

```

001 from a2a.server.agent_execution import AgentExecutor, RequestContext
002 from a2a.server.events import EventQueue
003 from a2a.server.tasks import TaskUpdater
004 from a2a.types import (InternalError, InvalidParamsError, Part, TaskState,
005                          TextPart, UnsupportedOperationError)
006 from a2a.utils import new_agent_text_message, new_task
007 from a2a.utils.errors import ServerError
008 from .agent import CurrencyAgent
009
010
011 class CurrencyAgentExecutor (AgentExecutor):
012
013     def __init__(self):
014         self.agent = CurrencyAgent()
015
016     async def execute(
017         self,
018         context: RequestContext,
019         event_queue: EventQueue,
020     ) -> None:
021
022         query = context.get_user_input()
023         task = context.current_task
024         if not task:
025             task = new_task(context.message) # type: ignore
026             await event_queue.enqueue_event(task)
027         updater = TaskUpdater(event_queue, task.id, task.context_id)
028         async for item in self.agent.stream(query, task.context_id):
029             is_task_complete = item['is_task_complete']
030             require_user_input = item['require_user_input']
031
032             if not is_task_complete and not require_user_input:
033                 await updater.update_status(
034                     TaskState.working,
035                     new_agent_text_message(
036                         item['content'],
037                         task.context_id,
038                         task.id,
039                     ),
040                 )
041             elif require_user_input:
042                 await updater.update_status(
043                     TaskState.input_required,
044                     new_agent_text_message(
045                         item['content'],
046                         task.context_id,
047                         task.id,
048                     ),
049                     final=True,
050                 )
051                 break
052             else:
053                 await updater.add_artifact(
054                     [Part(root=TextPart(text=item['content']))],
055                     name='conversion_result',
056                 )
057                 await updater.complete()
058                 break
059
060
061     async def cancel(
062         self, context: RequestContext, event_queue: EventQueue
063     ) -> None:
064         raise ServerError(error=UnsupportedOperationError())

```