

# Debug Mode API Validation Issue

## Issue Summary

The Kore.ai Agentic App Platform API's /runs/execute endpoint only accepts "thoughts" as a valid value for debug.debugMode, but rejects "all" and "function-call" with a validation error.

## Current Behavior

When making a request with debug.debugMode set to anything other than "thoughts", the API returns:

```
{
  "errors": [
    {
      "msg": "debug.debugMode must be thoughts",
      "code": 400
    }
  ],
  "timestamp": "2026-02-13T04:31:22.346Z",
  "path": "/api/v2/apps/{appId}/environments/{envName}/runs/execute"
}
```

## Test Results

debugMode Value	Result	HTTP Status
"thoughts"	✓ Success	200/201
"all"	✗ Rejected	400
"function-call"	✗ Rejected	400
(not specified)	✓ Success	200/201

## Expected Behavior

**Option 1:** Support all three debug modes

- "thoughts" - Thought process debugging
- "all" - Full debug information
- "function-call" - Function call debugging

**Option 2:** Clarify documentation

- If only "thoughts" is intended to be supported, update API documentation to reflect this
- Remove references to other debug modes from schemas/docs

**Option 3:** Improve error messages

- Provide more context about why certain modes are rejected
- Explain what modes are supported and when they might be available

## Reproduction Steps

### Quick Test (using curl)

*# This WORKS*

```
curl -X POST "https://agent-platform.kore.ai/api/v2/apps/{APP_ID}/environments/{ENV_NAME}/  
-H "Content-Type: application/json" \  
-H "x-api-key: {API_KEY}" \  
-d '{  
  "sessionIdentity": [  
    {"type": "userReference", "value": "test-user"},  
    {"type": "sessionReference", "value": "test-session"}  
  ],  
  "input": [{"type": "text", "content": "Test"}],  
  "debug": {"enable": true, "debugMode": "thoughts"}  
'
```

*# This FAILS with 400*

```
curl -X POST "https://agent-platform.kore.ai/api/v2/apps/{APP_ID}/environments/{ENV_NAME}/  
-H "Content-Type: application/json" \  
-H "x-api-key: {API_KEY}" \  
-d '{  
  "sessionIdentity": [  
    {"type": "userReference", "value": "test-user"},  
    {"type": "sessionReference", "value": "test-session"}  
  ],  
  "input": [{"type": "text", "content": "Test"}],  
  "debug": {"enable": true, "debugMode": "all"}  
'
```

### Comprehensive Test Script

A complete test script is available: `test_debug_mode_issue.sh`

1. Edit the script to add your credentials:

```
API_KEY="your-key-here"  
APP_ID="your-app-id"  
ENV_NAME="stage" # or "draft"
```

2. Run the script:

```
chmod +x test_debug_mode_issue.sh  
./test_debug_mode_issue.sh
```

The script will test all debug modes and provide a detailed report.

## Impact

### For API Users

- Limited debugging capabilities - only “thoughts” mode is available
- Confusion when trying to use other documented modes
- Need workarounds in client libraries to avoid validation errors

## For Client Implementations

- Must handle the limitation by either:
  - Only exposing “thoughts” mode (loses flexibility)
  - Allowing all modes but letting API reject them (better UX)
  - Not sending debugMode field by default (current workaround)

## Questions for Platform Team

1. **Is this intentional?**
  - Is the restriction to “thoughts” only by design?
  - Are “all” and “function-call” modes planned for future releases?
2. **Documentation alignment**
  - Should API documentation be updated to reflect only “thoughts” is valid?
  - Are there any circumstances where other modes would work?
3. **Error messaging**
  - Could the error message explain why only “thoughts” is accepted?
  - Could it indicate when other modes might become available?
4. **Future plans**
  - Will additional debug modes be supported?
  - Should clients prepare for multiple modes to be available?

## Workaround

For now, the recommended approach is:

- Use --debug without specifying a mode (sends {"debug": {"enable": true}})
- Only use --debug-mode thoughts when explicit thoughts debugging is needed
- Other modes are available in the CLI but will be rejected by the API

## Appendix: Complete Test Script

The following bash script can be used to reproduce and demonstrate the issue. Save it as test\_debug\_mode\_issue.sh, set your credentials, make it executable with `chmod +x test_debug_mode_issue.sh`, and run it.

```
#!/bin/bash
# =====
# Test Script: Debug Mode API Validation Issue
# =====
#
# Purpose: Demonstrate that the Kore.ai API only accepts "thoughts" as a valid
# debugMode value, rejecting "all" and "function-call"
#
# Usage:
# 1. Make this script executable: chmod +x test_debug_mode_issue.sh
# 2. Set your credentials below
# 3. Run: ./test_debug_mode_issue.sh
#
# =====
#
# -----
# Configuration - EDIT THESE VALUES
# -----
```

```

API_KEY="YOUR_API_KEY_HERE"
APP_ID="YOUR_APP_ID_HERE"
ENV_NAME="YOUR_ENV_NAME_HERE" # e.g., "stage" or "draft"

# -----
# Derived Variables
# -----
BASE_URL="https://agent-platform.kore.ai/api/v2"
ENDPOINT="${BASE_URL}/apps/${APP_ID}/environments/${ENV_NAME}/runs/execute"

# Colors for output
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
NC='\033[0m' # No Color

# -----
# Helper Functions
# -----
print_header() {
    echo ""
    echo "===== "
    echo "$1"
    echo "===== "
}

print_test() {
    echo ""
    echo "${YELLOW}Test: $1${NC}"
    echo "----- "
}

# -----
# Validation
# -----
if [ "$API_KEY" = "YOUR_API_KEY_HERE" ] || [ "$APP_ID" = "YOUR_APP_ID_HERE" ] || [ "$ENV_NAME" = "YOUR_ENV_NAME_HERE" ]
then
    echo "${RED}Error: Please set your API_KEY, APP_ID, and ENV_NAME in this script${NC}"
    exit 1
fi

print_header "Testing debugMode API Validation"
echo "Endpoint: $ENDPOINT"
echo "Testing different debugMode values..."

# -----
# Test 1: debugMode = "thoughts" (SHOULD SUCCEED)
# -----
print_test "1. debugMode = \"thoughts\" (Expected: SUCCESS ✓)"

RESPONSE=$(curl -s -w "\n%{http_code}" -X POST "$ENDPOINT" \
-H "Content-Type: application/json" \
-H "x-api-key: $API_KEY" \
-d '{

```

```

    "sessionIdentity": [
      {"type": "userReference", "value": "test-user-001"},
      {"type": "sessionReference", "value": "test-session-001"}
    ],
    "input": [
      {"type": "text", "content": "Test query for thoughts mode"}
    ],
    "debug": {
      "enable": true,
      "debugMode": "thoughts"
    }
  }')

HTTP_CODE=$(echo "$RESPONSE" | tail -n1)
BODY=$(echo "$RESPONSE" | sed '$_d')

if [ "$HTTP_CODE" = "200" ] || [ "$HTTP_CODE" = "201" ]; then
  echo "${GREEN}✓ SUCCESS - HTTP $HTTP_CODE${NC}"
  echo "Response (truncated):"
  echo "$BODY" | head -c 200
  echo "..."
else
  echo "${RED}✗ FAILED - HTTP $HTTP_CODE${NC}"
  echo "Response: $BODY"
fi

# -----
# Test 2: debugMode = "all" (EXPECTED TO FAIL)
# -----
print_test "2. debugMode = \"all\" (Expected: FAIL with 400 x)"

RESPONSE=$(curl -s -w "\n%{http_code}" -X POST "$ENDPOINT" \
  -H "Content-Type: application/json" \
  -H "x-api-key: $API_KEY" \
  -d '{
    "sessionIdentity": [
      {"type": "userReference", "value": "test-user-002"},
      {"type": "sessionReference", "value": "test-session-002"}
    ],
    "input": [
      {"type": "text", "content": "Test query for all mode"}
    ],
    "debug": {
      "enable": true,
      "debugMode": "all"
    }
  }')

HTTP_CODE=$(echo "$RESPONSE" | tail -n1)
BODY=$(echo "$RESPONSE" | sed '$_d')

if [ "$HTTP_CODE" = "400" ]; then
  echo "${YELLOW}✗ FAILED AS EXPECTED - HTTP $HTTP_CODE${NC}"

```

```

    echo "Error Response:"
    echo "$BODY" | python3 -m json.tool 2>/dev/null || echo "$BODY"
    echo ""
    echo "${RED}** ISSUE: API rejects debugMode=\"all\" **${NC}"
else
    echo "${GREEN}UNEXPECTED: Request succeeded with HTTP $HTTP_CODE${NC}"
    echo "Response: $BODY"
fi

# -----
# Test 3: debugMode = "function-call" (EXPECTED TO FAIL)
# -----
print_test "3. debugMode = \"function-call\" (Expected: FAIL with 400 x)"

RESPONSE=$(curl -s -w "\n%{http_code}" -X POST "$ENDPOINT" \
-H "Content-Type: application/json" \
-H "x-api-key: $API_KEY" \
-d '{
  "sessionId": [
    {"type": "userReference", "value": "test-user-003"},
    {"type": "sessionReference", "value": "test-session-003"}
  ],
  "input": [
    {"type": "text", "content": "Test query for function-call mode"}
  ],
  "debug": {
    "enable": true,
    "debugMode": "function-call"
  }
}')

HTTP_CODE=$(echo "$RESPONSE" | tail -n1)
BODY=$(echo "$RESPONSE" | sed '$_d')

if [ "$HTTP_CODE" = "400" ]; then
    echo "${YELLOW}x FAILED AS EXPECTED - HTTP $HTTP_CODE${NC}"
    echo "Error Response:"
    echo "$BODY" | python3 -m json.tool 2>/dev/null || echo "$BODY"
    echo ""
    echo "${RED}** ISSUE: API rejects debugMode=\"function-call\" **${NC}"
else
    echo "${GREEN}UNEXPECTED: Request succeeded with HTTP $HTTP_CODE${NC}"
    echo "Response: $BODY"
fi

# -----
# Test 4: debug without debugMode (SHOULD SUCCEED)
# -----
print_test "4. debug.enable=true without debugMode (Expected: SUCCESS ✓)"

RESPONSE=$(curl -s -w "\n%{http_code}" -X POST "$ENDPOINT" \
-H "Content-Type: application/json" \
-H "x-api-key: $API_KEY" \

```

```

-d '{
  "sessionIdentity": [
    {"type": "userReference", "value": "test-user-004"},
    {"type": "sessionReference", "value": "test-session-004"}
  ],
  "input": [
    {"type": "text", "content": "Test query without debugMode"}
  ],
  "debug": {
    "enable": true
  }
}'

HTTP_CODE=$(echo "$RESPONSE" | tail -n1)
BODY=$(echo "$RESPONSE" | sed 'd')

if [ "$HTTP_CODE" = "200" ] || [ "$HTTP_CODE" = "201" ]; then
  echo "${GREEN}✓ SUCCESS - HTTP $HTTP_CODE${NC}"
  echo "Response (truncated):"
  echo "$BODY" | head -c 200
  echo "..."
else
  echo "${RED}✗ FAILED - HTTP $HTTP_CODE${NC}"
  echo "Response: $BODY"
fi

# -----
# Summary
# -----
print_header "SUMMARY"
echo ""
echo "${GREEN}Working:${NC}"
echo "  - debugMode = \"thoughts\""
echo "  - debug.enable = true (without debugMode field)"
echo ""
echo "${RED}Not Working (returns 400 error):${NC}"
echo "  - debugMode = \"all\""
echo "  - debugMode = \"function-call\""
echo ""
echo "${YELLOW}Recommendation for Platform Team:${NC}"
echo "  1. Add support for \"all\" and \"function-call\" modes, OR"
echo "  2. Update API documentation to indicate only \"thoughts\" is valid, OR"
echo "  3. Provide more descriptive error messages explaining why modes are rejected"
echo ""
print_header "End of Tests"

```

## Environment Details

- **API Base URL:** <https://agent-platform.kore.ai/api/v2>
- **Endpoint:** /apps/{appId}/environments/{envName}/runs/execute
- **Method:** POST
- **Authentication:** x-api-key header
- **Content-Type:** application/json

---

**Date:** February 12, 2026

**Reported by:** CLI Tool User

**Severity:** Medium (Limitation, not blocking)

**Category:** API Validation / Feature Request