

iOS (<http://lib.csdn.net/base/ios>)

iOS (<http://lib.csdn.net/base/ios>) - 设计模式 (<http://lib.csdn.net/ios/node/673>) - 设计模式 (<http://lib.csdn.net/ios/knowledge/1462>)

👁 186 💬 0

iOS中MVC等设计模式详解

作者：ganjicijiao (<http://my.csdn.net/ganjicijiao>)

在iOS编程,利用设计模式可以大大提高你的开发效率,虽然在编写代码之初你需要花费较大时间把各种业务逻辑封装起来。(事实证明这是值得的!)

模型-视图-控制器(MVC)设计模式是被大家广为熟悉和使用的模式,实际上在移动开发中尤其ios开发中,这种模式被发挥到淋漓尽致

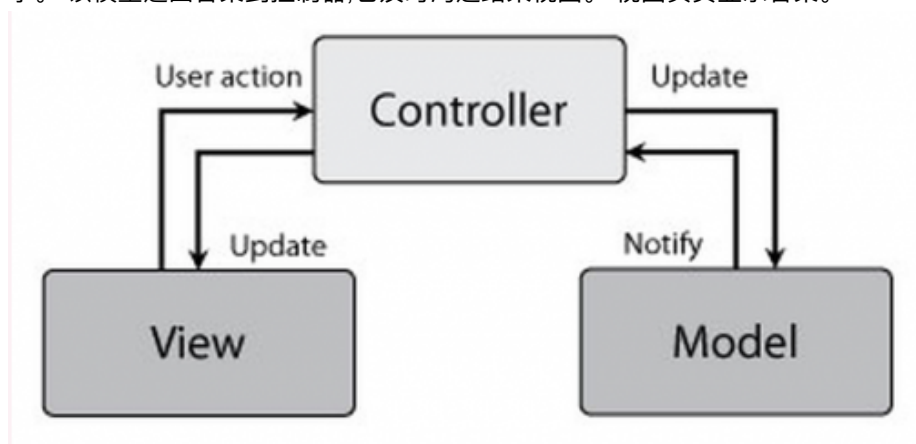
MVC设计模式包括三个部分:模型、视图和控制器。这个模型包含数据、信息,逻辑,或对象被认为是部分的业务层的iOS应用。视图包含所有的用户信息的组件,如文本区域,按钮,滑块,被认为是表示层的一个iOS应用。控制器是联络,或通信层,一个iOS应用。一个模型将永远不会直接沟通一个视图,反之亦然。控制器连接模型视图处理所有通信。

用户行为发生在一个视图,比如一个按钮点击。视图将消息发送给控制器。如果有必要,控制器与模型。控制器通过信息返回到视图。使用一个假想的计算器应用程序作为一个例子,MVC设计模式将应用如下:

模型——计算功能(加、减、乘、除)。

视图——用户界面组件(操作数和操作符按钮、计算文本)。

控制器——对象,倾听和沟通信息模型和视图。当平等的按钮被选中时,例如,控制器使请求模型执行数学。该模型返回答案到控制器,它及时沟通结果视图。视图负责显示答案。



二、简化你的设计模式

在Objective-C语言编程，添加到您的应用程序的特定行为的方式之一是通过继承。您可以创建一个现有类的子类，要么增强超类的属性和行为，或以某种方式修改它们。但也有其他的，更有活力的方式添加特定于应用程序的行为，不涉及子类。这些动态的技术和方法是基于设计模式。由于这篇文章解释说，在你的代码中采用的设计模式有助于类和框架类的可重用性和可扩展性。

设计模式：模板提供的设计，解决了一个编程问题

设计模式是面向对象的软件开发以及其他领域中使用抽象的工具。这是一个模板的设计，解决了一般情况下，在特定情况下经常发生的问题。因此，设计模式是一种特定的，具体的设计指南：“实例化”的模式，在一定意义上。有一定的灵活性，可以如何应用设计模式，事情往往可以决定如何应用该模式，如编程语言和现有架构。

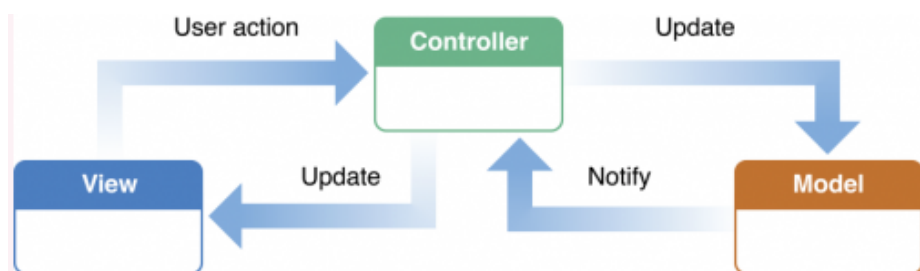
几个主题或设计影响设计模式的原则。这些设计原则，构建面向对象的系统，如拇指规则的“封装系统结构方面的变化”和“程序的接口，而不是执行。”他们表达了重要的见解。例如，封装原则说，如果你的系统有所不同，将它们封装隔离的部分，他们可以独立地改变系统的其他部分，特别是如果你为他们定义接口，不依赖于具体实现。您可以在以后改变或延长这些变量的部分，而不会影响系统的其他部分。从而消除依赖和减少零件之间的联轴器，因此系统变得更灵活，更容易改变。

这些好处，如使设计模式，一个重要的考虑因素，当你编写软件。如果你发现，相适应，在你的应用程序的设计和方式使用，程序的对象和类，它包括将更多的可重用性，可扩展性，更容易改变，未来的需求时，要求它。此外，基于设计模式的应用程序一般比应用程序是不是更优雅和高效，因为他们需要更少的代码来完成同样的目标。

你可以找到适应整个Cocoa Touch和Cocoa框架的设计模式，在Objective-C运行和语言。基于这些模式的一些机制，你会得到几乎是“免费”，但需要一些工作在你的一部分。你可以申请到自己的应用程序的代码，当形势需要时它的设计模式。如果您使用的可可触摸和Cocoa框架使用相同的模式，你的代码往往以适应更好的框架代码，并且工作更优雅。

最重要的设计模式：模型 – 视图 – 控制器

模型 – 视图 – 控制器设计模式，（俗称MVC）分配一个应用程序中的对象的三个角色之一：模型，视图，控制器。模式不仅定义应用程序中的对象扮演的角色，它定义了对象的方式与对方沟通。三种类型的对象中的每一个是从其他分离抽象的边界跨越这些边界的其他类型的对象进行通信。有时也被称为在一个应用程序中的目标的MVC类型的集合中的对象的层，例如，模型层。



MVC是一个好的设计任何iOS应用程序或Mac应用程序的核心。采用这种模式的好处是多方面的。很多在这些应用程序中的对象往往是可重复使用的，往往得到更好的界定及其接口。MVC设计的应用程序，也更容易比其他应用程序进行扩展。此外，您的应用程序可以使用的许多技术和架构是基于MVC和要求您的自定

义对象扮演一个MVC的角色。

您可能没有意识到这一点，但您已经创建了一个应用程序，是基于MVC：在你的iOS应用的HelloWorld。模型对象是用户名物业（NSString的对象）宣派及管理HelloWorldViewController类的。HelloWorldViewController和HelloWorldAppDelegate类的实例应用程序的控制器对象的应用程序的视图对象的文本字段，标签，按钮，背景图。

模型-视图-控制器的完整信息“模型-视图-控制器”的概念在Objective-C编程。

模型对象

模型对象封装了一个应用程序定义的数据操作和处理数据的逻辑和计算。例如，一个模型对象可能代表一个字符，在游戏或在地址簿中的联系人。有时，一个应用程序的模型层实际上是一个或多个图形相关的对象。大部分的数据，部分应用程序的持久状态（是否持久化状态存储在文件或数据库）应驻留在数据加载到应用程序的模型对象后。因为模型对象代表到一个特定的问题域相关的知识和专长，他们可以重复使用在类似的问题域。应该有一个“纯粹”的模型对象的视图对象，目前其数据，并允许用户编辑数据，它不应该被关注的用户界面和演示问题没有明确的连接。

在视图层创建或修改数据的用户操作在创建或更新模型对象通过一个控制器对象和结果传达。当一个模型对象的变化（例如，新的接收数据通过网络连接），它会通知控制器对象，更新相应的视图对象。

查看对象

一个视图对象是在一个应用程序，用户可以看到一个对象。一个视图对象知道如何绘制自己可能会响应用户的操作。视图对象的一个主要目的是显示数据从应用程序的模型对象，可以使编辑该数据。尽管如此，在一个MVC应用程序的视图对象模型对象通常脱钩。

因为你通常重用视图对象，对其进行重新配置，视图对象提供应用程序之间的一致性。对于iOS上，UIKit框架提供了视图类的集合;为OS X AppKit的框架提供了一个类似的集合。UIKit的视图对象的UIView类;最终继承了AppKit，它最终继承自NSView的类。

视图对象通过应用程序的控制器对象模型数据的变化，并了解用户发起的变化，例如，文本的文本字段中输入通过控制器对象应用程序的模型对象沟通。

控制器对象

控制器对象作为一个或多个应用程序的视图对象和一个或多个模型对象之间的中介。因此，控制器对象视图对象的管道了解变化的模型对象，反之亦然。控制器对象也可以执行一个应用程序的安装和协调的任务和其他对象的生命周期管理。

控制器对象解释用户视图对象中的行动和新的或更改的数据模型层。当模型对象的变化，控制器对象通信，新的模型数据的视图对象，使他们能够显示它。

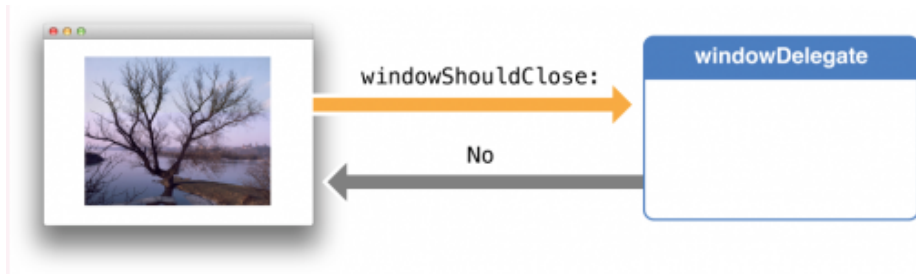
设计模式解决问题

面向对象的系统，作为一个应用程序是动态的。一个对象可以在运行时并不限定于在编译时设置的行为。一个对象可以发送邮件到其他对象，并可以有所不同运行情况下的目标相同的消息。一个对象也可以与其他对象的一个变量组在运行时，使用了多种技术，可以有效地完成工作的应用程序。对于一个对象，或对象的网络，做到这一点，它必须利用许多技术和框架体系结构的设计模式是适应的。

以下各节描述了许多这些技术和架构。考虑他们的Objective-C编程工具包的一部分。

代表团：代理代表另一个对象

代表团，一个对象调用代表的委托行为，并要求另一个对象。其他，委派，对象通常是一个框架对象。在一些点在执行时，它发送一个消息到其委托的消息告诉一些事件即将发生的委托，并要求一些回应。委托（通常是一个自定义类的一个实例）实现调用的方法的消息，并返回一个适当的值。通常情况下，该值是一个布尔值，告诉委托对象是否继续进行的动作。



因此，代表团是一个框架的运作注入特定于应用程序的行为的一种手段，无需子类类。它是一种常见的和强大的设计延伸和影响的行为框架。

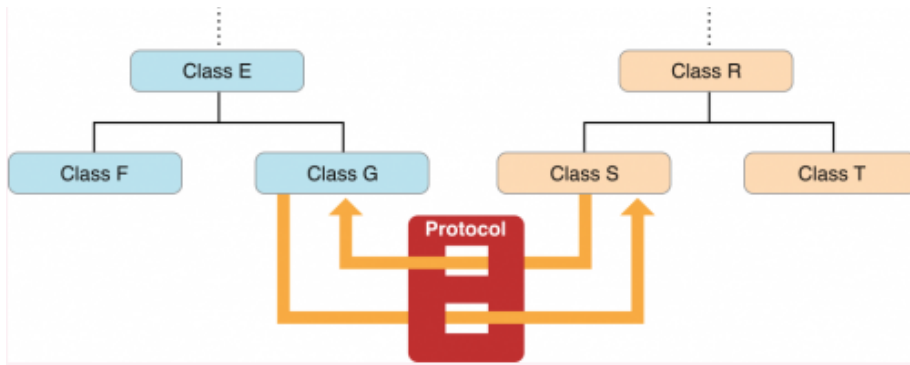
回想的HelloWorldAppDelegate对象的HelloWorld应用程序时创建工作通过你的第一个iOS应用。Xcode的自动分配，它是应用程序对象的委托，这是一个框架对象。应用程序委托可以处理应用：的didFinishLaunchingWithOptions：和其他代表团的发送消息给它的应用程序对象。

有两个纲领性组件代表团。委托类必须定义一个属性（按照惯例，名为委托）委托持有的参考。它也必须声明一个协议，必须采取委托类（请参阅下面的部分协议）。很多类的Cocoa Touch和Cocoa框架提供的代表团作为一种方法，应用程序与特定应用程序的东西，是可以采取增加框架的行为。

但是，代表团并不限于框架类。您可以实现在一个应用程序的两个自定义对象之间的代表团。在可可触摸应用程序使用一种常见的设计代表团允许子视图控制器进行通信其父视图控制器的一些值（通常是用户输入的值）的一种手段。

协议：启用通信对象之间的继承关系

协议是一声明的程序接口的方法，任何类都可以实现。与协议相关联的类的实例调用的方法的协议，并获得返回值类正式通过和实施该协议。这一个具体的目标，如解析XML的代码或复制对象，对象之间的沟通，进一步加强。协议接口的任一侧上的对象可以被遥远地相互关联的，通过继承。因此协议是，代表团，子类化的替代，往往是一个框架的实施代表团的一部分。

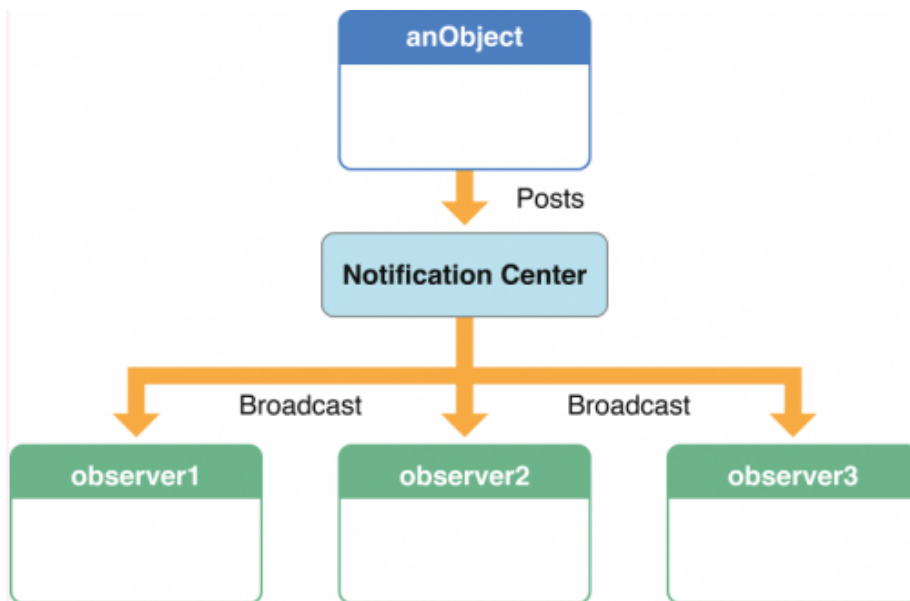


苹果提供的框架申报几十协议。此外，您的应用程序可以声明你的类可以采用自定义协议。协议是你的编程工具包的一部分。与Objective-C语言编程，给出了一个全面的协议描述。

通知中心：通知有关观察员事件

Foundation框架的通知中心是一个子系统，广播消息通知到一个应用程序中的所有对象注册事件的观察员。

（以编程方式，这是一个NSNotificationCenter类的实例。）事件可以发生在任何一个应用程序的应用程序进入背景状态，例如，或用户开始键入的文本字段中。一个通知，通知该事件已发生或即将发生的观察员，从而使观察者以适当的方式回应的机会。通知中心的广播通知是一种方法来增加一个应用程序对象之间的合作和凝聚力。



例如，在iOS应用程序的视图控制器可以观察的通知UIKeyboardWillShowNotification调整其视图的几何形状，以适应虚拟键盘。这个例子表明，通知是一个对象，它有一个名字，表明一个特定的事件，该事件是否已经发生或即将发生。它还带有一个参考对象的职位（或发送）的通知，通知中心，它可以包含一个字典的补充资料。

任何对象都可以观察到一个通知，但这样做的，它必须注册后才能收到它。在注册时，必须指定一个选择，确定要调用的方法的交付通知，方法的签名必须有一个参数：通知对象。注册时，观察者也可以指定发布对象。

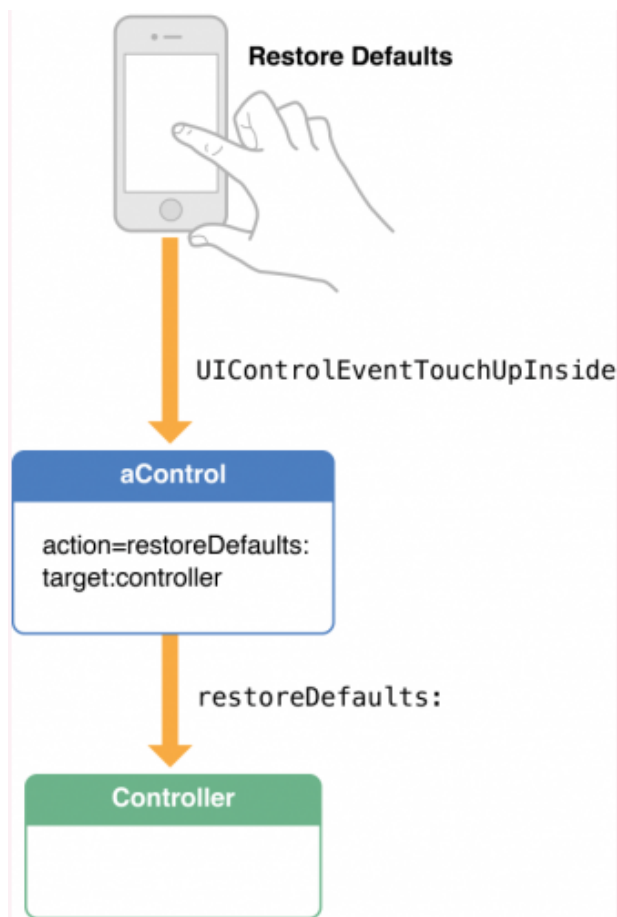
通知中心通知代表团消息被发送到任意对象的特定事件发生时。但是，处理通知的方法，委派方法不同，不能返回一个值。通过通知中心的通知是同步的，就像代表团。

在您的应用程序的自定义对象可以定义和发布自己的通知，以及其他定制的对象在你的应用程序中，可以观察到通知。

目标行动：当事件发生时要发送的消息封装

target-action设计的概念很简单。对象存储的元素，使讯息表达，当某个事件发生时，把这些元素组合在一起，并发送一条消息。的元素是吞吐量的消息（action），接收该消息的对象（target）的一个选择器。目标的类方法实现了相应的行动和目标，当它接收到的消息，在运行时，它响应此事件的执行方法。

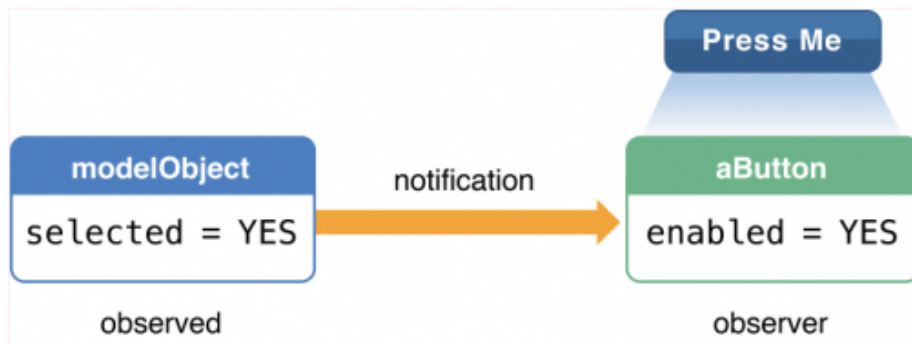
target-action主要是控制在Cocoa Touch和Cocoa框架的一个特点。控件是一个用户界面对象，如按钮，滑块，或切换用户操作（通过点击，拖动，等等），以表示它们打算到一个应用程序。Cocoa Touch控制存储的行动和目标都搭配一个或多个单元格对象存储目标和行动，大多数Cocoa Touch。



一些框架使用target-action控制以外的对象。例如，UIKit框架使用目标的行动，在其设计的手势识别。当一个手势识别对象识别手势，动作消息发送到目标对象。

键 - 值观察：当一个值发生变化时通知观察员

键 - 值观察允许一个对象在观察的另一个对象的一个属性。观察对象时，会通知该属性的值的变化。了解新的值，以及旧的，如果观测到的属性是一个对多的关系（例如，一个数组），它也得知涉及其中所包含的对象中的变化。志愿帮助应用程序变得更有凝聚力，保持对象模型，控制器，查看图层的变化同步。



类似NSNotificationCenter通知，多个志愿观察员可以观察到一个单一的财产。此外，KVO是动态的，因为它允许观察的对象，而不需要任何新的API，如通知名称的任意属性。国际志愿者组织是一个轻量级的点至点的通信机制，不允许观察的特定属性的所有实例。

基于设计模式的框架设计

CocoaKit和Cocoa框架还包括其他的设计基于设计模式，包括以下内容：

查看层次结构。的意见被安排在一个应用程序提出基于视觉的层次组织遏制。这种模式允许应用对待个人的观点和意见的成分均匀。在层次结构的根是一个窗口对象，每个视图中，根目录下有一个父视图和零个或更多孩子的意见。家长的意见附上孩子的意见。视图层次结构中的结构成分，两者的绘图和事件处理。

响应链。响应链是一系列的对象大多意见，也有窗口，视图控制器，本身以及其中的应用对象可以通过事件或动作消息，直到一个链中的对象处理该事件。因此，它是一个合作的事件处理机制。响应链密切相关的视图层次。

查看控制器。虽然两者的UIKit和AppKit的框架视图控制器类，他们在iOS中是特别重要的。视图控制器是一种特殊的控制器对象提出意见和管理一组。查看控制器对象提供的基础设施管理内容相关的意见和协调它们的显示和隐藏。视图控制器管理子层次结构的应用程序视图。

类别为您提供一种方式来扩展一个类添加方法。与delegate一样，它使您可以自定义行为没有子类。分类是一个Objective-C的功能描述中写Objective-C代码。

[查看原文>> \(http://blog.csdn.net/ganjicijiao/article/details/18412815\)](http://blog.csdn.net/ganjicijiao/article/details/18412815)



看过本文的人也看了：

- iOS知识结构图
(<http://lib.csdn.net/base/ios/structure>)
- iOS开发系列--通知与消息机制
(<http://lib.csdn.net/article/ios/42091>)
- 【IOS 开发】IOS 开发 简介 (IOS项...
(<http://lib.csdn.net/article/ios/35935>)
- iOS单例设计模式详解教程
(<http://lib.csdn.net/article/ios/42106>)
- ios_obj-c KVC,谓词,KVO和通知
(<http://lib.csdn.net/article/ios/42126>)
- 设计模式深入学习IOS版 (7) 观察者模...
(<http://lib.csdn.net/article/ios/42124>)

发表评论

输入评论内容

发表

0个评论

公司简介 (<http://www.csdn.net/company/about.html>) | 招贤纳士 (<http://www.csdn.net/company/recruit.html>) |
广告服务 (<http://www.csdn.net/company/marketing.html>) | 银行汇款帐号 (<http://www.csdn.net/company/account.html>)
| 联系方式 (<http://www.csdn.net/company/contact.html>) | 版权声明 (<http://www.csdn.net/company/statement.html>) |
法律顾问 (<http://www.csdn.net/company/layer.html>) | 问题报告 (<mailto:webmaster@csdn.net>) |
合作伙伴 (<http://www.csdn.net/friendlink.html>) | 论坛反馈 (<http://bbs.csdn.net/forums/Service>)

网站客服 杂志客服 (<http://wpa.qq.com/msgrd?v=3&uin=2251809102&site=qq&menu=yes>)

微博客服 (<http://e.weibo.com/csdnsupport/profile>) webmaster@csdn.net (<mailto:webmaster@csdn.net>) 400-600-2320 |

北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

 (<http://www.hd315.gov.cn/beian/view.asp?bianhao=010202001032100010>)