

Strands Troubleshooting Guide

Common Issues and Solutions

Issue 1: Agent Not Responding

Symptoms: Agent returns None or empty responses

Solutions:

1. Check model configuration:

- Verify API keys are set correctly
- Ensure model ID is valid
- Check network connectivity

2. Validate input:

- Ensure message is not empty
- Check for special characters
- Verify encoding is UTF-8

3. Review system prompt:

- Avoid contradictory instructions
- Keep prompts clear and concise
- Test with minimal prompt first

Issue 2: Tool Execution Failures

Symptoms: Agent tries to use tools but they fail

Common Causes:

1. Missing type hints:

- All parameters must have type annotations
- Return type must be specified
- Use proper types (int, str, List, etc.)

2. Invalid docstring:

- Tool must have a docstring
- Docstring should describe the function clearly
- Include parameter descriptions

3. Error handling:

- Tools should handle exceptions
- Return meaningful error messages
- Don't let tools crash the agent

Example of correct tool:

@tool

```
def calculate(x: int, y: int) -> int:
```

```
    """Add two numbers together."""
```

```
    return x + y
```

Issue 3: AWS Bedrock Access Denied

Symptoms: 'Access Denied' or '403 Forbidden' errors

Checklist:

1. Model Access:

- Log into AWS Console
- Navigate to Bedrock > Model access
- Ensure Claude 3.7 Sonnet is enabled
- Wait for approval (can take minutes)

2. IAM Permissions:

- User/role needs 'bedrock:InvokeModel'
- Check policy attachments
- Verify resource permissions

3. Region Settings:

- Bedrock not available in all regions
- Check your AWS_DEFAULT_REGION
- Try us-west-2 or us-east-1

4. Credentials:

- Run 'aws configure list' to verify
- Check environment variables
- Ensure credentials are not expired

Performance Optimization

Tips for Faster Agents:

1. Model Selection:

- Use smaller models for simple tasks
- Claude 3.7 Haiku for speed
- Local models for zero latency

2. Streaming Responses:

- Enable streaming for long outputs
- Improves perceived performance
- Better user experience

3. Caching:

- Cache common responses
- Store tool results when appropriate
- Use memory systems effectively

4. Batch Processing:

- Group similar requests
- Process documents in batches
- Parallelize when possible