bytedance / **deer-flow** Public

Code | Issues 451 | Pull requests 223 | Discussions | Actions | Projects

Commit 2176b2b

Hinotoi-agent authored 2 days ago · 6 / 6 · Verified

```
fix: validate bootstrap agent names before filesystem writes (#2274)

* fix: validate bootstrap agent names before filesystem writes

* fix: tighten bootstrap agent-name validation
```

main (#2274)

1 parent 8e35913 commit 2176b2b

5 files changed +78 -5 lines changed

Top

Filter files…

- backend
  - packages/harness/deerflow
    - agents/lead_agent
      - agent.py
    - config
      - agents_config.py
    - tools/builtins
      - setup_agent_tool.py
  - tests
    - test_lead_agent_model_resolution.py
    - test_setup_agent_tool.py

5 files changed +78 -5 lines changed

Search within code

...harness/deerflow/agents/lead_agent/agent.py

```
@@ -17,7 +17,7 @@
from deerflow.agents.middlewares.tool_error_handling_middleware import build_lead_runtime_middlewares
from deerflow.agents.middlewares.view_image_middleware import ViewImageMiddleware
from deerflow.agents.thread_state import ThreadState
- from deerflow.config.agents_config import load_agent_config
+ from deerflow.config.agents_config import load_agent_config, validate_agent_name
from deerflow.config.app_config import get_app_config
from deerflow.config.memory_config import get_memory_config
from deerflow.config.summarization_config import get_summarization_config
@@ -291,7 +291,7 @@ def make_lead_agent(config: RunnableConfig):
    subagent_enabled = cfg.get("subagent_enabled", False)
    max_concurrent_subagents = cfg.get("max_concurrent_subagents", 3)
    is_bootstrap = cfg.get("is_bootstrap", False)
-    agent_name = cfg.get("agent_name")
+    agent_name = validate_agent_name(cfg.get("agent_name"))

    agent_config = load_agent_config(agent_name) if not is_bootstrap else None
    # Custom agent model from agent config (if any), or None to let _resolve_model_name pick the default
```

...es/harness/deerflow/config/agents_config.py

```
@@ -15,6 +15,17 @@
AGENT_NAME_PATTERN = re.compile(r"^[A-Za-z0-9-]+$")

+ def validate_agent_name(name: str | None) -> str | None:
+     """Validate a custom agent name before using it in filesystem paths."""
+     if name is None:
+         return None
+     if not isinstance(name, str):
+         raise ValueError("Invalid agent name. Expected a string or None.")
+     if not AGENT_NAME_PATTERN.fullmatch(name):
```

```
      25 +         raise ValueError(f"Invalid agent name '{name}'. Must match pattern:
         {AGENT_NAME_PATTERN.pattern}")
      26 +     return name
      27 +
      28 +
18    29   class AgentConfig(BaseModel):
19    30       """Configuration for a custom agent."""
20    31
           @@ -46,8 +57,7 @@ def load_agent_config(name: str | None) -> AgentConfig |
           None:
46    57       if name is None:
47    58           return None
48    59
49       -     if not AGENT_NAME_PATTERN.match(name):
50       -         raise ValueError(f"Invalid agent name '{name}'. Must match pattern:
         {AGENT_NAME_PATTERN.pattern}")
      60 +     name = validate_agent_name(name)
51    61       agent_dir = get_paths().agent_dir(name)
52    62       config_file = agent_dir / "config.yaml"
53    63
```

...deerflow/tools/builtins/setup_agent_tool.py

```
           @@ -6,6 +6,7 @@
6     6    from langgraph.prebuilt import ToolRuntime
7     7    from langgraph.types import Command
8     8
      9 + from deerflow.config.agents_config import validate_agent_name
9     10   from deerflow.config.paths import get_paths
10    11
11    12   logger = logging.getLogger(__name__)
           @@ -25,8 +26,10 @@ def setup_agent(
25    26       """
26    27
27    28       agent_name: str | None = runtime.context.get("agent_name") if
         runtime.context else None
      29 +     agent_dir = None
28    30
29    31       try:
      32 +         agent_name = validate_agent_name(agent_name)
```

```
30  33          paths = get_paths()
31  34          agent_dir = paths.agent_dir(agent_name) if agent_name else paths.base_dir
32  35          agent_dir.mkdir(parents=True, exist_ok=True)
    ↕   @@ -55,7 +58,7 @@ def setup_agent(
55  58      except Exception as e:
56  59          import shutil
57  60
58      -       if agent_name and agent_dir.exists():
    61  +       if agent_name and agent_dir is not None and agent_dir.exists():
59  62              # Cleanup the custom agent directory only if it was created but an error occurred during setup
60  63              shutil.rmtree(agent_dir)
61  64          logger.error(f"[agent_creator] Failed to create agent '{agent_name}': {e}", exc_info=True)
    ...
```

...d/tests/test_lead_agent_model_resolution.py

```
    ↕   @@ -113,6 +113,26 @@ def _fake_create_chat_model(*, name, thinking_enabled, reasoning_effort=None):
113 113     assert result["model"] is not None
114 114
115 115
    116 + def test_make_lead_agent_rejects_invalid_bootstrap_agent_name(monkeypatch):
    117 +     app_config = _make_app_config([_make_model("safe-model", supports_thinking=False)])
    118 +
    119 +     monkeypatch.setattr(lead_agent_module, "get_app_config", lambda: app_config)
    120 +
    121 +     with pytest.raises(ValueError, match="Invalid agent name"):
    122 +         lead_agent_module.make_lead_agent(
    123 +             {
    124 +                 "configurable": {
    125 +                     "model_name": "safe-model",
    126 +                     "thinking_enabled": False,
    127 +                     "is_plan_mode": False,
    128 +                     "subagent_enabled": False,
    129 +                     "is_bootstrap": True,
```

```
+                    "agent_name": "../../../tmp/evil",
+                }
+            }
+        )
+
+
    def test_build_middlewares_uses_resolved_model_name_for_vision(monkeypatch):
        app_config = _make_app_config(
            [
```

backend/tests/test_setup_agent_tool.py

```
@@ -0,0 +1,40 @@
+ from __future__ import annotations
+
+ from pathlib import Path
+ from types import SimpleNamespace
+
+ from deerflow.tools.builtins.setup_agent_tool import setup_agent
+
+
+ class _DummyRuntime(SimpleNamespace):
+     context: dict
+     tool_call_id: str
+
+
+ def test_setup_agent_rejects_invalid_agent_name_before_writing(tmp_path, monkeypatch):
+     monkeypatch.setenv("DEER_FLOW_HOME", str(tmp_path))
+     outside_dir = tmp_path.parent / "outside-target"
+     traversal_agent = f"../../../{outside_dir.name}/evil"
+     runtime = _DummyRuntime(context={"agent_name": traversal_agent}, tool_call_id="tool-1")
+
+     result = setup_agent.func(soul="test soul", description="desc", runtime=runtime)
+
+     messages = result.update["messages"]
+     assert len(messages) == 1
+     assert "Invalid agent name" in messages[0].content
```

```
25  +     assert not (tmp_path / "agents").exists()
26  +     assert not (outside_dir / "evil" / "SOUL.md").exists()
27  +
28  +
29  + def test_setup_agent_rejects_absolute_agent_name_before_writing(tmp_path,
      monkeypatch):
30  +     monkeypatch.setenv("DEER_FLOW_HOME", str(tmp_path))
31  +     absolute_agent = str(tmp_path / "outside-agent")
32  +     runtime = _DummyRuntime(context={"agent_name": absolute_agent},
      tool_call_id="tool-2")
33  +
34  +     result = setup_agent.func(soul="test soul", description="desc",
      runtime=runtime)
35  +
36  +     messages = result.update["messages"]
37  +     assert len(messages) == 1
38  +     assert "Invalid agent name" in messages[0].content
39  +     assert not (tmp_path / "agents").exists()
40  +     assert not (Path(absolute_agent) / "SOUL.md").exists()
```

Comments 0

Please sign in to comment.