

## Factory Method pattern in ASP.NET

**Factory Method** is a creational pattern where a **base “Creator” class defines a method for creating objects**, but lets **subclasses decide which concrete “Product” to instantiate**.

**Key point:** client code depends on the *abstraction*, and the choice of concrete type is moved into the creator subclasses.

---

### Why we use it in real systems

Factory Method is useful when:

- You want to **hide construction complexity**
  - You want to **vary what gets created** without changing client code
  - You want to **support extension** (add new discount types with minimal existing changes)
  - You want to **encapsulate rules** for choosing/creating products
- 

## Real-world examples in ASP.NET apps

### 1) Payment provider selection

You might have:

- IPaymentGateway (Product)
- StripeGateway, AdyenGateway, PaypalGateway (Concrete products)
- PaymentGatewayFactory chooses which one based on country, currency, tenant config

This is extremely common in multi-tenant SaaS.

---

## **2) File storage provider**

- IStorageService
  - LocalStorageService, S3StorageService, AzureBlobStorageService
  - Factory creates the right one based on environment/config.
- 

## **3) Message sender**

- INotificationSender
- EmailSender, SmsSender, PushSender
- Factory picks based on message type + user preferences.