

[Proxy Design Pattern]

↳ Placeholder

The **Proxy Design Pattern** is a structural design pattern that provides a surrogate or placeholder for another object to control access to it. It enables additional functionalities (like access control, lazy initialization, logging, etc.) while keeping the actual object's interface intact.

Key Concepts:

- **Proxy Object:** Represents the real object and controls access to it.
- **Real Subject:** The actual object that the proxy represents.
- **Client:** Interacts with the proxy as if it were the real object.

Types of Proxies:

1. **Remote Proxy:** Manages interaction with a remote object.
2. **Virtual Proxy:** Delays the creation or initialization of a resource until it is needed.
3. **Protection Proxy:** Manages access control to the real object based on permissions.
4. **Smart Proxy:** Adds additional behaviors, like reference counting, logging, or caching.

