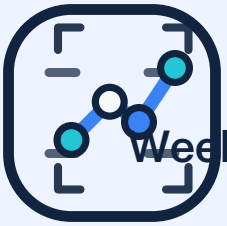


Week06 | 课时 2 | Dagster Assets 与资产图：从任务流到资产流

Table of contents

Asset graph 不是流程图，而是数据产品的运行边界图

这一讲把 Week06 的核心对象讲清楚：

asset key、依赖、Definitions、resources、external asset、observation 和 materialization 必须分清。

进入课时 3 返回 Week06 总览

下载讲义

提供适合离线阅读的 PDF 版和适合批注整理的 Word 版。

PDF 版 · 打印 / 离线阅读 Word 版 · 批注 / 二次整理

这节课解决什么问题

很多人第一次用 Dagster，会把每个 Python 函数都拆成 asset，或者把所有脚本直接串进一个 job。Week06 不能这么做。

这节课要解决：

- 哪些东西应该变成 asset
- 哪些只是 source / external observation
- asset key 如何避免和旧 assets 冲突
- 为什么 source observation job 和 materialization job 不能混在一起

参考学习时间

45–55 分钟

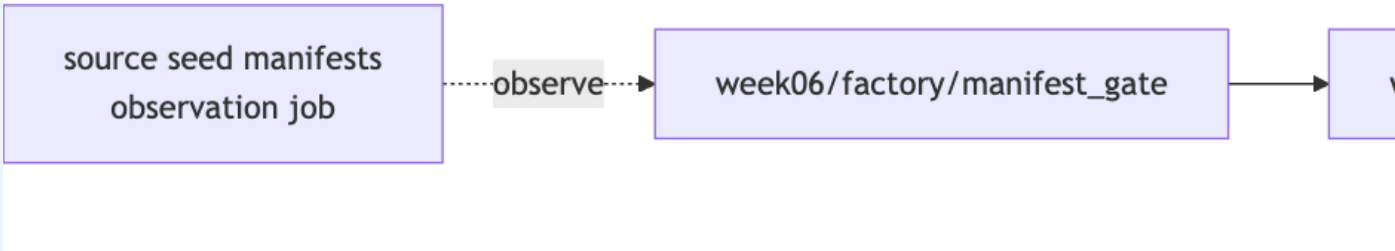
学完这一讲，你应该能做到什么

1. 读懂最小 Dagster asset graph。
2. 设计 week06/* 命名空间，避免 asset key 冲突。
3. 区分 observable source asset、external asset 和 materializable asset。
4. 解释 Definitions 为什么是 Dagster 入口。
5. 说明 resources / config 为什么不能硬编码本机路径。

本课产出

- docs/blueprints/week06/week06-asset-graph.md
- docs/blueprints/week06/week06-data-factory-blueprint.md

先看一张最小资产图



这张图有两个重点：

- source observation 可以单独观测输入，但不要和 materialization job 混选
- Week04 / Week05 依赖可以先作为 optional / external，不要伪造通过

1. Week06 asset key 怎么命名

建议采用明确前缀：

资产	推荐 key	说明
manifest gate	week06/factory/manifest_gate	读取并校验本次输入声明
ticket raw asset	week06/ingestion/raw_ticket_events_partitioned	薄包装 Week03 ticket ingest
ticket fact asset	week06/silver/ticket_fact_partitioned	观察或物化结构化事实层
optional lakehouse	week06/external/lakehouse_state	未完成时 skipped / not_available
optional semantic mart	week06/external/support_kpi_mart	未完成时 skipped / not_available
backfill plan	week06/ops/backfill_plan	为一个 daily partition 生成 dry-run 计划
run evidence	week06/ops/run_evidence_report	汇总运行证据
delivery summary	week06/ops/data_factory_delivery_summary	给实验 / 作业收口

不要直接复用旧的 raw_ticket_events、ticket_facts key，避免同名或语义接近的资产有两套依赖。

2. 什么该 materialize，什么该 observe

类型	适合对象	Week06 做法
materializable asset	本周负责生成或更新的资产	manifest_gate、partitioned ticket assets、run evidence
observable source asset	外部输入状态	seed manifest、source files，可单独 job
external asset	由其他系统负责，但本图需要依赖	lakehouse / semantic mart
placeholder asset	前置周未完成时保留接口	status 写 skipped，不阻塞主线

⚠ 不要混选 source observation 和 core materialization

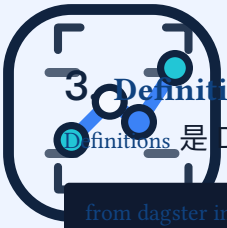
Week06 的 core job 只选择 regular materializable assets。source observation 应单独定义 job，或在 Student Core 中让 `manifest_gate` 直接读取 manifest 文件。

💡 少量关键资产 + 薄包装

Student Core 不是把每个 helper 函数都变成 asset。优先把 manifest gate、partitioned raw、partitioned fact、backfill plan、run evidence、delivery summary 做清楚，再扩展 observation 和 external assets。

Asset card 怎么写

字段	要回答的问题	Week06 示例
`asset_key`	这个数据产品稳定叫什么	`week06/silver/ticket_fact_partitioned`
`group`	它属于哪条运行边界	`week06_data_factory`
`owner`	谁负责解释和恢复	`course-team` 或你的名字
`partition_def`	失败边界是什么	daily partition，默认 `2026-04-17`
`upstream`	依赖哪些资产	manifest gate、raw ticket events
`downstream`	谁会消费它	run evidence、Week07/Week08
`materialization_type`	生成、观察还是外部依赖	materializable / external / optional
`checks`	怎么验收	row count、duplicate、null rate、partition completeness
`evidence`	证据落在哪里	`reports/week06/run_evidence/*.json`
`failure_mode`	失败后怎么恢复	retry / replay / backfill / hold



3. Definitions 是什么

Definitions 是 Dagster 的项目入口。Week06 不需要另起一套入口，而是在现有入口中注册：

```
from dagster import Definitions

defs = Definitions(
    assets=[...],
    asset_checks=[...],
    jobs=[week06_data_factory_job],
    resources={...},
)
```

课程重点不是背 API，而是理解：

- assets 定义“有哪些数据产品”
- jobs 定义“哪些资产可以被一起运行”
- resources 定义“运行时如何访问外部系统”
- checks 定义“资产是否可验收”

4. Resources / config 不能硬编码

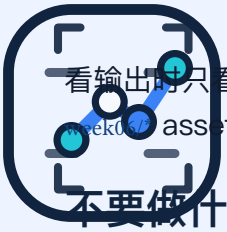
Week06 需要的配置应来自 env 或 run config：

配置	示例
Database	DATABASE_URL
Object storage	MINIO_ENDPOINT、bucket
Reports	WEEK06_REPORT_DIR
Partition start	WEEK06_PARTITION_START_DATE
Trace / release	TRACE_ID、RELEASE_ID
Optional dependency	LAKEHOUSE_ENABLED、ANALYTICS_ENABLED

硬编码本机绝对路径会让 devbox、Dagster 容器和同学机器全部不一致。

可执行的项目侧落点以 omnisupport-copilot 当前实现为准：

```
# 可执行：已与项目 runbook 对齐
docker compose --profile tools --env-file infra/env/.env.local -f infra/docker-compose.yml run --rm devbox \
  pytest tests/integration/test_week06_definitions_loadable.py -q
```



看输出时只看一个判断：week06_postgres、week06_minio、week06_reports 这些 resources 是否和 week06 assets 一起注册成功。

不要做什么

- 不要新建一个平行的 Dagster entrypoint，Week06 继续走 pipelines/definitions.py。
- 不要复制 Week03 ingest 的 SQL / 写入逻辑，asset wrapper 只做薄包装。
- 不要因为 Week04 / Week05 optional 依赖未完成就写成 passed。
- 不要把每个 Python helper 拆成 asset，先保证少量资产能被解释、验证和交接。

5. 本课最重要判断

> **Week06 的资产图要少而准：关键资产、清晰命名、明确依赖、可解释状态，比复杂图更重要。**

自检清单

- ☐ 我能解释 asset key 为什么要有 week06/* namespace
- ☐ 我能区分 source observation 和 materialization
- ☐ 我知道 Week04 / Week05 未完成时不能 fake passed
- ☐ 我能说出 Definitions 注册哪些对象
- ☐ 我知道哪些配置不能硬编码

课后最小行动

补一份 asset graph 草稿：

```
## Week06 asset graph draft

- Core materializable assets:
- Observable source assets:
- External / optional assets:
- Jobs:
- Required resources:
- Known skipped dependencies:
```