<https://blog.octo.com/en/hexagonal-architecture-three-principles-and-an-implementation-example/>

由Alistair Cockburn在2005年记录的六边形架构是一种软件架构，它有很多优点，自2015年以来又重新引起了人们的兴趣。

六角形建筑的初衷是:

允许应用程序同样由用户、程序、自动化测试或批处理脚本驱动，并在与最终运行时设备和数据库隔离的情况下进行开发和测试。

为了探索通过自动化测试或独立于数据库开发和测试应用程序的好处，我们建议您阅读我们最近发布的关于测试金字塔的系列博文:实践中的测试金字塔。

这个承诺非常有吸引力，而且它还有另一个好处:它允许隔离应用程序的核心业务，并独立于其他一切自动测试其行为。这可能是这个体系结构吸引领域驱动设计(DDD)实践者的原因。但是要注意，*DDD和六角形结构是两个截然不同的概念，它们可以相互加强，但不一定一起使用。但这是另一个话题了!*

最后，这个体系结构的设置并不复杂。它基于一些简单的规则和原则。让我们探讨一下这些原则，看看它们在实践中意味着什么。

1 六边形架构的原理

2 细节:代码内部和外部是如何组织的?

3 Detail: At the Runtime

4 Detail: 依赖倒置在右边

5 Detail: 为什么界面在左边?

6 测试六角形架构

7 为了进一步推进

8 References

一 、六边形架构的原理

六边形建筑基于三个原则和技术:

显式分离用户端、业务逻辑和服务器端

依赖关系从用户端和服务器端转移到业务逻辑

我们通过使用端口和适配器来隔离边界

词汇表注意:在本文的其余部分，将使用“用户端”、“业务逻辑”和“服务器端”这几个词。这些词来自原始文章，在下一节中定义。

1 原则:分开用户端、业务逻辑和服务器端

第一个原则是显式地将代码分为三个形式化的大区域。

二、 细节:代码内部和外部是如何组织的?

三、 Detail: At the Runtime

四、 Detail: 依赖倒置在右边

五、 Detail: 为什么界面在左边?

六、 测试六角形架构

七、 为了进一步推进

八、 References