Agent Lifecycle Hooks (AgentHooks) — MCQs

**1. What is the purpose of AgentHooks in the Agents SDK?**

A. To manage agent memory  
B. To receive callbacks on lifecycle events for a specific agent  
C. To generate prompts dynamically  
D. To enforce guardrails

**Answer:** B — It allows you to hook into events in an agent's lifecycle for a specific agent

**2. Which base class does AgentHooks alias?**

A. RunHooksBase  
B. AgentHooksBase  
C. AgentLifecycleBase  
D. RunHooks

**Answer:** B — AgentHooks = AgentHooksBase[TContext, Agent]

**3. Which method in AgentHooks runs before an agent is invoked?**

A. on\_agent\_end  
B. on\_start  
C. on\_handoff  
D. on\_tool\_start

**Answer:** B — on\_start() is called before the agent is invoked

**4. What does the on\_end hook indicate?**

A. Agent initialization  
B. Agent produced a final output  
C. Tool has started  
D. Handoff started

**Answer:** B — on\_end() is called when the agent produces a final output

**5. Which hook is called when an agent is handed off to?**

A. on\_handoff  
B. on\_agent\_end  
C. on\_handoff\_start  
D. on\_tool\_start

**Answer:** A — on\_handoff() is called when the agent is being handed off to

**6. In AgentHooks, what are the parameters for on\_handoff?**

A. (context, agent, tool)  
B. (context, agent, source)  
C. (context, from\_agent, to\_agent)  
D. (context, agent, response)

**Answer:** B — It receives context, agent, and source, where source is the agent handing off to this agent

**7. Which hook runs when a tool is invoked by an agent?**

A. on\_start  
B. on\_tool\_start  
C. on\_handoff  
D. on\_end

**Answer:** B — on\_tool\_start() is triggered concurrently with tool invocation

**8. What parameters does on\_tool\_end receive?**

A. context, agent, tool, result  
B. context, agent, tool  
C. context, agent, input\_items  
D. context, agent, response

**Answer:** A — It gets context, agent, tool, and the result (usually a string)

**9. Which two hooks are called for LLM invocations?**

A. on\_llm\_invoke\_start, on\_llm\_invoke\_end  
B. on\_llm\_before, on\_llm\_after  
C. on\_llm\_start, on\_llm\_end  
D. on\_generation\_start, on\_generation\_end

**Answer:** C — Hooks used are on\_llm\_start() and on\_llm\_end()

**10. What does on\_llm\_start receive as parameters?**

A. context, agent, tool, result  
B. context, agent, *no arguments*  
C. context, agent, system\_prompt, input\_items  
D. context, agent, response

**Answer:** C — It gets context, agent, system\_prompt, and input\_items

**11. What triggers the on\_llm\_end method?**

A. After tool execution  
B. After agent output  
C. Immediately after the LLM call returns for this agent  
D. At agent start

**Answer:** C — It's called right after the LLM returns a response

**12. Which callback is part of RunHooksBase but not AgentHooksBase?**

A. on\_start  
B. on\_agent\_start  
C. on\_tool\_end  
D. on\_handoff

**Answer:** B — on\_agent\_start is part of RunHooksBase, not AgentHooksBase

**13. In RunHooksBase, when is on\_agent\_end called?**

A. At tool invocation start  
B. When the agent produces a final output  
C. When handoff occurs  
D. At LLM invocation

**Answer:** B — Called when the agent produces final output

**14. Which hook is not part of AgentHooksBase?**

A. on\_start  
B. on\_handoff  
C. on\_agent\_end  
D. on\_llm\_end

**Answer:** C — on\_agent\_end is part of RunHooksBase, not AgentHooksBase

**15. True or False: on\_tool\_start in AgentHooks triggers before tool invocation.**

A. True  
B. False

**Answer:** A — It’s called concurrently as the tool is being invoked

**16. What is the purpose of context (RunContextWrapper) passed to all hooks?**

A. Tracking run metadata and shared state  
B. Pre-generating prompts  
C. Generating final output  
D. Logging errors only

**Answer:** A — It wraps run-level context and dependencies passed to hooks

**17. Which hook would be appropriate for logging every time an agent starts running?**

A. on\_tool\_start  
B. on\_start  
C. on\_agent\_end  
D. on\_llm\_end

**Answer:** B — on\_start runs each time the agent starts and is ideal for logging

**18. What signature does on\_agent\_end in RunHooksBase follow?**

A. on\_agent\_end(context, agent, output)  
B. on\_agent\_end(context, agent, result)  
C. on\_agent\_end(context)  
D. on\_agent\_end(agent, output)

**Answer:** A — Receives context, agent, and output

**19. If you want to preprocess data before each agent runs, which hook should you override?**

A. on\_tool\_end  
B. on\_llm\_end  
C. on\_start (AgentHooksBase)  
D. on\_tool\_start

**Answer:** C — on\_start allows you to run preprocessing before agent invocation

**20. If you need both general and per-agent lifecycle behavior, you should:**

A. Use AgentHooksBase only  
B. Use RunHooksBase only  
C. Combine both RunHooks and AgentHooks  
D. Use neither

**Answer:** C — Combine both for global and per-agent callback behaviors