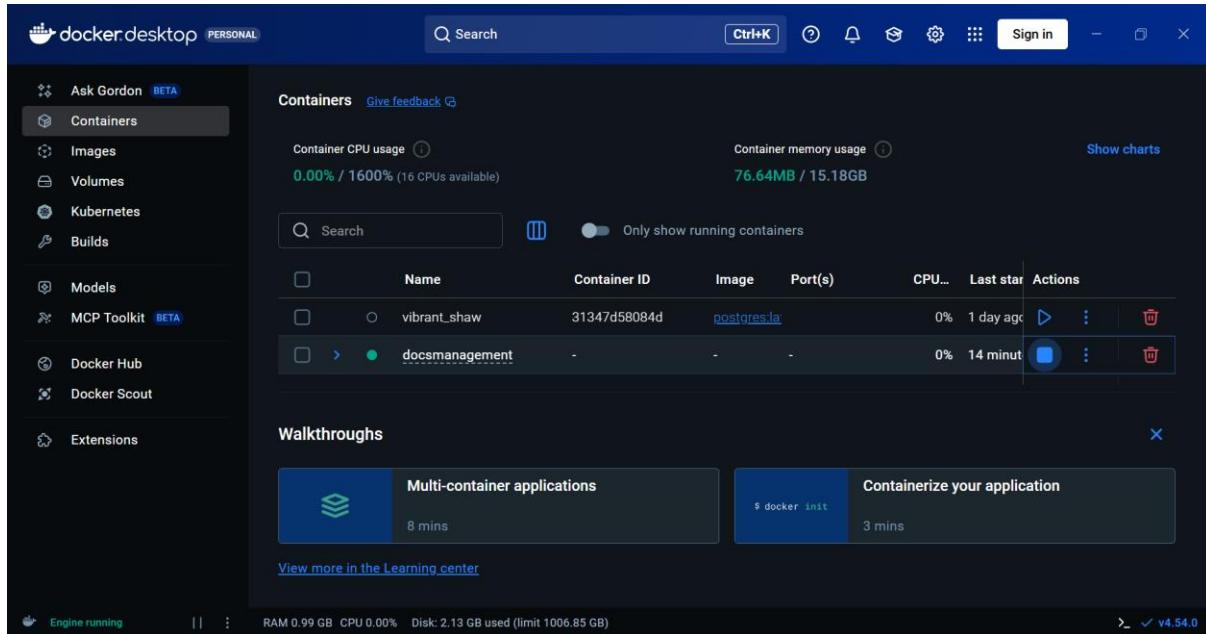
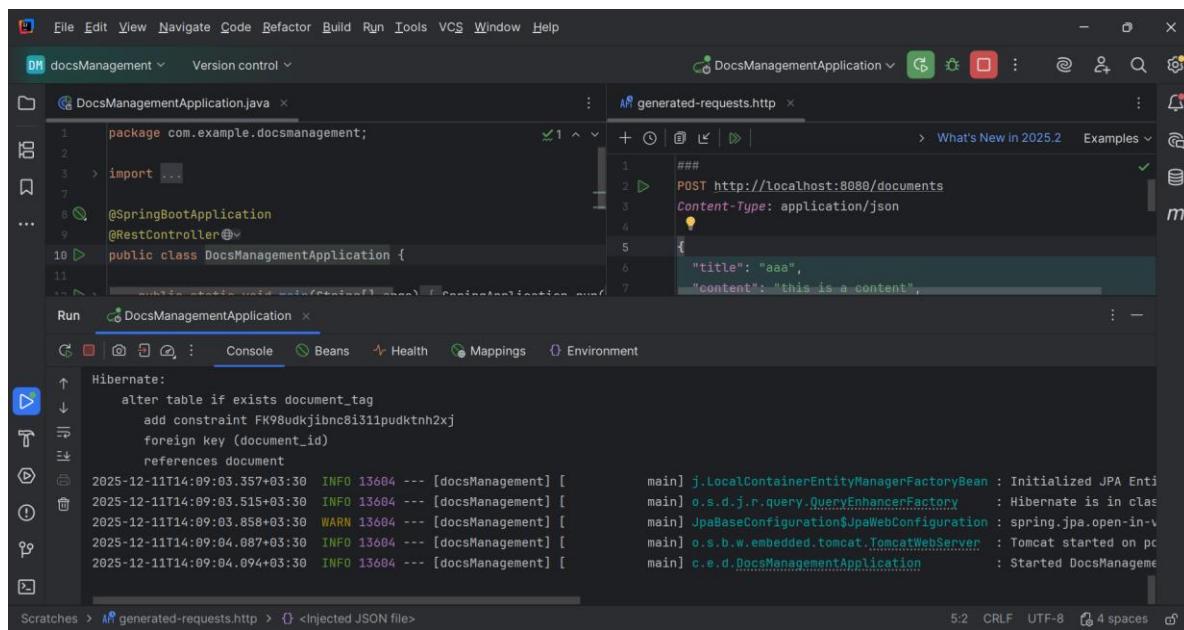


## اجرای کد بک اند برای سرویس کوچک مدیریت اسناد

ابتدا کانتینر شامل پایگاه داده و سپس اپلیکشن را اجرا می‌کنیم، سپس دستورات را از طریق فایل generated-`sequence.java` اجرا می‌کنیم.



شکل ۱: کانتینر در حال اجرا



شکل ۲: اجرای اپلیکیشن

The screenshot shows the JetBrains IDE interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Icons for Open, Save, Undo, Redo, Cut, Copy, Paste, Find, and others.
- Project Bar:** docsManagement, Version control.
- Code Editor:** A file named "generated-requests.http" is open. The code is a POST request to "http://localhost:8080/documents".

```
##  
POST http://localhost:8080/documents  
Content-Type: application/json  
  
{  
    "title": "aaa",  
    "content": "this is a content",  
    "tags": ["simple", "text"]  
}
```
- Run Bar:** Run with: No Environment.
- Right Panel:** Shows "What's New in 2025.2" and "Examples".
- Bottom Panels:**
  - Services:** A list of services including "HTTP Request".
  - HTTP Request Details:** Shows an incoming POST request for "generated-requests" with status 200 and duration 455 ms. It also shows an outgoing GET request for "generated-requests" and a redacted "generated-requests" entry.
  - Response Log:** Shows the response body: "Response file saved." followed by a redacted URL.
  - Console:** Shows the response code: 200; Time: 455ms (455 ms); Content length: 1 bytes (\ B).

شکل ۳: اجرای یک دستور post برای ایجاد یک سند (عدد نمایش داده شده شناسه ایجاد شده برای سند است)

The screenshot shows the API Management interface. In the top navigation bar, 'File', 'Edit', 'View', 'Navigate', 'Code', 'Refactor', 'Build', 'Run', 'Tools', 'VCS', and 'Help' are visible. The 'File' menu is open, showing 'docsManagement' and 'Version control'. On the right, there are icons for 'DocsManagementApplication', a green checkmark, a red square, a blue square, a double arrow, a person icon, and a search icon.

The main workspace displays an 'API generated-requests.http' file. The code content is as follows:

```
##  
POST http://localhost:8080/documents  
Content-Type: application/json  
  
{  
    "title": "bbb",  
    "content": "Deep learning models such as CNN, RNN...",  
    "tags": ["deep", "ai"]  
}
```

To the right of the code editor, there's a 'What's New in 2025.2' section and an 'Examples' dropdown. Below the code editor is a 'Services' panel with various icons for API management tasks like creating, deleting, and modifying services.

The bottom status bar shows 'Scratches > API generated-requests.http > <Injected JSON file>' and '7:28 CRLF UTF-8 4 spaces'.

#### شکل ۴: ایجاد یک سند دیگر

```

Terminal - docsManagement
Terminal Local + ▾
documentsdb=# 
documentsdb=# 
PS C:\Users\el\IdeaProjects\docsManagement> docker exec -it postgres-documents bash
root@043d6fc5f0088:/# psql -U user
psql (18.1 (Debian 18.1-1.pgdg13+2))
Type "help" for help.

template1 | user | UTF8 | libc | en_US.utf8 | en_US.utf8 | | | user=CTc/user
          |     |      |    |       |       |   |   | =c/user
user     | user | UTF8 | libc | en_US.utf8 | en_US.utf8 | | | user=CTc/user
(5 rows)

user=# \c documentsdb;
You are now connected to database "documentsdb" as user "user".
documentsdb=# \dt
           List of tables
 Schema |   Name    | Type  | Owner
-----+-----+-----+-----+
 public | document | table | user
 public | document_tag | table | user
 public | tag | table | user
(3 rows)

documentsdb=# SELECT * FROM document;
   created_at   | id | content | title
-----+-----+-----+-----+
 2025-12-11 14:17:28.53338 | 1 | this is a content | aaa
 2025-12-11 14:20:28.537398 | 2 | Deep learning models such as CNN, RNN... | bbb
(2 rows)

documentsdb=#

```

شکل ۵: نمایش پایگاه داده و اسناد ایجاد شده

```

GET http://localhost:8080/documents/search?query=aaa&mode=title
Show Request

HTTP/1.1 200
> (Headers) ...Content-Type: application/json...

[
  {
    "id": 1,
    "title": "aaa",
    "content": "this is a content",
    "createdAt": "2025-12-11T14:17:28.53338",
    "tags": [
      "simple",
      "text"
    ]
  }
]
Response file saved.
> ۲۰۲۵-۱۲-۱۱T۱۴۲۲۱۹.200.json

```

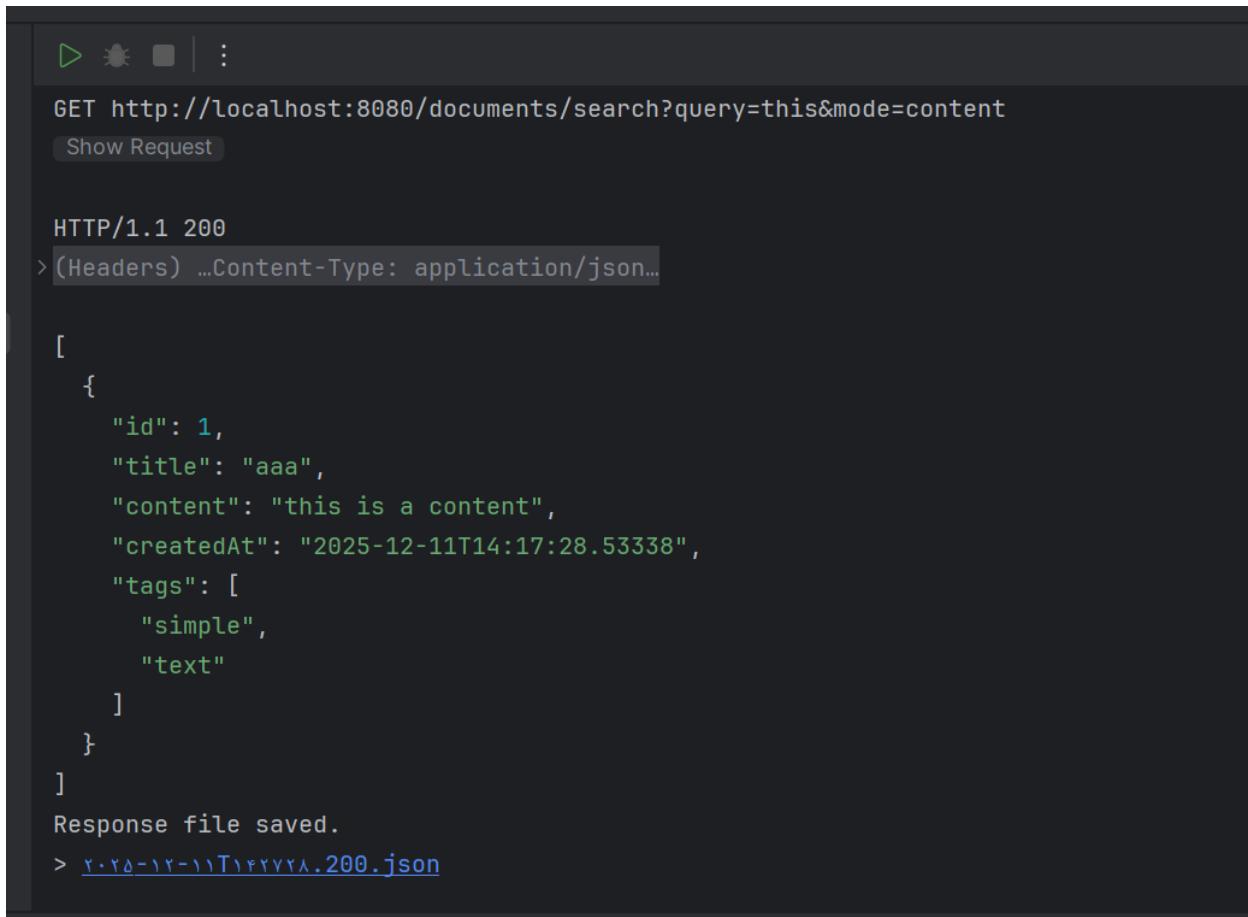
شکل ۶: جستجوی سند بر اساس عنوان (یکی از اسناد بازگردانده شده)

```
▶ ⚡ 🗃 | :  
GET http://localhost:8080/documents/search?query=aaa&mode=tag  
Show Request  
  
HTTP/1.1 200  
> (Headers) ...Content-Type: application/json...  
  
[]  
Response file saved.  
> ۲۰۲۵-۱۲-۱۱T۱۴۲۴.۵.200.json  
  
Response code: 200; Time: 11ms (11 ms); Content length: 2 bytes (۲ B)
```

شکل ۷: جستجو بر اساس برچسب (یک برچسب ناموجود)

```
▶ ⚡ 🗃 | :  
GET http://localhost:8080/documents/search?query=deep&mode=tag  
Show Request  
  
HTTP/1.1 200  
> (Headers) ...Content-Type: application/json...  
  
[  
 {  
   "id": 2,  
   "title": "bbb",  
   "content": "Deep learning models such as CNN, RNN...",  
   "createdAt": "2025-12-11T14:20:28.537398",  
   "tags": [  
     "deep",  
     "ai"  
   ]  
 }  
]  
Response file saved.  
> ۲۰۲۵-۱۲-۱۱T۱۴۲۵۲۲.200.json
```

شکل ۸: جستجو بر اساس برچسب (یک برچسب موجود)



```
GET http://localhost:8080/documents/search?query=this&mode=content
Show Request

HTTP/1.1 200
> (Headers) ...Content-Type: application/json...

[
  {
    "id": 1,
    "title": "aaa",
    "content": "this is a content",
    "createdAt": "2025-12-11T14:17:28.53338",
    "tags": [
      "simple",
      "text"
    ]
  }
]
Response file saved.
> ٢٠٢٥-١٢-١١T١٤:٢٨.٢٠٠.json
```

شكل ٩: جستجو بر اساس متن اسناد

```
GET http://localhost:8080/documents/search?query=a&mode=all
Show Request

HTTP/1.1 200
> (Headers) ...Content-Type: application/json...

[
  {
    "id": 2,
    "title": "bbb",
    "content": "Deep learning models such as CNN, RNN...",
    "createdAt": "2025-12-11T14:20:28.537398",
    "tags": [
      "deep",
      "ai"
    ]
  },
  {
    "id": 1,
    "title": "aaa",
    "content": "this is a content".
  }
]
```

شکل ۱۰: حالت جستجوی all (جستجوی یک حرف a)