Open-Source AI Toolkit for Developers

Petyo Dimitrov

2024-11-14

Qinshift

About me

17 years in Software Engineering

Senior Software Architect
in Musala Soft

Head of Data & AI Service Offer in
Qinshift

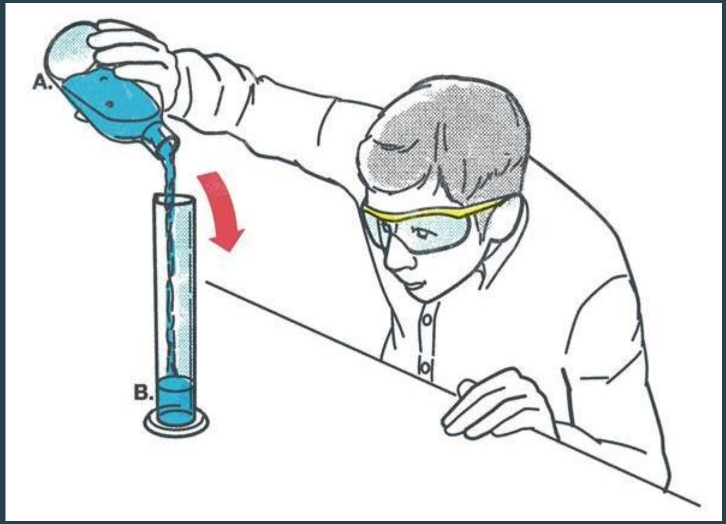
Background

AI in Action: Enhancing Software Development

Petyo Dimitrov, Senior Software Architect

2023-6-22

2023



2024

Niki Uzunov

Agenda

01 Commercial leaders

02 Development use cases & issues

03 Open-source alternatives

04 Trade-offs

05 Next steps

This will age like fine...

Which AI dev tools do you use?

bit.ly/DevAITools

Commercial leaders

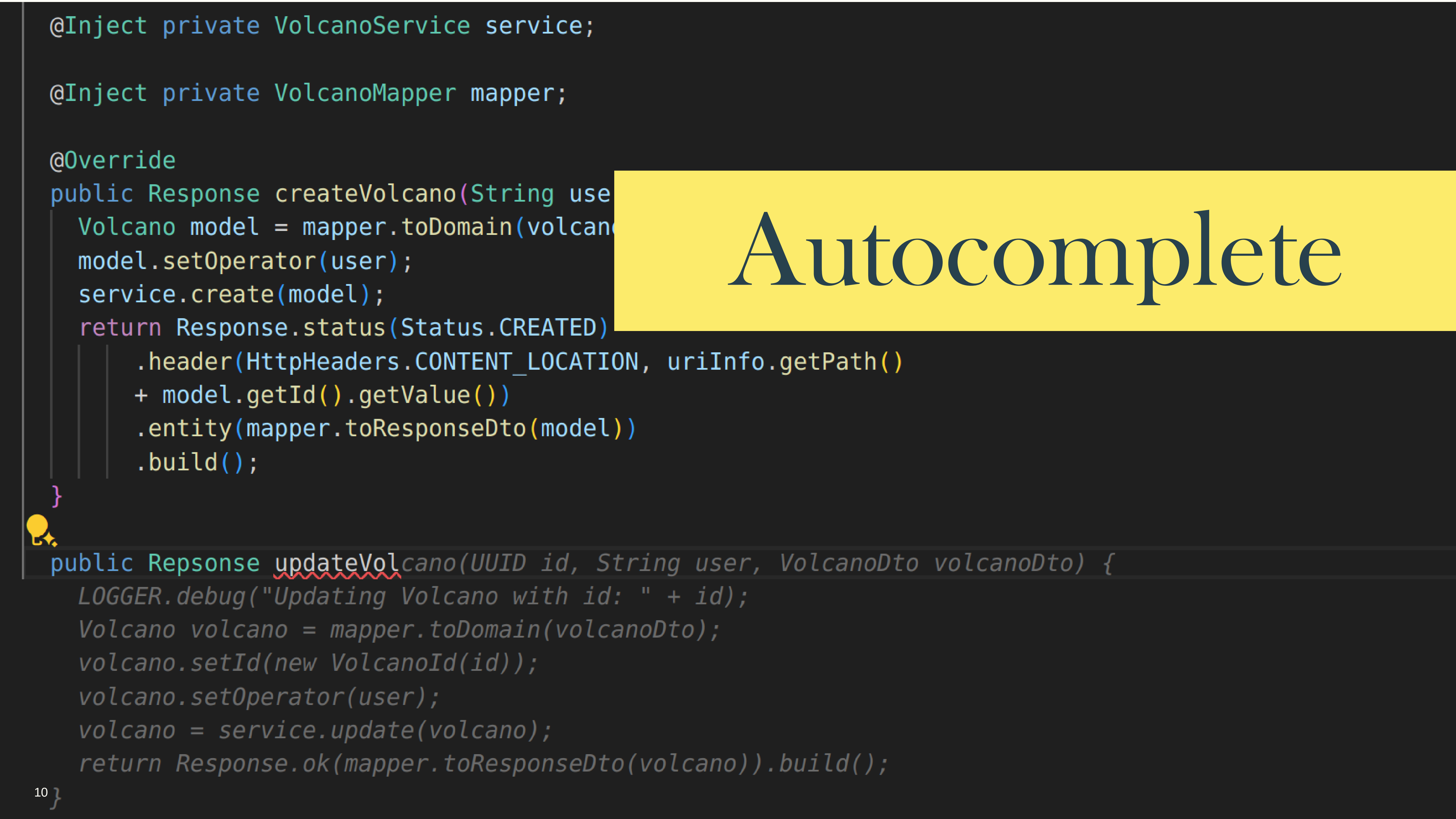
ChatGPT

GitHub Copilot

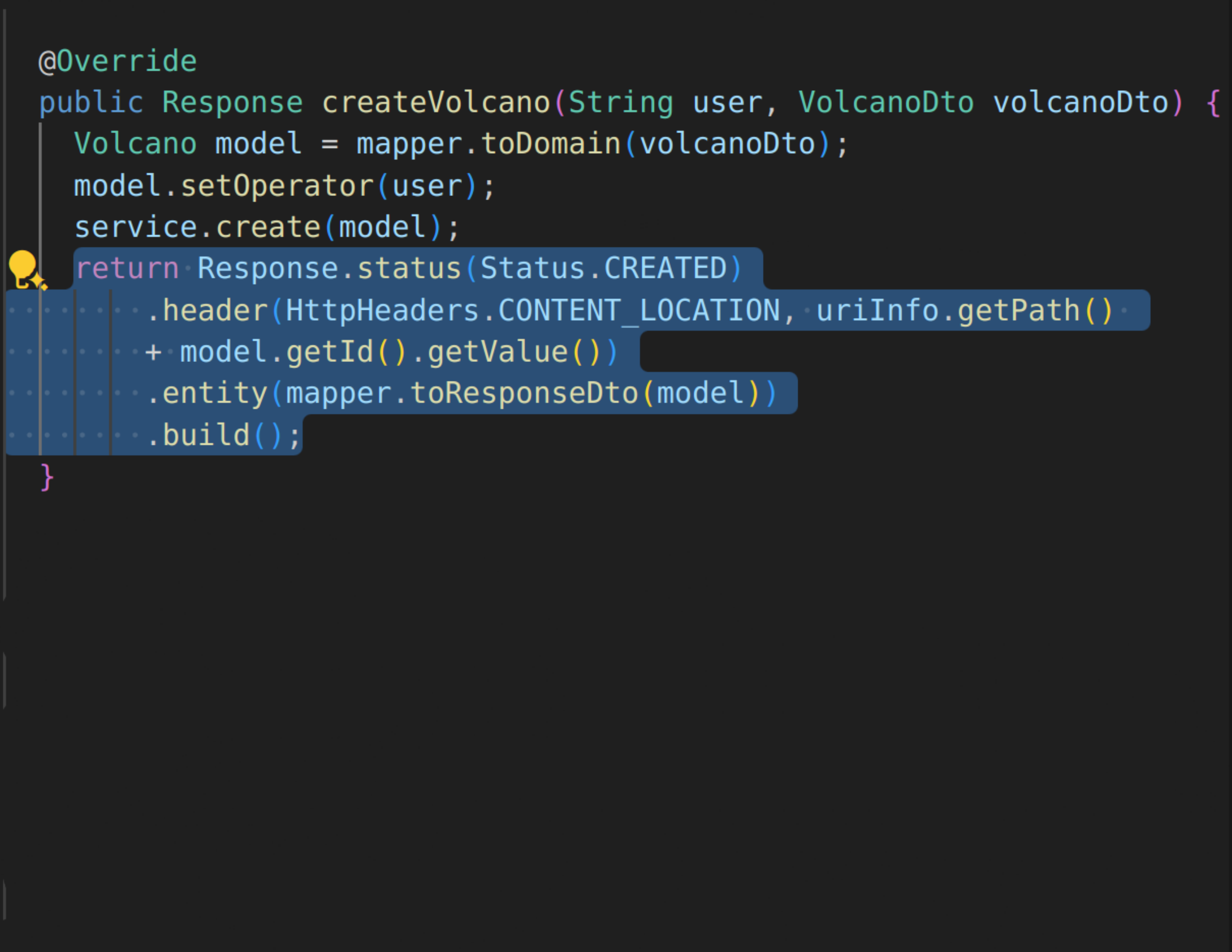
Development use cases

Qinshift

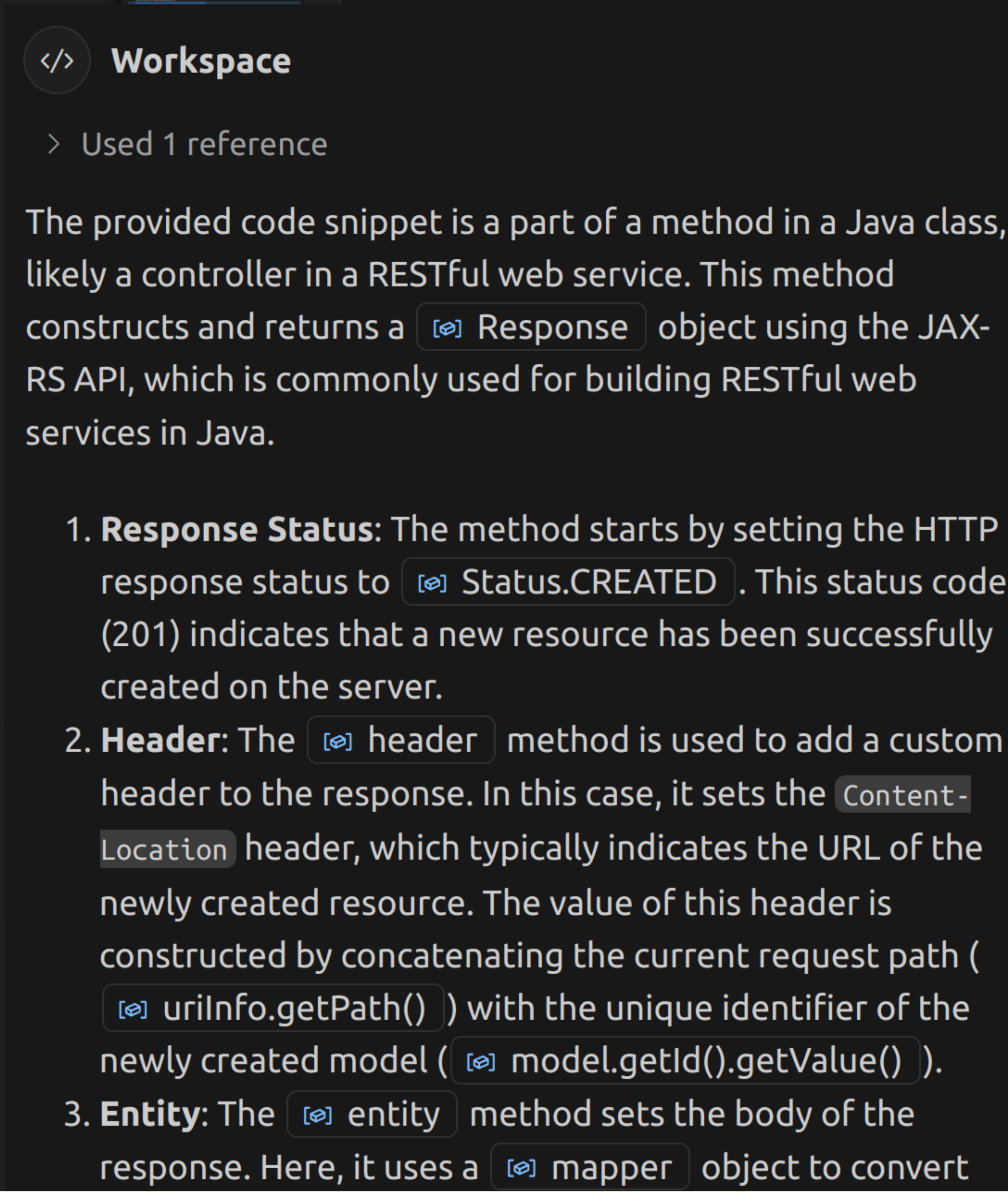
Autocomplete



```
@Inject private VolcanoService service;

@Inject private VolcanoMapper mapper;

@Override
public Response createVolcano(String use
  Volcano model = mapper.toDomain(volcan
  model.setOperator(user);
  service.create(model);
  return Response.status(Status.CREATED)
      .header(HttpHeaders.CONTENT_LOCATION, uriInfo.getPath()
      + model.getId().getValue())
      .entity(mapper.toResponseDto(model))
      .build();
}

public Repsonse updateVolcano(UUID id, String user, VolcanoDto volcanoDto) {
  LOGGER.debug("Updating Volcano with id: " + id);
  Volcano volcano = mapper.toDomain(volcanoDto);
  volcano.setId(new VolcanoId(id));
  volcano.setOperator(user);
  volcano = service.update(volcano);
  return Response.ok(mapper.toResponseDto(volcano)).build();
}
```



```
@Override
public Response createVolcano(String user, VolcanoDto volcanoDto) {
    Volcano model = mapper.toDomain(volcanoDto);
    model.setOperator(user);
    service.create(model);
    return Response.status(Status.CREATED)
            .header(HttpHeaders.CONTENT_LOCATION, uriInfo.getPath()
            + model.getId().getValue())
            .entity(mapper.toResponseDto(model))
            .build();
}
```

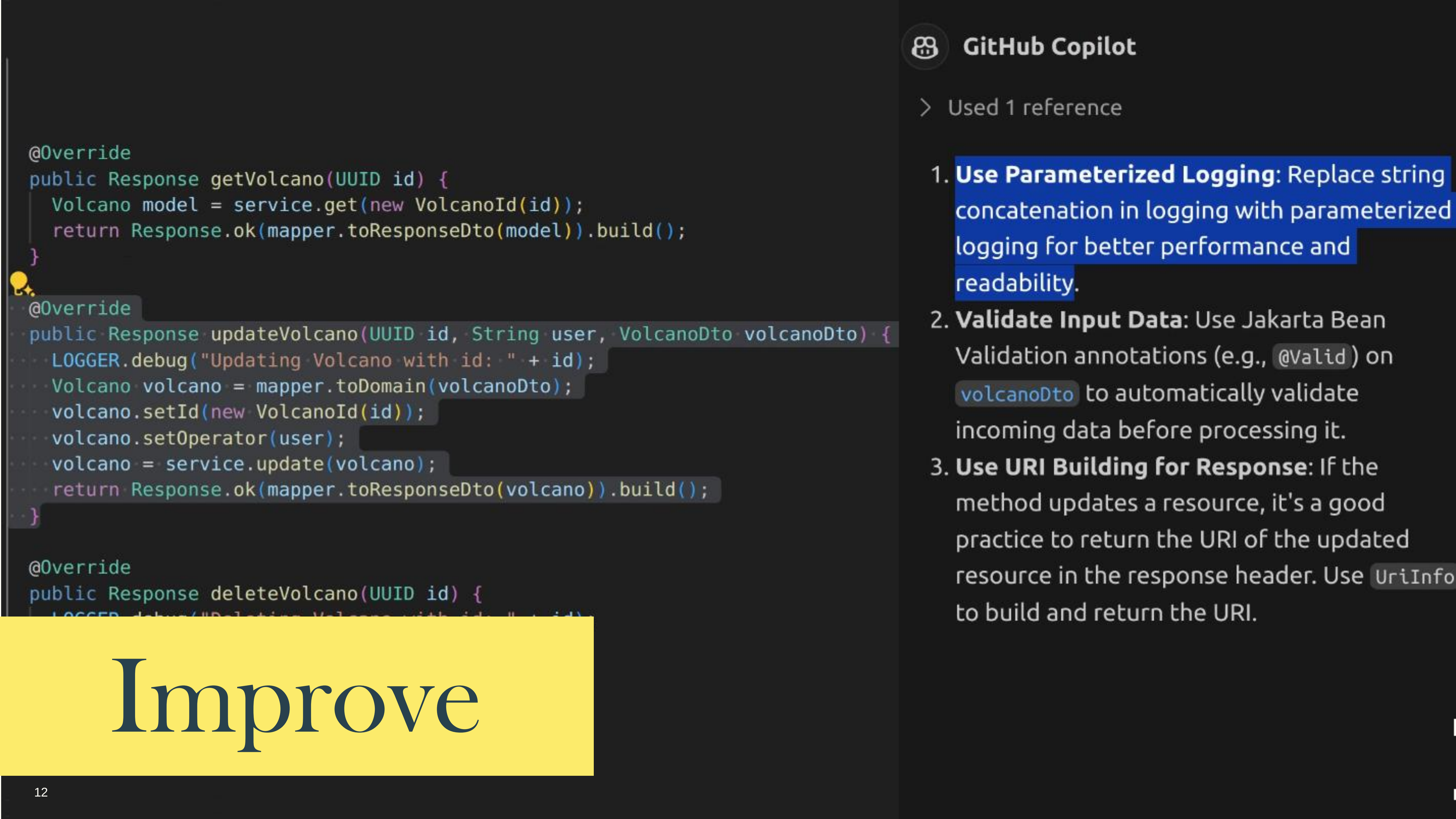


The provided code snippet is a part of a method in a Java class, likely a controller in a RESTful web service. This method constructs and returns a Response object using the JAX-RS API, which is commonly used for building RESTful web services in Java.

1. **Response Status**: The method starts by setting the HTTP response status to Status.CREATED. This status code (201) indicates that a new resource has been successfully created on the server.
2. **Header**: The header method is used to add a custom header to the response. In this case, it sets the `Content-Location` header, which typically indicates the URL of the newly created resource. The value of this header is constructed by concatenating the current request path (uriInfo.getPath()) with the unique identifier of the newly created model (model.getId().getValue()).
3. **Entity**: The entity method sets the body of the response. Here, it uses a mapper object to convert

Explain

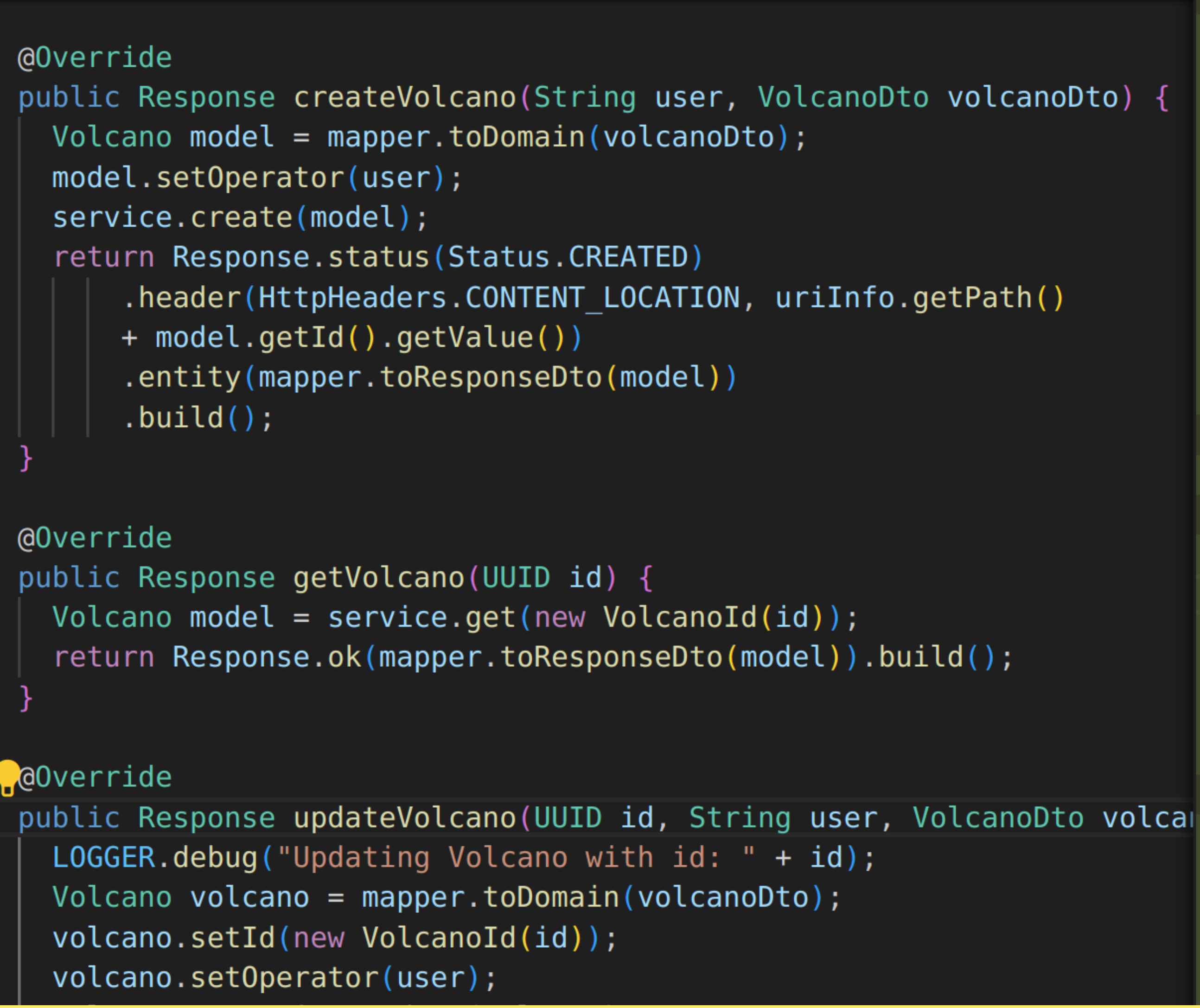
Improve



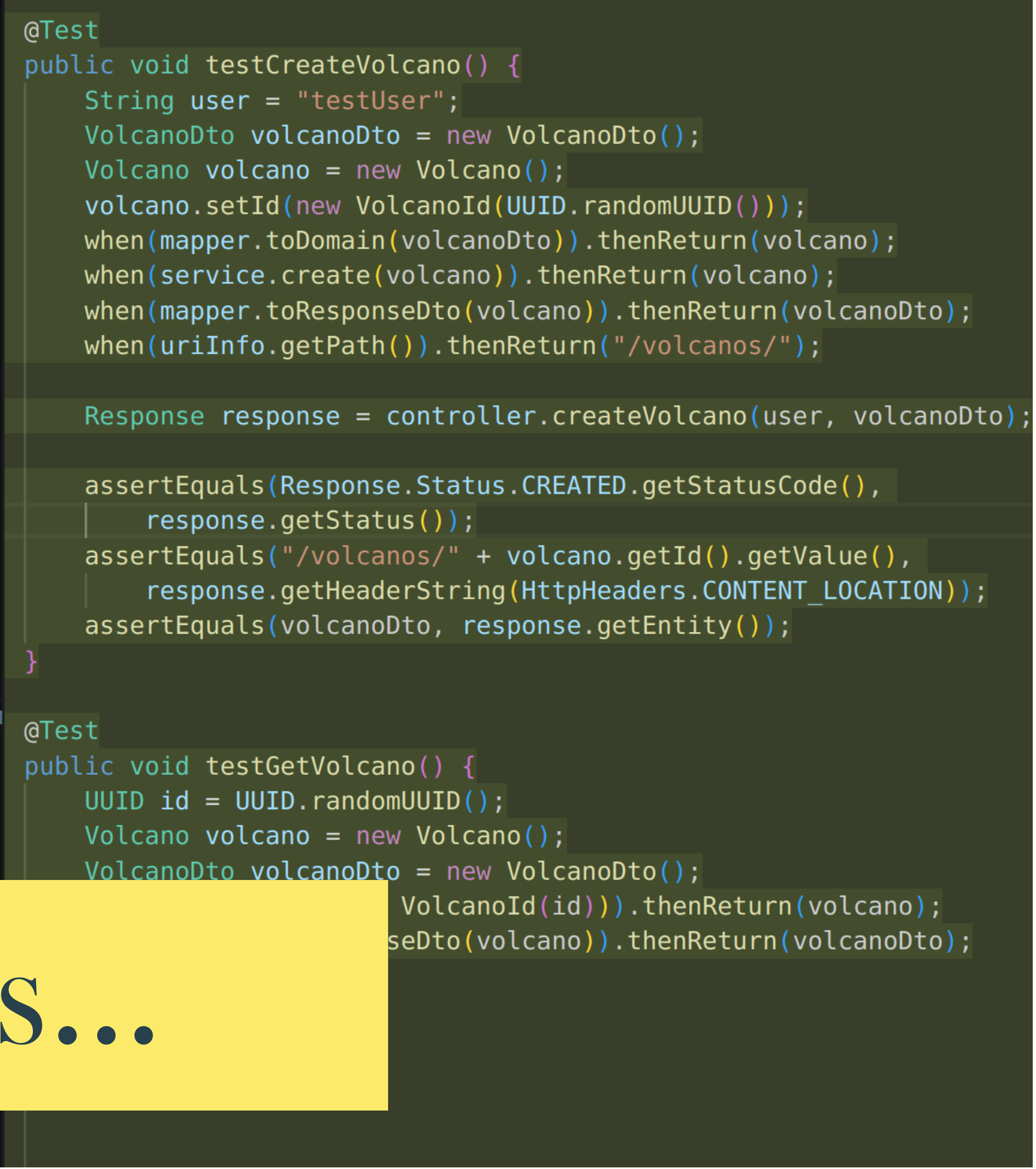
```
@Override
public Response getVolcano(UUID id) {
    Volcano model = service.get(new VolcanoId(id));
    return Response.ok(mapper.toResponseDto(model)).build();
}

@Override
public Response updateVolcano(UUID id, String user, VolcanoDto volcanoDto) {
    LOGGER.debug("Updating Volcano with id: " + id);
    Volcano volcano = mapper.toDomain(volcanoDto);
    volcano.setId(new VolcanoId(id));
    volcano.setOperator(user);
    volcano = service.update(volcano);
    return Response.ok(mapper.toResponseDto(volcano)).build();
}

@Override
public Response deleteVolcano(UUID id) {
```

1. **Use Parameterized Logging**: Replace string concatenation in logging with parameterized logging for better performance and readability.
2. **Validate Input Data**: Use Jakarta Bean Validation annotations (e.g., `@Valid`) on `volcanoDto` to automatically validate incoming data before processing it.
3. **Use URI Building for Response**: If the method updates a resource, it's a good practice to return the URI of the updated resource in the response header. Use `UriInfo` to build and return the URI.

Generate tests...

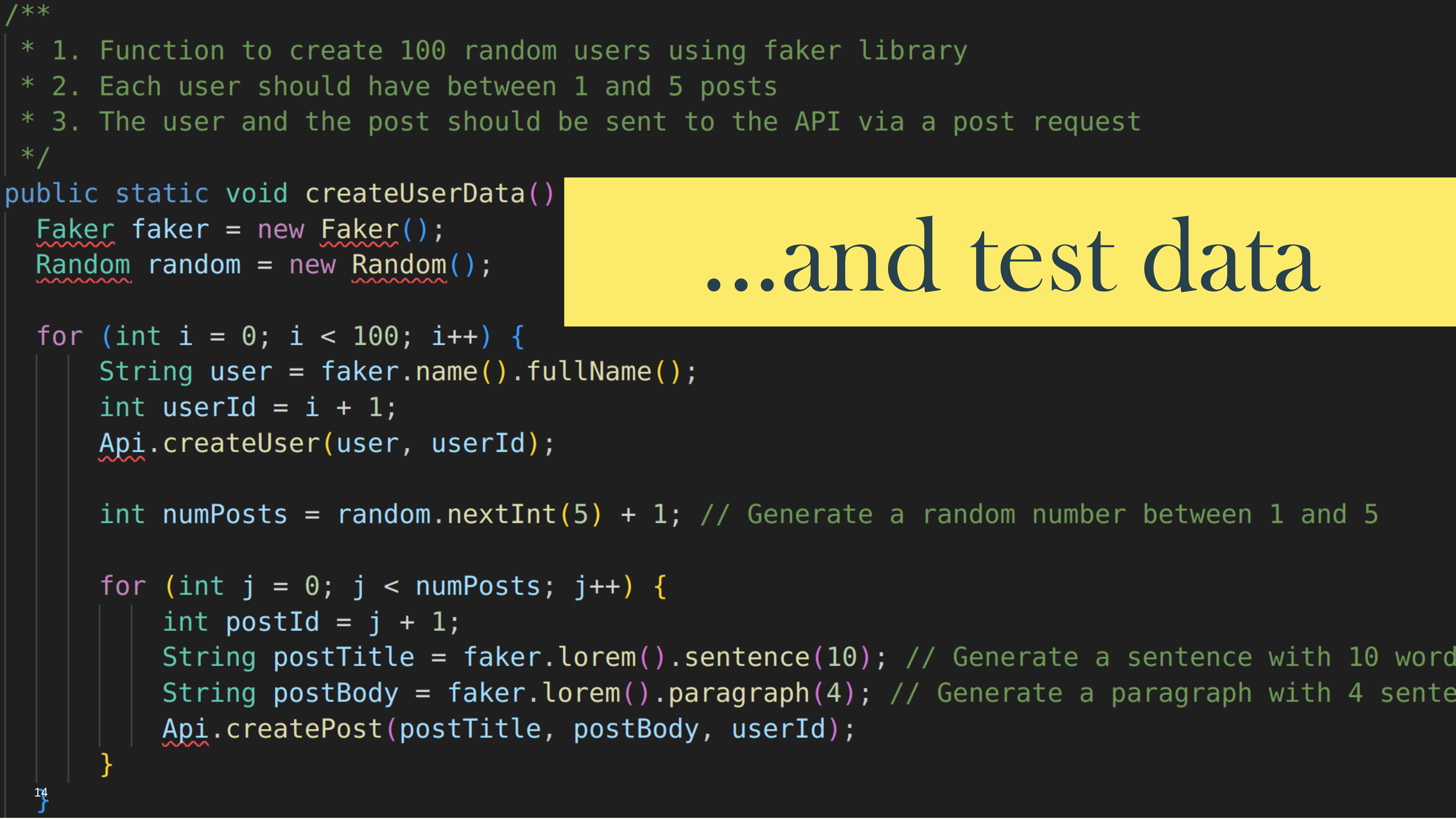


```
@Override
public Response createVolcano(String user, VolcanoDto volcanoDto) {
  Volcano model = mapper.toDomain(volcanoDto);
  model.setOperator(user);
  service.create(model);
  return Response.status(Status.CREATED)
      .header(HttpHeaders.CONTENT_LOCATION, uriInfo.getPath()
      + model.getId().getValue())
      .entity(mapper.toResponseDto(model))
      .build();
}

@Override
public Response getVolcano(UUID id) {
  Volcano model = service.get(new VolcanoId(id));
  return Response.ok(mapper.toResponseDto(model)).build();
}

@Override
public Response updateVolcano(UUID id, String user, VolcanoDto volca
  LOGGER.debug("Updating Volcano with id: " + id);
  Volcano volcano = mapper.toDomain(volcanoDto);
  volcano.setId(new VolcanoId(id));
  volcano.setOperator(user);
```

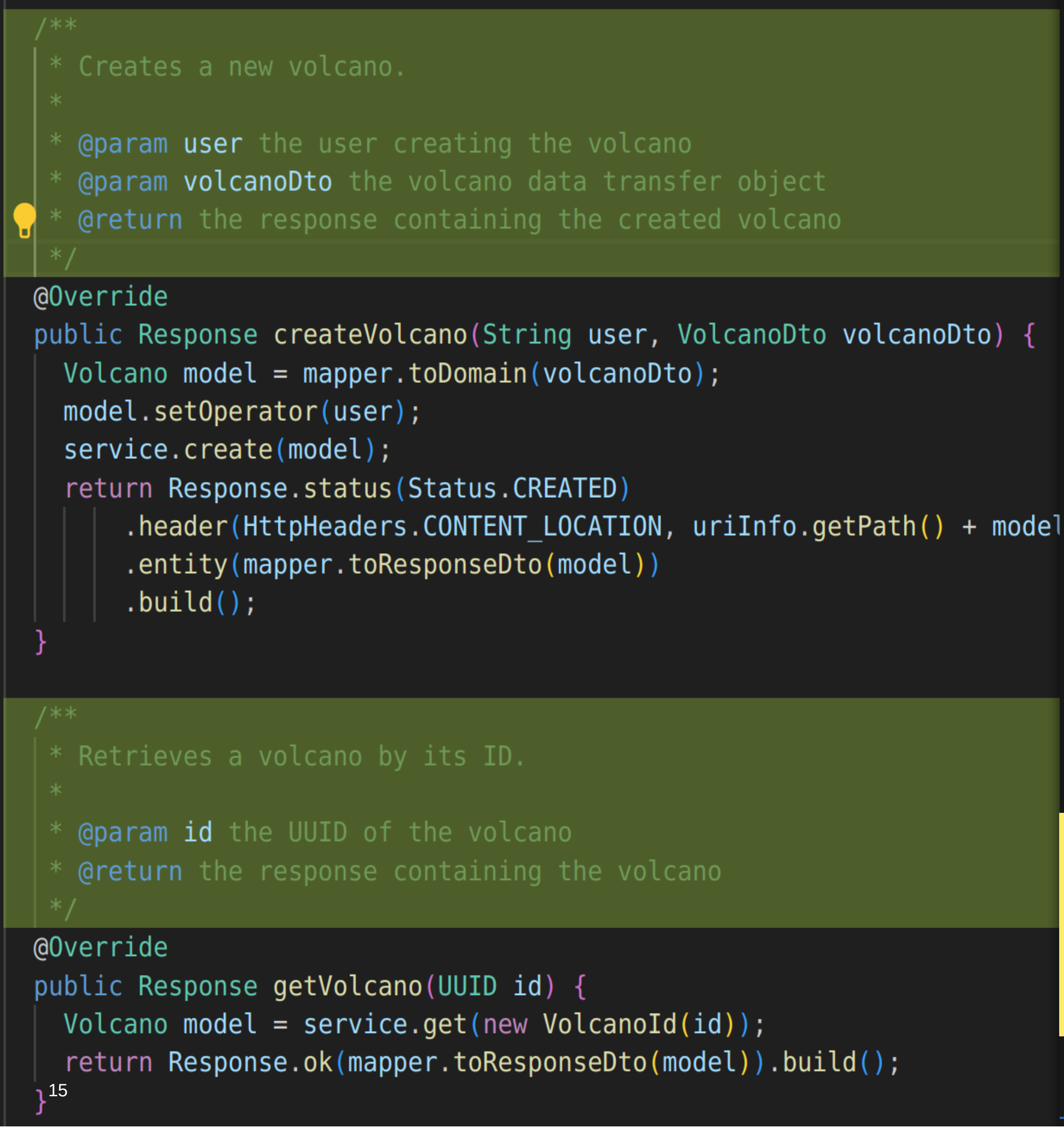


```
@Test
public void testCreateVolcano() {
    String user = "testUser";
    VolcanoDto volcanoDto = new VolcanoDto();
    Volcano volcano = new Volcano();
    volcano.setId(new VolcanoId(UUID.randomUUID()));
    when(mapper.toDomain(volcanoDto)).thenReturn(volcano);
    when(service.create(volcano)).thenReturn(volcano);
    when(mapper.toResponseDto(volcano)).thenReturn(volcanoDto);
    when(uriInfo.getPath()).thenReturn("/volcanos/");

    Response response = controller.createVolcano(user, volcanoDto);

    assertEquals(Response.Status.CREATED.getStatusCode(),
        response.getStatus());
    assertEquals("/volcanos/" + volcano.getId().getValue(),
        response.getHeaderString(HttpHeaders.CONTENT_LOCATION));
    assertEquals(volcanoDto, response.getEntity());
}

@Test
public void testGetVolcano() {
    UUID id = UUID.randomUUID();
    Volcano volcano = new Volcano();
    VolcanoDto volcanoDto = new VolcanoDto();
                        VolcanoId(id))).thenReturn(volcano);
                       seDto(volcano)).thenReturn(volcanoDto);
```

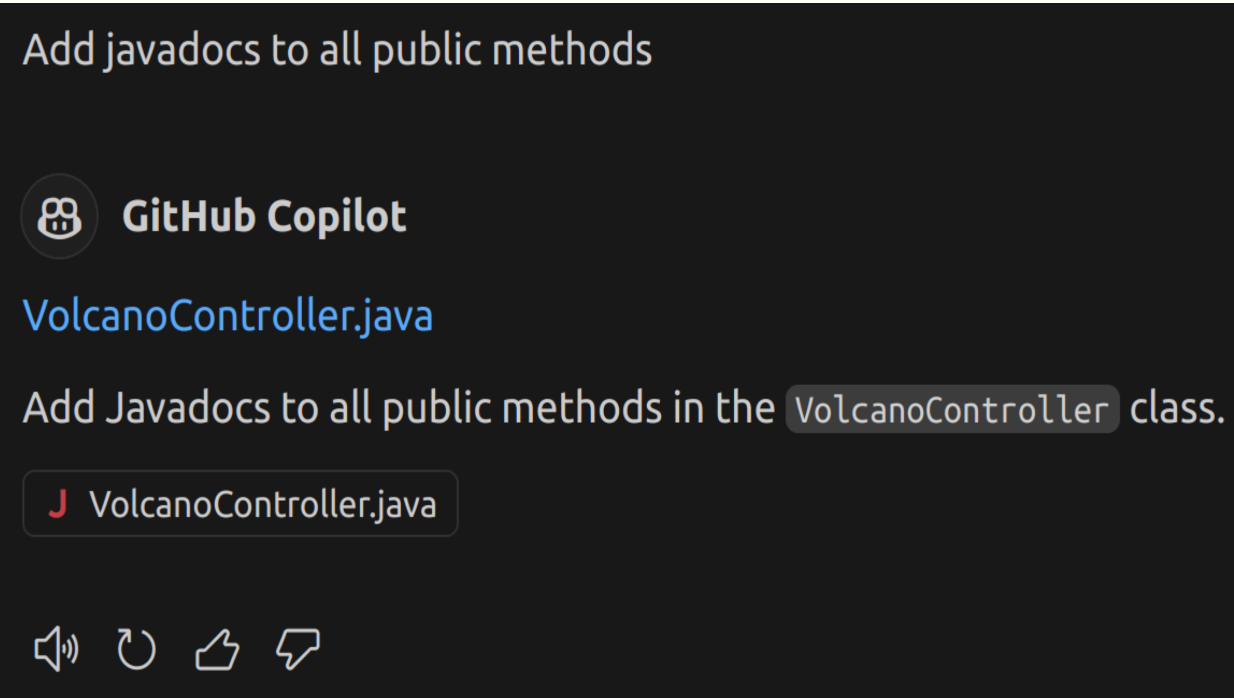
...and test data



```
/**
 * 1. Function to create 100 random users using faker library
 * 2. Each user should have between 1 and 5 posts
 * 3. The user and the post should be sent to the API via a post request
 */
public static void createUserData()
  Faker faker = new Faker();
  Random random = new Random();

  for (int i = 0; i < 100; i++) {
      String user = faker.name().fullName();
      int userId = i + 1;
      Api.createUser(user, userId);

      int numPosts = random.nextInt(5) + 1; // Generate a random number between 1 and 5

      for (int j = 0; j < numPosts; j++) {
          int postId = j + 1;
          String postTitle = faker.lorem().sentence(10); // Generate a sentence with 10 word
          String postBody = faker.lorem().paragraph(4); // Generate a paragraph with 4 sente
          Api.createPost(postTitle, postBody, userId);
      }
  }
```



```java
/**
 * Creates a new volcano.
 *
 * @param user the user creating the volcano
 * @param volcanoDto the volcano data transfer object
 * @return the response containing the created volcano
 */
@Override
public Response createVolcano(String user, VolcanoDto volcanoDto) {
  Volcano model = mapper.toDomain(volcanoDto);
  model.setOperator(user);
  service.create(model);
  return Response.status(Status.CREATED)
      .header(HttpHeaders.CONTENT_LOCATION, uriInfo.getPath() + model
      .entity(mapper.toResponseDto(model))
      .build();
}

/**
 * Retrieves a volcano by its ID.
 *
 * @param id the UUID of the volcano
 * @return the response containing the volcano
 */
@Override
public Response getVolcano(UUID id) {
  Volcano model = service.get(new VolcanoId(id));
  return Response.ok(mapper.toResponseDto(model)).build();
}
```



Add javadocs to all public methods

GitHub Copilot

VolcanoController.java

Add Javadocs to all public methods in the `VolcanoController` class.

Document

Expected impact

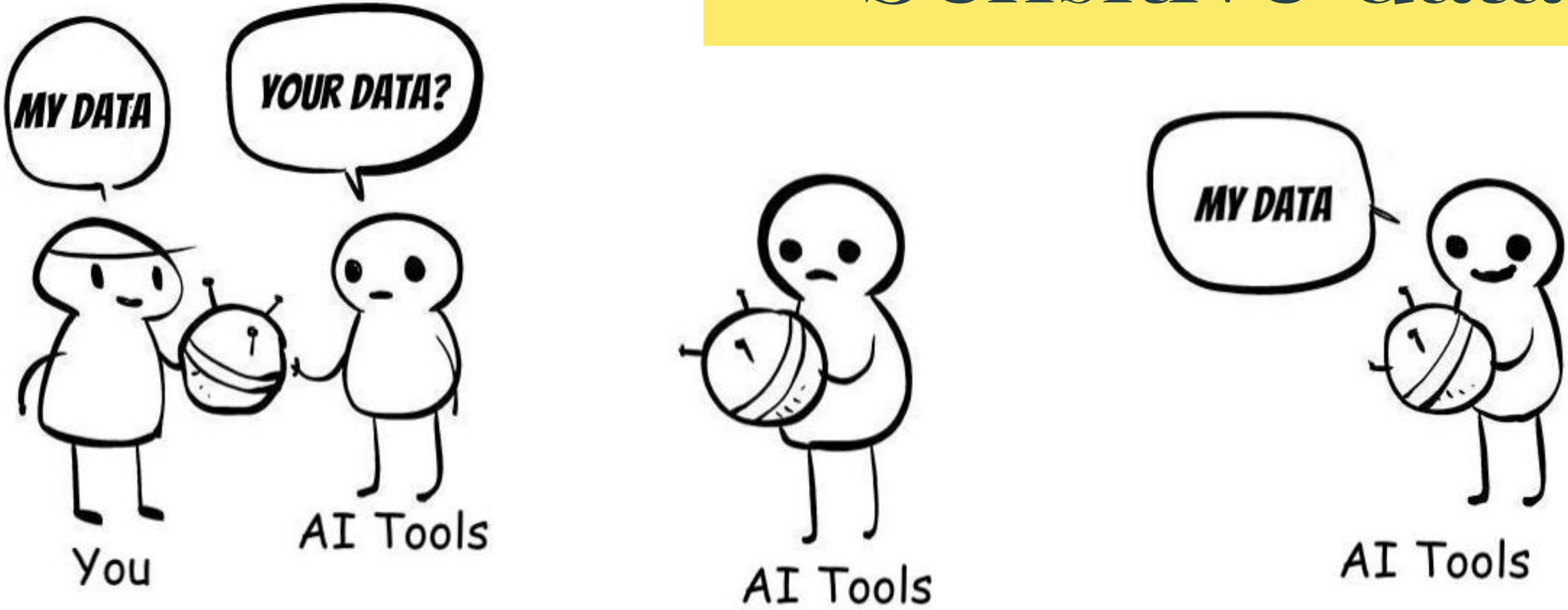
~25%

(std. error ~10%)*

[The Effects of Generative AI on High Skilled Work: Evidence from Three Field Experiments with Software Developers (Sep 2024)]

Issues

Sensitive data



Lack of control

Open-source alternatives

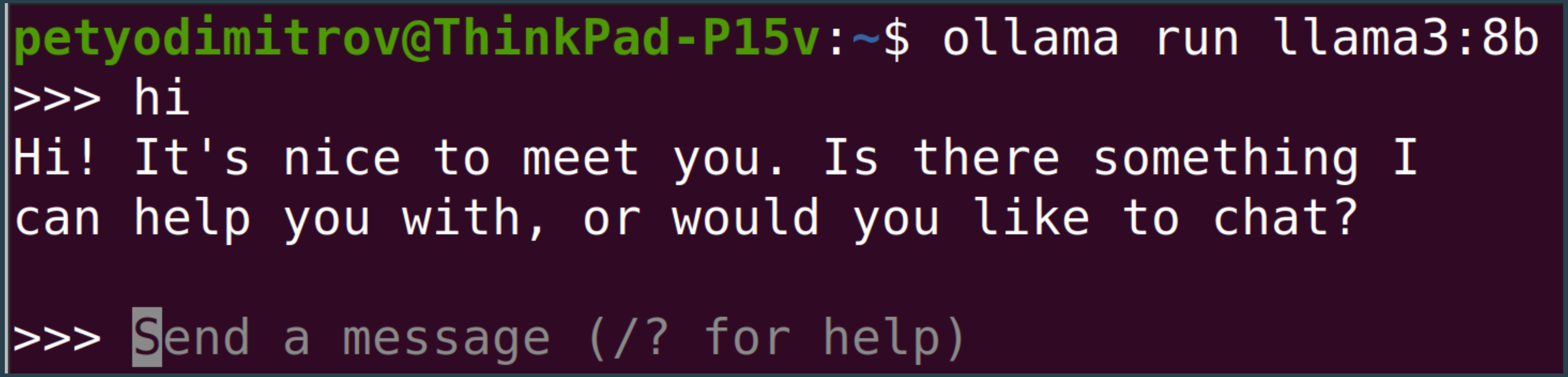
Ollama

Continue.dev

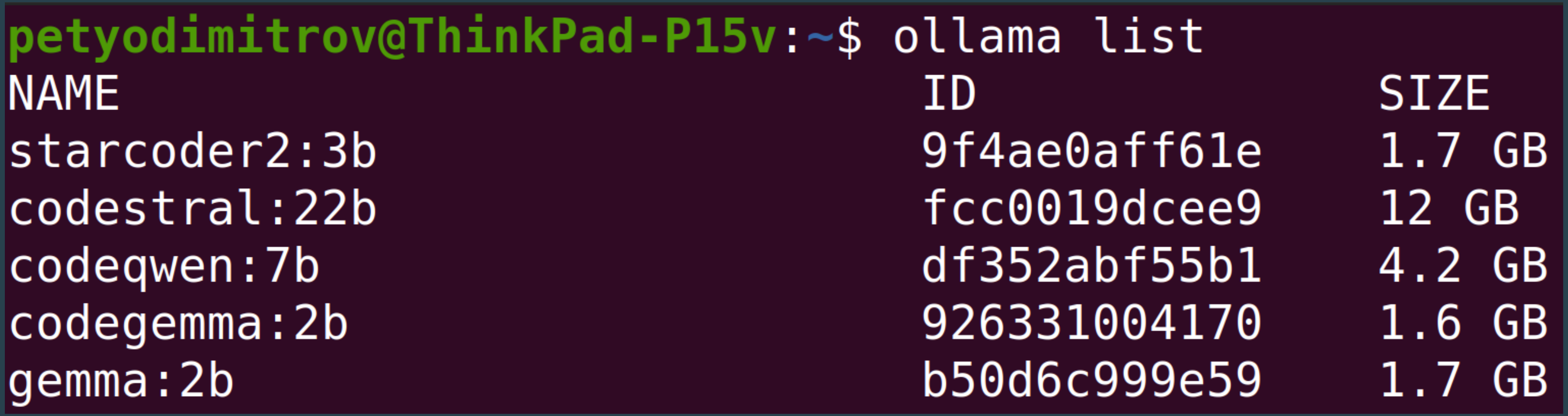
Open WebUI

Ollama

Docker for LLMs



```
petyodimitrov@ThinkPad-P15v:~$ ollama run llama3:8b
>>> hi
Hi! It's nice to meet you. Is there something I
can help you with, or would you like to chat?

>>> Send a message (/? for help)
```



```
petyodimitrov@ThinkPad-P15v:~$ ollama list
NAME                           ID              SIZE
starcoder2:3b                  9f4ae0aff61e    1.7 GB
codestral:22b                  fcc0019dcee9    12 GB
codeqwen:7b                    df352abf55b1    4.2 GB
codegemma:2b                   926331004170    1.6 GB
gemma:2b                       b50d6c999e59    1.7 GB
```

Specifics

Supported on Linux, MacOS & Windows

Works with GPU and CPU

Loads and unloads models dynamically

Related to llama.cpp, llamafile, vLLM, etc.

Resource requirements

Parameters	**Via CPU (RAM)**	**Via GPU (VRAM)**	**GPU card**
3B	8GB	4-6GB	-
7B	16GB	6GB	RTX 2060
13B	*32GB*	10-12GB	RTX 3060/3080
20B	64GB	*16GB*	RTX 3090, A100
65B+	128GB	40GB	dual RTX 3090, A100

* Apple M1 chips with 16GB RAM handle up to ~13B parameters

LLM models

Autocomplete:

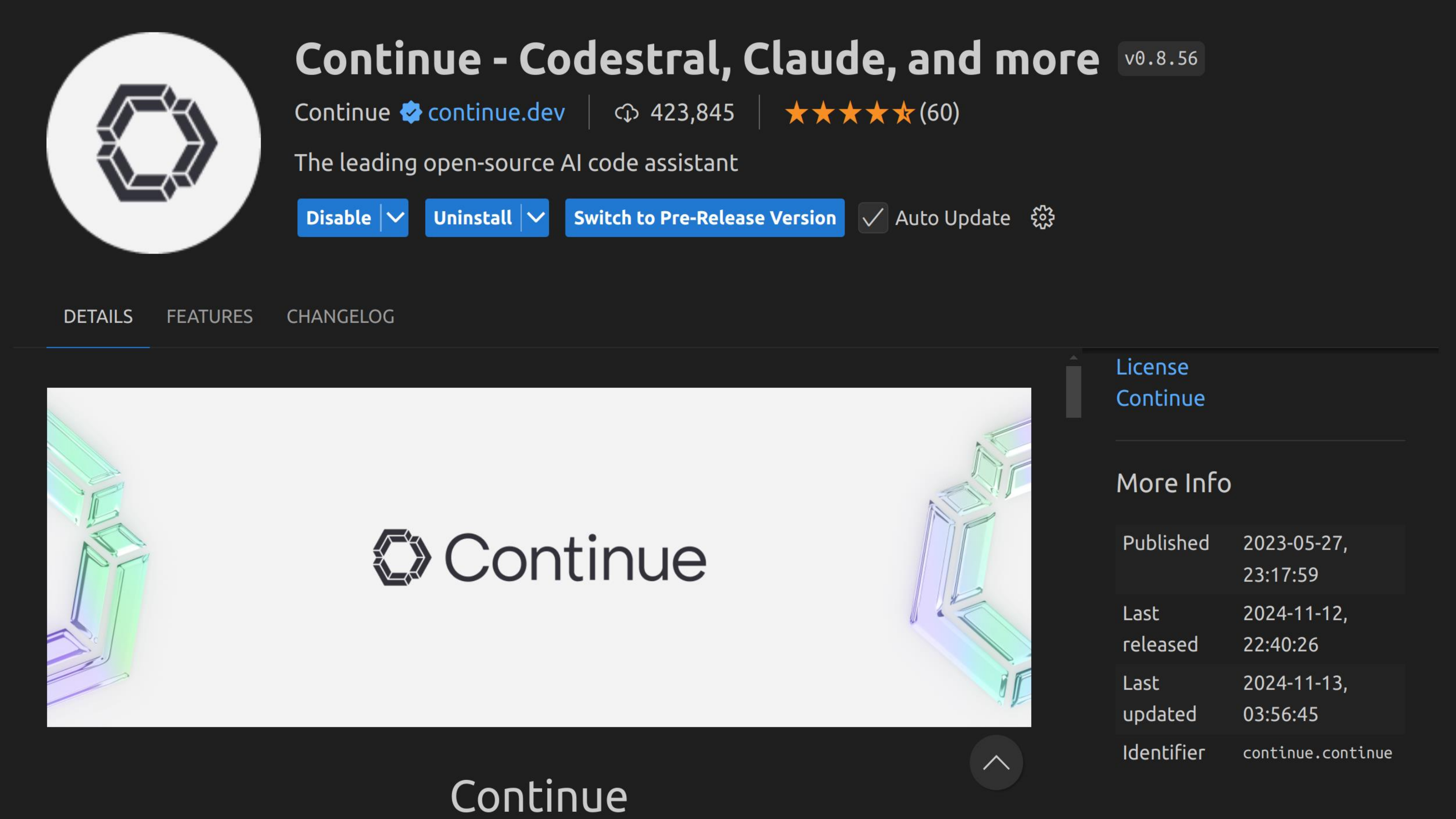
- **codestral:22b**
- llama3:8b
- deepseek-coder:6.7b
- starcoder2:3B
- deepseek-coder:1.3b

Chat:

- **Llama 3.1 405B**
- DeepSeek Coder 2 16B
- Llama 3.1 8B

https://evalplus.github.io/leaderboard.html
https://aider.chat/docs/leaderboards/#llm-code-editing-skill-by-model-release-date

Continue.dev



Continue - Codestral, Claude, and more

v0.8.56

Continue continue.dev | 423,845 | ★★★★☆ (60)

The leading open-source AI code assistant

More Info

Published	2023-05-27, 23:17:59
Last released	2024-11-12, 22:40:26
Last updated	2024-11-13, 03:56:45
Identifier	continue.continue

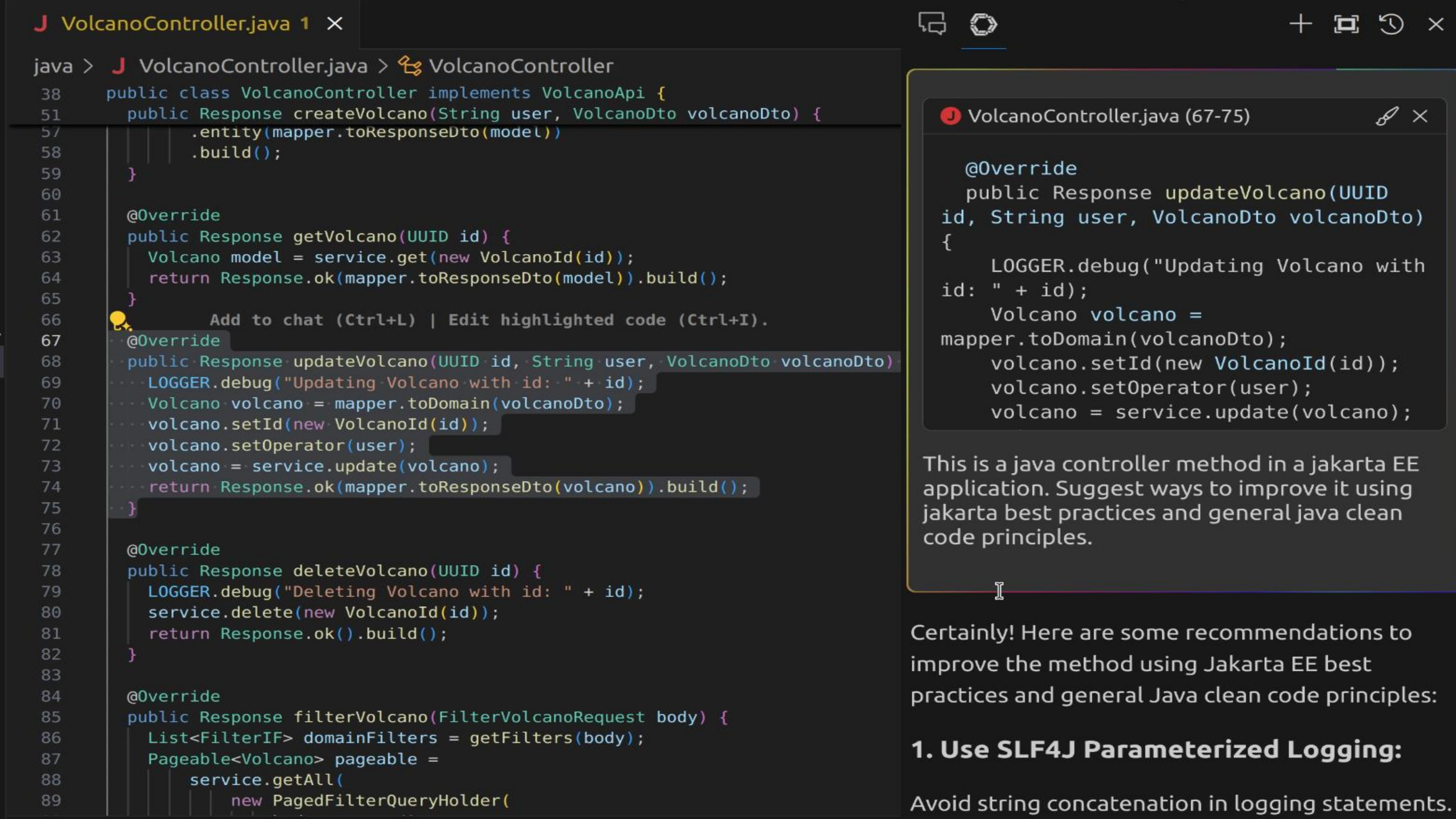
Continue

Specifics

Supported for VS Code and IntelliJ

Works with local and cloud LLMs

Features: autocomplete, chat, edit,
shortcuts, indexing workspace



```
38  public class VolcanoController implements VolcanoApi {
51    public Response createVolcano(String user, VolcanoDto volcanoDto) {
57          .entity(mapper.toResponseDto(model))
58          .build();
59    }
60
61    @Override
62    public Response getVolcano(UUID id) {
63      Volcano model = service.get(new VolcanoId(id));
64      return Response.ok(mapper.toResponseDto(model)).build();
65    }
66          Add to chat (Ctrl+L) | Edit highlighted code (Ctrl+I).
67    @Override
68    public Response updateVolcano(UUID id, String user, VolcanoDto volcanoDto)
69      LOGGER.debug("Updating Volcano with id: " + id);
70      Volcano volcano = mapper.toDomain(volcanoDto);
71      volcano.setId(new VolcanoId(id));
72      volcano.setOperator(user);
73      volcano = service.update(volcano);
74      return Response.ok(mapper.toResponseDto(volcano)).build();
75    }
76
77    @Override
78    public Response deleteVolcano(UUID id) {
79      LOGGER.debug("Deleting Volcano with id: " + id);
80      service.delete(new VolcanoId(id));
81      return Response.ok().build();
82    }
83
84    @Override
85    public Response filterVolcano(FilterVolcanoRequest body) {
86      List<FilterIF> domainFilters = getFilters(body);
87      Pageable<Volcano> pageable =
88          service.getAll(
89              new PagedFilterQueryHolder(
```

VolcanoController.java (67-75)

```
  @Override
  public Response updateVolcano(UUID id, String user, VolcanoDto volcanoDto) {
    LOGGER.debug("Updating Volcano with id: " + id);
    Volcano volcano = mapper.toDomain(volcanoDto);
    volcano.setId(new VolcanoId(id));
    volcano.setOperator(user);
    volcano = service.update(volcano);
```

This is a java controller method in a jakarta EE application. Suggest ways to improve it using jakarta best practices and general java clean code principles.

Certainly! Here are some recommendations to improve the method using Jakarta EE best practices and general Java clean code principles:

1. Use SLF4J Parameterized Logging:

Avoid string concatenation in logging statements.

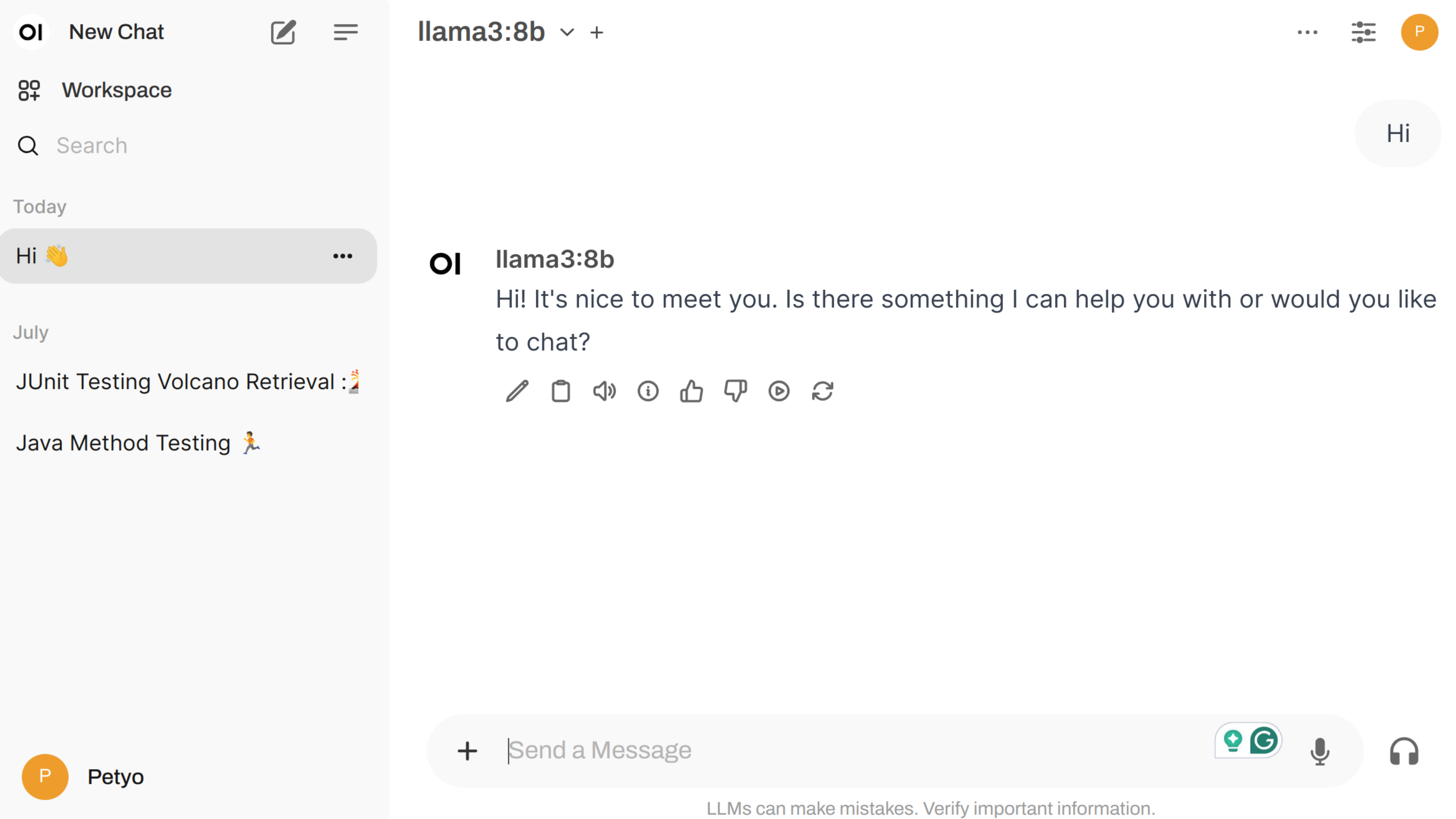
Open WebUI

Specifics

ChatGPT-like UI

Runs via Docker

Supports chat, multi-modality, RAG



Hi 👋

JUnit Testing Volcano Retrieval :🌋

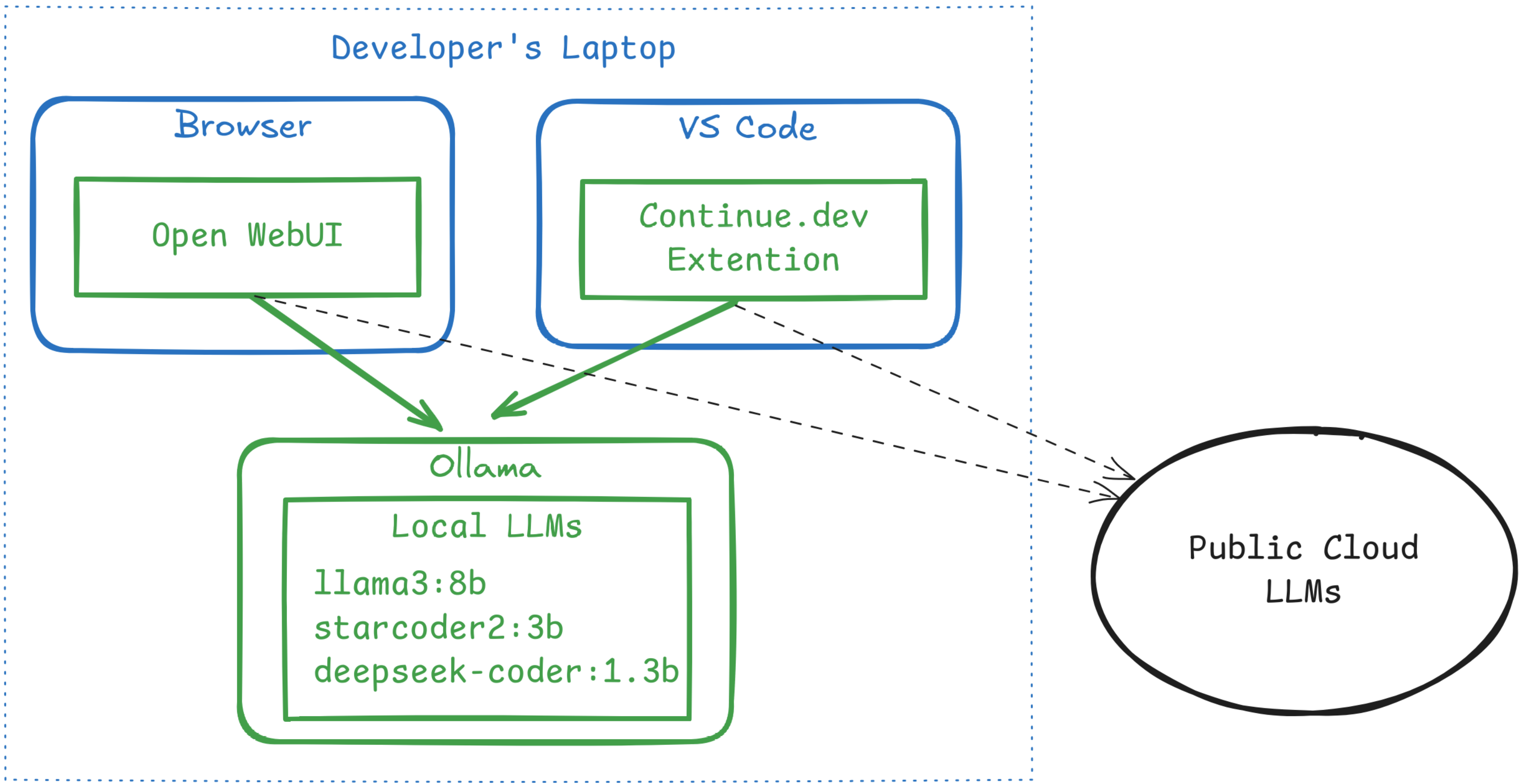
Java Method Testing 🏃

Hi

llama3:8b

Hi! It's nice to meet you. Is there something I can help you with or would you like to chat?

Developer toolkit



Developer's Laptop

Browser

Open WebUI

VS Code

Continue.dev
Extention

Ollama

Local LLMs

llama3:8b
starcoder2:3b
deepseek-coder:1.3b

Public Cloud
LLMs

Trade-offs

Requires GPU resource

Harder to scale (cost)

Requires more expertise & training

Behind state-of-the-art models

Honorable mentions

Cursor

Aider

Next steps

Next steps

Experiment with any of these*

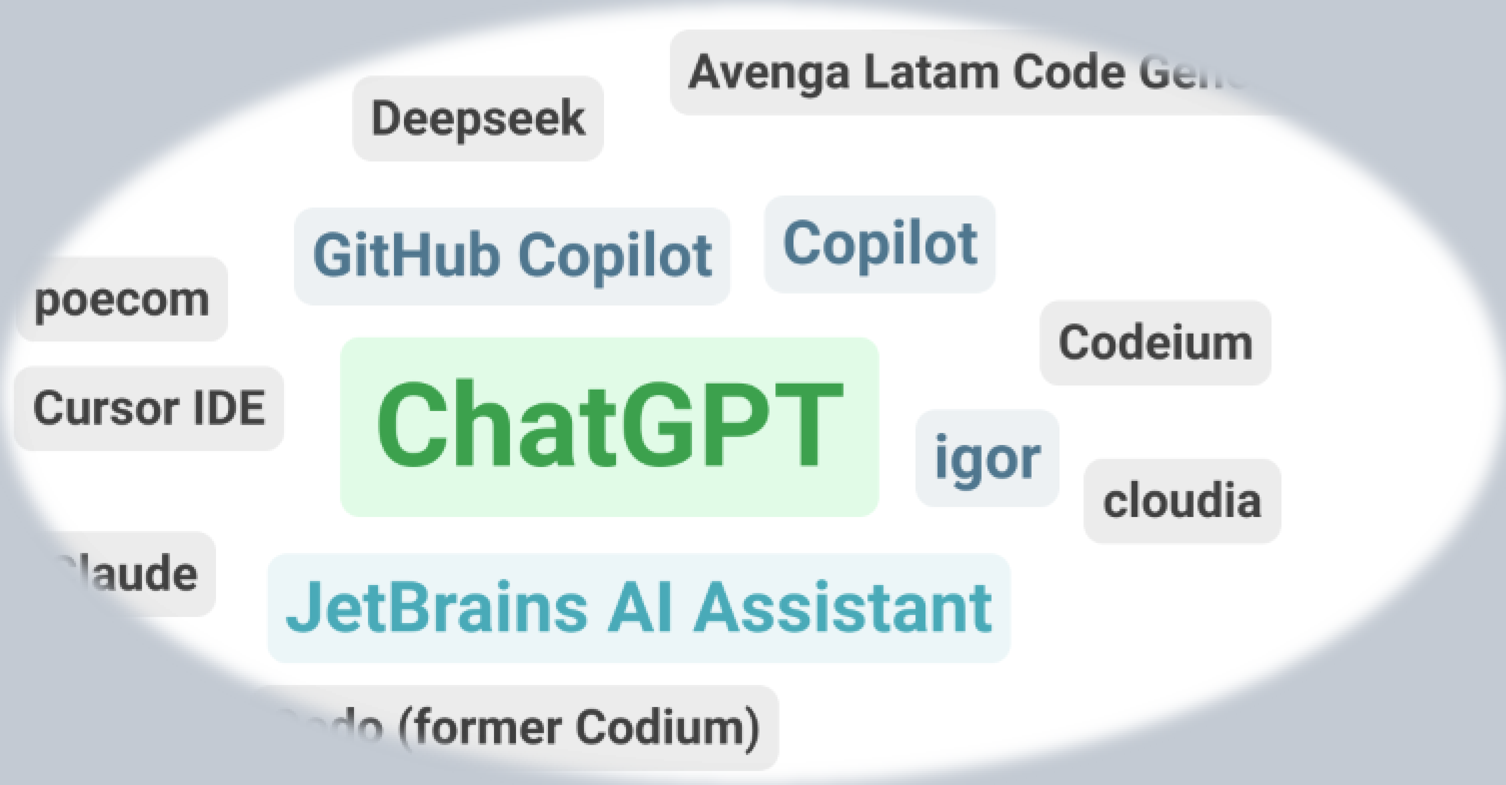
Run locally on CPU/GPU

Use Groq / MistralAI

Use a VM

Questions?

Tools used by TechTalk's audience



Thanks!

Petyo Dimitrov

Senior Software Architect, Musala Soft

Head of Data & AI, Qinshift

petyo.dimitrov@qinshift.com

Qinshift