

Infonetica – Software Engineer Intern Take-Home Exercise

Topic: *Configurable Workflow Engine (State-Machine API)*

Expected effort: ≤ 2 hours **Submission:** *public GitHub repo only*

1. Objective

Design and implement a **minimal backend service** that lets a client:

1. **Define** one or more workflows as configurable **state machines** (states + actions).
2. **Start** workflow *instances* from a chosen definition.
3. **Execute actions** to move an instance between states, with full validation.
4. **Inspect / list** states, actions, definitions, and running instances.

There is **no single “right” design**—we will evaluate your structure, naming, clarity, and trade-offs.

2. Core Concepts

Concept	Required attributes	Notes
State	id / name, isInitial (bool), isFinal (bool), enabled (bool)	You may add others (e.g., description).
Action (transition)	id / name, enabled (bool), fromStates (collection of state IDs), toState (single state ID)	One action may originate from multiple states but always ends in one.
Workflow definition	Collection of <i>States</i> and <i>Actions</i>	Must contain exactly one isInitial == true state.
Workflow instance	Reference to its definition, current state, basic history (action + timestamp)	Starts at the definition’s initial state.

3. Functional Requirements (what your API must expose)

Your service should provide *API operations* (you choose the exact HTTP routes, verbs, payload shapes, etc.) that enable:

Area	Required capabilities
Workflow configuration	<ul style="list-style-type: none">- Create a new workflow definition (provide states + actions in one go or incrementally).- Retrieve an existing definition.
Runtime	<ul style="list-style-type: none">- Start a new workflow instance for a chosen definition.

Area	Required capabilities
	<ul style="list-style-type: none"> - Execute an <i>action</i> on a given instance, moving it to the target state only if: <ul style="list-style-type: none"> o the action belongs to the instance's definition o the action is enabled o the current state is in fromStates. - Retrieve the current state (and basic history) of an instance.
Validation rules	Reject invalid definitions (e.g., duplicate IDs, missing initial state). Reject invalid action executions (e.g., disabled action, wrong source state, transition to unknown state, attempts to act on final states).
Persistence	Keep it simple: in-memory data or a lightweight local file (JSON/YAML/etc.). No database required.

4. Technical Guidelines

- **Language / stack:** .NET 8 / C#.
 - **Keep dependencies minimal.** If using ASP.NET Core, minimal-API style is fine.
 - **Time-box yourself to ≈2 hours.** It is perfectly acceptable to leave TODO comments for polish you would add with more time.
 - **Document assumptions** briefly in your README or code comments.
 - Unit tests are welcome but **not required**—add only if they help you clarify design quickly.
-

5. Evaluation Criteria

Aspect	What we care about
Design & readability	Clear class / module boundaries, sensible names, tidy project layout.
Correctness	State-machine rules enforced, invalid operations blocked gracefully, helpful error messages.
Maintainability	Code that could be extended (e.g., adding more flags, validation, persistence) without major rewrites.
Pragmatism	Appropriate level of abstraction for a small scope—avoid over-engineering.
Documentation	Concise README and targeted comments that make the project easy to run and review.

In later interview rounds we will discuss your decisions—be ready to explain the why behind your design.

6. Deliverables & Submission

1. **Public GitHub repository** containing:
 - Source code.
 - **README.md** with:
 - Quick-start instructions (e.g., dotnet run, sample build/run commands, environment notes).
 - Any assumptions, shortcuts, or known limitations.
 2. Share the **repository URL** as your submission. Zipped submissions will **not** be accepted.
-

7. Need clarification?

If anything is ambiguous, note your assumption in the README and proceed. We value clear reasoning over guessing what we want. Happy coding, and we look forward to reviewing your work!