

Code Review Complete! 🎉

This feedback should help you improve your skills.

🚫 The feedback uses Qwen/Qwen2.5-Coder-32B-Instruct to compare your response to a gold standard solution. As we know, LLMs are not perfect. You should compare your work against the assessment criteria if you doubt the feedback.

Here's your detailed feedback:### Question 1: Create a Basic Code Agent with Web Search Capability
Your Solution:

```
# Create a CodeAgent with DuckDuckGo search capability
from smolagents import (
    CodeAgent,
    DuckDuckGoSearchTool,
    HfApiModel,
    FinalAnswerTool,
    Tool,
    tool,
)

agent = CodeAgent(
    tools=[DuckDuckGoSearchTool()], # Add search tool here
    model=HfApiModel(), # Add model here
)
```

Reference Solution:

```
from smolagents import CodeAgent, DuckDuckGoSearchTool, HfApiModel

agent = CodeAgent(
    tools=[DuckDuckGoSearchTool()],
    model=HfApiModel("Qwen/Qwen2.5-Coder-32B-Instruct")
)
```

Assessment Criteria:

- Correct imports are included
- DuckDuckGoSearchTool is added to tools list
- HfApiModel is properly configured
- Model ID is correctly specified

Feedback:

Overall Assessment

The student's solution is mostly correct but has a few issues that need to be addressed to fully meet the assessment criteria.

Correct imports are included

✅ The student has imported all necessary modules, including the additional ones that are not required for the basic solution. This is acceptable as long as they are not used elsewhere in the code.

DuckDuckGoSearchTool is added to tools list

✅ The student has correctly added the DuckDuckGoSearchTool to the tools list.

HfApiModel is properly configured

- ✗ The student has not specified the model ID for the HfApiModel, which is required for the model to function correctly.
Model ID is correctly specified
- ✗ The student has not provided the model ID 'Qwen/Qwen2.5-Coder-32B-Instruct' as specified in the reference solution.

Question 2: Set Up a Multi-Agent System with Manager and Web Search Agents

Your Solution:

```
# Create web agent and manager agent structure
model = HfApiModel(
    "Qwen/Qwen2.5-Coder-32B-Instruct", provider="together", max_tokens=8096
)
web_agent = ToolCallingAgent(
    tools=[DuckDuckGoSearchTool()], # Add required tools
    model=model, # Add model
    max_steps=10, # Adjust steps
    name="web_agent", # Add name
    description="Browses the web to find information", # Add description
    verbosity_level=0,
)

manager_agent = CodeAgent(
    model=HfApiModel("deepseek-ai/DeepSeek-R1", provider="together", max_tokens=8096),
    # tools=[calculate_cargo_travel_time],
    managed_agents=[web_agent],
    additional_authorized_imports=[
        "geopandas",
        "plotly",
        "shapely",
        "json",
        "pandas",
        "numpy",
    ],
    planning_interval=5,
    verbosity_level=2,
    # final_answer_checks=[check_reasoning_and_plot],
    max_steps=15,
)
```

Reference Solution:

```
web_agent = ToolCallingAgent(
    tools=[DuckDuckGoSearchTool(), visit_webpage],
```

```

model=model,
max_steps=10,
name="search",
description="Runs web searches for you."
)

manager_agent = CodeAgent(
    tools=[],
    model=model,
    managed_agents=[web_agent],
    additional_authorized_imports=["time", "numpy", "pandas"]
)

```

Assessment Criteria:

- Web agent has correct tools configured
- Manager agent properly references web agent
- Appropriate max_steps value is set
- Required imports are authorized

Feedback:

Overall Assessment

The student's solution meets the assessment criteria with some minor differences in approach and tool configuration. The web agent has the correct tools configured, the manager agent properly references the web agent, and the required imports are authorized. However, there are some discrepancies in the model used for the web agent and the manager agent, and the max_steps value is set differently for each agent. Additionally, the verbosity_level and planning_interval parameters are set in the student's solution, which are not present in the reference solution. These differences do not affect the overall functionality of the solution, but they should be noted for consistency and clarity.

Web agent has correct tools configured

✅ The web agent is configured with the DuckDuckGoSearchTool, which is the correct tool for web searches.

Manager agent properly references web agent

✅ The manager agent is configured with the web agent as a managed agent, which is the correct reference.

Appropriate max_steps value is set

❌ The web agent has a max_steps value of 10, while the manager agent has a max_steps value of 15. These values are different from the reference solution, which sets the max_steps value to 10 for the web agent. The difference in max_steps values between the web agent and the manager agent should be consistent with the intended functionality of the system.

Required imports are authorized

✅ The manager agent is configured with the required imports, including 'geopandas', 'plotly', 'shapely', 'json', 'pandas', and 'numpy'.

Question 3: Configure Agent Security Settings

Your Solution:

```

pip install 'smolagents[e2b]'
# Set up secure code execution environment

```



```
from smolagents import HfApiModel, CodeAgent

agent = CodeAgent(
    tools=[],
    model=model,
    executor_type="e2b"# Add security configuration
)
```

Reference Solution:

```
from smolagents import CodeAgent, E2BSandbox

agent = CodeAgent(
    tools=[],
    model=model,
    sandbox=E2BSandbox(),
    additional_authorized_imports=["numpy"]
)
```

Assessment Criteria:

- E2B sandbox is properly configured
- Authorized imports are appropriately limited
- Security settings are correctly implemented
- Basic agent configuration is maintained

Feedback:

Overall Assessment

The student's solution has some issues that need to be addressed to meet the assessment criteria.

E2B sandbox is properly configured

✗ The student's solution does not use the E2BSandbox class as specified in the reference solution. Instead, it uses an executor_type parameter which is not mentioned in the reference solution. This indicates that the E2B sandbox is not properly configured.

Authorized imports are appropriately limited

✗ The student's solution does not specify any authorized imports, which is a requirement according to the reference solution. The reference solution specifies 'numpy' as an additional authorized import, but this is missing in the student's solution.

Security settings are correctly implemented

✗ The student's solution does not include any security settings. The reference solution does not explicitly mention security settings, but the use of the E2BSandbox class and the specification of authorized imports can be considered as part of the security settings. The student's solution does not meet these requirements.

Basic agent configuration is maintained

✗ The student's solution does not maintain the basic agent configuration as specified in the reference solution. The reference solution specifies the use of the CodeAgent class with a model parameter, but the student's solution does not provide a value for the model parameter. Additionally, the student's solution uses an executor_type parameter which is not mentioned in the reference solution.

Question 4: Implement a Tool-Calling Agent

Your Solution:

```
# Create a tool-calling agent
from smolagents import ToolCallingAgent

agent = ToolCallingAgent(
    tools=[], # Add configuration here
    model=model,
    name="tool_calling",
    description="call a specific tool",
    verbosity_level=0,
    max_steps=10,
)
```

Reference Solution:

```
from smolagents import ToolCallingAgent

agent = ToolCallingAgent(
    tools=[custom_tool],
    model=model,
    max_steps=5,
    name="tool_agent",
    description="Executes specific tools based on input"
)
```

Assessment Criteria:

- Tools are properly configured
- Step limit is set appropriately
- Agent name and description are provided
- Basic configuration is complete

Feedback:

Overall Assessment

The student's solution is mostly correct but has some areas for improvement.

Tools are properly configured

✗ The tools list is empty, which means the agent will not be able to call any tools. The reference solution includes a custom_tool, which should be added to the tools list.

Step limit is set appropriately

✓ The step limit is set to 10, which is a reasonable number of steps for the agent to take. This meets the criteria.

Agent name and description are provided

✓ The agent name is set to 'tool_calling' and the description is set to 'call a specific tool'. These values are provided and meet the criteria.

Basic configuration is complete

✗ While the agent name, description, and step limit are provided, the tools list is empty, which is a critical part of the configuration. Therefore, the basic configuration is not complete.

Question 5: Set Up Model Integration

Your Solution:

```
# Configure model integration
from smolagents import CodeAgent, HfApiModel, DuckDuckGoSearchTool

model = HfApiModel()

web_agent = CodeAgent(
    tools=[DuckDuckGoSearchTool()],
    model=model,
    name="web_search",
    description="Runs web searches for you. Give it your query as an argument.",
)

manager_agent = CodeAgent(tools=[], model=model, managed_agents=[web_agent])
```

Reference Solution:

```
from smolagents import HfApiModel, LiteLLMModel

# Hugging Face model
hf_model = HfApiModel("Qwen/Qwen2.5-Coder-32B-Instruct")

# Alternative model via LiteLLM
other_model = LiteLLMModel("anthropic/claude-3-sonnet")
```

Assessment Criteria:

- o Correct model imports are included
- o Model is properly initialized
- o Model ID is correctly specified
- o Alternative model option is provided

Feedback:

Overall Assessment