# TÀI LIỆU HƯỚNG DẪN KẾT NỐI SMS BRANDNAME

Version 2.0 (11/03/2013)

# Table of Contents

# 1. Giới thiệu tài liệu

Tài liệu mô tả API dành cho các đối tác và các đơn vị liên quan kết nối hệ thống SMS Brandname Vivas để gửi tin nhắn Brandname

# 2. Yêu cầu kỹ thuật

Đối tác và đơn vị liên quan cần chuẩn bị những hạng mục sau

| Yêu cầu | Chú thích | Mô tả |
|---|---|---|
| Kết nối Internet | Bắt buộc | Dành cho đối tác kết nối với hệ thống SMS Brandname của Vivas thông qua Internet |
| Kết nối metronet | Không bắt buộc | Dành cho đối tác kết nối với hệ thống SMS Brandname của Vivas thông qua đường riêng |
| IP tĩnh | Khuyến nghị | Đối tác nên sử dụng IP tĩnh để kết nối với hệ thống SMS Brandname của Vivas |
| Khả năng lập trình | Bắt buộc | Để kết nối với hệ thống SMS Brandname của Vivas qua API, nhân lực đối tác yêu cầu phải có khả năng lập trình và triển khai theo tài liệu API |

# 3. Các bước kết nối kỹ thuật

B1: Vivas gửi tài liệu hướng dẫn kết nối cho đối tác

B2: Vivas gửi tài khoản và Brandname test cho đối tác

Đối với đối tác sử dụng MD5: Vivas cung cấp sharekey

Đối với đối tác yêu cầu bảo mật SHA1withRSA: đối tác cung cấp public key tương ứng với private key mà đối tác sẽ sử dụng trong kết nối. Chú ý: đối tác không phải cấp cho Vivas private key của mình

B3: Đối tác tiến hành thử nghiệm kết nối, gửi Brandname test và tiến hành các hạng mục lập trình và triển khai phía đối tác

B4: Vivas gửi BB nghiệm thu kết nối kỹ thuật cho đối tác

B5: Đối tác tiến hành test nghiệm thu kết nối kỹ thuật

B6: Vivas gửi tài khoản và Brandname theo HĐ kinh doanh cho đối tác

B7: Vivas hỗ trợ kỹ thuật trong quá trình 2 bên hợp tác kinh doanh

# 4. Đặc tả API

Hệ thống SMS Brandname của Vivas cung cấp bộ API, giúp đối tác có thể kết nối hệ thống phía đối tác và hệ thống Vivas

- Đơn giản, dễ triển khai
- Hỗ trợ giao thức bảo mật trong truyền tin: HTTPS
- Có đầy đủ các hàm chức năng cho việc gửi tin và kiểm tra trạng thái gửi
- Có cơ chế đảm bảo an toàn thông tin và toàn vẹn dữ liệu trong quá trình gửi nhận request, phù hợp với các mức độ yêu cầu của đối tác

Bộ API kết nối bao gồm 4 hàm chức năng sau

| API | Mô tả chức năng |
|---|---|
| login | Đăng nhập hệ thống, xin session kết nối<br>Khi chưa đăng nhập, chỉ cho phép đối tác gửi một request đồng thời đến API<br>Sau khi đăng nhập, đối tác được phép gửi nhiều request đồng thời đến API trong giới han cho phép |
| send_sms | Gửi tin<br>Hỗ trợ gửi nhiều cùng nội dung tin nhắn đến nhiều số điện thoại, tối đa 1000 số |
| verify | Kiểm tra trạng thái gửi tin<br>Kiểm tra trạng thái các tin đã gửi ứng với API send_sms |
| logout | Đăng xuất hệ thống |

## a. login

| Tham số | Giá trị |
|---|---|
| Protocol | HTTP |
| Method | POST |
| URL | http://mkt.vivas.vn:9080/SMSBNAPI/login |
| Data | XML (CHÚ Ý ESCAPE CÁC KÝ TỰ " ' < > &) |
| Response | Chú ý lấy header "Set-Cookie" để giữ SessionID khi gọi các API tiếp theo |

### 1. Request format

```
Request
<RQST>
<USERNAME>username</USERNAME>
<PASSWORD>password generated by SHA1</PASSWORD>
</RQST>
```

| Tham số | Mô tả | Bắt buộc | Yêu cầu |
|---|---|---|---|
| USERNAME | Tên đăng nhập hệ thống | Có | Case sensitive |
| PASSWORD | Mật khẩu được băm | Có | Tham khảo phụ lục về hàm băm |

| | | | |
|---|---|---|---|
| | sử dụng SHA1 | | |

### 2. Response format

```
Response
<RPLY>
<STATUS>login status</STATUS>
</RPLY>
```

| Login Status | Giá trị |
|---|---|
| 0 | Đăng nhập thành công |
| 1 | Sai username |
| 2 | Sai password |
| 21 | Request bị từ chối vì quá số lượng request đồng thời đến hệ thống |
| 50 | Lỗi xử lý |
| 51 | Lỗi xử lý |
| 52 | Lỗi xử lý |
| 98 | Lỗi sai protocol gọi request |
| 99 | Lỗi thiếu tham số gọi request |

## b. send_sms

| Tham số | Giá trị |
|---|---|
| Protocol | HTTP |
| Method | POST |
| URL | http://mkt.vivas.vn:9080/SMSBNAPI/send_sms |
| Data | XML (CHÚ Ý ESCAPE CÁC KÝ Tự " ' < > &) |
| Request | Chú ý gắn header "Cookie" với SessionID thu được khi login để giữ session |

### 1. Request format

```
Request
<RQST>
	<REQID>remote request id</REQID>
	<BRANDNAME>brandname</BRANDNAME>
	<TEXTMSG>msg</TEXTMSG>
	<SENDTIME>yyyyMMddHHmmss</SENDTIME>
	<TYPE>1: CSKH hoặc 2: QC</TYPE>
	// loop
	<DESTINATION>
		<MSGID>remote msg id</MSGID>
		<MSISDN>msisdn</MSISDN>
		<CHECKSUM>checksum</CHECKSUM>
	</DESTINATION>
	// end loop
</RQST>
```

| Tham số | Mô tả | Bắt buộc | Yêu cầu |
|---|---|---|---|

| | | | |
|---|---|---|---|
| REQID | ID của request của hệ thống phía bên đối tác | Có | Là chuỗi ký tự, độ dài tối đa 255 ký tự, không được trùng nhau cho mỗi request |
| BRANDNAME | Tên Brandname | Có | Case sensitive |
| TEXTMSG | Nội dung tin nhắn | Có | Không chứa các ký tự đặc biệt charcode lớn hơn 127 |
| SENDTIME | Thời gian gửi tin | Có | Theo format yyyyMMddHHmmss |
| TYPE | Loại SMS | Có | 1: CSKH<br>2:QC |
| DESTINATION | Mỗi số điện thoại nhận tin nhắn được gửi trong một khối DESTINATION | Có | |
| MSGID | ID của SMS phía hệ thống đối tác | Có | Là chuỗi ký tự, độ dài tối đa 255 ký tự, không được trùng nhau cho mỗi SMS trong một request |
| MSISDN | Số điện thoại nhận SMS | Có | Format bắt đầu 84 |
| CHECKSUM | CHECKSUM của SMS | Có | Chuỗi checksum sinh ra từ các thông số của request và SMS, sử dụng MD5 hoặc SHA1withRSA tùy theo đối tác<br>Tham khảo phục lục về cách tính checksum |

## 2. Response format

Failed Response

```
<RPLY>
	<STATUS>request status</STATUS>
</RPLY>
```

Success response

```
<RPLY>
	<REQID>remote request id</REQID>
	<STATUS>request status</STATUS>
</RPLY>
```

| Send Status | Giá trị |
|---|---|
| 0 | Request được tiếp nhận thành công |
| 3 | Request bị từ chối vì Brandname không tồn tại hoặc không thuộc sở hữu |
| 4 | Request bị từ chối vì không tìm thấy template hoặc không đúng template |
| 5 | Request bị từ chối vì chứa một checksum sai |
| 6 | Request bị từ chối vì trùng ID |
| 8 | Request bị từ chối vì vượt hạn mức gửi tin |
| 9 | Request bị từ thối vì thiếu loại SMS |
| 10 | Request bị từ chối vì thiếu thời gian gửi |
| 12 | Request bị từ chối vì trùng msgid |
| 13 | Request bị từ chối vì vượt quá số lượng số điện thoại trong |

| | |
|---|---|
| | request |
| 14 | Request bị từ chối vì chứa số điện thoại sai |
| 20 | Request bị từ chối vì chưa đăng nhập hoặc mất session |
| 21 | Request bị từ chối vì quá số lượng request đồng thời đến hệ thống |
| 50 | Lỗi xử lý |
| 51 | Lỗi xử lý |
| 52 | Lỗi xử lý |
| 98 | Lỗi sai protocol gọi request |
| 99 | Lỗi thiếu tham số gọi request |

## c. verify

| Tham số | Giá trị |
|---|---|
| Protocol | HTTP |
| Method | POST |
| URL | http://mkt.vivas.vn:9080/SMSBNAPI/verify |
| Data | XML (CHÚ Ý ESCAPE CÁC KÝ TỰ " ' < > &) |
| Request | Chú ý gắn header "Cookie" với SessionID thu được khi login để giữ session |

### 1. Request format

```
Request
<RQST>
	<REQID>remote request id</REQID>
</RQST>
```

| Tham số | Mô tả | Bắt buộc | Yêu cầu |
|---|---|---|---|
| REQID | ID của request của hệ thống phía bên đối tác | Có | Là chuỗi ký tự, độ dài tối đa 255 ký tự, là ID của request gửi tin trước đó |

### 2. Response format

```
Success Response
<RPLY>
	<REQID>remote request id</REQID>
	<STATUS>request status</STATUS>
	// loop
	<DESTINATION>
	<MSGID>remote msg id</MSGID>
	<MSISDN>msisdn</MSISDN>
	<RESULT>Msg result</RESULT>
	</DESTINATION>
	// end loop
</RPLY>
```

```
Failed response
<RPLY>
      <REQID>remote request id</REQID>
      <STATUS>verify status</STATUS>
</RPLY>
```

| Verify Status | Giá trị |
|---|---|
| 7 | Sai request ID |
| 20 | Request bị từ chối vì chưa đăng nhập hoặc mất session |
| 21 | Request bị từ chối vì quá số lượng request đồng thời đến hệ thống |
| 50 | Lỗi xử lý |
| 51 | Lỗi xử lý |
| 52 | Lỗi xử lý |
| 98 | Lỗi sai protocol gọi request |
| 99 | Lỗi thiếu tham số gọi request |

Status của SMS được quy định như sau

| SMS Status | Giá trị |
|---|---|
| 0 | Thành công, Gửi thành công đến Gateway |
| 1 | Chờ, Đang chờ xử lý |
| 2 | Chờ, Đang được gửi |
| 3 | Lỗi, Gửi thất bại |
| 4 | Lỗi, Bị hủy |
| 5 | Chờ, Bị Gateway từ chối, chờ gửi lại |
| 6 | Đã gửi, không nhận được phản hồi từ Gateway |
| 7 | Lỗi, Tin nhắn không hợp lệ |
| 8 | Lỗi, Tin nhắn vượt hạn mức |
| 9 | Lỗi, Tin nhắn không tìm thấy gateway để gửi |
| 10 | Lỗi khác |

## d. logout

| Tham số | Giá trị |
|---|---|
| Protocol | HTTP |
| Method | POST, GET |
| URL | http://mkt.vivas.vn:9080/SMSBNAPI/logout |
| Data | XML (CHÚ Ý ESCAPE CÁC KÝ TỰ “ ‘ < > &) |
| Request | Chú ý gắn header “Cookie” với SessionID thu được khi login để giữ session |

### 1. Response format

```
Response
<RPLY>
        <STATUS>logout status</STATUS>
</RPLY>
```

| Logout Status | Giá trị |
|---|---|
| 0 | Đăng xuất thành công |
| 20 | Request bị từ chối vì chưa đăng nhập hoặc mất session |
| 21 | Request bị từ chối vì quá số lượng request đồng thời đến hệ thống |
| 50 | Lỗi xử lý |
| 51 | Lỗi xử lý |
| 52 | Lỗi xử lý |
| 98 | Lỗi sai protocol gọi request |
| 99 | Lỗi thiếu tham số gọi request |

## 5. Yêu cầu khác

Đối tác cần tuân thủ các giới hạn sau đây khi sử dụng API kết nối hệ thống SMS Brandname

- Giới hạn số lượng kết nối đồng thời đến hệ thống: 20
- Giới hạn số lượng SMS trong một request: 1000
- Không sử dụng ID trùng nhau cho các request
- Không sử dụng ID trùng nhau cho các SMS trong request
- Truyền đầy đủ và chính xác các tham số yêu cầu

Các lỗi gây ra bởi việc vi phạm các quy định trên sẽ không được hỗ trợ bởi Vivas

Ngoài ra các hành vi không hợp lệ sẽ bị cảnh cáo và xử lý

Đối tác nên sử dụng domain name như quy định và DNS để kết nối với Vivas thay bằng IP, để tiện việc quy hoạch IP phía Vivas. Tuy nhiên Vivas sẽ thông báo với đối tác khi có sự thay đổi về IP phía Vivas.

Đối với các đối tác sử dụng firewall, đối tác phải cấu hình firewall cho phép các request từ CP sang địa chỉ IP hiện thời của Vivas và có trách nghiệm cập nhật IP này khi có thay đổi. Đối với các đối tác không sử dụng DNS, đối tác phải cấu hình hosts với địa chỉ IP hiện thời và có trách nhiệm cập nhật IP khi có thay đổi.

## 6. Phụ lục

### a. Tạo checksum sử dụng MD5

Checksum yêu cầu tham số sau

- username: tên đăng nhập
- password: mật khẩu băm sử dụng SHA1
- brandname: Brandname gửi tin
- sendtime: Thời gian gửi tin, format yyyyMMddHHmmss
- msgid: ID của tin nhắn trong request
- msg: nội dung tin nhắn
- msisdn: số điện thoại nhận tin, bắt đầu 84
- sharekey: key bí mật được trao cho đối tác

Chuỗi checksum được sinh ra bằng cách băm MD5 từ chuỗi ghép các tham số theo thứ tự nêu trên, cách nhau bằng dấu '&', giữa tham số và giá trị cách nhau bằng dấu '='

Ví dụ:
username=VIVAS&password=SHA1PASSWORD&brandname=COCACOLA&sendtime=20120415163000&msgid=1&msg=Hello&msisdn=84901234567&sharekey=PRESHAREDKEY

Sau đó checksum sẽ được sinh ra lại ở phía Vivas để so sánh với chuỗi đối tác gửi sang nhằm kiểm tra tính toàn vẹn và chính xác của thông tin

## b. Tạo checksum sử dụng SHA1withRSA

Checksum yêu cầu tham số sau

- username: tên đăng nhập
- password: mật khẩu băm sử dụng SHA1
- brandname: Brandname gửi tin
- sendtime: Thời gian gửi tin, format yyyyMMddHHmmss
- msgid: ID của tin nhắn trong request
- msg: nội dung tin nhắn
- msisdn: số điện thoại nhận tin, bắt đầu 84

Chuỗi checksum được sinh ra bằng cách băm MD5 từ chuỗi ghép các tham số theo thứ tự nêu trên, cách nhau bằng dấu '&', giữa tham số và giá trị cách nhau bằng dấu '='

Ví dụ:

username=VIVAS&password=SHA1PASSWORD&brandname=COCACOLA&sendtime=20120415163000&msgid=1&msg=Hello&msisdn=84901234567

Sau đó checksum được mã hóa bởi private key của đối tác sử dụng SHA1withRSA, và sẽ được giải mã bằng public key ở phía Vivas nhằm kiểm tra tính toàn vẹn và chính xác của thông tin

## c. Băm mật khẩu sử dụng SHA1

Mật khẩu khi gửi qua API được băm sử dụng SHA1 tiêu chuẩn

## d. Java example

### 1. Băm MD5

```
	/**
	 * @param input: original text
	 * @return hased string with MD5
	*/
public static String getMD5(String input){
	StringBuffer sb = new StringBuffer();
	try {
		MessageDigest md = MessageDigest.getInstance("MD5");
		md.update(input.getBytes());
		byte[] output = md.digest();
		for (int i = 0; i < output.length; i++) {
			sb.append(Integer.toString((output[i] & 0xff) + 0x100,
16).substring(1));
		}
	} catch (NoSuchAlgorithmException e) {
		e.printStackTrace();
	}
	return sb.toString();
}
```

### 2. Băm SHA1withRSA với private key

```
	/**
	 * @param private_key_hex_string: private key as hex string
	 * @param msg: original text
	 * @return hased string with SHA1withRSA
	 * @throws NoSuchAlgorithmException
	 * @throws InvalidKeySpecException
	 * @throws InvalidKeyException
	 * @throws SignatureException
	 */
	public static String getSignature(String private_key_hex_string, String
msg)
			throws NoSuchAlgorithmException,
			InvalidKeySpecException,
			InvalidKeyException,
			SignatureException{

		Signature kiemtra = Signature.getInstance("SHA1withRSA");
		PrivateKey pk = getPrivateKeyFromHexString(private_key_hex_string);
		kiemtra.initSign(pk);
		kiemtra.update(msg.getBytes());
		return byteArrayToHexString(kiemtra.sign());
	}
```

### 3. Kiểm tra SHA1withRSA với public key

```
	/**
	 * @param public_key_hex_string: public key as hex string
	 * @param msg: original text
	 * @param hash: hashed as hex string
	 * @return
	 * @throws InvalidKeySpecException
	 * @throws NoSuchAlgorithmException
	 * @throws InvalidKeyException
	 * @throws SignatureException
	 */
	public static boolean verifySignature(String public_key_hex_string, String
msg, String hash)
			throws InvalidKeySpecException,
			NoSuchAlgorithmException,
			InvalidKeyException,
			SignatureException{

		byte[] signature = hexStringToByteArray(hash);
		byte[] public_key_byte =
hexStringToByteArray(public_key_hex_string);
		PublicKey publicKey = KeyFactory.getInstance("RSA")
				.generatePublic(new
X509EncodedKeySpec(public_key_byte));

		Signature kiemtra = Signature.getInstance("SHA1withRSA");
		kiemtra.initVerify(publicKey);
		kiemtra.update(msg.getBytes());
		return kiemtra.verify(signature);
	}
```

### 4. Các method chung khác

```
	public static PrivateKey getPrivateKeyFromHexString(String
private_key_hex_string)
			throws NoSuchAlgorithmException,
			InvalidKeySpecException{

		KeyFactory kf = KeyFactory.getInstance("RSA");
		byte[] tmp = hexStringToByteArray(private_key_hex_string);
		PKCS8EncodedKeySpec ks = new PKCS8EncodedKeySpec(tmp);
		return kf.generatePrivate(ks);
	}
```

```
	public static String getPrivateKeyHexString(KeyPair keyPair){
		PrivateKey privateKey = keyPair.getPrivate();
		byte[] private_key = privateKey.getEncoded();

		return byteArrayToHexString(private_key);
	}

	public static String getPublicKeyHexString(KeyPair keyPair){
		PublicKey publicKey = keyPair.getPublic();
		byte[] public_key = publicKey.getEncoded();

		return byteArrayToHexString(public_key);
	}

	public static KeyPair genKeyPair() throws NoSuchAlgorithmException,
NoSuchProviderException {
		KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
		keyGen.initialize(1024);

		return keyGen.genKeyPair();
	}

	public static String byteArrayToHexString(byte[] bytes) {
		final char[] hexArray =
{'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
		char[] hexChars = new char[bytes.length * 2];
		int v;
		for ( int j = 0; j < bytes.length; j++ ) {
			v = bytes[j] & 0xFF;
			hexChars[j * 2] = hexArray[v >>> 4];
			hexChars[j * 2 + 1] = hexArray[v & 0x0F];
		}
		return new String(hexChars);
	}

	public static byte[] hexStringToByteArray(String s) {
		int len = s.length();
		byte[] data = new byte[len / 2];
		for (int i = 0; i < len; i += 2) {
			data[i / 2] = (byte) ((Character.digit(s.charAt(i), 16) << 4)
					+ Character.digit(s.charAt(i+1), 16));
		}
		return data;
	}
```

### e. Giữ session

HTTP Session hoạt động dựa trên nguyên lý như sau

B1. CP thực hiện http request đầu tiên đến API, server chèn header Set-Cookie vào response

B2: Nếu login thành công, CP lưu lại giá trị của header này

B3: Các request sau này từ CP trong cùng 1 phiên, được gắn header Cookie với giá trị trong B2

B4: Phía server sẽ phát hiện ra các request trong cùng một session qua header trên

```
/*Ví dụ Java, sử dụng java.net.**/
URL url = new URL(api_url);
URLConnection conn = url.openConnection();
/* xử lý login request*/
String cookie = conn.getHeaderField("Set-Cookie");

/* xử lý send sms request*/
URL url = new URL(api_url);
URLConnection conn = url.openConnection();
conn.setRequestProperty("Cookie", cookie);
```

Session có hiệu lực trong 30 phút

### f. Các ký tự escape XML và cách thay thế

| Ký tự | Thay thế |
|---|---|
| & | & |
| < | < |
| > | > |
| ' | ' |
| " | " |
| \r |  |
| \n | 
 |