

# Building Real-time Applications with Kuzzle: An Open-Source Backend Solution

---

By

Raju Karmuri



SAINT LOUIS  
UNIVERSITY™



# What is Kuzzle?

---

- Kuzzle is an API that is an open-source ready-to-use backend platform designed to facilitate the development of scalable, real-time applications in no time.
- Kuzzle is built using Node.js and uses Elastic Search(*Scalable*) and Redis services(*Fast*).
- Kuzzle can leverage Elasticsearch as its primary data storage backend. Elasticsearch is well-suited for handling and querying large amounts of data, making it an excellent choice for applications that require efficient searching and filtering.
- Kuzzle uses Redis services for caching frequently accessed data to improve performance or acting as a message broker to facilitate communication between different parts of an application.

# Benefits of Kuzzle

---

When you start Kuzzle, you automatically have access to an API exposing a wide range of features:

- Data Storage and access 
- Advanced permission system 
- Multi Authentication
- Multi Protocol API
- Real-time Engine 
- Integrated cluster mode

# Install Kuzzle

---

- Prerequisites: Node.js, Docker, Kourou, and Docker compose
- Kuzzle can be run in 2 ways. They are:
  - Kuzzle CLI i.e., using Kourou: `npm install -g kourou` (or try: `npm install -g kuzzle`)
  - Kuzzle Admin Console: <https://next-console.kuzzle.io/#/create-connection>
- *Note: You must have Kuzzle running in the background to access Admin console*

# Install Kuzzle using CLI

- Command: `kourou app:scaffold playground`

```
rajukarmuri@Rajus-Air Kuzzle Workshop % kourou app:scaffold playground
(node:14864) [DEP0040] DeprecationWarning: The `punycode` module is deprecated.
Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)

🚀 Kourou – Scaffolds a new Kuzzle application

generic
✓ Checking destination
✓ Prepare temporary folder
✓ Cloning template repository
✓ Copying template files
✓ Cleaning up

[✓] Scaffolding complete!
[✓] Use cd playground && docker compose up -d to start your Kuzzle stack.
rajukarmuri@Rajus-Air Kuzzle Workshop %
```

# Run Kuzzle using CLI

```
[rajukarmuri@Rajus-Air playground % kourou app:start-services
(node:15110) [DEP0040] DeprecationWarning: The `punycode` module is deprecated.
Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)]
```

🚀 **Kourou** – Starts Kuzzle services (Elasticsearch and Redis)

Writing the Docker Compose file to /Users/rajukarmuri/.kourou/kuzzle-services.yml...

[✓] Elasticsearch and Redis are booting in the background right now.

To watch the logs, run

```
docker compose -f /Users/rajukarmuri/.kourou/kuzzle-services.yml logs -f
```

Elasticsearch port: 9200

Redis port: 6379

```
rajukarmuri@Rajus-Air playground %
```

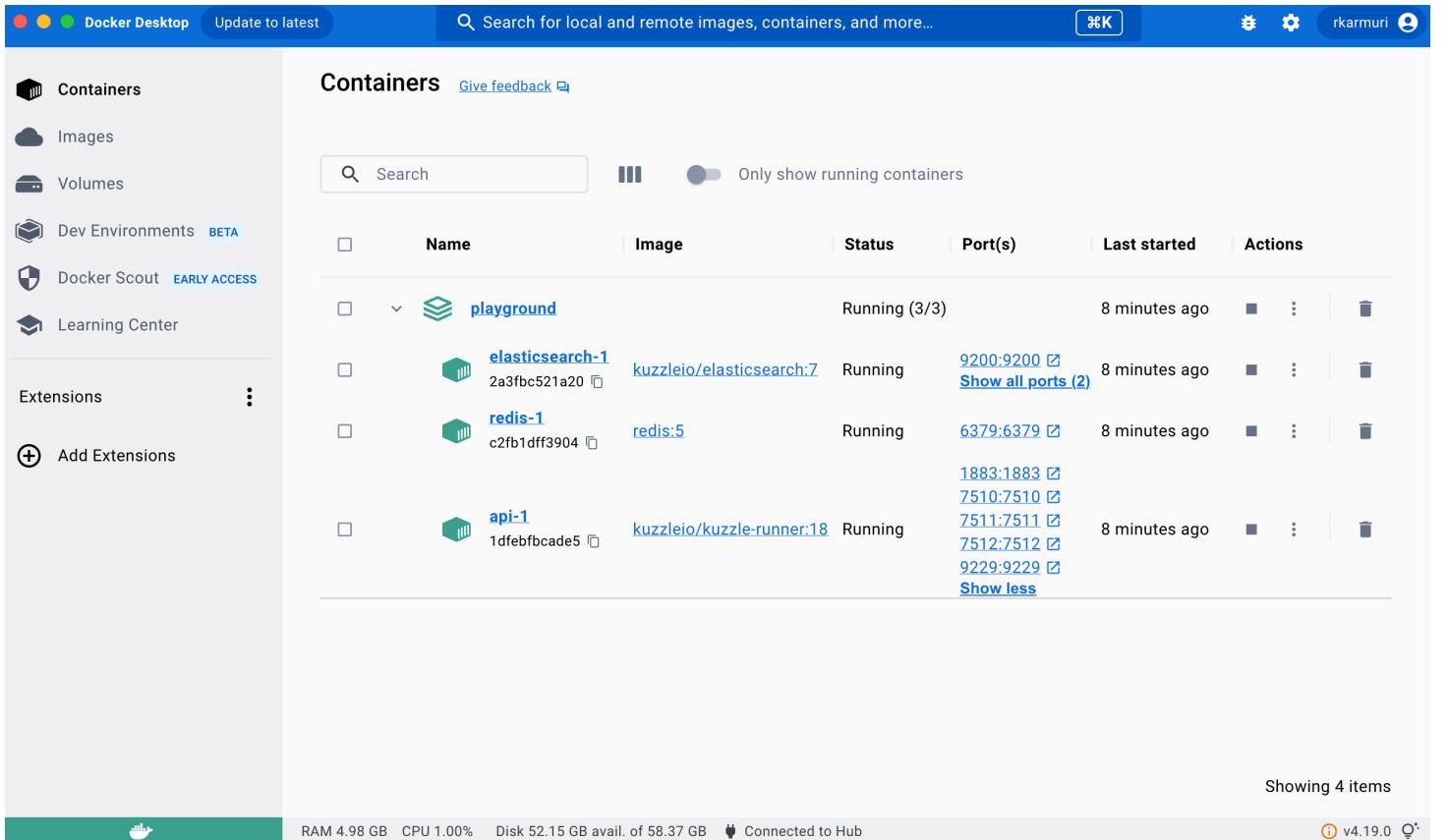


# Run Kuzzle using CLI

```
[✓] Scaffolding complete!
[✓] Use cd playground && docker compose up -d to start your Kuzzle stack.
rajukarmuri@Rajus-Air Kuzzle Workshop % cd playground
rajukarmuri@Rajus-Air playground % docker compose up -d
[+] Running 28/3
[✓] api 9 layers [██████████]    0B/0B    Pulled          24.2s
[✓] redis 6 layers [██████]     0B/0B    Pulled          4.0s
[✓] elasticsearch 10 layers [██████████████]   0B/0B    Pulled         19.2s
[+] Running 4/4
[✓] Network playground_default      C...          0.1s
[✓] Container playground-elasticsearch-1 Healthy      17.0s
[✓] Container playground-redis-1    Healthy      6.5s
[✓] Container playground-api-1     S...          17.0s
rajukarmuri@Rajus-Air playground % docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
1dfebfbcade5      kuzzleio/kuzzle-runner:18 "docker-entrypoint.s..." 3 minutes ago   Up 3 minutes (healthy)  0.0.0.0:18
[83->1883/tcp, 0.0.0.0:7510-7512->7510-7512/tcp, 0.0.0.0:9229->9229/tcp] playground-api-1
2a3fbcc521a20     kuzzleio/elasticsearch:7  "/bin/tini -- /usr/l..." 3 minutes ago   Up 3 minutes (healthy)  0.0.0.0:92
00->9200/tcp, 0.0.0.0:9300->9300/tcp                                         playground-elasticsearch-1
c2fb1dff3904      redis:5                  "docker-entrypoint.s..." 3 minutes ago   Up 3 minutes (healthy)  0.0.0.0:63
79->6379/tcp
rajukarmuri@Rajus-Air playground %
```

# Run Kuzzle using CLI

Open *localhost:7512*



The screenshot shows the Docker Desktop interface with the following details:

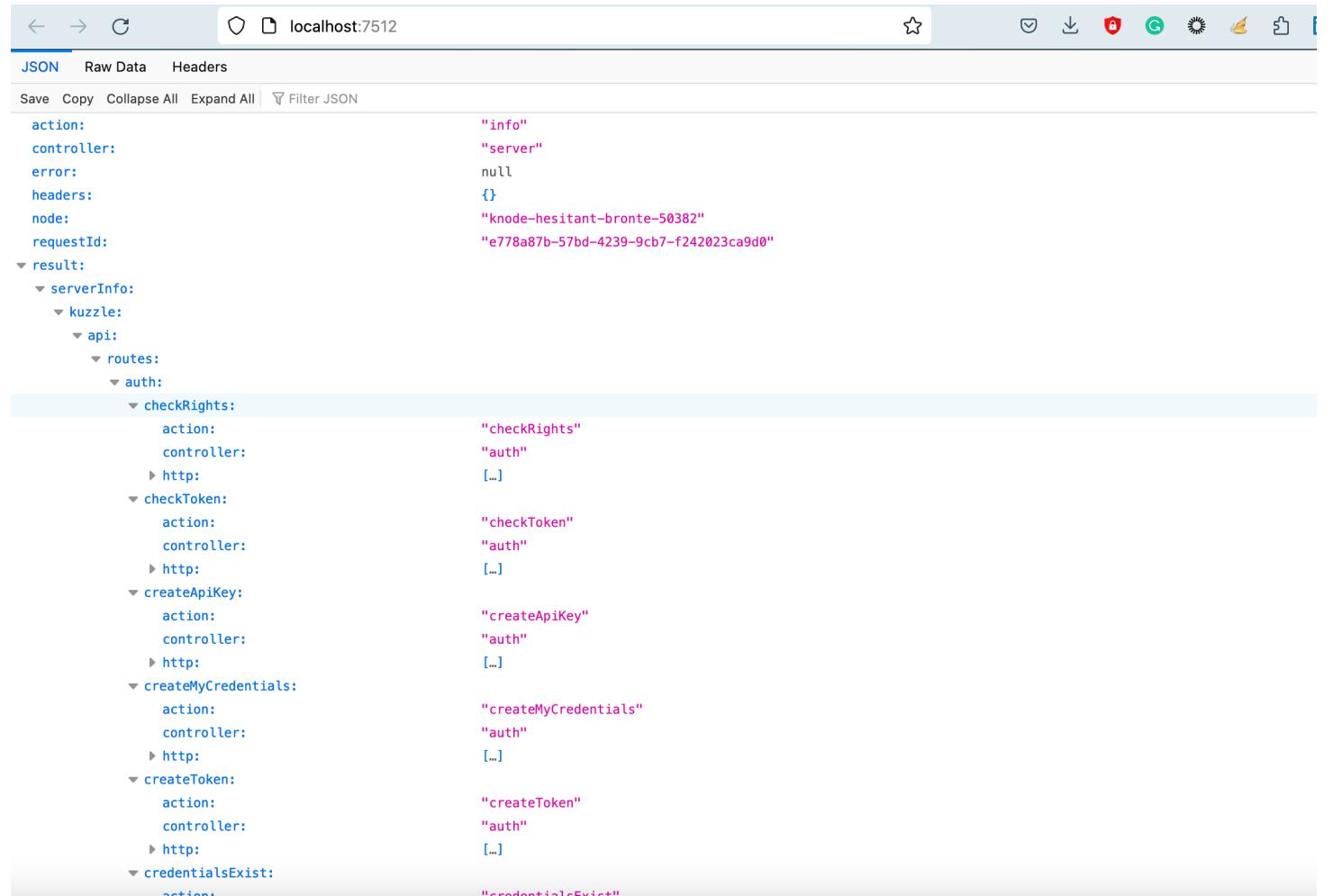
- Containers** sidebar: Includes options for Containers, Images, Volumes, Dev Environments (BETA), Docker Scout (EARLY ACCESS), and Learning Center.
- Extensions** sidebar: Shows "Add Extensions".
- Containers Table**: A list of running containers with the following data:

Name	Image	Status	Port(s)	Last started	Actions
playground	kuzzleio/kuzzle-runner:18	Running (3/3)	8 minutes ago	<span>... </span>	
elasticsearch-1	kuzzleio/elasticsearch:7	Running	9200:9200 ↗ <a href="#">Show all ports (2)</a>	8 minutes ago	<span>... </span>
redis-1	redis:5	Running	6379:6379 ↗	8 minutes ago	<span>... </span>
api-1	kuzzleio/kuzzle-runner:18	Running	1883:1883 ↗ 7510:7510 ↗ 7511:7511 ↗ 7512:7512 ↗ 9229:9229 ↗ <a href="#">Show less</a>	8 minutes ago	<span>... </span>
- Footer**: RAM 4.98 GB, CPU 1.00%, Disk 52.15 GB avail. of 58.37 GB, Connected to Hub, v4.19.0



# Run Kuzzle using CLI

Successfully connected!!!



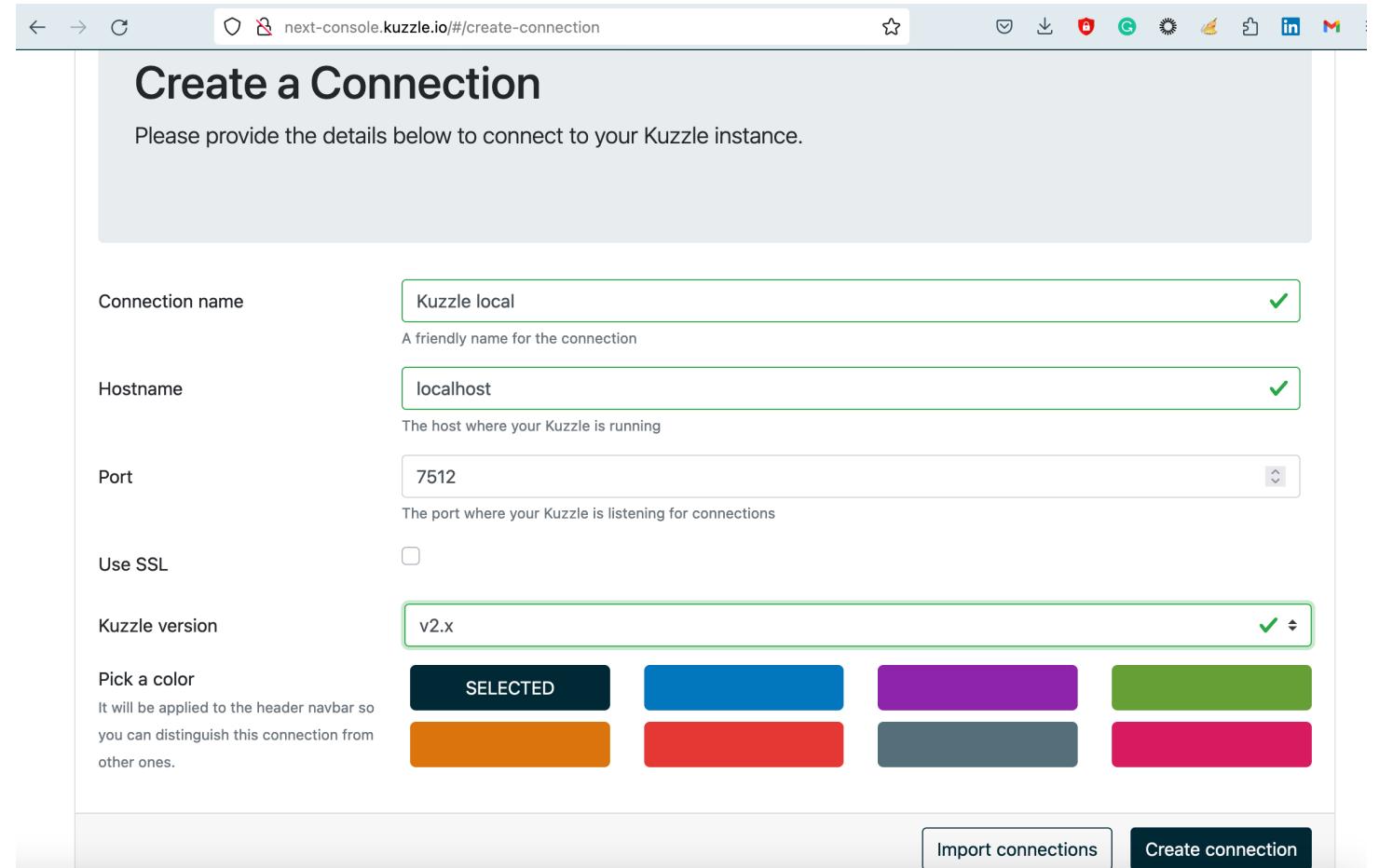
```
action: "info"
controller: "server"
error: null
headers: {}
node: "knode-hesitant-bronte-50382"
requestId: "e778a87b-57bd-4239-9cb7-f242023ca9d0"
result:
  serverInfo:
    kuzzle:
      api:
        routes:
          auth:
            checkRights:
              action: "checkRights"
              controller: "auth"
              http: [...]
            checkToken:
              action: "checkToken"
              controller: "auth"
              http: [...]
            createApiKey:
              action: "createApiKey"
              controller: "auth"
              http: [...]
            createMyCredentials:
              action: "createMyCredentials"
              controller: "auth"
              http: [...]
            createToken:
              action: "createToken"
              controller: "auth"
              http: [...]
            credentialsExist:
              action: "credentialsExist"
```



# Login to Kuzzle Admin Console



You must have Kuzzle running!!!  
Login as anonymous!!!



The screenshot shows a web browser window with the URL `next-console.kuzzle.io/#/create-connection`. The page title is "Create a Connection" and the sub-instruction is "Please provide the details below to connect to your Kuzzle instance." The form fields are as follows:

- Connection name:** Kuzzle local (with a green checkmark)
- Hostname:** localhost (with a green checkmark)
- Port:** 7512
- Use SSL:** An unchecked checkbox.
- Kuzzle version:** v2.x (highlighted with a green border and a green checkmark)
- Pick a color:** A note stating "It will be applied to the header navbar so you can distinguish this connection from other ones." Below this are five color swatches: dark blue (selected), blue, purple, orange, red, grey, and pink.

At the bottom right are two buttons: "Import connections" and "Create connection".



# Initial Kuzzle Admin Console

The screenshot shows the Kuzzle Admin Console interface. At the top, there is a navigation bar with the Kuzzle logo, a "Data" menu item, and user authentication information ("Anonymous on Kuzzle local"). On the left, a sidebar titled "Indexes" lists "All indexes" and shows a single entry: "chat" under the "data" category. A search bar at the top left is labeled "Search index & collection". The main content area is titled "Indexes" and displays a table with one row. The table has columns for "Name" and "Collections". The "Name" column contains "chat", and the "Collections" column contains "--". To the right of the table are three icons: a magnifying glass, a plus sign, and a trash can.

Name	Collections
chat	--

# Storing and Accessing data

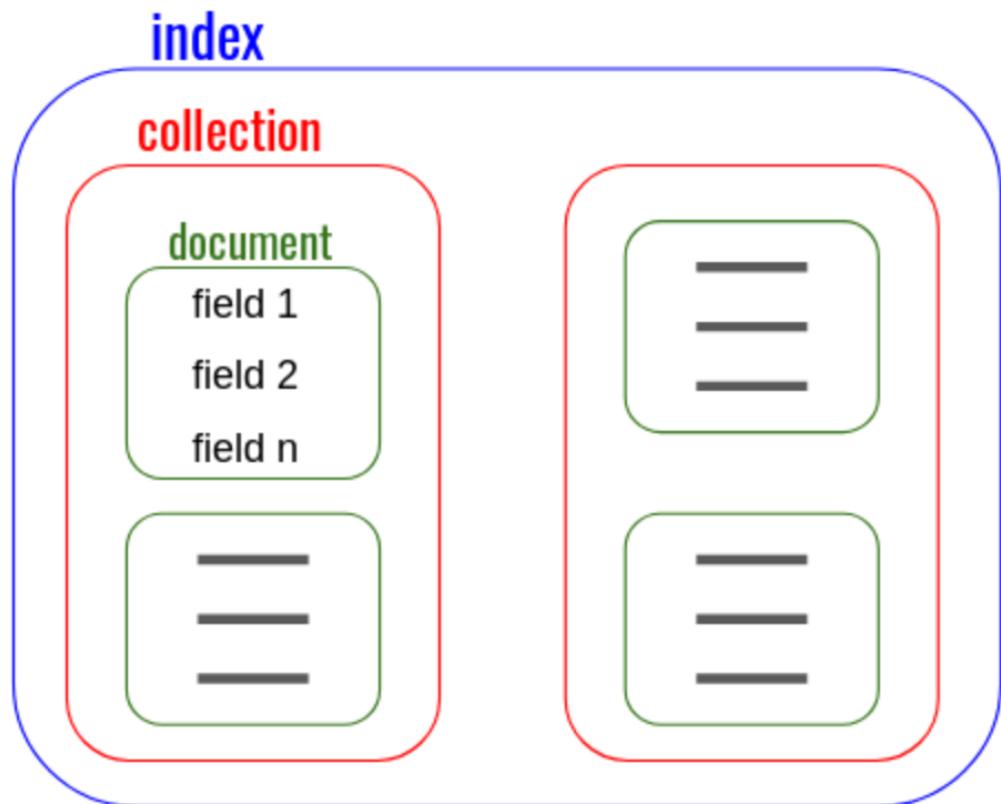
---



# Automatic Database Access

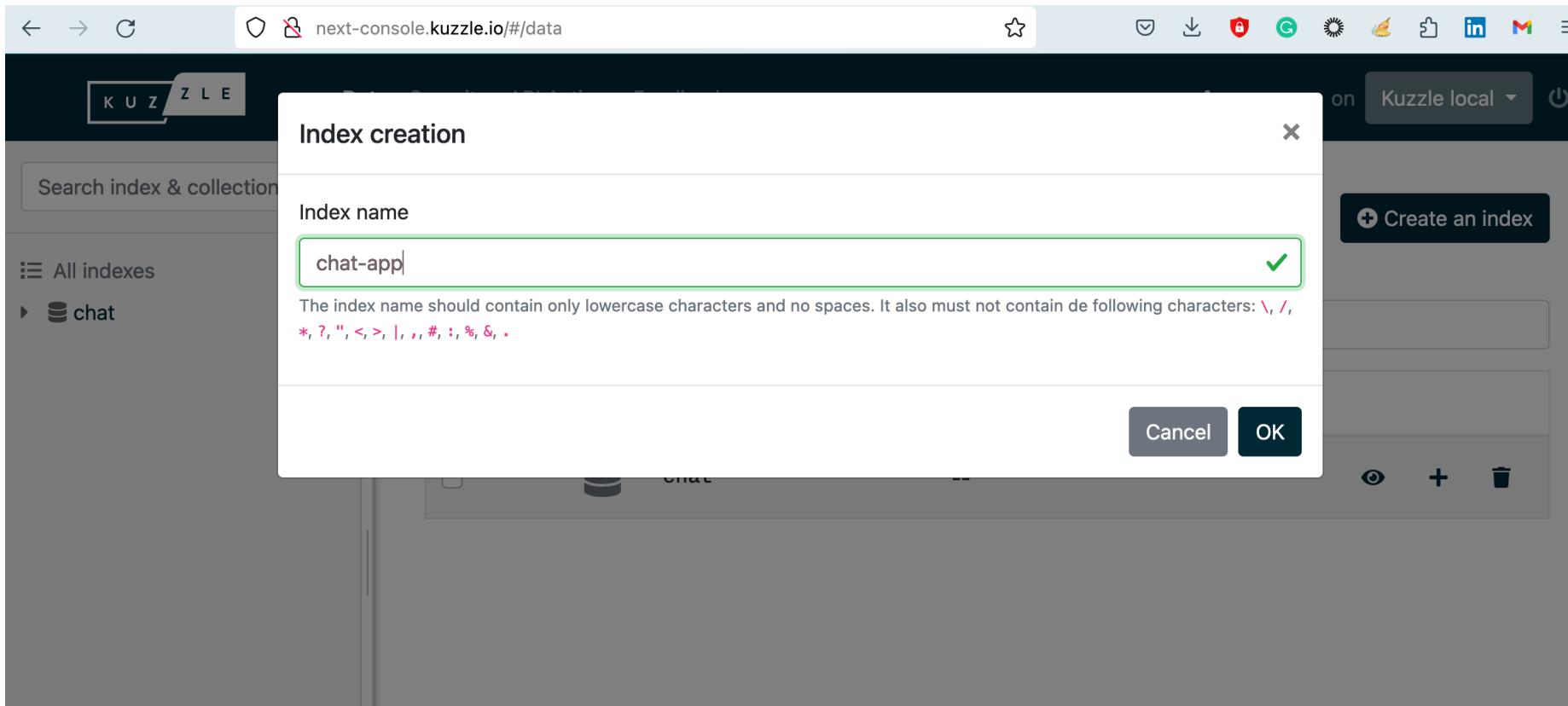
---

- Now that Kuzzle is running, we are going to create some documents.
- Kuzzle organizes the data storage in 4 levels (*comparing with RDBMS*):
  - Indexes (*similar to Database*)
  - Collections (*similar to tables*)
  - Documents (*similar to lines*)
  - Fields (*similar to columns*)



# Create an Index

Command: kourou index:create chat-app  
Or using a code in an SDK similar to command  
Or using Admin Console



# Create a collection

```
Command: kourou collection:create chat-app  
messaging-app '{  
    mappings: {  
        properties: {  
            name: { type: "keyword" },  
            city: { type: "keyword" },  
            age: { type: "integer" }  
        }  
    }'  
}'
```

The screenshot shows the Kuzzle interface for creating a new collection. The top navigation bar includes 'Data', 'Security', 'API Action', 'Feedback', 'Anonymous on', 'Kuzzle local', and a power icon. On the left, a sidebar shows 'All indexes' with 'chat (1)' and 'chat-app (0) no collections'. The main area is titled 'chat - app > Create a new collection' under 'data > chat-app'. It has a 'Collection name' input field containing 'messaging-app' with a green checkmark and a note 'This field is mandatory'. Below it is a JSON editor with the following code:

```
1 {  
2   "properties": {}  
3 }
```

To the right of the editor is a note: 'You can (optionally) use this editor to define the mappings for this collection.' Another note below it says: 'The mappings of a collection is the definition of how each document in the collection (and its fields) are stored and indexed. [Read more about mappings](#)'. At the bottom are 'Cancel' and 'Create' buttons.

# Creating Indexes and Collections from CLI

```
rajukarmuri@Rajus-Air playground % kourou index:create chat-app
(node:24876) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
[i] Unknown command "index:create", fallback to API action

🚀 Kourou – Executes an API query.

[i] Connecting to ws://localhost:7512 ...
undefined
[✓] Successfully executed "index:create"
rajukarmuri@Rajus-Air playground % kourou collection:create chat-app messaging-app '{
  mappings: {
    properties: {
      name: { type: "keyword" },
      city: { type: "keyword" },
      age: { type: "integer" }
    }
  }
}'
(node:24891) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)

🚀 Kourou – Creates a collection

[i] Connecting to ws://localhost:7512 ...
[✓] Collection "chat-app":"messaging-app" created
```

# Creating Documents

## Create a new document

data > chat-app > messaging-app

Document ID  Leave blank to let Kuzzle auto-generate the ID

Form view

```
1 {  
2   "age": "25",  
3   "city": "Basil",  
4   "name": "Roger"  
5 }
```

**Mapping**

- ▼ Object
  - ▼ age: Object
    - type: "integer"
  - ▼ city: Object
    - type: "keyword"
  - ▼ name: Object
    - type: "keyword"

**Cancel** **+ Create**

Command: kourou document:create chat-app messaging-app '{  
 name: "Roger",  
 city: "Basil",  
 age: 25  
}'

```
rajukarmuri@Rajus-Air playground % kourou document:create chat-app messaging-app '{  
  name: "Roger",  
  city: "Basil",  
  age: 25  
}'  
  
(node:25050) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use   
ad.  
(Use `node --trace-deprecation ...` to show where the warning was created)  
[i] Unknown command "document:create", fallback to API action  
  
🚀 Kourou – Executes an API query.  
  
[i] Connecting to ws://localhost:7512 ...  
{  
  "_id": "PtMIR4wBP4NKLKgzcTi7",  
  "_source": {  
    "name": "Roger",  
    "city": "Basil",  
    "age": 25,  
    "_kuzzle_info": {  
      "author": "-1",  
      "createdAt": 1701998784947,  
      "updatedAt": null,  
      "updater": null  
    }  
  },  
  "_version": 1  
}  
[✓] Successfully executed "document:create"  
rajukarmuri@Rajus-Air playground %
```

# messaging-app

[Create New Document](#) [Refresh](#)

data > chat-app > messaging-app

[View](#)[Advanced query...](#)[Quick Search](#)[Reset](#) Toggle all [Delete](#)

Show 25 of 1 total items.

▼  PtMIR4wBP4NKLKgzcTi7



▼ Object

name: "Roger"

city: "Basil"

age: 25

▼ \_kuzzle\_info: Object

author: "-1"

createdAt: "07/12/2023, 19:26:24"

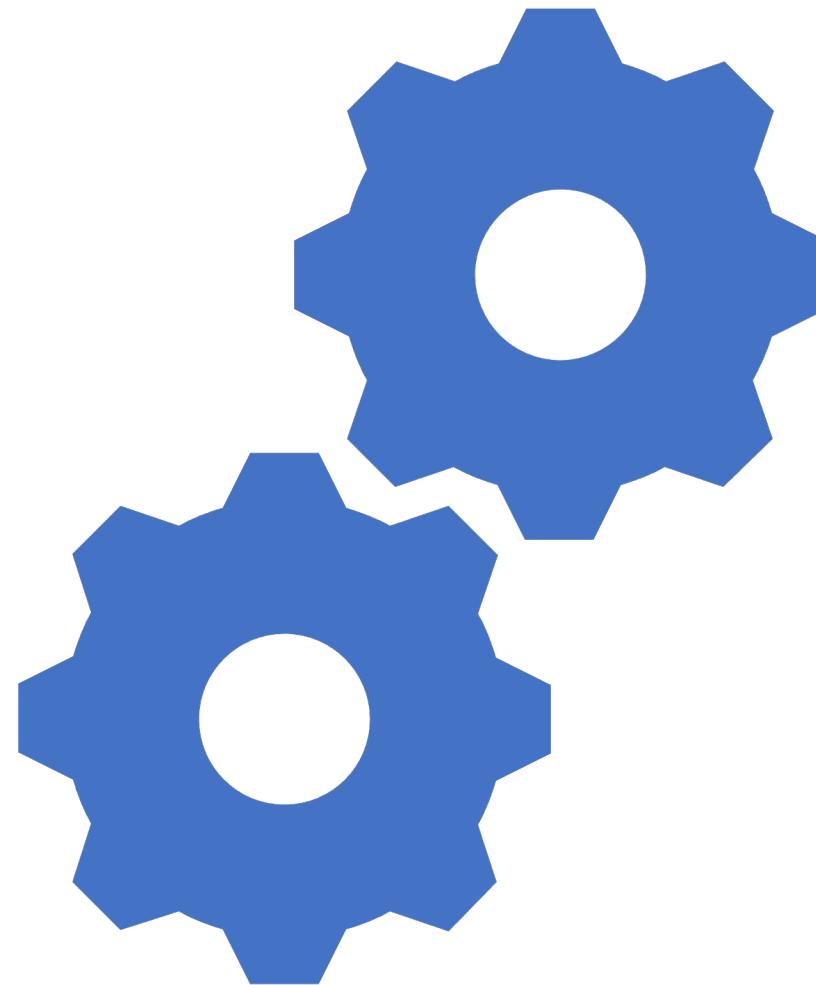
updatedAt: null

updateAt: null

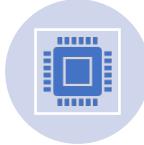
K U Z Z L E

# Real-time Engine

---



# Real-time Notifications



Kuzzle integrates an **advanced real-time engine**. It can work in a classic pub/sub mode but also as a real-time database notification engine.



Subscription to real-time notifications is **performed client-side and does not require additional server-side code**.



**Any change occurring on the database can generate a real-time notification** about the creation, modification and deletion of documents.

## Command: kourou realtime:subscribe chat-app messaging-app

```
rajukarmuri@Rajus-Air playground % kourou realtime:subscribe chat-app messaging-app
(node:25150) [DEP0040] DeprecationWarning: The `punycode` module is deprecated.
Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
```

⚡ **Kourou** – Subscribes to realtime notifications

```
[i] Connecting to ws://localhost:7512 ...
[i] Waiting for realtime notifications on "chat-app":"messaging-app" ...
```

# Real-time Notifications

---

```
🚀 Kourou – Subscribes to realtime notifications

[i] Connecting to ws://localhost:7512 ...
[i] Waiting for realtime notifications on "chat-app":"messaging-app" ...
[i] New notification triggered by API action "realtime:subscribe"
{
  "count": 2
}
[i] New notification triggered by API action "realtime:unsubscribe"
{
  "count": 1
}
[i] New notification triggered by API action "document:create"
{
  "_id": "P9MiR4wBP4NKLKgzJziO",
  "_source": {
    "name": "Stephen",
    "city": "San Francisco",
    "age": 26,
    "_kuzzle_info": {
      "author": "-1",
      "createdAt": 1702000469892,
      "updatedAt": null,
      "update": null
    }
  },
  "_version": 1
}
[i] New notification triggered by API action "realtime:subscribe"
{
  "count": 2
}
```

Advanced  
permission  
system

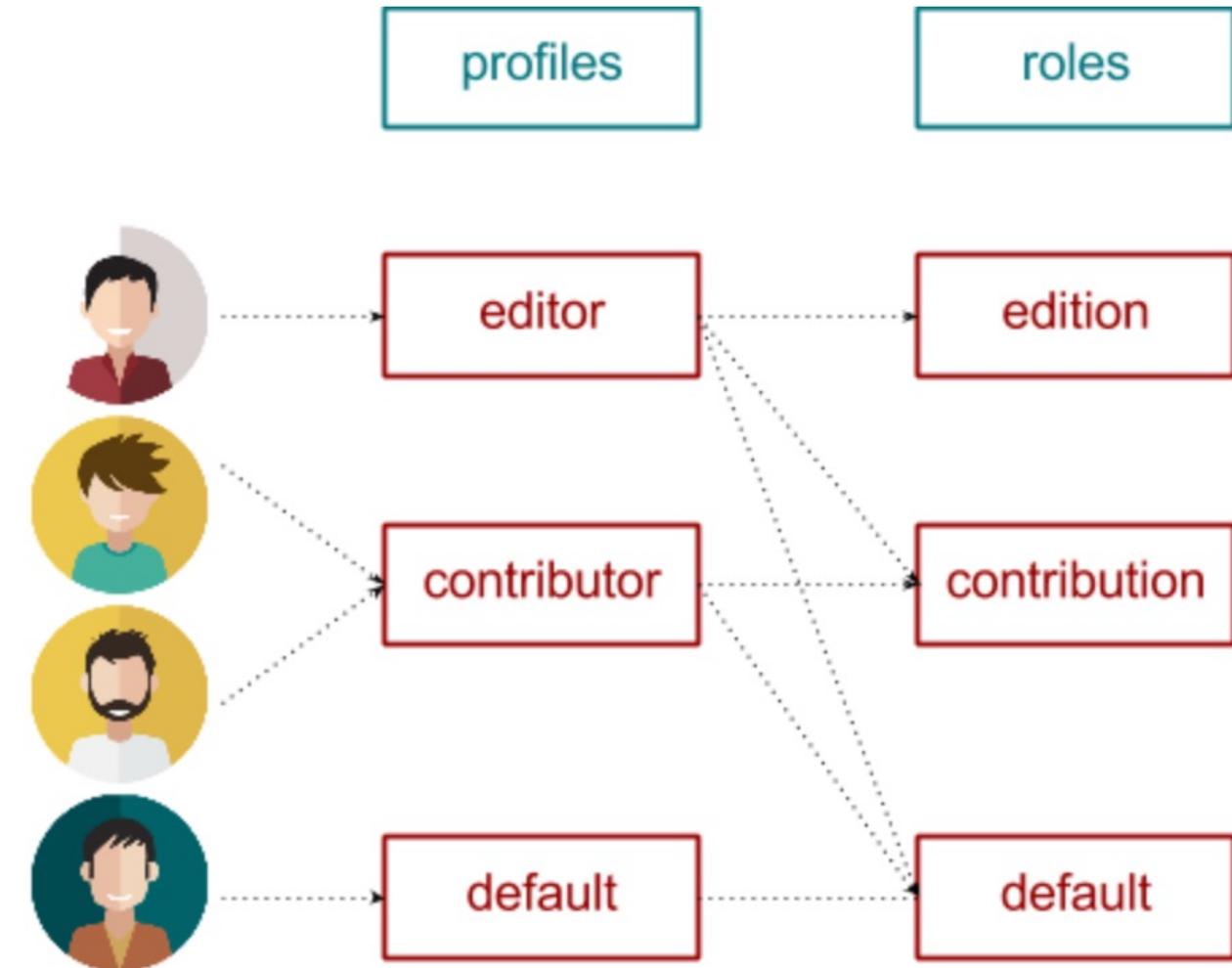
---



# Set up Permissions

---

- As in any backend, Kuzzle allows you to **restrict access to its features and data**, depending on the querying users.
- The permissions system is designed following a standard model and is structured in 3 dimensions:
  - Roles
  - Profiles
  - Users



# Create a Role

```
kourou security:createRole '{  
  controllers: {  
    auth: {  
      actions: {  
        getCurrentUser: true } },  
    server: { actions: { now: true } } } }'  
--id dummyRole
```

🚀 **Kourou** – Executes an API query.

```
[i] Connecting to ws://localhost:7512 ...  
{  
  "_id": "dummyRole",  
  "_source": {  
    "controllers": {  
      "auth": {  
        "actions": {  
          "getCurrentUser": true  
        }  
      },  
      "server": {  
        "actions": {  
          "now": true  
        }  
      }  
    },  
    "_kuzzle_info": {  
      "author": null,  
      "createdAt": 1702054328652,  
      "updatedAt": null,  
      "updater": null  
    }  
  }  
[✓] Successfully executed "security:createRole"
```

The screenshot shows the Kuzzle UI interface. At the top, there's a navigation bar with tabs for Data, Security, API Action, and Feedback. On the right, it shows 'Anonymous' and 'Kuzzle local'. Below the navigation, there's a sidebar with links for Users, Profiles, and Roles. The main content area is titled 'Roles' and shows a list of roles under 'security > roles'. There's a search bar at the top of the list table, with dropdowns for 'Controllers' and 'Reset'. The table lists four items: 'admin', 'anonymous', 'default', and 'dummyRole'. Each item has a checkbox to its left and edit and delete icons to its right. At the bottom of the table, there are pagination controls: '<<', '<', '1', '>', and '>>'. The footer of the page also features the 'KUZZLE' logo.



# Create a Profile

```
kourou security:createProfile '{  
  policies: [  
    { roleId: "dummyRole" }  
  ]  
}' --id dummyProfile
```

The screenshot shows the Kuzzle Security interface with the 'Profiles' tab selected. The left sidebar has 'Users', 'Profiles' (selected), and 'Roles'. The main area is titled 'Profiles' and shows a list of profiles: 'admin', 'default', 'anonymous', and 'dummyProfile'. Each profile entry has a 'Toggle all' checkbox, a 'Delete selected' button, and edit/delete icons. A search bar at the top allows filtering by role. The bottom right shows pagination controls.

Profile	Action
admin	<input type="checkbox"/> <span style="color: red;">Delete selected</span> <span style="float: right;">edit delete</span>
default	<input type="checkbox"/> <span style="color: red;">Delete selected</span> <span style="float: right;">edit delete</span>
anonymous	<input type="checkbox"/> <span style="color: red;">Delete selected</span> <span style="float: right;">edit delete</span>
dummyProfile	<input type="checkbox"/> <span style="color: red;">Delete selected</span> <span style="float: right;">edit delete</span>

# Create a User

```
kourou security:createUser '{  
    content: {  
        profileIds: ["dummyProfile"]  
    },  
    credentials: {  
        local: { username: "melis",  
            password: "password" } } }
```

The screenshot shows the Kuzzle UI interface. At the top, there's a dark header with the Kuzzle logo, navigation links for Data, Security, API Action, and Feedback, and a user status bar indicating 'Anonymous' on 'Kuzzle local'. Below the header is a sidebar with three items: 'Users' (selected), 'Profiles', and 'Roles'. The main content area is titled 'Users' and shows a list of users under the 'security' index. The list includes a search bar, a toolbar with 'Advanced query...', 'Quick Search', and 'Reset' buttons, and a pagination section showing '25' items. One user entry is visible: 'kuid-closed-parrot-45234' with the profile 'melis' and the role 'dummyProfile'. The bottom right corner features the Kuzzle logo.

# Real World Applications



IOT (INTERNET OF  
THINGS) SOLUTIONS



CHAT APPLICATIONS



GEOLOCATION  
SERVICES



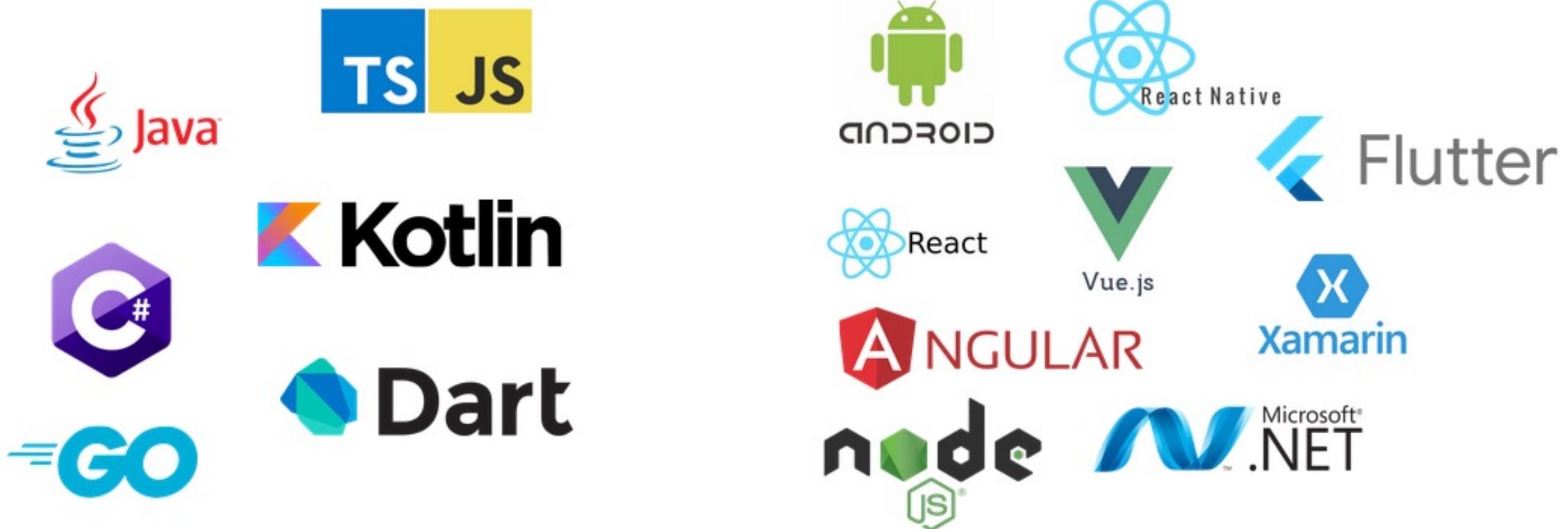
GAMING PLATFORMS



ANY REAL-TIME  
COLLABORATIVE  
APPLICATIONS

# SDKs to run Kuzzle

- We can leverage the benefits of Kuzzle through various SDKs.



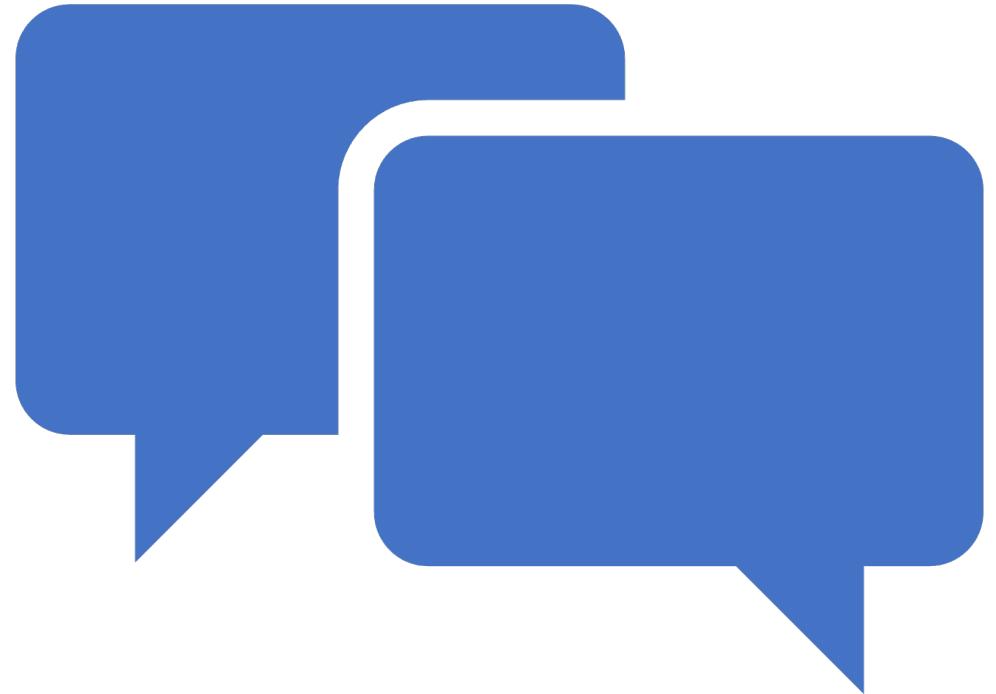
# Install Kuzzle SDKs

Command: `npm install kuzzle-sdk`



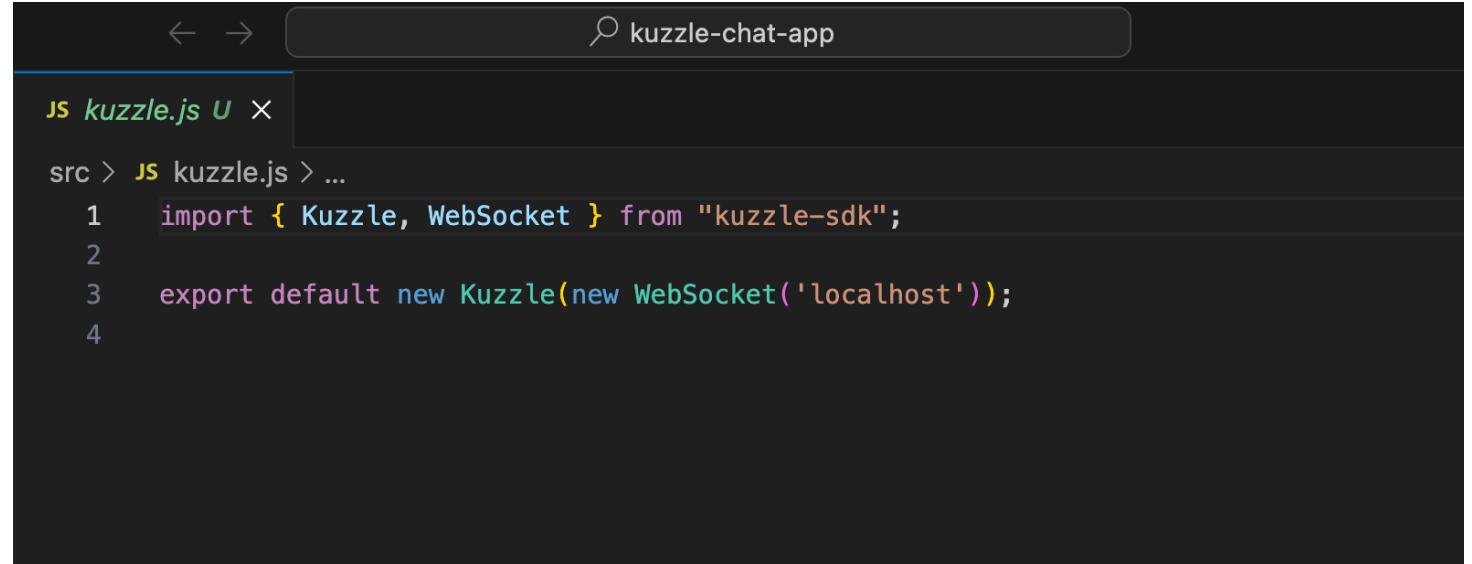
# Message App using React.js and Kuzzle

---



# Message App using React.js and Kuzzle

- We should be connected to Kuzzle
- We can connect to the kuzzle server using WebSocket protocol
- WebSocket protocol allows us to use all features of Kuzzle
- Make sure your Docker containers are up and running
- Make sure you can connect to your localhost port 7512



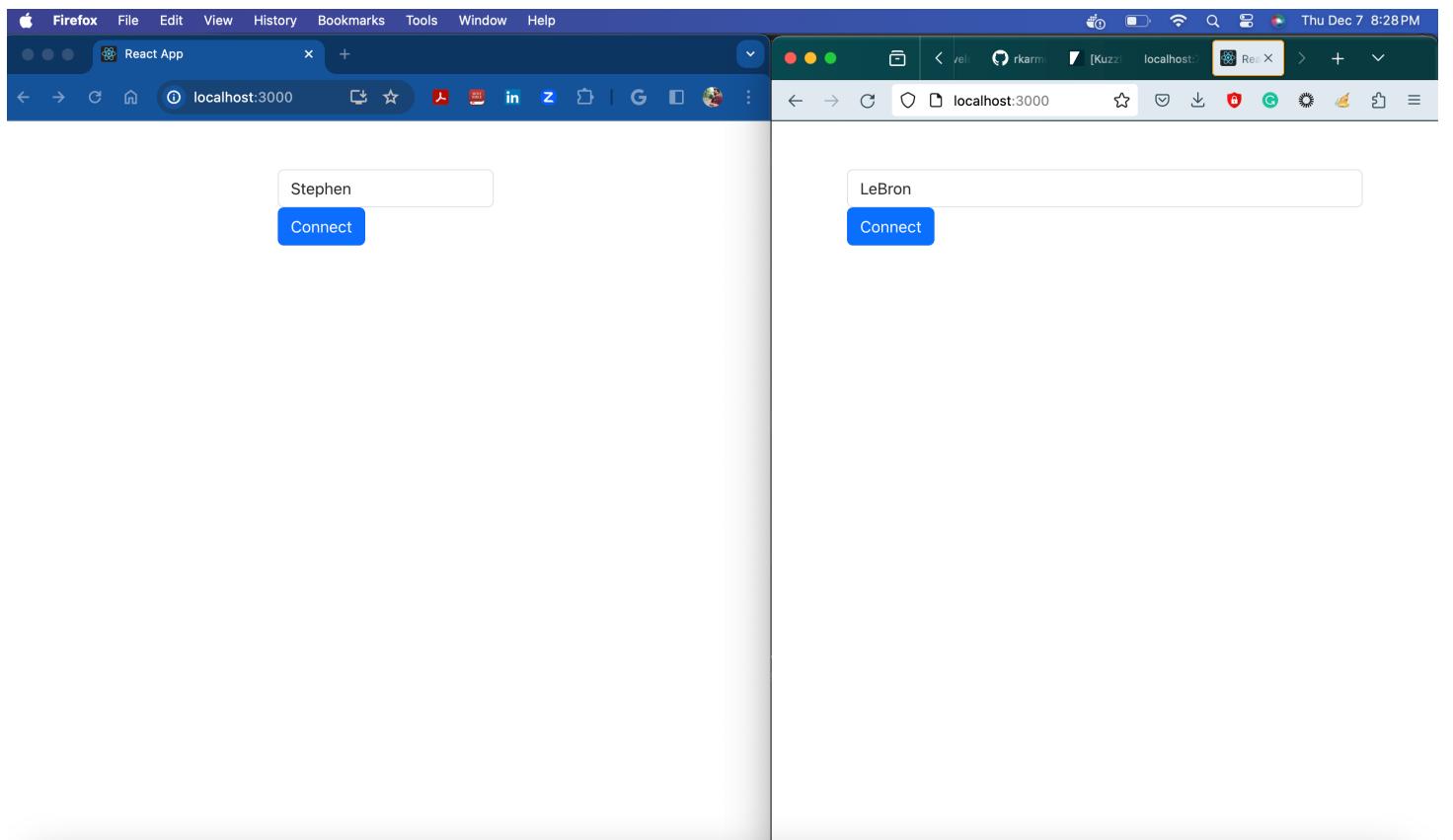
A screenshot of a code editor window titled 'kuzzle-chat-app'. The file 'kuzzle.js' is open, showing the following code:

```
src > JS kuzzle.js > ...
1 import { Kuzzle, WebSocket } from "kuzzle-sdk";
2
3 export default new Kuzzle(new WebSocket('localhost'));
4
```

GitHub Repo: <https://github.com/rkarmuri/kuzzle-workshop>

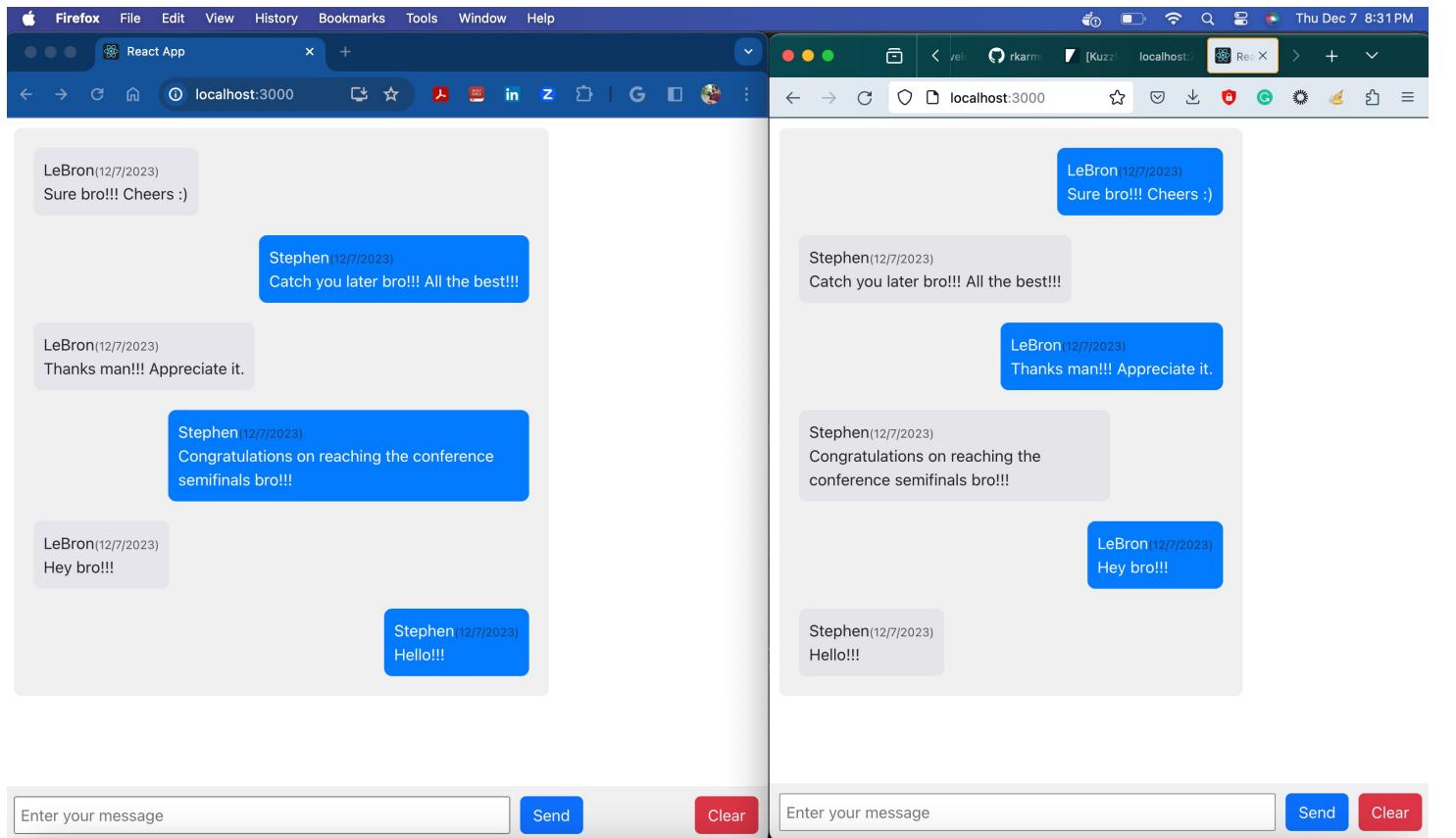
# Message App using React.js and Kuzzle

Open `localhost:3000` in 2  
different tabs and connect  
using any name.



# Message App using React.js and Kuzzle

We can see that the messages are instantly delivered across the WebSocket between 2 users.



# Message App using React.js and Kuzzle

We can also see that the messages are stored in the database backend on the Kuzzle storage.

The screenshot shows the Kuzzle Data Explorer interface. On the left, there's a sidebar with a search bar and a tree view of indexes: 'All indexes' (chat (1) with messages, chat-app (1) with messaging-app). The main area displays a list of 6 items. Each item has a checkbox, a delete button, and edit/pencil icons. The first message is from 'LeBron' with the value 'Sure bro!!! Cheers :)'. The second message is from 'Stephen' with the value 'Catch you later bro!!! All the best!!!'. Both messages have createdAt and updatedAt fields set to '07/12/2023, 20:30:43' and '07/12/2023, 20:30:26' respectively, and null update and updater fields.

ID	Author	Value	createdAt	updatedAt
RdNDR4wBP4NKLKgzUzhx	LeBron	Sure bro!!! Cheers :)	07/12/2023, 20:30:43	null
RNNDR4wBP4NKLKgzDjhe	Stephen	Catch you later bro!!! All the best!!!	07/12/2023, 20:30:26	null

# Questions?

---



Thank you

---

