

Logic

August 19, 2026

Logic

Kind: Module

Source: Pams/Logic (line 1)

Part of: Pams

.NET project in solution

Pams/Logic is a .NET project in the Pams solution. It contains application logic that other projects can reference, keeping host and transport code separate from business rules.

Diagram

```
graph TD
  Client[TypeScript client] --> Host[Application host]
  Host --> Logic[Pams/Logic]
  Logic --> Rules[Application logic]
```

Usage

```
type LogicRequest = {
  action: string;
  input: Record<string, unknown>;
};

async function runLogic(request: LogicRequest) {
  const response = await fetch("/api/logic", {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(request),
  });
}
```

```

    if (!response.ok) {
        throw new Error(`Logic request failed: ${response.status}`);
    }

    return response.json();
}

const result = await runLogic({
    action: "process",
    input: { id: "example-id" },
});

console.log(result);

```

AI Coding Instructions

- Keep business rules in `Pams/Logic` rather than placing them in API, UI, or infrastructure projects.
- Reference `Pams/Logic` from the application host that exposes HTTP, messaging, or UI entry points.
- Avoid adding transport-specific types, such as HTTP request objects, to logic-layer APIs.
- Match existing solution conventions for namespaces, project references, dependency injection, and error handling.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Declared in (1)

- `Pams` — `Pams/Pams.sln` :1