# 简介

服务定位器是一个了解如何提供各种应用所需的服务（或组件）的对象。

在服务定位器中， 每个组件都只有一个单独的实例，并通过ID 唯一地标识。 用这个 ID 就能从服务定位器中得到这个组件。

在 Yii 中，服务定位器是 [yii\di\ServiceLocator](https://www.yiichina.com/doc/api/2.0/yii-di-servicelocator) 或其子类的一个实例。

最常用的服务定位器是application（应用）对象，可以通过 \Yii::$app 访问。

它所提供的服务被称为application components（应用组件）， 比如：request、response、urlManager 组件。

可以通过服务定位器所提供的功能， 非常容易地配置这些组件，或甚至是用你自己的实现替换掉他们。

除了 application 对象，每个模块对象本身也是一个服务定位器。

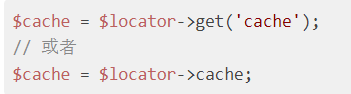
# 使用服务定位器

要使用服务定位器，第一步是要注册相关组件。

组件可以通过 [yii\di\ServiceLocator::set()](https://www.yiichina.com/doc/api/2.0/yii-di-servicelocator" \l "set()-detail) 方法进行注册。 以下的方法展示了注册组件的不同方法：



一旦组件被注册成功，你可以任选以下两种方式之一，通过它的 ID 访问它：



如上所示， [yii\di\ServiceLocator](https://www.yiichina.com/doc/api/2.0/yii-di-servicelocator) 允许通过组件 ID 像访问一个属性值那样访问一个组件。

 当你第一次访问某组件时，[yii\di\ServiceLocator](https://www.yiichina.com/doc/api/2.0/yii-di-servicelocator) 会通过该组件的注册信息创建一个该组件的实例，并返回它。 之后，如果再次访问，则服务定位器会返回同一个实例。

# 配置服务定位器

因为服务定位器，经常会在创建时附带[配置信息](https://www.yiichina.com/doc/guide/2.0/concept-configurations)， 因此我们提供了一个可写的属性，名为 [components](https://www.yiichina.com/doc/api/2.0/yii-di-servicelocator" \l "setComponents()-detail)， 这样就可以配置该属性，或一次性注册多个组件。

下面的代码展示了如何用一个配置数组，配置一个应用并注册 db，cache，tz 和 search 组件：



在上面的代码中，有一个替代方法来配置 search 组件。 而不是直接写一个 PHP 回调建立 SolrService实例，你可以使用一个静态类方法来返回这样的回调， 如下所示：



当你发布一个 Yii 组件封装一些非 Yii 第三方库时，这种替代方法是最好的。 当您使用如上所示的静态方法来表示构建复杂逻辑的第三方对象时， 您的组件用户只需要调用静态方法来配置组件。

# 遍历树（Tree traversal）

模块允许任意嵌套; Yii 应用程序本质上是一个模块树。

由于这些模块中的每一个都是服务定位器，所以子模块有权限访问其父模块。

这允许模块使用 $this->get('db') 而不是引用根服务定位器 Yii::$app->get('db')。 增加的好处是开发人员可以覆盖模块中的配置。