


Jenkins



Jenkins

Smart-Ticket-Issue-Mana...

#11

Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#11'

Timings

Git Build Data

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

Timestamps

View as plain text

☒ System clock time

☒ Use browser timezone

☐ Elapsed time

☐ None

Console Output

Download

Copy

View as plain text

Started by user unknown or anonymous

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\Smart-Ticket-Issue-Management-System

[Pipeline] {

[Pipeline] timestamps

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Tool Install)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Checkout)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] git

07:42:16 The recommended git tool is: NONE

07:42:16 No credentials specified

07:42:16 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\Smart-Ticket-Issue-Management-System\.git # timeout=10

07:42:16 Fetching changes from the remote Git repository

Eureka Server

localhost:8761

Chat

spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment	test	Current time	2026-01-06T17:02:57 +0000
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	13
		Renews (last min)	0

DS Replicas

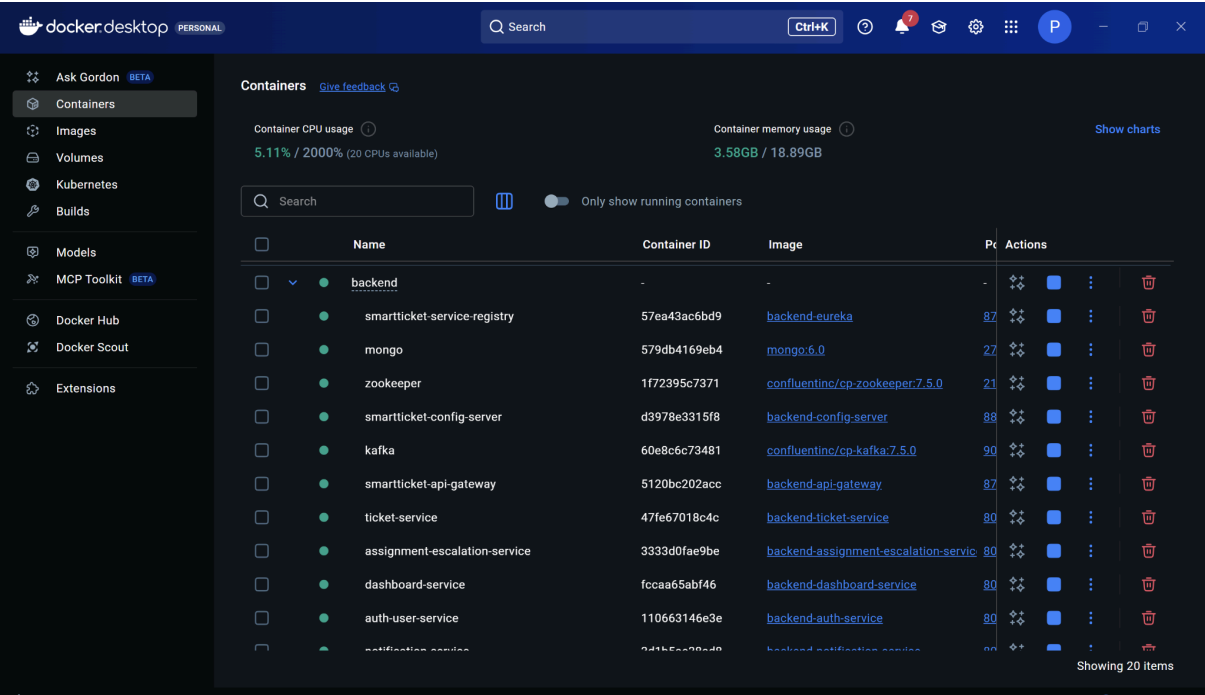
Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ASSIGNMENT-ESCALATION-SERVICE	n/a (1)	(1)	UP (1) - 26668733eec2assignment-escalation-service:8084
AUTH-USER-SERVICE	n/a (1)	(1)	UP (1) - 7854dea1b26aauth-user-service:8081
DASHBOARD-SERVICE	n/a (1)	(1)	UP (1) - 566e16fcb735dashboard-service:8082
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - 22a995699c84notification-service:8083
SMARTTICKET-API-GATEWAY	n/a (1)	(1)	UP (1) - 9f6e84bb008e-smartticket-api-gateway:8765
SMARTTICKET-CONFIG-SERVER	n/a (1)	(1)	UP (1) - 832260a2872c-smartticket-config-server:8888
TICKET-SERVICE	n/a (1)	(1)	UP (1) - e0a7f6df1025ticket-service:8085

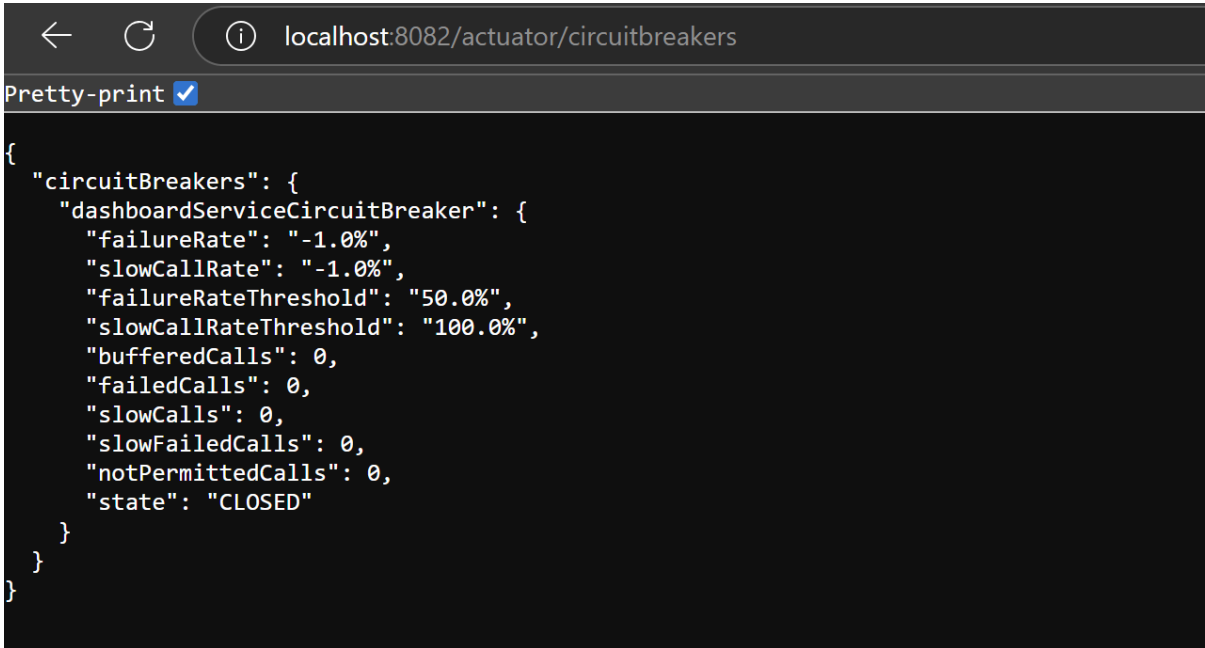
General Info

Name	Value
total-avail-memory	160mb

Docker

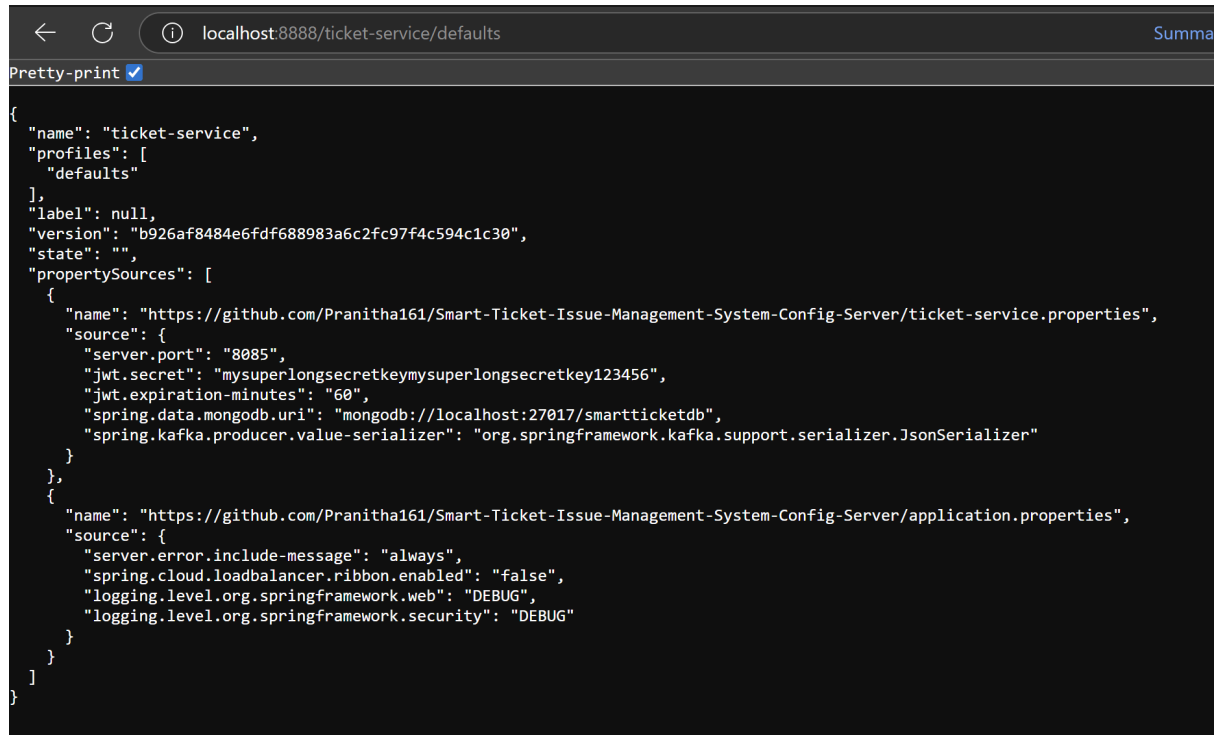


Circuit Breaker



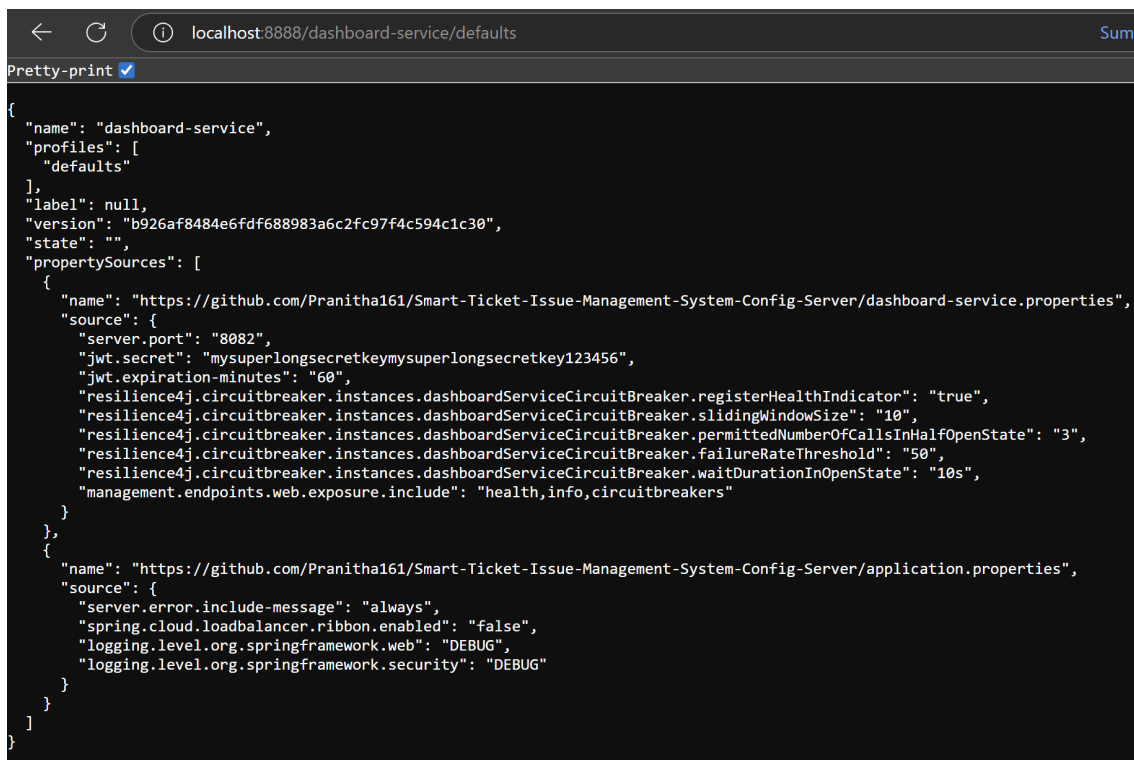
Config Server

Ticket-Service Defaults



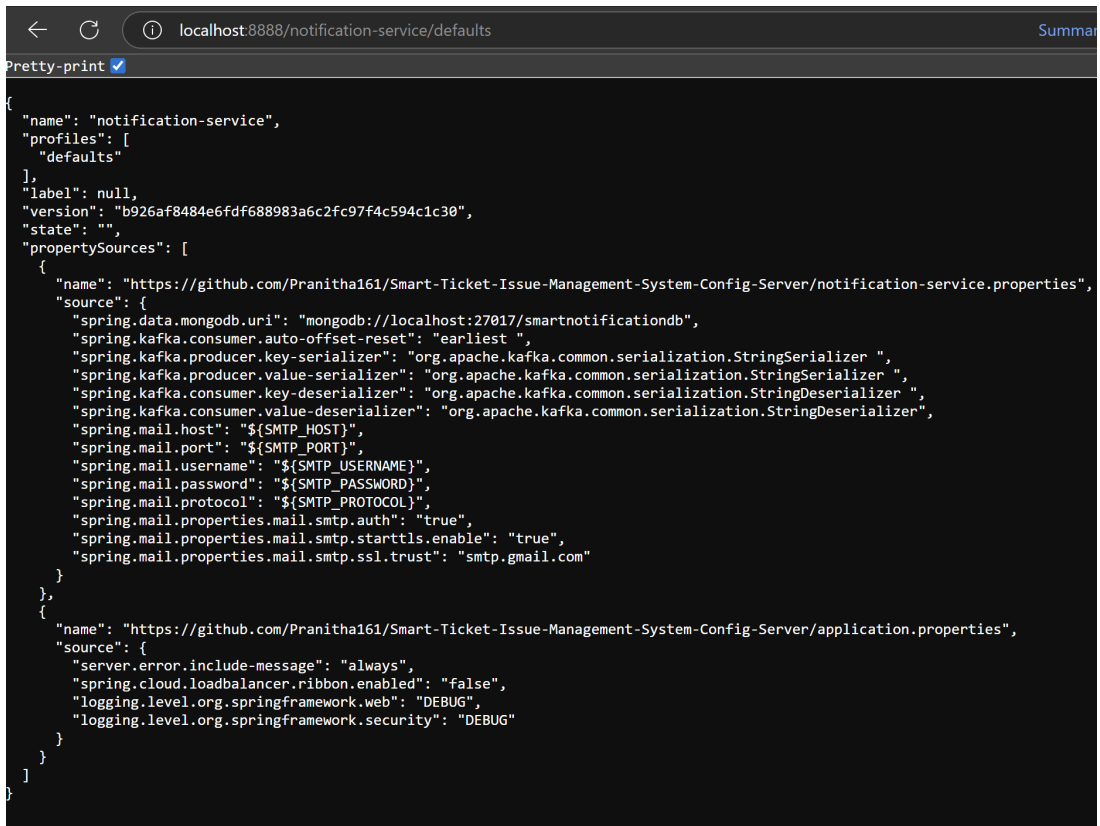
```
{
  "name": "ticket-service",
  "profiles": [
    "defaults"
  ],
  "label": null,
  "version": "b926af8484e6fdf688983a6c2fc97f4c594c1c30",
  "state": "",
  "propertySources": [
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/ticket-service.properties",
      "source": {
        "server.port": "8085",
        "jwt.secret": "mysuperlongsecretkeymysuperlongsecretkey123456",
        "jwt.expiration-minutes": "60",
        "spring.data.mongodb.uri": "mongodb://localhost:27017/smartticketdb",
        "spring.kafka.producer.value-serializer": "org.springframework.kafka.support.serializer.JsonSerializer"
      }
    },
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/application.properties",
      "source": {
        "server.error.include-message": "always",
        "spring.cloud.loadbalancer.ribbon.enabled": "false",
        "logging.level.org.springframework.web": "DEBUG",
        "logging.level.org.springframework.security": "DEBUG"
      }
    }
  ]
}
```

Dashboard Service Defaults



```
{
  "name": "dashboard-service",
  "profiles": [
    "defaults"
  ],
  "label": null,
  "version": "b926af8484e6fdf688983a6c2fc97f4c594c1c30",
  "state": "",
  "propertySources": [
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/dashboard-service.properties",
      "source": {
        "server.port": "8082",
        "jwt.secret": "mysuperlongsecretkeymysuperlongsecretkey123456",
        "jwt.expiration-minutes": "60",
        "resilience4j.circuitbreaker.instances.dashboardServiceCircuitBreaker.registerHealthIndicator": "true",
        "resilience4j.circuitbreaker.instances.dashboardServiceCircuitBreaker.slidingWindowSize": "10",
        "resilience4j.circuitbreaker.instances.dashboardServiceCircuitBreaker.permittedNumberOfCallsInHalfOpenState": "3",
        "resilience4j.circuitbreaker.instances.dashboardServiceCircuitBreaker.failureRateThreshold": "50",
        "resilience4j.circuitbreaker.instances.dashboardServiceCircuitBreaker.waitDurationInOpenState": "10s",
        "management.endpoints.web.exposure.include": "health,info,circuitbreakers"
      }
    },
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/application.properties",
      "source": {
        "server.error.include-message": "always",
        "spring.cloud.loadbalancer.ribbon.enabled": "false",
        "logging.level.org.springframework.web": "DEBUG",
        "logging.level.org.springframework.security": "DEBUG"
      }
    }
  ]
}
```

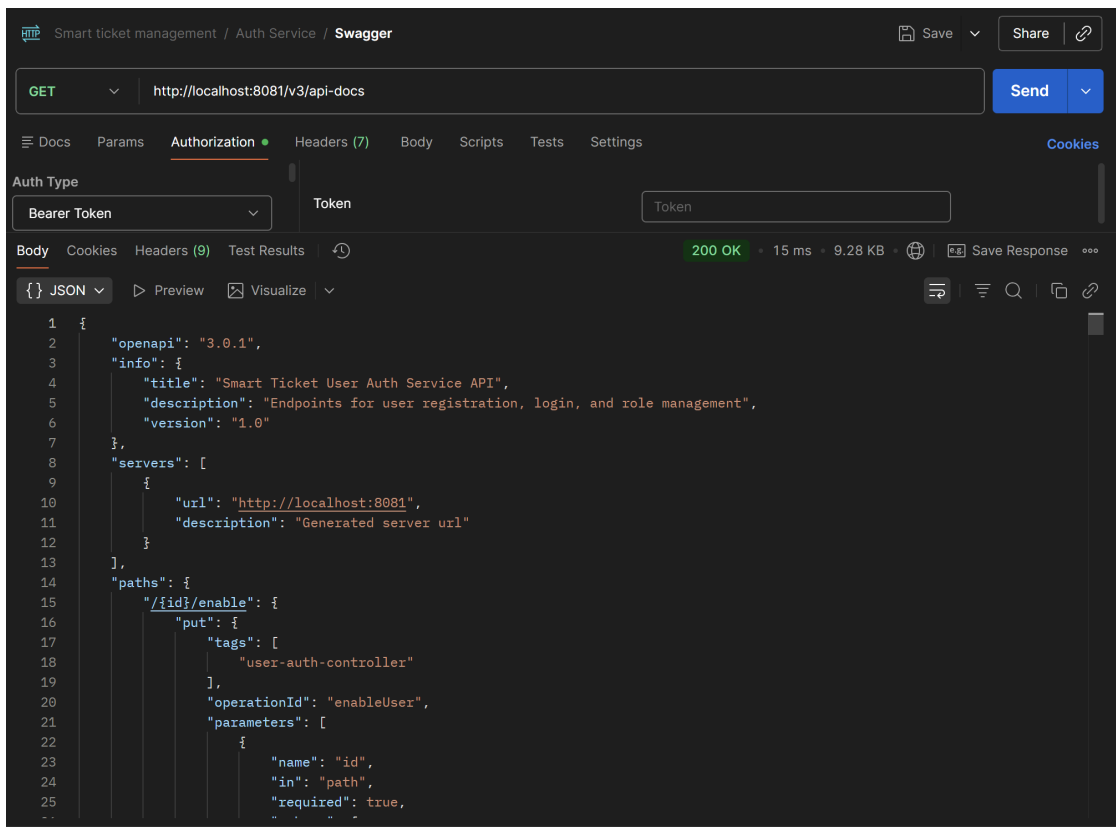
Notification Service Defaults



The screenshot shows a web browser at the URL `localhost:8888/notification-service/defaults`. The page displays a JSON configuration for the notification service. The configuration includes a name, profiles, a label, a version, a state, and property sources. The first property source is for the notification service properties, and the second is for the application properties.

```
{
  "name": "notification-service",
  "profiles": [
    "defaults"
  ],
  "label": null,
  "version": "b926af8484e6fdf688983a6c2fc97f4c594c1c30",
  "state": "",
  "propertySources": [
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/notification-service.properties",
      "source": {
        "spring.data.mongodb.uri": "mongodb://localhost:27017/smartnotificationdb",
        "spring.kafka.consumer.auto-offset-reset": "earliest",
        "spring.kafka.producer.key-serializer": "org.apache.kafka.common.serialization.StringSerializer",
        "spring.kafka.producer.value-serializer": "org.apache.kafka.common.serialization.StringSerializer",
        "spring.kafka.consumer.key-deserializer": "org.apache.kafka.common.serialization.StringDeserializer",
        "spring.kafka.consumer.value-deserializer": "org.apache.kafka.common.serialization.StringDeserializer",
        "spring.mail.host": "${SMTP_HOST}",
        "spring.mail.port": "${SMTP_PORT}",
        "spring.mail.username": "${SMTP_USERNAME}",
        "spring.mail.password": "${SMTP_PASSWORD}",
        "spring.mail.protocol": "${SMTP_PROTOCOL}",
        "spring.mail.properties.mail.smtp.auth": "true",
        "spring.mail.properties.mail.smtp.starttls.enable": "true",
        "spring.mail.properties.mail.smtp.ssl.trust": "smtp.gmail.com"
      }
    },
    {
      "name": "https://github.com/Pranitha161/Smart-Ticket-Issue-Management-System-Config-Server/application.properties",
      "source": {
        "server.error.include-message": "always",
        "spring.cloud.loadbalancer.ribbon.enabled": "false",
        "logging.level.org.springframework.web": "DEBUG",
        "logging.level.org.springframework.security": "DEBUG"
      }
    }
  ]
}
```

Swagger



The screenshot shows the Swagger UI for the Smart Ticket User Auth Service API. The interface includes a top bar with the API name and a 'Send' button. Below the top bar, there are tabs for 'Docs', 'Params', 'Authorization', 'Headers (7)', 'Body', 'Scripts', 'Tests', and 'Settings'. The 'Authorization' tab is selected, showing a 'Bearer Token' authentication type. The 'Body' tab is also visible, showing a JSON response. The JSON response is a Swagger specification for the API, including the title, description, version, servers, and paths.

```
{
  "openapi": "3.0.1",
  "info": {
    "title": "Smart Ticket User Auth Service API",
    "description": "Endpoints for user registration, login, and role management",
    "version": "1.0"
  },
  "servers": [
    {
      "url": "http://localhost:8081",
      "description": "Generated server url"
    }
  ],
  "paths": {
    "/{id}/enable": {
      "put": {
        "tags": [
          "user-auth-controller"
        ],
        "operationId": "enableUser",
        "parameters": [
          {
            "name": "id",
            "in": "path",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ]
      }
    }
  }
}
```