

Jenkins

The screenshot shows the Jenkins interface for a build named "Smart-Ticket-Issue-Mana...". The build number is "#11". The current tab is "Console Output". The console output window displays the following log entries:

```
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
[Pipeline] 
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
```

On the left sidebar, under "Console Output", the "View as plain text" option is selected. Below it, there are timestamping options: "System clock time" (selected), "Use browser timezone" (checked), "Elapsed time", and "None".

Eureka Server

The screenshot shows the Spring Eureka dashboard at localhost:8761. The top navigation bar includes links for "HOME" and "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) - 26668733eec2:assignment-escalation-service:8084
AUTH-USER-SERVICE	n/a (1)	(1)	UP (1) - 7854de1b26a:auth-user-service:8081
DASHBOARD-SERVICE	n/a (1)	(1)	UP (1) - 566e16fc735:dashboard-service:8082
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - 22a995699c84:notification-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) - e0a7f6df1025:ticket-service:8085

General Info

Name	Value
total-avail-memory	160mb

Docker

A screenshot of the Docker Desktop application interface. The left sidebar shows various sections like Ask Gordon, Containers, Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit, Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' with a 'Give feedback' button. It displays container usage statistics: 'Container CPU usage' at 5.11% / 2000% (20 CPUs available) and 'Container memory usage' at 3.58GB / 18.89GB. A search bar and a 'Only show running containers' toggle are present. A table lists 20 running containers with columns for Name, Container ID, Image, and Actions. The containers listed are: backend, smartticket-service-registry, mongo, zookeeper, smartticket-config-server, kafka, smartticket-api-gateway, ticket-service, assignment-escalation-service, dashboard-service, auth-user-service, and notification-service.

Name	Container ID	Image	P	Actions
backend	-	-	-	⋮
smartticket-service-registry	57ea43ac6bd9	backend-eureka	87	⋮
mongo	579db4169eb4	mongo:6.0	27	⋮
zookeeper	1f72395e7371	confluentinc/cp-zookeeper:7.5.0	21	⋮
smartticket-config-server	d3978e3315f8	backend-config-server	88	⋮
kafka	60e8c6c73481	confluentinc/cp-kafka:7.5.0	90	⋮
smartticket-api-gateway	5120bc202acc	backend-api-gateway	87	⋮
ticket-service	47fe67018c4c	backend-ticket-service	80	⋮
assignment-escalation-service	3333d0fae9be	backend-assignment-escalation-service	80	⋮
dashboard-service	fccaa65abf46	backend-dashboard-service	80	⋮
auth-user-service	110663146e3e	backend-auth-service	80	⋮
notification-service	ad1bca200d20	backend-notification-service	0	⋮

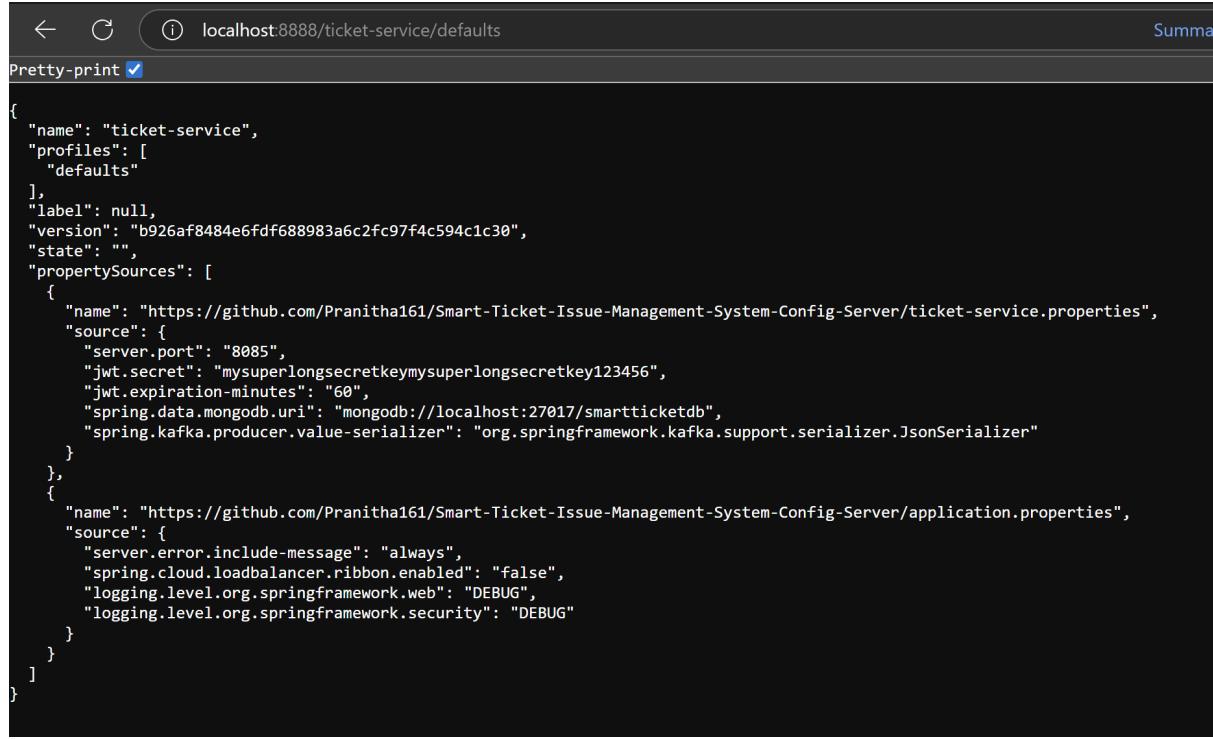
Circuit Breaker

A screenshot of a web browser displaying the URL `localhost:8082/actuator/circuitbreakers`. The page shows a JSON response for the 'circuitBreakers' endpoint. The response indicates that the 'dashboardServiceCircuitBreaker' is in a 'CLOSED' state, with configuration parameters: failureRate: "-1.0%", slowCallRate: "-1.0%", failureRateThreshold: "50.0%", slowCallRateThreshold: "100.0%", bufferedCalls: 0, failedCalls: 0, slowCalls: 0, slowFailedCalls: 0, and notPermittedCalls: 0.

```
{  
  "circuitBreakers": {  
    "dashboardServiceCircuitBreaker": {  
      "failureRate": "-1.0%",  
      "slowCallRate": "-1.0%",  
      "failureRateThreshold": "50.0%",  
      "slowCallRateThreshold": "100.0%",  
      "bufferedCalls": 0,  
      "failedCalls": 0,  
      "slowCalls": 0,  
      "slowFailedCalls": 0,  
      "notPermittedCalls": 0,  
      "state": "CLOSED"  
    }  
  }  
}
```

Config Server

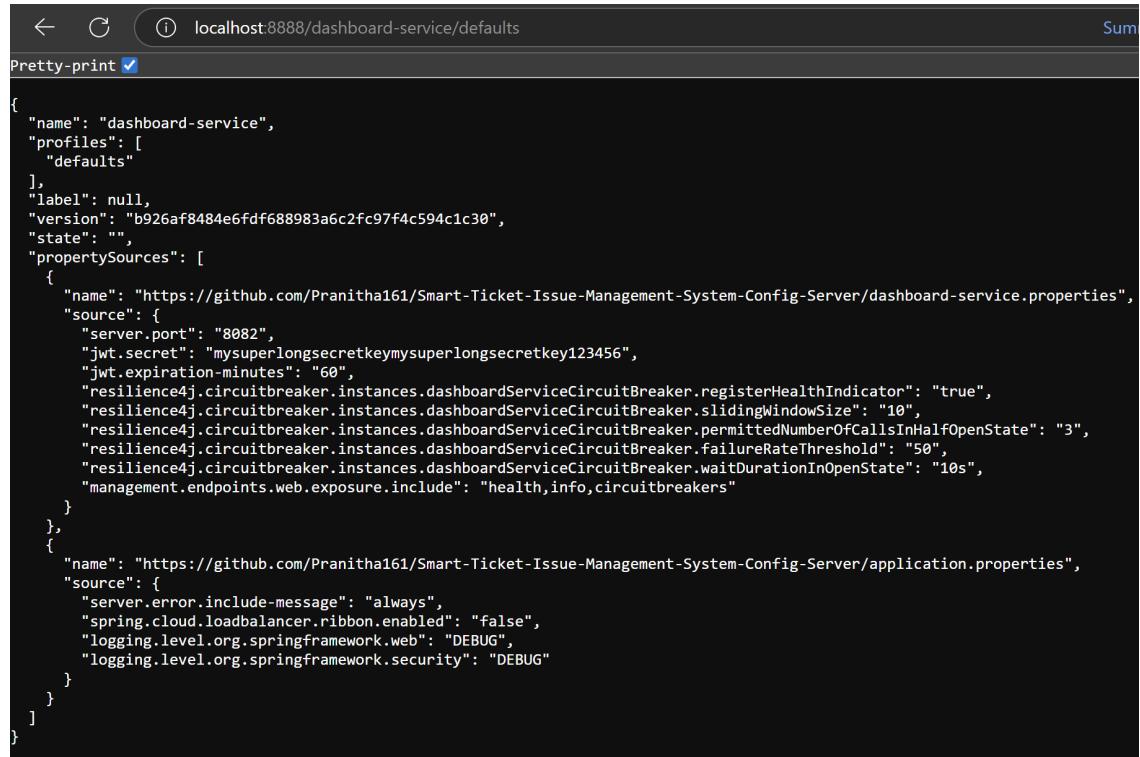
Ticket-Service Defaults



A screenshot of a web browser displaying the JSON configuration for the Ticket-Service Defaults. The URL is `localhost:8888/ticket-service/defaults`. The JSON output is as follows:

```
{  
  "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"  
        }  
      }  
    ]  
  ]  
}
```

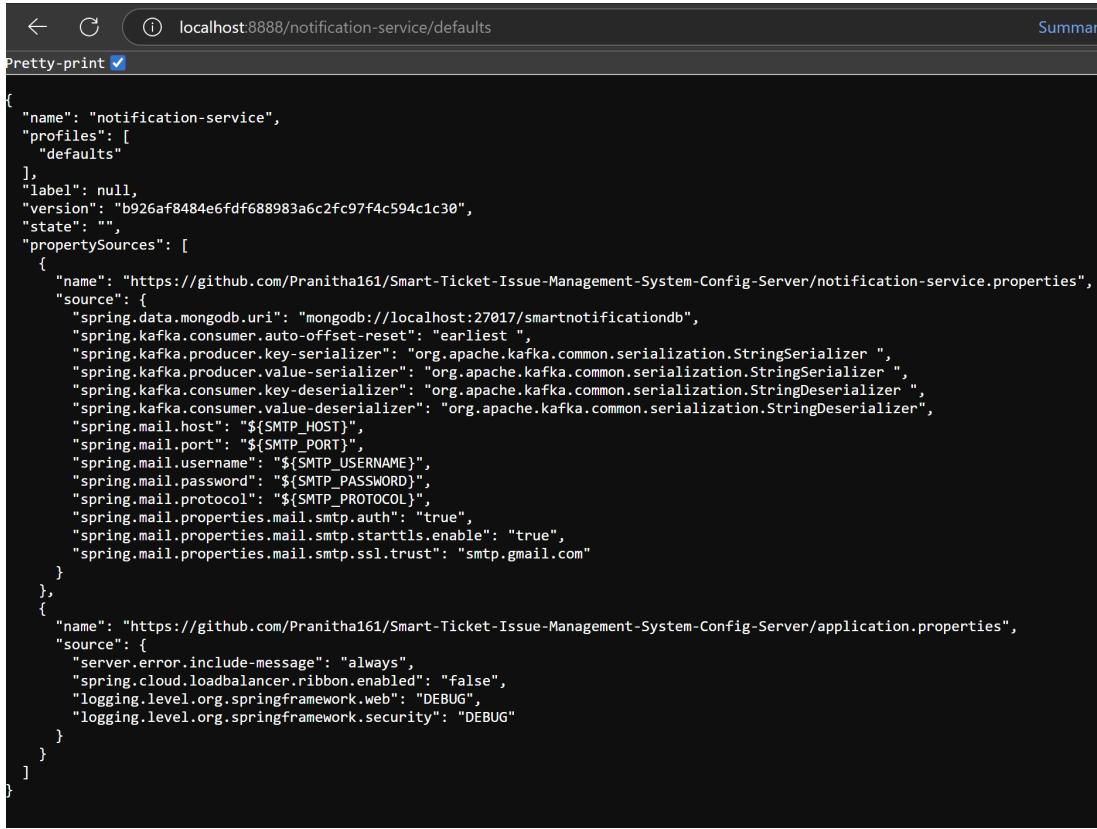
Dashboard Service Defaults



A screenshot of a web browser displaying the JSON configuration for the Dashboard Service Defaults. The URL is `localhost:8888/dashboard-service/defaults`. The JSON output is as follows:

```
{  
  "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"  
        }  
      }  
    ]  
  ]  
}
```

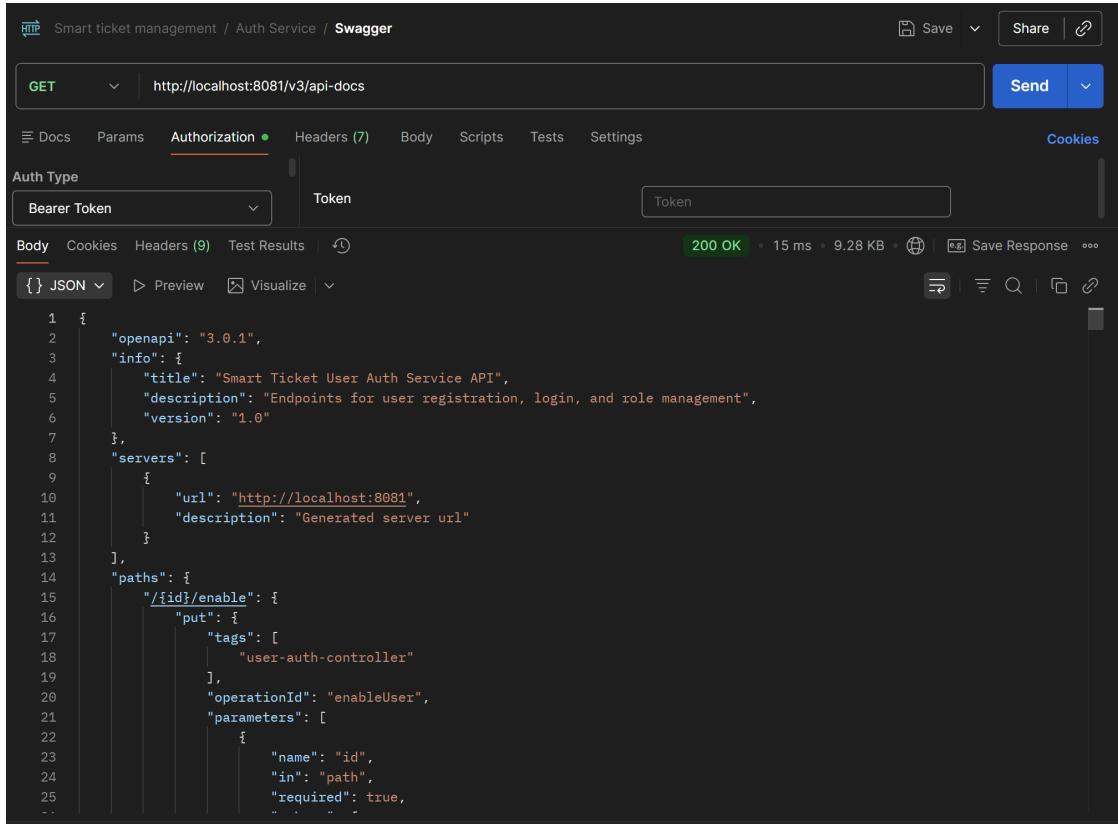
Notification Service Defaults



```
Pretty-print ✓

{
  "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 interface for the Smart ticket management / Auth Service API. The URL in the address bar is `http://localhost:8081/v3/api-docs`. The top navigation bar includes links for Docs, Params, Authorization (selected), Headers (7), Body, Scripts, Tests, Settings, Cookies, and a Send button. The Authorization section is set to "Bearer Token" and has a "Token" input field. Below the navigation, the response status is shown as 200 OK with 15 ms, 9.28 KB, and a Save Response button. The main content area displays the API's OpenAPI specification. The code block below shows the first few lines of the JSON:

```
1  {
2    "openapi": "3.0.1",
3    "info": {
4      "title": "Smart Ticket User Auth Service API",
5      "description": "Endpoints for user registration, login, and role management",
6      "version": "1.0"
7    },
8    "servers": [
9      {
10        "url": "http://localhost:8081",
11        "description": "Generated server url"
12      }
13    ],
14    "paths": {
15      "/{id}/enable": {
16        "put": {
17          "tags": [
18            "user-auth-controller"
19          ],
20          "operationId": "enableUser",
21          "parameters": [
22            {
23              "name": "id",
24              "in": "path",
25              "required": true,
26              "type": "string"
27            }
28          ],
29          "responses": {
30            "200": {
31              "description": "Success response"
32            }
33          }
34        }
35      }
36    }
37  }
```