Design Patterns in PHP

Introduction

What are design patterns?

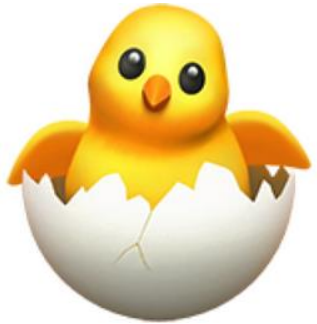
- Solutions to recurring problems
- Guidelines on how to tackle certain problems
- Not a silver bullet to all your problems
- Do not try to force them
- Are not solutions looking for problems

Types of design patterns

- Creational Design Patterns
- Structural Design Patterns
- Behavioral Design Patterns

Creational Design Patterns

Creational patterns are focused towards how to instantiate an object or group of related objects.



Creational Design Patterns

- ☝ Singleton
- 🏠 Simple Factory
- 🏭 Factory Method
- 🔨 Abstract Factory
- 👷 Builder
- 🐑 Prototype

Structural Design Patterns

Structural patterns are mostly concerned with object composition or in other words how the entities can use each other.

Structural Design Patterns

- 🔌 Adapter
- 🌉 Bridge
- 🌿 Composite
- ☕ Decorator
- 🗃️ Facade
- 💦 Flyweight
- 🔐 Proxy

Behavioral Design Patterns

Behavioral design patterns are design patterns that identify common communication patterns between objects and realize these patterns.

Behavioral Design Patterns

- ⛓️ Chain of Responsibility
- 👮 Command
- 💤 Iterator
- 📩 Mediator
- 🔙 Memento
- 🐵 Observer
- 🏃 Visitor
- 🤲 Strategy
- 💢 State
- 📙 Template Method

Singleton Design pattern

In plain words

Ensures that only one object of a particular class is ever created.

Wikipedia says

In software engineering, the singleton pattern is a software design pattern that restricts the instantiation of a class to one object. This is useful when exactly one object is needed to coordinate actions across the system.

Real world Example

There can only be one president of a country at a time. The same president has to be brought to action, whenever duty calls. President here is singleton.

Code Example

https://github.com/Bootsity/design-patterns-php/tree/master/Creational/singleton

Generalized UML