

Facilitator guide: Module 6, FinTech and data products

A teacher-facing companion to Module 6. The module is the full lesson; this guide adds the facilitation layer: a preparation checklist, timing cues, guidance for the discussion questions, and notes on what to emphasize and where students get stuck. Slides and a printable version are below.

At a glance

- **Length:** 90 minutes. A 60 minute and a 110 minute variant are noted under Teaching notes.
- **Level:** MBA or graduate students in FinTech product design, data product management, or technology entrepreneurship. Pairs well with Module 4 in a course covering both the investment and product-design perspectives on the same dataset.
- **Access:** the no-code sandbox at sandbox.altfndata.com for the core session, plus the documentation site at docs.altfndata.com. The session is built around the product itself, so it also covers a code extension: an instructor class API key, requested from info@altfndata.com in advance, lets you demo a live production API request; the extension is optional and no coding background is required for the core session.
- **Goal:** students leave able to name the three access surfaces of a data product, explain the tradeoff behind self-serve versus manually issued access, read an API request and response shape, and sketch an access-tier design of their own.

Before class

- Register your own sandbox account and run the demo query in the "Datasets and queries used" section end to end, so you have known-good results on screen.
- Request an instructor class API key from info@altfndata.com with enough lead time if you plan to run the live code extension, and test the request with the Python client beforehand.
- Open docs.altfndata.com once and note the quickstart, task-based tutorials, and the Evaluation Guide at [use-cases.html](https://docs.altfndata.com/use-cases.html), so you can navigate them smoothly on screen.
- Load the slides and confirm the projector or screen share shows the sandbox and, if using the extension, a terminal or notebook clearly.
- Set up a whiteboard or shared document ahead of time for the small-group access-tier design exercise.

Timed agenda with cues

Time	Segment	What to do	Watch for
0 to 15	Three access surfaces	Introduce the sandbox, documentation, and production API, and ask students why a vendor maintains all three rather than just one.	Students assuming one surface is strictly "better." Frame each as serving a different customer need, not a hierarchy.
15 to 30	Guided demo: sandbox	Register together if needed, then walk the SQL editor, coverage browser, data dictionary, and schema tabs, running the fine art query live.	Rushing past the schema tab. Pause there, since it sets up the later discussion of data dictionary versus raw table.

Time	Segment	What to do	Watch for
30 to 45	Guided demo: documentation and client library	Show the quickstart, tutorials, Evaluation Guide, and the downloadable Python client without running code yet.	Students unclear on audience. Ask directly who each page seems written for before moving on.
45 to 60	Guided demo: API request and response	Walk the request body and response envelope shape; optionally run a live request with the class API key.	Students conflating result_count with a total match count. Flag this explicitly the first time it appears on screen.
60 to 75	Small-group exercise	Groups sketch an access-tier structure for a hypothetical data product, justifying self-serve versus manual approval per tier.	Groups defaulting to "more tiers is better." Push them to justify each tier's access rule with a specific reason.
75 to 88	Group share-out	Each group presents its access-tier design in a minute or two.	Time overrun. Set a visible timer per group so every group gets to present.
88 to 90	Wrap-up and homework	Restate the one-sentence takeaway and hand out the homework assignment.	Leave the full 2 minutes for homework logistics, not zero.

Guidance for the discussion questions

Use these as the "what to listen for" behind each question in the module. They are talking points, not a graded key.

- 1. Why self-serve sandbox but manually issued production keys.** Listen for: the sandbox carries low risk and removes friction for exploration, while production access to real data at scale warrants a human check on who is requesting it and why.
- 2. What the result_count versus total-match separation reveals about pagination design.** Listen for: returning only the current page's count keeps responses light and fast, but it also means a consumer must paginate deliberately rather than assume one response tells them everything that matched.
- 3. The purpose of a data dictionary and coverage browser separate from the raw tables.** Listen for: these tools translate a raw schema into something a non-technical or first-time user can trust and use correctly, which is itself part of the product, not just the data.
- 4. What distinguishes an individual-analyst tier from an institutional-desk tier.** Listen for: differences in request volume, automation and rate limits, support expectations, and price, not simply "more data" for the higher tier.
- 5. What risk a vendor takes on by offering a no-code sandbox against real production data.** Listen for: exposing genuine records for free invites misuse or scraping at scale, which is part of why the production API is gated behind manual key issuance even though the sandbox is not.
- 6. How to decide what belongs in documentation versus in the tool itself.** Listen for: the tool itself should carry what a user needs in the moment to complete a task, while documentation carries context, rationale, and edge cases a user would only seek out deliberately.
- 7. What to measure to know an access-tier design is working.** Listen for: concrete signals such as self-serve conversion rate, time to first successful request, upgrade rate between tiers, and support burden per tier, rather than usage volume alone.

Teaching notes

- **Most common misconception:** students assume the production API is just as open as the sandbox once someone has an account. Contrast the two access flows side by side and name the manual key-issuance step as a deliberate gate, not an oversight.
- **Second misconception:** treating `result_count` as the total number of matching rows rather than the size of the current page. Run the same query twice with offset advanced to show pagination in practice.
- **60 minute variant:** compress the three guided demos into one pass covering sandbox and API together, skip the live code extension, and shorten the small-group exercise to a single shared whiteboard rather than separate groups.
- **110 minute variant:** after group share-out, have each group also sketch one example filter object per tier and discuss rate-limiting a self-serve tier as a class, using the Extension activities in the module.
- **If the sandbox or the live API demo is slow or blocked on the room network:** fall back to the pre-run screenshots and the static request and response examples in the module, and offer the live code extension as an optional follow-up for interested students.

For the full lesson content, queries, and homework, see Module 6. Questions or a class API key: info@altfndata.com.