local" showing the "Databases" tab. It lists 11 databases and 16 collections. The "admin" database is selected, showing 32.0KB storage size, 0 collections, and 1 index.
- mongosh**: A terminal window showing the output of a mongosh session. It includes connection details, MongoDB version, and a note about using Mongosh Beta. It also lists startup warnings related to access control, binding, and soft rlimits.
- VS Code**: A code editor window showing a TypeScript file with code related to "colorofberlin". The terminal tab at the bottom shows the command "switched to db colorofberlin".

**mongosh Terminal Output:**

```
Last login: Thu Apr 29 12:31:10 on ttys001
alenakhineika@Alenas-MBP ~ % mongosh
Current Mongosh Log ID: 608ae8679309ee6c19bd45be
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true
Using MongoDB: 4.4.4
Using Mongosh Beta: 0.7.7

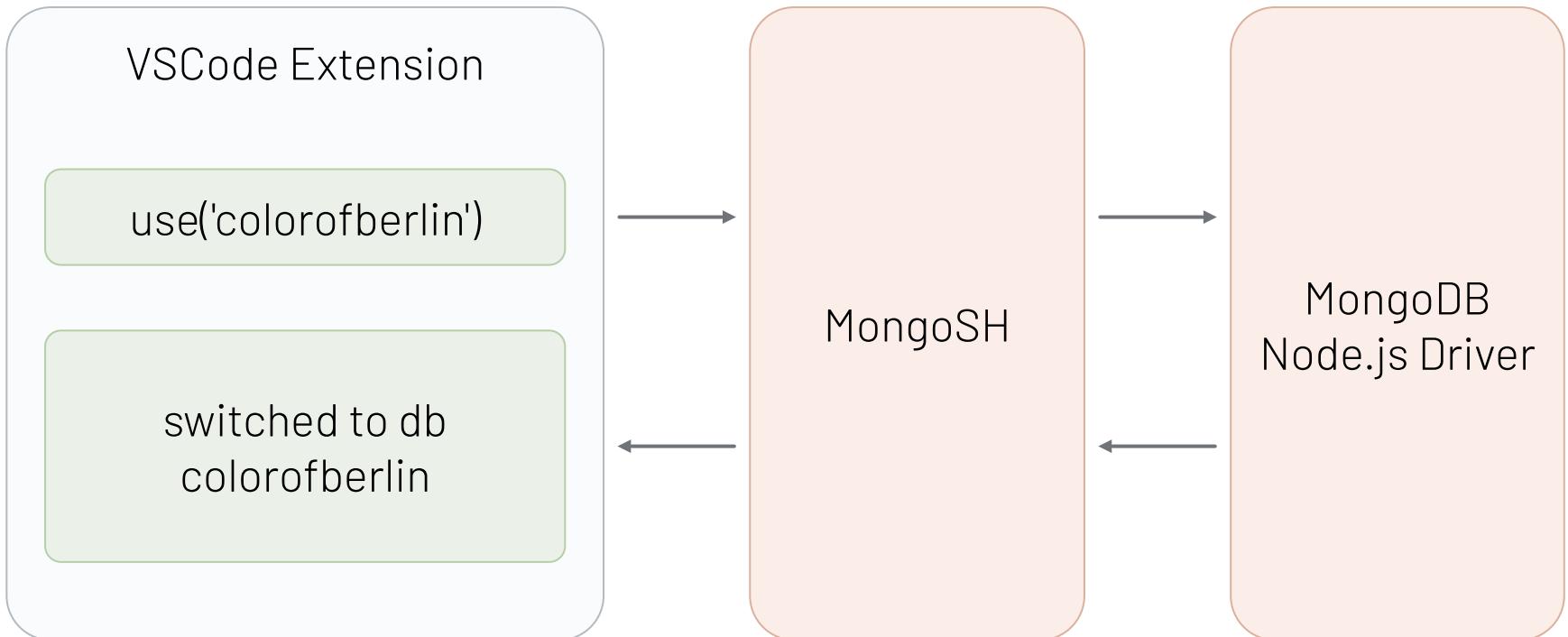
For more information about mongosh, please see our docs: https://docs.mongodb.com/mongodb-shell/
-----
The server generated these startup warnings when booting:
2021-04-29T19:09:18.057+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2021-04-29T19:09:18.057+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
2021-04-29T19:09:18.057+02:00: Soft rlimits too low
-----
```

**VS Code Terminal Output:**

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

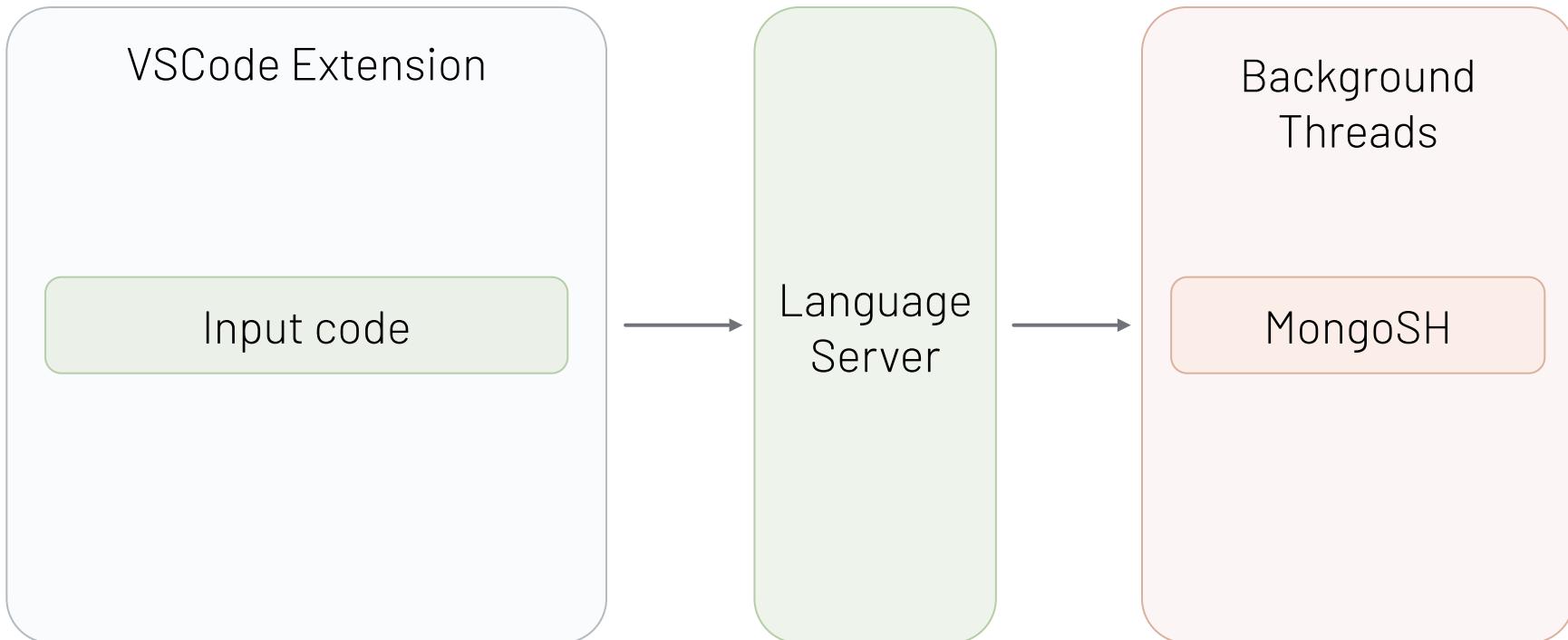
# Code Evaluation with MongoSH

---



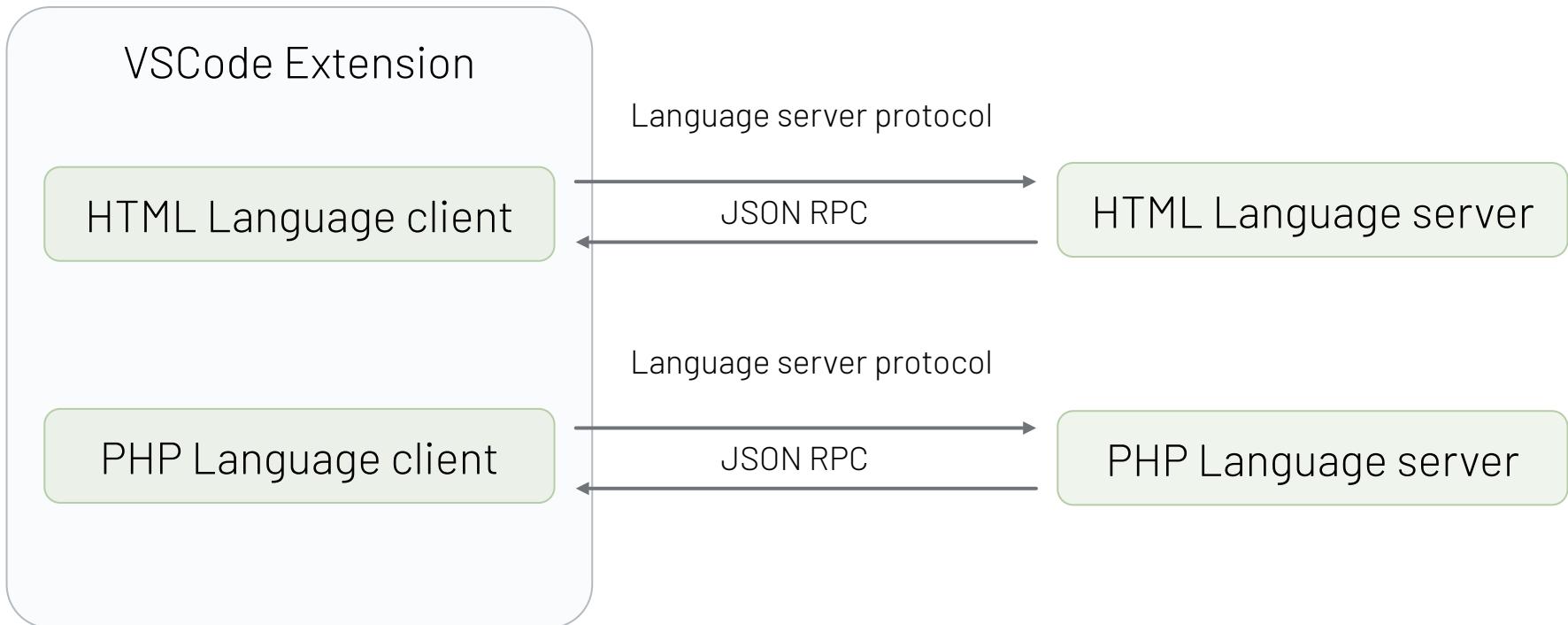
# Layers of application

---



# Language Server

---

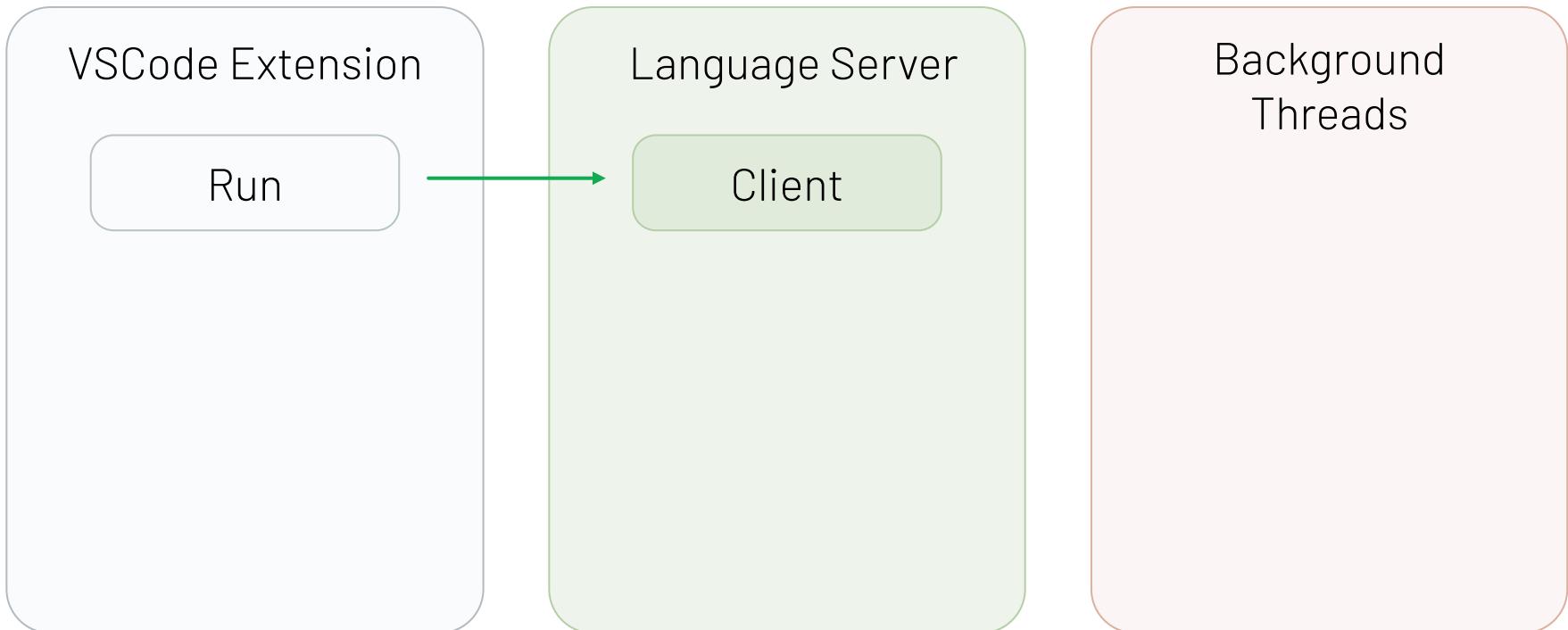


# Why background threads?

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

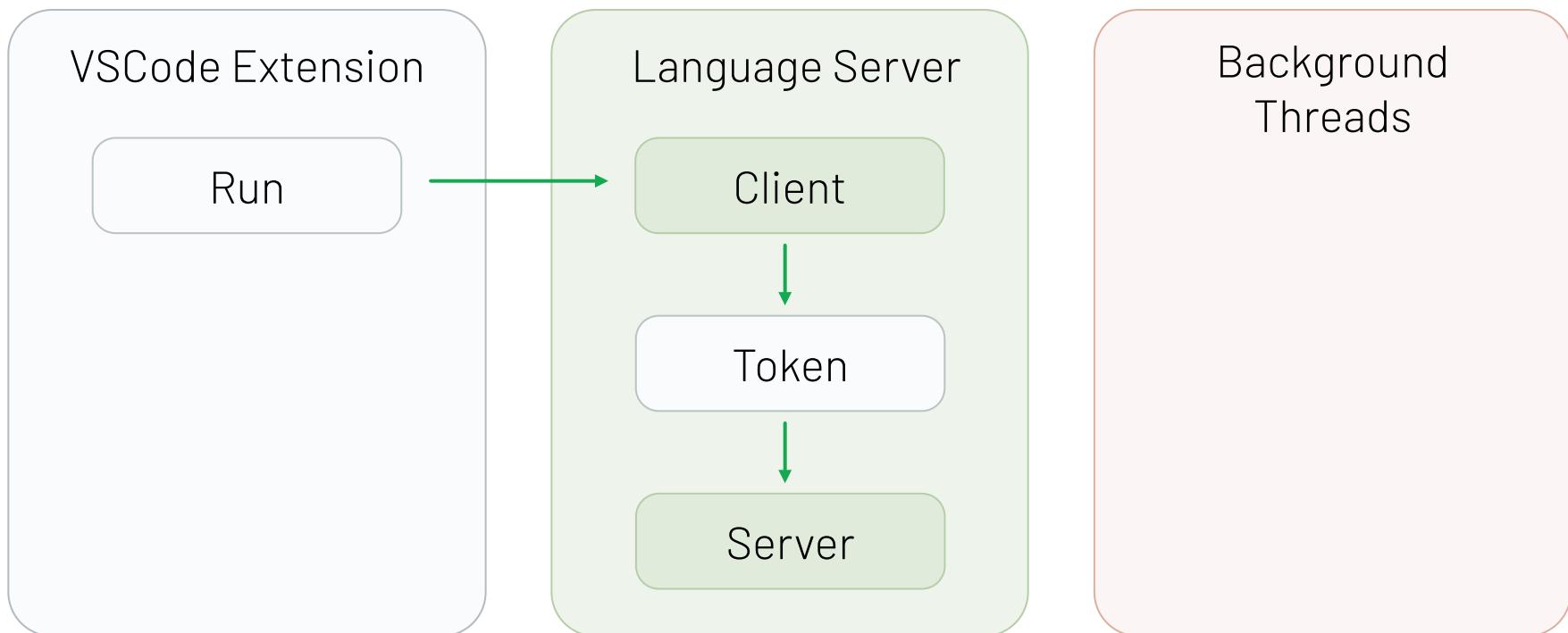
# Run Request

---



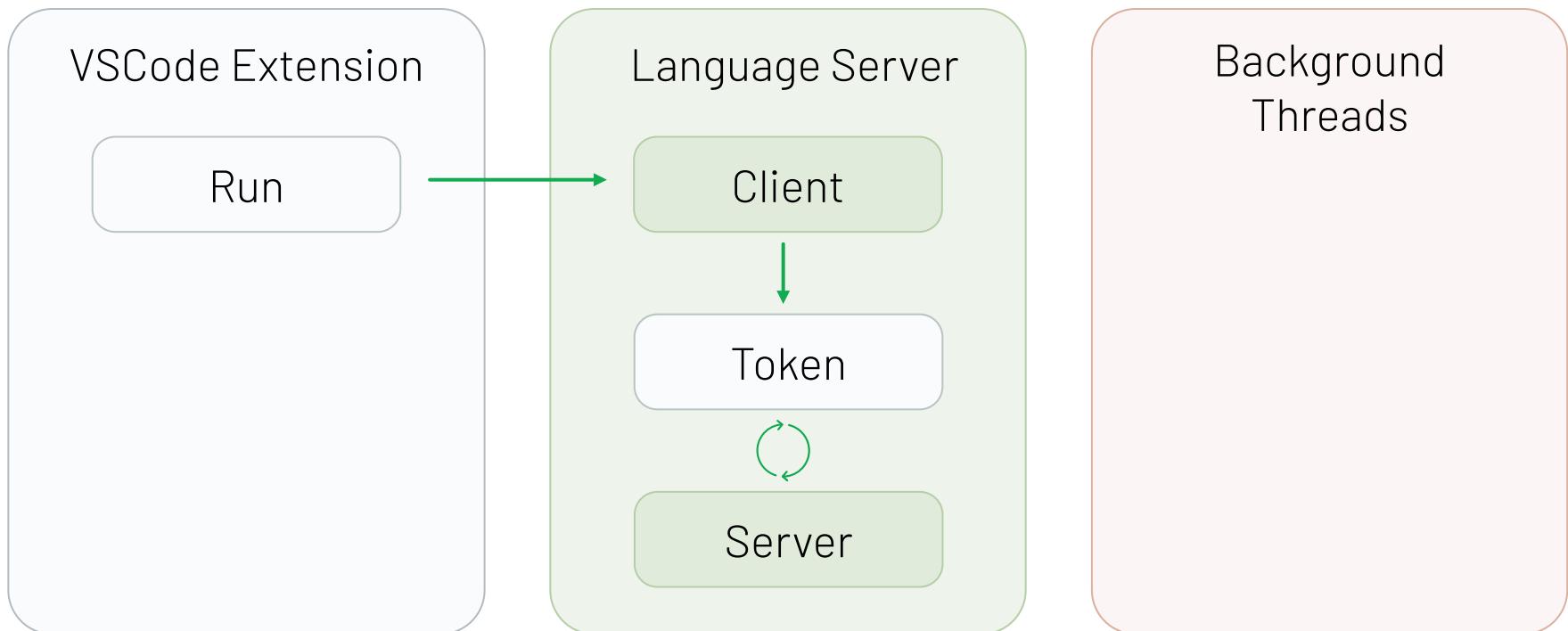
# Add cancellation token

---



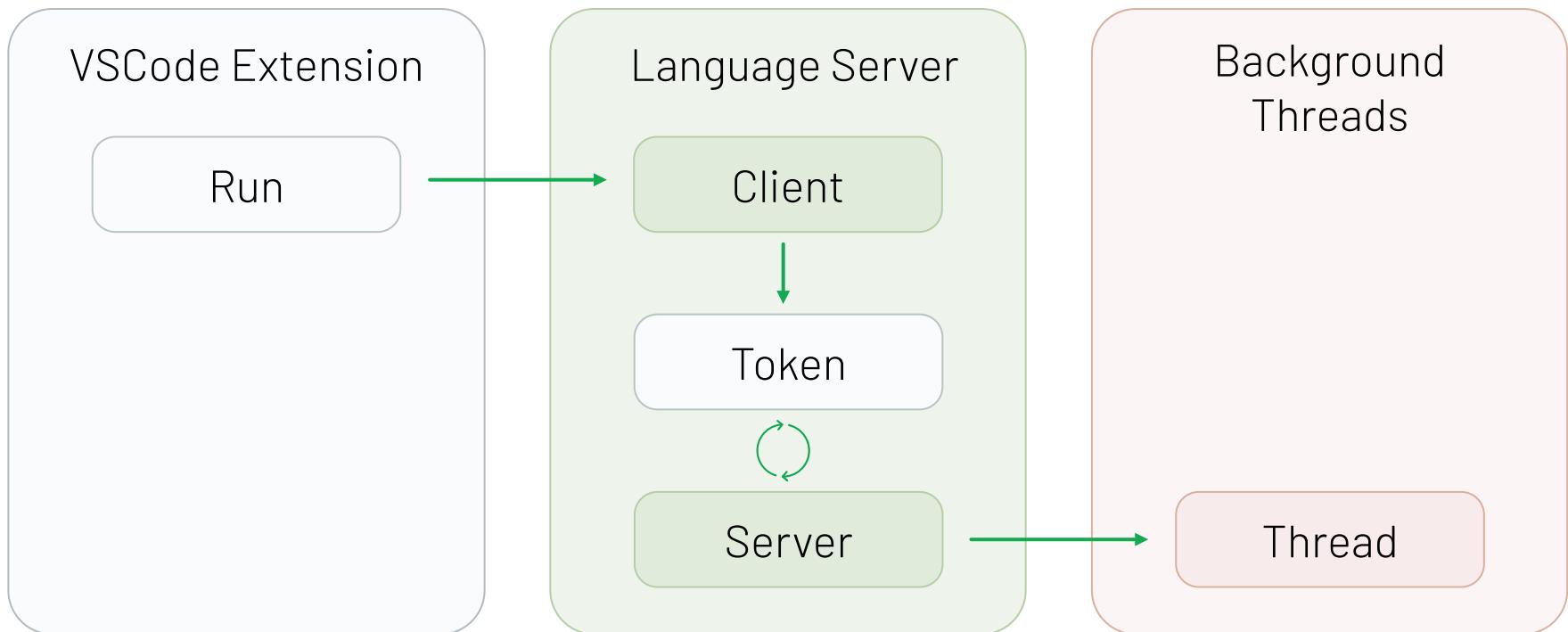
# Listen for cancellation events

---



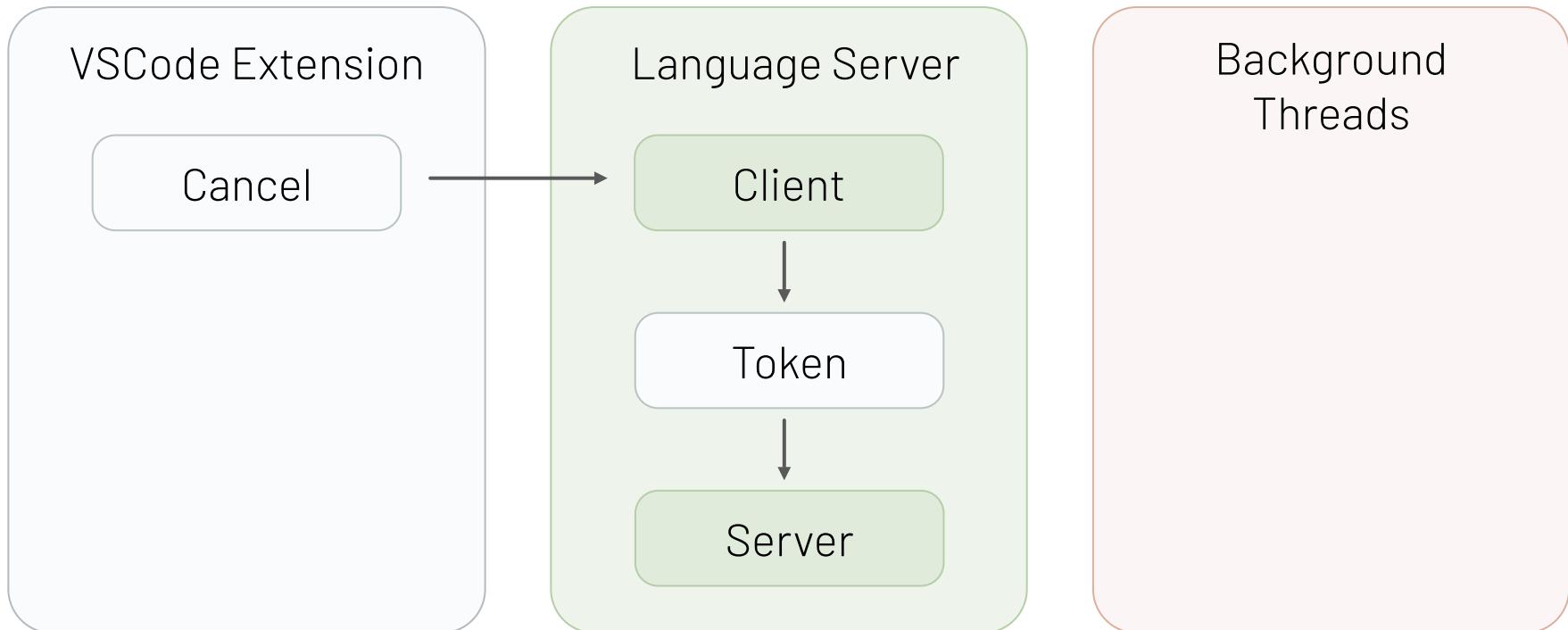
# Create background thread

---



# Cancel Request

---



A screenshot of a GitHub repository page for the MongoDB extension for VS Code. The repository is named "mongodb-js/vscode".

The page shows the following details:

- Code tab** is selected.
- Issues**: 21
- Pull requests**: 1
- Actions**
- Security**
- Insights**
- Settings**

The main content area displays a list of recent commits:

Author	Commit Message	Date
alenakhineika	fix: save big files without error and preserve play...	7 days 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

**About** section:

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...](#)

Tags:

- database
- mongodb
- terraform
- vscode
- vscode-extension
- mongodb-atlas

# VSCode

# MongoDB extension for VSCode

---

Connect to MongoDB and work with data (CRUD)

# MongoDB extension for VSCode

---

Connect to MongoDB and work with data (CRUD)

Benefit from playgrounds (powered by MongoSH and LSP)

# MongoDB extension for VSCode

---

Connect to MongoDB and work with data (CRUD)

Benefit from playgrounds (powered by MongoSH and LSP)

Onboard new developers (syntax highlighting, autocomplete, snippets, debug)

# MongoDB extension for VSCode

---

Connect to MongoDB and work with data (CRUD)

Benefit from playgrounds (powered by MongoSH and LSP)

Onboard new developers (syntax highlighting, autocomplete, snippets, debug)

Use embedded MongoDB Shell

# MongoDB extension for VSCode

---

Connect to MongoDB and work with data (CRUD)

Benefit from playgrounds (powered by MongoSH and LSP)

Onboard new developers (syntax highlighting, autocomplete, snippets, debug)

Use embedded MongoDB Shell

Feel productive and relaxed when you build applications

Thank you!