

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

iOS (<http://lib.csdn.net/base/ios>) - 经典场景 (<http://lib.csdn.net/ios/node/743>) - 推送 (<http://lib.csdn.net/ios/knowledge/1687>)

👁 364 💬 9

IOS消息推送之APNS

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

一、背景概述：

1, 环境配置

APNS : Apple Push Notification Service。本文对推送相关概念不再赘述，只侧重完整流程。

Demo 开发环境 : Mac os 10.9.4 , Xcode 6.0.1 ; 测试设备 : iphone 4s (ios 7.1)

服务端开发环境 : mac 10.9.4 + php 5.4.24、

Demo 下载地址 : 点击打开链接 (<http://download.csdn.net/detail/jiajiayouba/8019667>)

2, APNS 相关博客

如对apns相关概念不清楚，可参考以下几个博客：(博客中部分内容重复，但总体来说，通读一遍，还是大有裨益的)

<http://cshbbrain.iteye.com/blog/1859810> =》IOS 基于APNS消息推送原理与实现(JAVA后台)

<http://www.cnblogs.com/qq78292959/archive/2012/07/16/2593651.html> =》iOS消息推送机制的实现

<http://blog.csdn.net/xunyn/article/details/8243573> =》APNS编程----iOS真机测试消息推送

<http://blog.csdn.net/wswqiang/article/details/8208581> =》IOS APNS 处理

<http://eric-gao.iteye.com/blog/1567777> =》IOS PEM 文件的生成

<http://www.36coder.com/study/996.html> =》PHP 实现APNS 推送

<http://blog.csdn.net/sxfctt/article/details/7939082> =》APNS 相关总结 (推荐)

3, APNS 接口

消息推送：

开发接口 : gateway.sandbox.push.apple.com:2195

发布接口 : gateway.push.apple.com:2195

反馈服务：

开发接口 : feedback.sandbox.push.apple.com : 2196

发布接口：产品接口：feedback.push.apple.com：2196

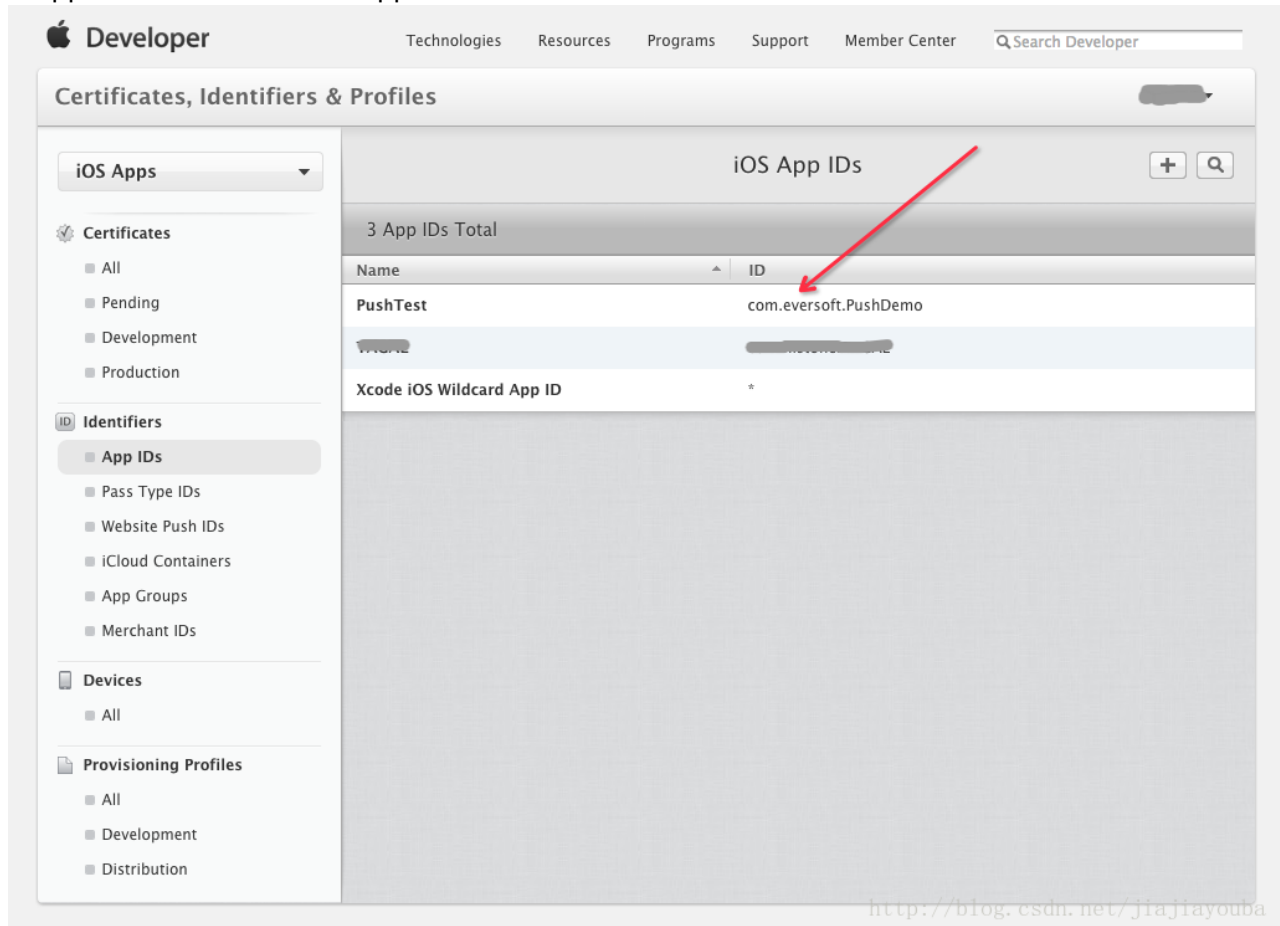
二、制作Push证书和Pem文件

1, 新建一个App ID

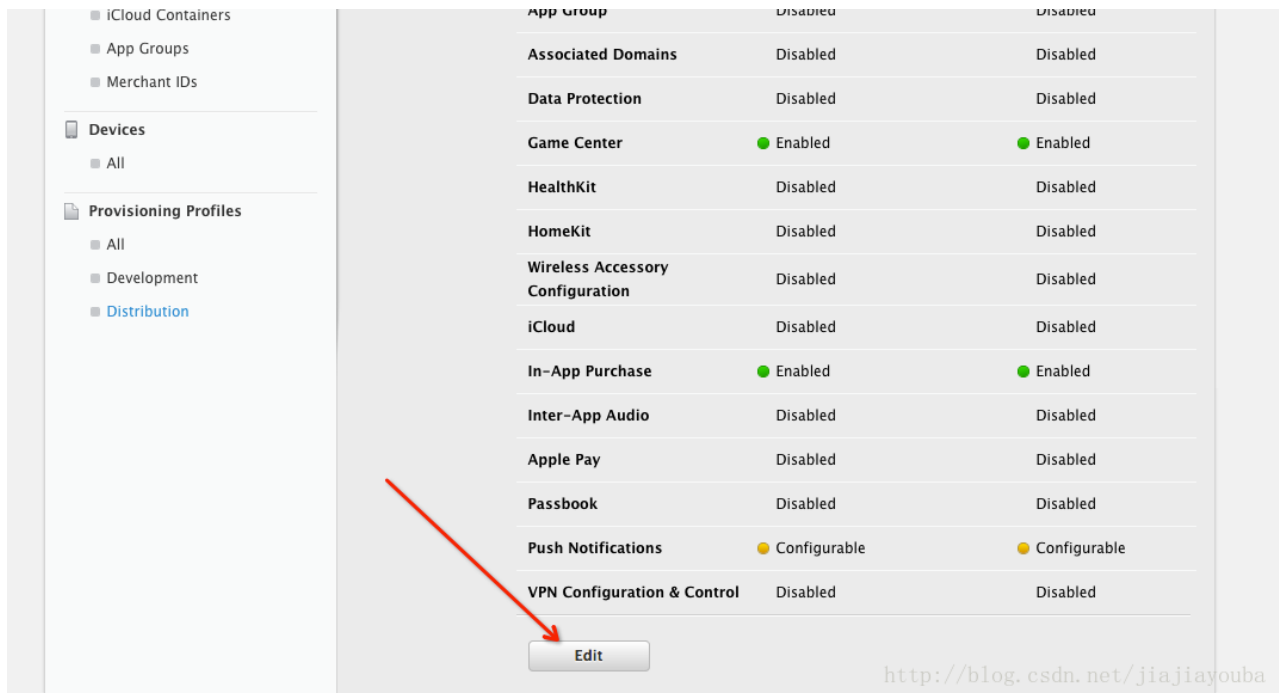
新建流程不再赘述，这里只提醒两点：1》App ID Suffix 中，一定要选择Explicit App ID；2》App Services 中，记得勾选Push Notifications。这里以新建一个id为：com.eversoft.PushDemo 为例。

2, 配置push开发证书

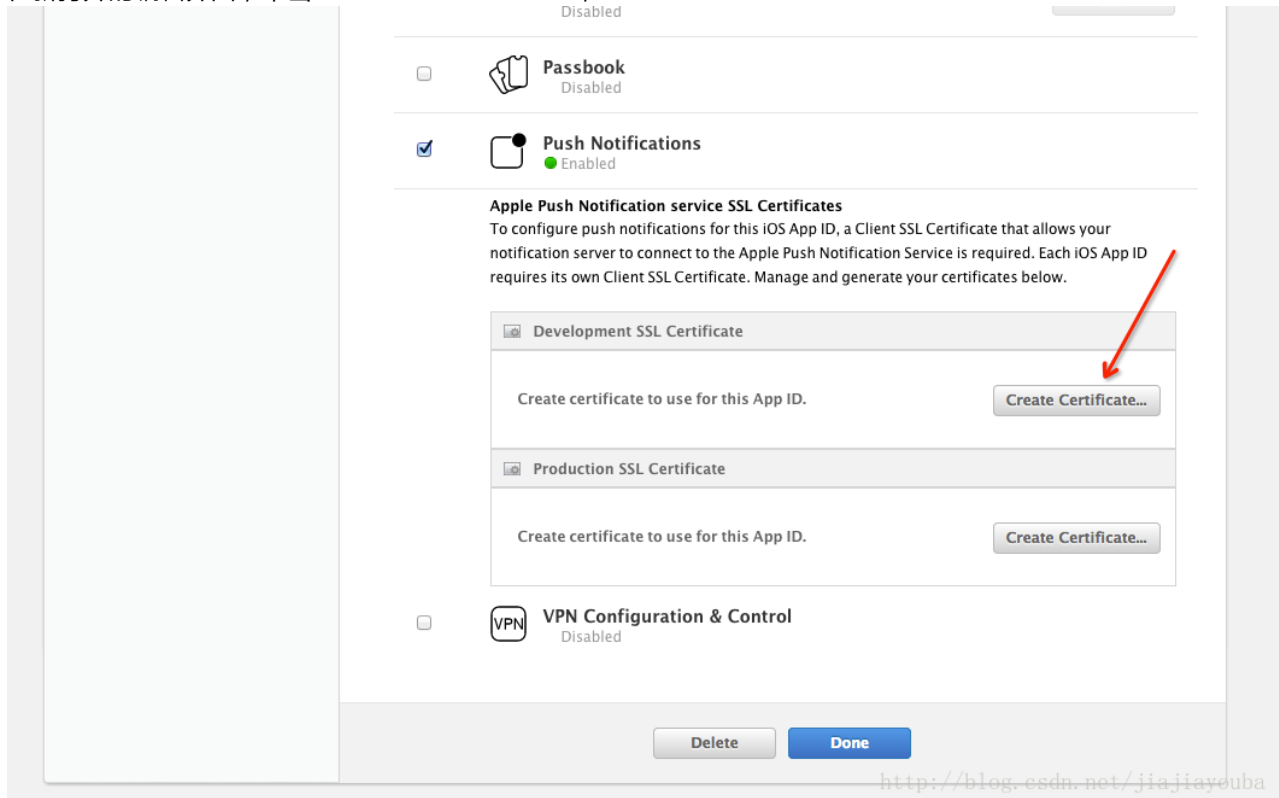
在App IDs中，选中刚才新建的App id：com.eversoft.PushDemo，单击，展开详细信息属性。



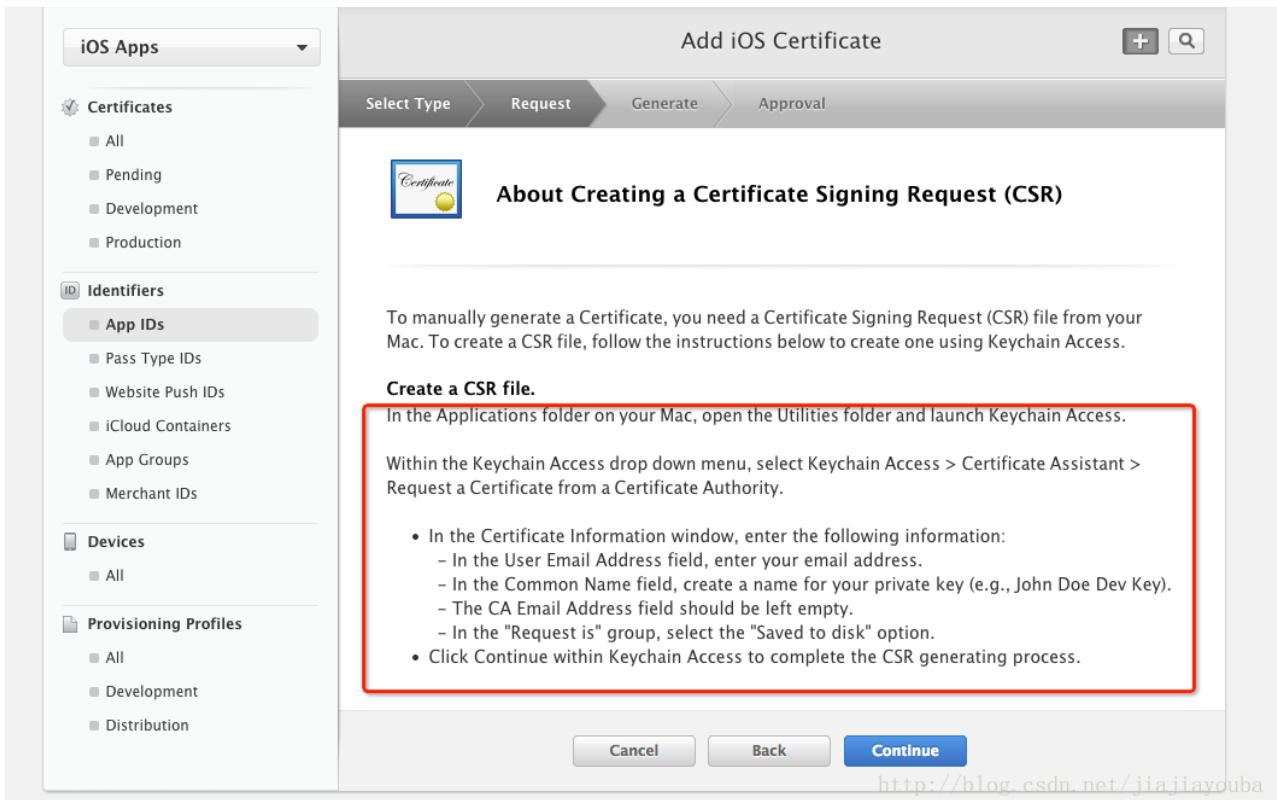
在详细信息属性中，单击下方的“Edit”按钮，



在新打开的编辑界面，单击“Create Certificate”，



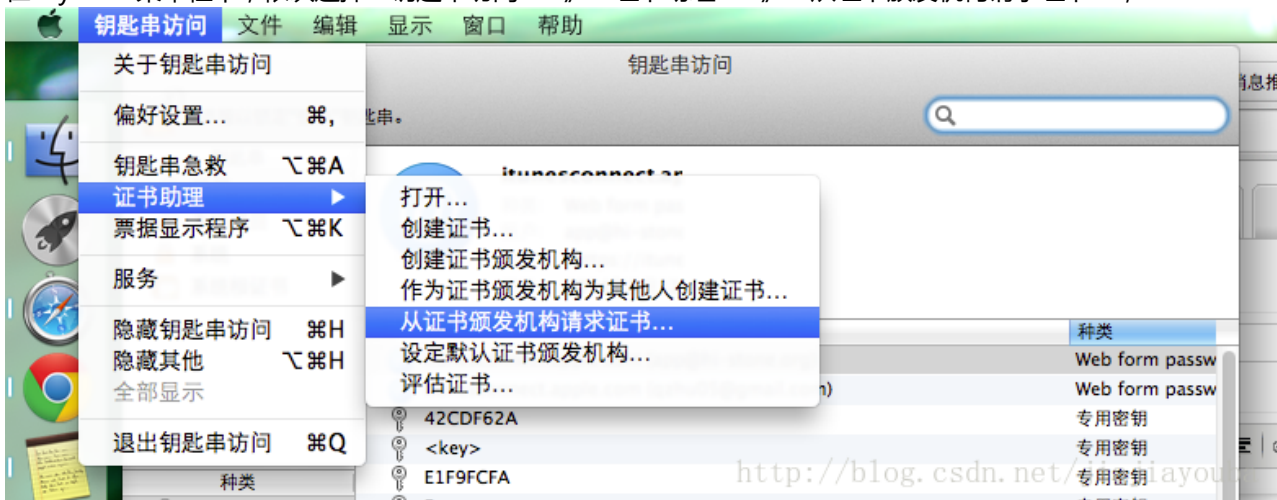
在新打开的界面中，会提示我们，创建一个csr 证书签名请求文件。具体的创建步骤，界面中已经给出了详细的英文说明。



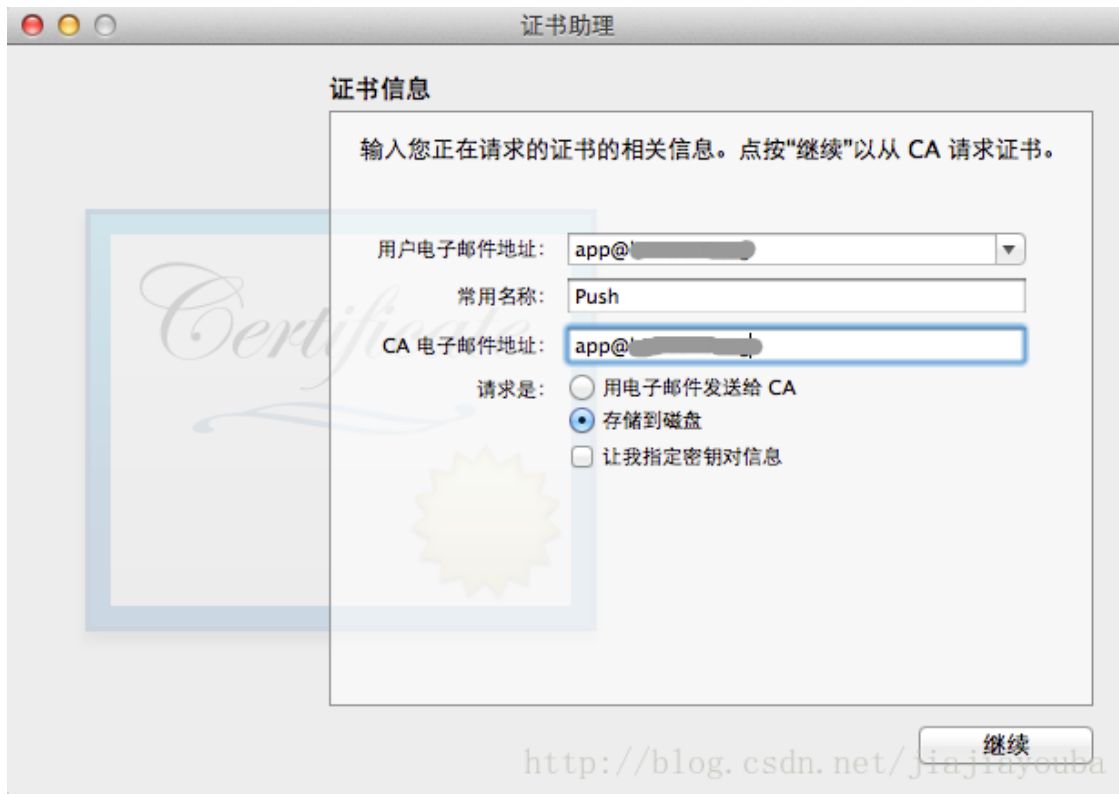
在进行下一步之前，我们先按照英文说明，创建一个 CSR 文件。

在mac电脑上，打开应用程序 keychain（钥匙串访问）；

在keychain菜单栏中，依次选择“钥匙串访问”=》“证书助理”=》“从证书颁发机构请求证书”；

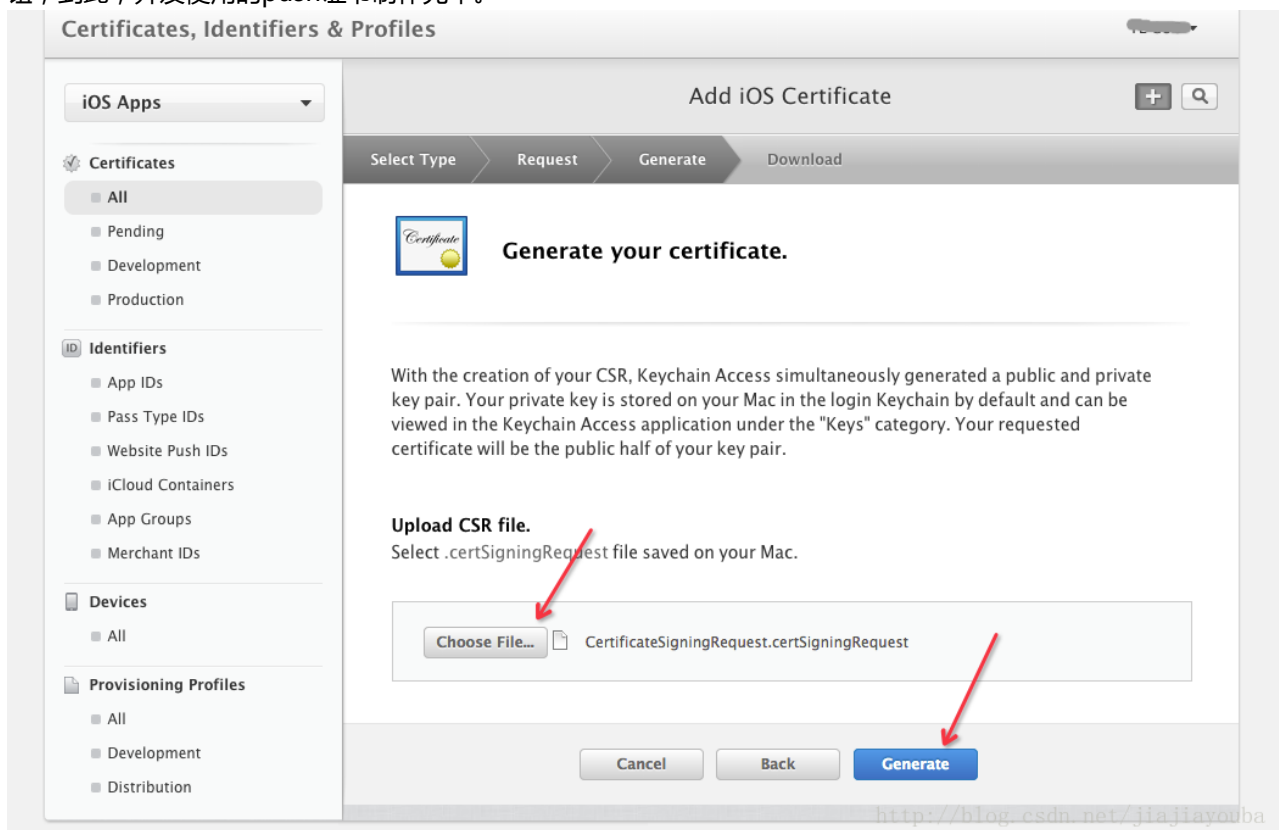


在新打开的“证书助理”界面中，填写用户电子邮件地址，常用名称，CA电子邮件地址，这两个邮件地址直接填写你的苹果账号的邮件地址即可，然后选择“存储到磁盘”，然后点击“继续”；



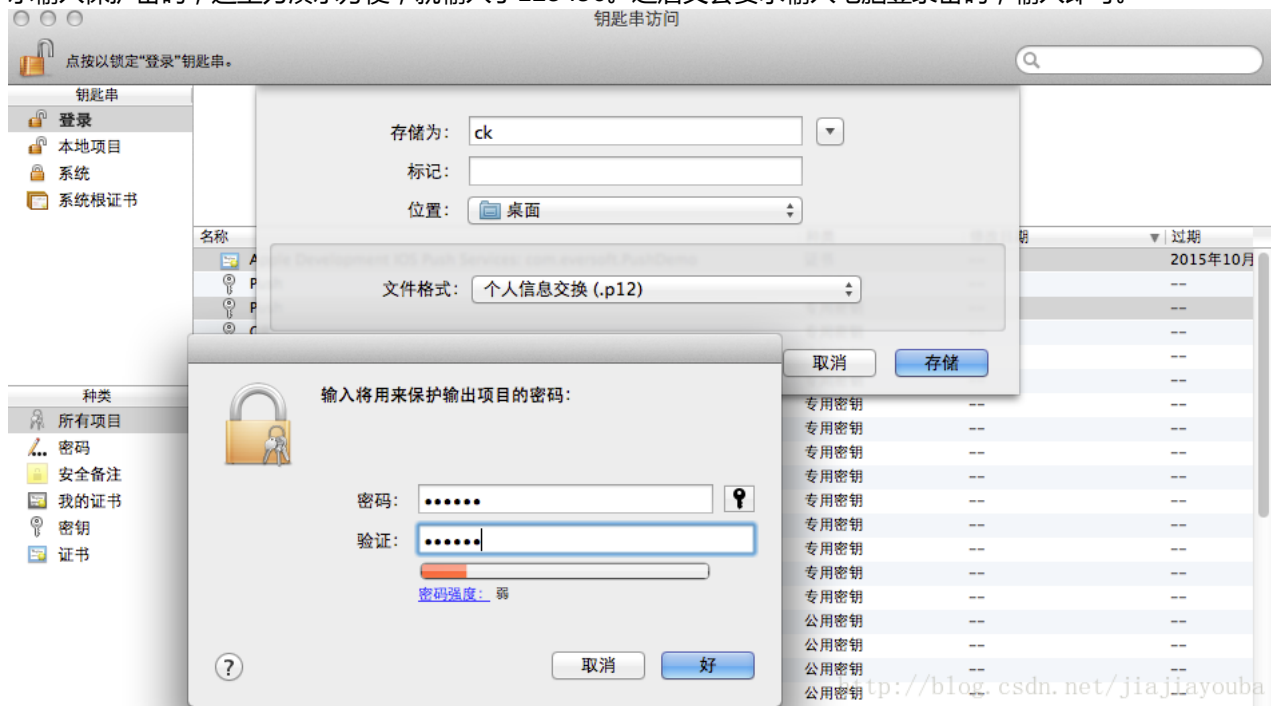
选择CSR文件保存位置，“存储”即可。至此，CSR文件，制作完成。

回到刚才我们的web页面上，点击“Continue”，进入下一页面；新的页面中，会要求我们上传刚才制作的csr文件，选择“Choose File”，找到我们刚才存储的csr文件，单击“打开”，最后，点击页面上的“Generate”按钮，到此，开发使用的push证书制作完毕。



证书生成成功后，选择“Download”，将制作好的证书下载到本地。然后双击下载的证书aps_development.cer，双击后，证书就自动导入到钥匙串中了。

打开 keychain，左侧钥匙串选择“登录”，种类选择“所有项目”，在右侧窗口中，选中刚才导入的Apple Development IOS Push Services证书(不用选中专用密钥)，右键，选择导出，命名为：ck.p12，存储时，会提示输入保护密码，这里为演示方便，就输入了123456。之后又会要求输入电脑登录密码，输入即可。



3, 生成PEM文件

最后，打开终端，执行以下命令，生成pem文件

```
openssl pkcs12 -in ck.p12 -out ck.pem -nodes
```

执行时，会要求输入导入密码，这里输入刚才的保护密码123456即可。



到此，php 服务端使用的pem证书就制作完毕了。

Development PP 文件制作不再赘述。

三、IOS 代码编写

首先，在AppDelegate.m 中：

1, 注册通知

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    ViewController *mainCtrl=[[ViewController alloc] init];
    self.window.rootViewController=mainCtrl;

    //注册通知
    if ([UIDevice currentDevice].systemVersion.doubleValue<8.0) {
        [[UIApplication sharedApplication] registerForRemoteNotificationTypes:(UIRemoteNotificationTypeAlert | UIRemoteNotificationTypeSound | UIRemoteNotificationTypeBadge)];
    }
    else {
        [[UIApplication sharedApplication] registerForRemoteNotifications];
        [[UIApplication sharedApplication] registerUserNotificationSettings:[UIUserNotificationSettings settingsForTypes:UIUserNotificationTypeBadge|UIUserNotificationTypeSound|UIUserNotificationTypeAlert categories:nil]];
    }

    //判断是否由远程消息通知触发应用程序启动
    if (launchOptions) {
        //获取应用程序消息通知标记数（即小红圈中的数字）
        NSInteger badge = [UIApplication sharedApplication].applicationIconBadgeNumber;
        if (badge>0) {
            //如果应用程序消息通知标记数（即小红圈中的数字）大于0，清除标记。
            badge--;
            //清除标记。清除小红圈中数字，小红圈中数字为0，小红圈才会消除。
            [UIApplication sharedApplication].applicationIconBadgeNumber = badge;
            NSDictionary *pushInfo = [launchOptions objectForKey:@"UIApplicationLaunchOptionsRemoteNotificationKey"];

            //获取推送详情
            NSString *pushString = [NSString stringWithFormat:@"%@",[pushInfo objectForKey:@"aps"]];
            UIAlertView *alert=[[UIAlertView alloc] initWithTitle:@"finish Loaunch" message:pushString delegate:nil cancelButtonTitle:@"cancel" otherButtonTitles:nil, nil];
            [alert show];
        }
    }

    return YES;
}
```

2, 注册通知后，获取device token

```

- (void)application:(UIApplication *)app didRegisterForRemoteNotificationsWithDeviceToken
:(NSData *)deviceToken {
    NSString *token = [NSString stringWithFormat:@"%@", deviceToken];
    NSLog(@"My token is:%@", token);
    //这里应将device token发送到服务器端
}

- (void)application:(UIApplication *)app didFailToRegisterForRemoteNotificationsWithError
:(NSError *)error {
    NSString *error_str = [NSString stringWithFormat: @"%@", error];
    NSLog(@"Failed to get token, error:%@", error_str);
}

```

3, 接收推送通知

```

- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo
{
    [UIApplication sharedApplication].applicationIconBadgeNumber=0;
    for (id key in userInfo) {
        NSLog(@"key: %@, value: %@", key, [userInfo objectForKey:key]);
    }
    /* eg.
    key: aps, value: {
        alert = "\U8fd9\U662f\U4e00\U6761\U6d4b\U8bd5\U4fe1\U606f";
        badge = 1;
        sound = default;
    }
    */
    UIAlertView *alert=[[UIAlertView alloc] initWithTitle:@"remote notification" message:
userInfo[@"aps"][@"alert"] delegate:nil cancelButtonTitle:@"cancel" otherButtonTitles:nil
, nil];
    [alert show];
}

```

注意：app 前台运行时，会调用 remote notification；app后台运行时，点击提醒框，会调用remote notification，点击app 图标，不调用remote notification，没反应；app 没有运行时，点击提醒框，finishLaunching 中，launchOptions 传参，点击app 图标，launchOptions 不传参，不调用remote notification。

四、服务器端代码编写

此章不在iOS程序员职责范围之内，故只给出示例代码，不做深入讨论。

1, php 源码：

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>APNS</title>
</head>
<body>
<?php
/**
 * @file apns.php

```



```

* @synopsis apple APNS class
* @author Yee, <rlk002@gmail.com>
* @version 1.0
* @date 2012-09-17 11:27:59
*/
class APNS
{
    const ENVIRONMENT_PRODUCTION = 0;
    const ENVIRONMENT_SANDBOX = 1;
    const DEVICE_BINARY_SIZE = 32;
    const CONNECT_RETRY_INTERVAL = 1000000;
    const SOCKET_SELECT_TIMEOUT = 1000000;
    const COMMAND_PUSH = 1;
    const STATUS_CODE_INTERNAL_ERROR = 999;
    const ERROR_RESPONSE_SIZE = 6;
    const ERROR_RESPONSE_COMMAND = 8;
    const PAYLOAD_MAXIMUM_SIZE = 256;
    const APPLE_RESERVED_NAMESPACE = 'aps';
    protected $_environment;
    protected $_providerCertificateFile;
    protected $_rootCertificationAuthorityFile;
    protected $_connectTimeout;
    protected $_connectRetryTimes = 3;
    protected $_connectRetryInterval;
    protected $_socketSelectTimeout;
    protected $_hSocket;
    protected $_deviceTokens = array();
    protected $_text;
    protected $_badge;
    protected $_sound;
    protected $_customProperties;
    protected $_expiryValue = 604800;
    protected $_customIdentifier;
    protected $_autoAdjustLongPayload = true;
    protected $asurls = array('ssl://gateway.push.apple.com:2195', 'ssl://gateway.sand
box.push.apple.com:2195');
    protected $_errorResponseMessages = array
    (
        0 => 'No errors encountered',
        1 => 'Processing error',
        2 => 'Missing device token',
        3 => 'Missing topic',
        4 => 'Missing payload',
        5 => 'Invalid token size',
        6 => 'Invalid topic size',
        7 => 'Invalid payload size',
        8 => 'Invalid token',
        self::STATUS_CODE_INTERNAL_ERROR => 'Internal error'
    );

    function __construct($environment, $providerCertificateFile)
    {
        if($environment != self::ENVIRONMENT_PRODUCTION && $environment != self::ENVI
RONMENT_SANDBOX)
        {
            throw new Exception(

```

```

        "Invalid environment '{$environment}'"
    );
}
$this->_environment = $environment;

if(!is_readable($providerCertificateFile))
{
    throw new Exception(
        "Unable to read certificate file '{$providerCertificateFile}'"
    );
}
$this->_providerCertificateFile = $providerCertificateFile;

$this->_connectTimeout = @ini_get("default_socket_timeout");
$this->_connectRetryInterval = self::CONNECT_RETRY_INTERVAL;
$this->_socketSelectTimeout = self::SOCKET_SELECT_TIMEOUT;
}

public function setRCA($rootCertificationAuthorityFile)
{
    if(!is_readable($rootCertificationAuthorityFile))
    {
        throw new Exception(
            "Unable to read Certificate Authority file '{$rootCertificationAuthor
ityFile}'"
        );
    }
    $this->_rootCertificationAuthorityFile = $rootCertificationAuthorityFile;
}

public function getRCA()
{
    return $this->_rootCertificationAuthorityFile;
}

protected function _connect()
{
    $sURL = $this->asurls[$this->_environment];
    $streamContext = stream_context_create(
        array
        (
            'ssl' => array
            (
                'verify_peer' => isset($this->_rootCertificationAuthorityFile
),
                'cafile' => $this->_rootCertificationAuthorityFile,
                'local_cert' => $this->_providerCertificateFile
            )
        )
    );

    $this->_hSocket = @stream_socket_client($sURL,$nError,$sError,$this->_connect
Timeout,STREAM_CLIENT_CONNECT, $streamContext);

    if (!$this->_hSocket)
    {

```

```

        throw new Exception
        (
            "Unable to connect to '{$sURL}': {$sError} ({$nError})"
        );
    }
    stream_set_blocking($this->_hSocket, 0);
    stream_set_write_buffer($this->_hSocket, 0);
    return true;
}

public function connect()
{
    $bConnected = false;
    $retry = 0;
    while(!$bConnected)
    {
        try
        {
            $bConnected = $this->_connect();
        } catch (Exception $e)
        {
            if ($nRetry >= $this->_connectRetryTimes)
            {
                throw $e;
            } else
            {
                usleep($this->_nConnectRetryInterval);
            }
        }
        $retry++;
    }
}

public function disconnect()
{
    if (is_resource($this->_hSocket))
    {
        return fclose($this->_hSocket);
    }
    return false;
}

protected function getBinaryNotification($deviceToken, $payload, $messageID = 0,
$Expire = 604800)
{
    $tokenLength = strlen($deviceToken);
    $payloadLength = strlen($payload);

    $ret = pack('CNNnH*', self::COMMAND_PUSH, $messageID, $Expire > 0 ? time() +
$Expire : 0, self::DEVICE_BINARY_SIZE, $deviceToken);
    $ret .= pack('n', $payloadLength);
    $ret .= $payload;
    return $ret;
}

protected function readErrorMessage()

```

```

    {
        $errorResponse = @fread($this->_hSocket, self::ERROR_RESPONSE_SIZE);
        if ($errorResponse === false || strlen($errorResponse) != self::ERROR_RESPONSE_SIZE)
        {
            return;
        }
        $errorResponse = $this->parseErrorMessage($errorResponse);
        if (!is_array($errorResponse) || empty($errorResponse))
        {
            return;
        }
        if (!isset($errorResponse['command'], $errorResponse['statusCode'], $errorResponse['identifier']))
        {
            return;
        }
        if ($errorResponse['command'] != self::ERROR_RESPONSE_COMMAND)
        {
            return;
        }
        $errorResponse['timeline'] = time();
        $errorResponse['statusMessage'] = 'None (unknown)';
        if (isset($this->_aErrorResponseMessages[$errorResponse['statusCode']]))
        {
            $errorResponse['statusMessage'] = $this->_aErrorResponseMessages[$errorResponse['statusCode']];
        }
        return $errorResponse;
    }

    protected function parseErrorMessage($errorMessage)
    {
        return unpack('Ccommand/CstatusCode/Nidentifier', $errorMessage);
    }

    public function send()
    {
        if (!$this->_hSocket)
        {
            throw new Exception(
                'Not connected to Push Notification Service'
            );
        }
        $sendCount = $this->getDTNumber();
        $messagePayload = $this->getPayload();
        foreach($this->_deviceTokens AS $key => $value)
        {
            $apnsMessage = $this->getBinaryNotification($value, $messagePayload, $messageID = 0, $expire = 604800);
            $nLen = strlen($apnsMessage);
            $aErrorMessage = null;
            if ($nLen != ($nWritten = (int)@fwrite($this->_hSocket, $apnsMessage)))
            {
                $aErrorMessage = array

```

```

        (
            'identifier' => $key,
            'statusCode' => self::STATUS_CODE_INTERNAL_ERROR,
            'statusMessage' => sprintf('%s (%d bytes written instead of %d bytes)', $this->_errorResponseMessages[self::STATUS_CODE_INTERNAL_ERROR], $nWritten, $nLen)
        );
    }
}

public function addDT($deviceToken)
{
    if (!preg_match('~^[a-f0-9]{64}$~i', $deviceToken))
    {
        throw new Exception
        (
            "Invalid device token '{$deviceToken}'"
        );
    }
    $this->_deviceTokens[] = $deviceToken;
}

public function getDTNumber()
{
    return count($this->_deviceTokens);
}

public function setText($text)
{
    $this->_text = $text;
}

public function getText()
{
    return $this->_text;
}

public function setBadge($badge)
{
    if (!is_int($badge))
    {
        throw new Exception
        (
            "Invalid badge number '{$badge}'"
        );
    }
    $this->_badge = $badge;
}

public function getBadge()
{
    return $this->_badge;
}

public function setSound($sound = 'default')

```

```

    {
        $this->_sound = $sound;
    }

    public function getSound()
    {
        return $this->_sound;
    }

    public function setCP($name, $value)
    {
        if ($name == self::APPLE_RESERVED_NAMESPACE)
        {
            throw new Exception
            (
                "Property name '" . self::APPLE_RESERVED_NAMESPACE . "' can not be us
ed for custom property."
            );
        }
        $this->_customProperties[trim($name)] = $value;
    }

    protected function _getPayload()
    {
        $aPayload[self::APPLE_RESERVED_NAMESPACE] = array();

        if (isset($this->_text))
        {
            $aPayload[self::APPLE_RESERVED_NAMESPACE]['alert'] = (string)$this->_text
;
        }
        if (isset($this->_badge) && $this->_badge > 0)
        {
            $aPayload[self::APPLE_RESERVED_NAMESPACE]['badge'] = (int)$this->_badge;
        }
        if (isset($this->_sound))
        {
            $aPayload[self::APPLE_RESERVED_NAMESPACE]['sound'] = (string)$this->_soun
d;
        }

        if (is_array($this->_customProperties))
        {
            foreach($this->_customProperties as $propertyName => $propertyValue)
            {
                $aPayload[$propertyName] = $propertyValue;
            }
        }
        return $aPayload;
    }

    public function setExpiry($expiryValue)
    {
        if (!is_int($expiryValue))
        {
            throw new Exception

```

```

        (
            "Invalid seconds number '{$expiryValue}'"
        );
    }
    $this->_expiryValue = $expiryValue;
}

public function getExpiry()
{
    return $this->_expiryValue;
}

public function setCustomIdentifier($customIdentifier)
{
    $this->_customIdentifier = $customIdentifier;
}

public function getCustomIdentifier()
{
    return $this->_customIdentifier;
}

public function getPayload()
{
    $sJSONPayload = str_replace
    (
        '"' . self::APPLE_RESERVED_NAMESPACES . '":[]',
        '"' . self::APPLE_RESERVED_NAMESPACES . '":{}',
        json_encode($this->_getPayload())
    );
    $nJSONPayloadLen = strlen($sJSONPayload);

    if ($nJSONPayloadLen > self::PAYLOAD_MAXIMUM_SIZE)
    {
        if ($this->_autoAdjustLongPayload)
        {
            $maxTextLen = $textLen = strlen($this->_text) - ($nJSONPayloadLen - s
            elf::PAYLOAD_MAXIMUM_SIZE);
            if ($nMaxTextLen > 0)
            {
                while (strlen($this->_text = mb_substr($this->_text, 0, --$textLe
            n, 'UTF-8')) > $maxTextLen);
                return $this->getPayload();
            }
        }
        else
        {
            throw new Exception
            (
                "JSON Payload is too long: {$nJSONPayloadLen} bytes. Maximum
            size is " .
                self::PAYLOAD_MAXIMUM_SIZE . " bytes. The message text can no
            t be auto-adjusted."
            );
        }
    }
    else
    {
        throw new Exception

```

```

        (
            "JSON Payload is too long: {$nJSONPayloadLen} bytes. Maximum size
is " .
            self::PAYLOAD_MAXIMUM_SIZE . " bytes"
        );
    }
}
return $sJSONPayload;
}
}

?>
<?php
date_default_timezone_set('PRC');
echo "we are young,test apns.  -".date('Y-m-d h:i:s',time());

$rootpath = 'entrust_root_certification_authority.pem'; //ROOT证书地址
$cp = 'ck.pem'; //provider证书地址
$apns = new APNS(1,$cp);
try
{
    //$apns->setRCA($rootpath); //设置ROOT证书
    $apns->connect(); //连接
    $apns->addDT('acc5150a4df26507a84f19ba145ca3c1be5842a6177511ce7c43d01badb1bd96'); //
加入deviceToken
    $apns->setText('这是一条测试信息'); //发送内容
    $apns->setBadge(1); //设置图标数
    $apns->setSound(); //设置声音
    $apns->setExpiry(3600); //过期时间
    $apns->setCP('custom operation',array('type' => '1','url' => 'http://www.google.com.h
k')); //自定义操作
    $apns->send(); //发送
    echo ' sent ok';
} catch(Exception $e)
{
    echo $e;
}
?>

</body>
</html>

```

2, 启动 Apache

mac 自带apache ,可直接运行php。

打开 “终端(terminal)” , 输入 `sudo apachectl -v` , 可显示Apache的版本 ;

输入 `sudo apachectl start` , 这样Apache就启动了。

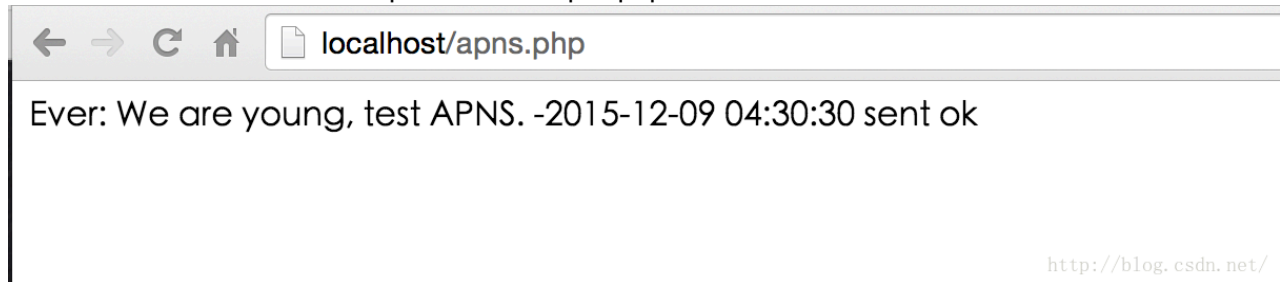
编辑文件 `/etc/apache2/httpd.conf` , 把 `LoadModule php5_module libexec/apache2/libphp5.so` 前面的注释去掉 ; 然后重启apache : `sudo apachectl restart`

打开Safari浏览器地址栏输入 “`http://localhost`” , 可以看到内容为 “It works!” 的页面。其位

于 "/Library/WebServer/Documents/" 下，这就是Apache的默认根目录。

3, 如何调试

将服务器端写好的apns.php 文件以及生成的 ck.pem 文件，直接拷贝到 /Library/WebServer/Documents/ 下，在浏览器中，直接浏览：<http://localhost/apns.php> 。这样消息就发送到了苹果服务器。



[查看原文>> \(http://blog.csdn.net/jiajiayouba/article/details/39926017\)](http://blog.csdn.net/jiajiayouba/article/details/39926017)



看过本文的人也看了：

- iOS知识结构图
(<http://lib.csdn.net/base/ios/structure>)
- iOS经典讲解之实现App消息推送功能(二...
(<http://lib.csdn.net/article/ios/43788>)
- iOS收到推送消息后跳到指定的页面
(<http://lib.csdn.net/article/ios/43841>)
- ios push推送
(<http://lib.csdn.net/article/ios/43836>)
- iOS 通知——推送通知
(<http://lib.csdn.net/article/ios/43800>)
- iOS远程推送
(<http://lib.csdn.net/article/ios/43822>)

发表评论

输入评论内容

发表

9个评论



(<http://my.csdn.net/u013936053>)

u013936053 (<http://my.csdn.net/u013936053>)

66666

2016-09-23 18:30:13

回复



(<http://my.csdn.net/u011519618>)

u011519618 (<http://my.csdn.net/u011519618>)

66666666

2016-07-06 15:25:24

回复



(<http://my.csdn.net/jinbingchenying>)

jinbingchenying (<http://my.csdn.net/jinbingchenying>)

为什么运行php出错了，全复制有问题么

2016-06-07 11:13:13

回复



(http://my.csdn.net/pengyuan_D)

pengyuan_D (http://my.csdn.net/pengyuan_D)

好

2015-11-11 16:18:24

回复



(<http://my.csdn.net/Sqingfeng>)

Sqingfeng (<http://my.csdn.net/Sqingfeng>)

大神，求指导，<http://localhost/apns.php> 在浏览器中调试后，怎么没有消息推送到手机上

2015-10-15 11:54:43

回复

加载更多

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>)