Automated Fault Tracing - Sample Inputs and Outputs

This document provides comprehensive sample inputs and expected outputs for the Automated Fault Tracing system, demonstrating the various failure scenarios and API capabilities.

Sample 1: Test Assertion Failure

Input (POST /ingest)


```
{
  "build_id": "B001_TEST_ASSERTION",
  "log": "2025-09-14 10:00:00 [INFO] Starting test suite execution\n2025-09-14 10:00:05 [
  "metadata": {
    "job_name": "unit-tests",
    "branch": "feature/auth-improvements",
    "build_number": "156"
  },
  "commits": [
    {
      "sha": "a1b2c3d4",
      "author": "john.doe@example.com",
      "message": "Fix user session management and authentication flow",
      "changed_files": ["src/auth/session_manager.py", "src/auth/authenticator.py", "test
    }
  ]
}
```


Expected Output


```
{
  "build_id": "B001_TEST_ASSERTION",
  "label": "Test:Failure:Assertion",
  "confidence": 0.85,
  "summary": "Label: Test:Failure:Assertion (0.85) | Exception: AssertionError | Test: te
  "attribution": {
    "sha": "a1b2c3d4",
    "author": "john.doe@example.com",
    "score": 8,
    "changed_files": ["src/auth/session_manager.py", "src/auth/authenticator.py", "tests/
    "tests_detected": ["test_user_authentication"]
  },
  "status": "success"
}
```


Key Features Demonstrated:

- High confidence classification (0.85)
- Clear assertion error detection
- Strong commit attribution (score 8) based on changed test file
- Exception and test extraction from logs

Sample 2: Network Timeout Infrastructure Issue

Input (POST /ingest)


```
{
  "build_id": "B002_NETWORK_TIMEOUT",
  "log": "2025-09-14 14:30:00 [INFO] Initializing integration test suite\n2025-09-14 14:3
  "metadata": {
    "job_name": "integration-tests",
    "branch": "main",
    "build_number": "789"
  },
  "commits": [
    {
      "sha": "x9y8z7w6",
      "author": "network.team@example.com",
      "message": "Update network timeout configuration and retry logic",
      "changed_files": ["src/network/http_client.py", "src/network/retry_handler.py", "co
    }
  ]
}
```


Expected Output


```
{
  "build_id": "B002_NETWORK_TIMEOUT",
  "label": "Infra:Network:Timeout",
  "confidence": 0.92,
  "summary": "Label: Infra:Network:Timeout (0.92) | Exception: SocketTimeoutException | T
  "attribution": {
    "sha": "x9y8z7w6",
    "author": "network.team@example.com",
    "score": 7,
    "changed_files": ["src/network/http_client.py", "src/network/retry_handler.py", "conf
    "tests_detected": ["test_api_integration"]
  },
  "status": "success"
}
```


Key Features Demonstrated:

- Very high confidence (0.92) for clear network timeout patterns
- Infrastructure categorization vs test failure

- Network-related semantic boost in commit attribution
- Java stack trace parsing

Sample 3: Build Compilation Error

Input (POST /ingest)


```
{
  "build_id": "B003_COMPILE_ERROR",
  "log": "2025-09-14 09:15:00 [INFO] Starting Maven build process\n2025-09-14 09:15:05 [I
  "metadata": {
    "job_name": "maven-build",
    "branch": "feature/user-service-refactor",
    "build_number": "234"
  },
  "commits": [
    {
      "sha": "m5n6o7p8",
      "author": "dev.team@example.com",
      "message": "Refactor UserService to use new repository pattern",
      "changed_files": ["src/main/java/com/example/UserService.java", "src/main/java/com/
    }
  ]
}
```


Expected Output


```
{
  "build_id": "B003_COMPILE_ERROR",
  "label": "Infra:Build:Compilation",
  "confidence": 0.89,
  "summary": "Label: Infra:Build:Compilation (0.89) | Compilation error in UserService.ja
  "attribution": {
    "sha": "m5n6o7p8",
    "author": "dev.team@example.com",
    "score": 6,
    "changed_files": ["src/main/java/com/example/UserService.java", "src/main/java/com/ex
    "tests_detected": []
  },
  "status": "success"
}
```


Key Features Demonstrated:

- Build-time vs runtime failure classification
- File path extraction from compilation errors
- Attribution based on matching changed files with error locations
- Maven/Java specific error pattern recognition

Sample 4: Low Confidence / Unclassified Case

Input (POST /ingest)


```
{
  "build_id": "B004_UNCLASSIFIED",
  "log": "2025-09-14 16:45:00 [INFO] Starting deployment process\n2025-09-14 16:45:05 [IN
  "metadata": {
    "job_name": "deployment",
    "branch": "release/v2.1.0",
    "build_number": "445"
  },
  "commits": [
    {
      "sha": "q1r2s3t4",
      "author": "release.team@example.com",
      "message": "Prepare release v2.1.0 with minor bug fixes",
      "changed_files": ["version.txt", "CHANGELOG.md", "deployment/staging.yaml"]
    }
  ]
}
```


Expected Output


```
{
  "build_id": "B004_UNCLASSIFIED",
  "label": "UNCLASSIFIED",
  "confidence": 0.23,
  "summary": "Label: UNCLASSIFIED (0.23) | Generic deployment failure | No specific erro
  "attribution": {
    "sha": "q1r2s3t4",
    "author": "release.team@example.com",
    "score": 1,
    "changed_files": ["version.txt", "CHANGELOG.md", "deployment/staging.yaml"],
    "tests_detected": []
  },
  "status": "success"
}
```


Key Features Demonstrated:

- Low confidence triggering potential ML fallback
- Generic error patterns falling back to UNCLASSIFIED
- Weak attribution due to vague error messages
- Deployment-specific context handling

ML Training Data Sample

For training the optional ML classifier, here are sample labeled texts:

```
[
  {"text": "AssertionError: expected 2 got 3", "label": "Test:Failure:Assertion"},
  {"text": "assert user.is_active == True failed", "label": "Test:Failure:Assertion"},
  {"text": "Connection timed out after 30 seconds", "label": "Infra:Network:Timeout"},
  {"text": "java.net.SocketTimeoutException: Read timed out", "label": "Infra:Network:Tim
  {"text": "DNS resolution failed for hostname api.example.com", "label": "Infra:Network:
  {"text": "compilation failed: cannot find symbol", "label": "Infra:Build:Compilation"},
  {"text": "BUILD FAILED due to compile errors", "label": "Infra:Build:Compilation"},
  {"text": "ModuleNotFoundError: No module named 'requests'", "label": "Infra:Build:Deper
  {"text": "package not found: com.example.utils", "label": "Infra:Build:Dependencies"},
  {"text": "Test timeout: test_long_running_operation", "label": "Test:Failure:Timeout"},
  {"text": "OutOfMemoryError: Java heap space", "label": "Infra:Resource:Memory"},
  {"text": "authentication failed: invalid credentials", "label": "Security:Authenticatic
  {"text": "SSL certificate verification failed", "label": "Security:Certificate"},
  {"text": "database connection timeout", "label": "Database:Connection"},
  {"text": "SQL syntax error near 'SELECT'", "label": "Database:Query"}
]
```

Other API Endpoint Examples

GET /features Response

```
{
  "log_normalization": "active",
  "rule_classification": "active",
  "summarization": "active",
  "commit_attribution": "active",
  "reporting": "active",
  "ml_classification": "available"
}
```

GET /health Response

```
{
  "status": "healthy",
  "builds_stored": 4,
  "ml_enabled": true,
  "timestamp": 1694700000.456
}
```

Sample Markdown Report Output

```
# Build Analysis Report

**Build ID:** B001_TEST_ASSERTION
**Analysis Date:** 2025-09-14 10:00:00

## Classification Results

- **Label:** Test:Failure:Assertion
- **Confidence:** 0.85

### Score Breakdown

- **Test:Failure:Assertion:** 14
- **Test:Failure:Exception:** 4
- **UNCLASSIFIED:** 2

## Summary

Label: Test:Failure:Assertion (0.85) | Exception: AssertionError | Test: test_user_auther

### Exceptions Found

- AssertionError

### Tests Involved

- test_user_authentication

### Assertion Details
```

assert len(active_sessions) == 2, f"Expected 2 active sessions, got {len(active_sessions)}"

```

## Suspected Commit

- **SHA:** a1b2c3d4
- **Author:** john.doe@example.com
- **Attribution Score:** 8

### Changed Files

- src/auth/session_manager.py
- src/auth/authenticator.py
- tests/test_authentication.py

### Related Tests

- test_user_authentication
```

```
---
*Generated by Automated Fault Tracing System*
```

CURL Command Examples

```
# 1. Ingest a build log
curl -X POST http://127.0.0.1:8000/ingest \
  -H "Content-Type: application/json" \
  -d '{
    "build_id": "B001_TEST_ASSERTION",
    "log": "2025-09-14 10:00:00 [INFO] Starting test suite...",
    "metadata": {"job_name": "unit-tests"},
    "commits": [{"sha": "a1b2c3d4", "author": "john.doe", "message": "Fix auth", "changed
  }'

# 2. Get build analysis
curl http://127.0.0.1:8000/build/B001_TEST_ASSERTION

# 3. Get markdown report
curl http://127.0.0.1:8000/build/B001_TEST_ASSERTION/report.md

# 4. Train ML model (if ENABLE_ML=1)
curl -X POST http://127.0.0.1:8000/train \
  -H "Content-Type: application/json" \
  -d '[
    {"text": "AssertionError: expected 2 got 3", "label": "Test:Failure:Assertion"},
    {"text": "Connection timed out", "label": "Infra:Network:Timeout"}
  ]'

# 5. Get feature status
curl http://127.0.0.1:8000/features

# 6. Health check
curl http://127.0.0.1:8000/health
```

Summary Statistics

- **Total test cases:** 4 comprehensive scenarios
- **ML training samples:** 15 labeled examples across 6 categories
- **Failure categories covered:** Test Failures, Network Issues, Build Issues, Resource Issues, Security Issues, Database Issues
- **Confidence range:** 0.23 (low, triggers ML fallback) to 0.92 (high, strong rule match)
- **Attribution scores:** 1 (weak) to 8 (strong correlation between commit changes and failure)
- **Languages supported:** Python, Java, SQL error patterns
- **Build systems:** Maven, generic CI/CD, deployment pipelines

This comprehensive sample set demonstrates the system's ability to handle diverse failure scenarios while maintaining explainable, deterministic classification with optional ML

augmentation for edge cases.