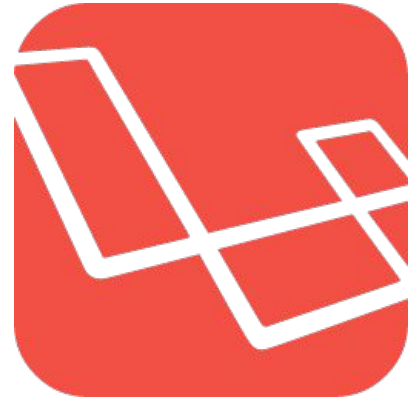


Laravel singleton

Algirdas Dumskis (algirdasdum@gmail.com)

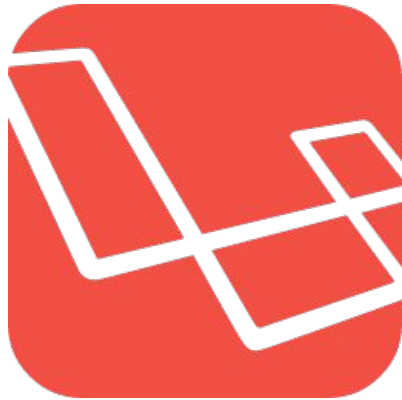
IoC container

Inversion of Control container

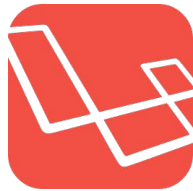


Singleton

One instance for all calls

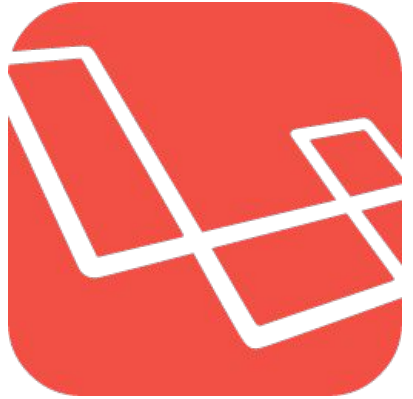


Singleton

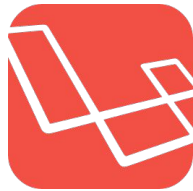


```
App::singleton('foo', function()  
{  
    return new FooBar;  
});
```

Singleton vs Bind

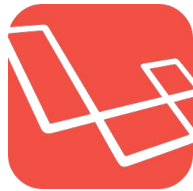


Singleton vs Bind



Bind - resolving a type from the container;

Singleton - binding a "shared" type into the container;

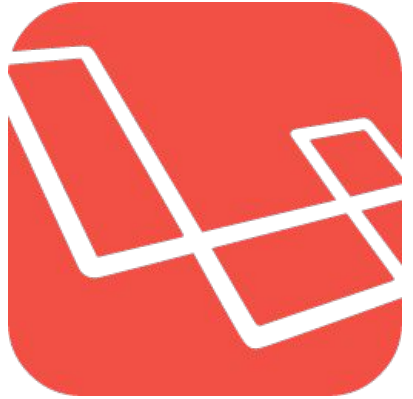


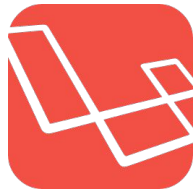
Singleton vs Bind

```
public function bind($abstract, $concrete = null, $shared = false)
{
    ...
}

public function singleton($abstract, $concrete = null)
{
    $this->bind($abstract, $concrete, true);
}
```

Singleton preparation





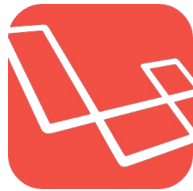
Facade

```
class AdvertFacade extends Facade
{
    /**
     * Name of the binding in the IoC container
     *
     * @return string
     */
    protected static function getFacadeAccessor()
    {
        return 'advert_manager';
    }
}
```

Object



```
class AdvertManager {  
  
    private static $instance;  
  
    /**  
     * @return AdvertManager  
     */  
    public static function getInstance()  
    {  
        return static::$instance ?: (static::$instance = new self());  
    }  
}
```

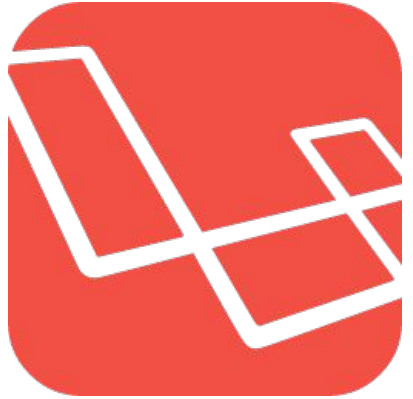


Service provider

```
class AdvertServiceProvider extends ServiceProvider
{
    public function register()
    {
        include __DIR__ . '/routes.php';
        $this->app->make('Adumskis\LaravelAdvert\AdvertManagerController');
        $this->app->singleton('advert_manager', function() {
            return new AdvertManager();
        });
    }

    public function provides()
    {
        return ['advert_manager'];
    }
}
```

Usage



Thank you!

