



Gotadi Webview Integration

Version: 1.5

Lịch sử thay đổi

Phiên bản	Ngày phát hành	Thay đổi
1.0	23-Apr-2019	Version đầu tiên
1.5	11-Sep-2019	Version thứ năm (Update PNR)

Mục lục

I.	Tổng quan.....	3
1.	Mục đích của tài liệu.....	3
2.	Phạm vi sử dụng	3
3.	Tài liệu liên quan	3
4.	Thuật ngữ và viết tắt.....	3
II.	Yêu cầu bảo mật.....	4
1.	Kênh truyền SSL/HTTPS	4
2.	Header bảo mật và thông kê lưu lượng truyền	4
3.	Xác thực mã truy cập.....	4
4.	Xác thực chữ ký điện tử.....	4
5.	Mã hóa dữ liệu truyền.....	4
III.	Quy trình kết nối	6
1.	Quy trình kết nối.....	6
2.	Mô tả luồng dữ liệu.....	7
3.	Mô tả chi tiết kết nối và trao đổi dữ liệu.....	8
a.	Khởi tạo liên kết webview.....	8
b.	URL thanh toán	12
c.	Yêu cầu xuất vé/phòng.....	13
d.	Truy vấn kết quả giao dịch.....	15
IV.	Phụ lục.....	18
1.	Tạo chữ ký bằng RSA-SHA256	18
2.	Verify chữ ký bằng RSA-SHA256	18
3.	Encrypt dữ liệu bằng RSA	18
4.	Decrypt dữ liệu bằng RSA.....	18
5.	Encrypt dữ liệu bằng TripleDes.....	19
6.	Decrypt dữ liệu bằng TripleDes.....	19
7.	Bảng chú thích mã lỗi.....	20
8.	Bảng chú thích thông tin chi tiết của vé máy bay.....	20

9.	Bảng chú thích thông tin chi tiết của phòng khách sạn	21
----	---	----

I. Tổng quan

1. Mục đích của tài liệu

Tài liệu mô tả kết nối Webview giữa đối tác và Gotadi. Cho phép khách hàng có thể hoàn toàn đặt vé/phòng khách sạn của Gotadi và thanh toán trên ứng dụng và website của đối tác.

2. Phạm vi sử dụng

Tài liệu mô tả ngắn gọn mô hình kết nối, sơ đồ tuần tự và các hàm API của các nghiệp vụ liên quan đến kết nối giữa hai hệ thống.

3. Tài liệu liên quan

- Thông tin kết nối của đối tác được Gotadi gửi qua e-mail.
- Kịch bản kiểm tra kết quả kết nối.
- Source code mẫu.

4. Thuật ngữ và viết tắt

STT	Thuật ngữ	Giải thích
1	URL	Uniform Resource: được dùng để tham chiếu tới tài nguyên trên Internet.
2	SSL	Secure Sockets Layer: là các giao thức mật mã được thiết kế để cung cấp truyền thông an toàn qua Internet.
3	HTTPS	Hypertext Transfer Protocol Secure: là một giao thức kết hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet.
4	3DES	Triple DES (3DES hay TDES): là một thuật toán khóa đối xứng, áp dụng thuật toán mã hóa DES ba lần cho mỗi khối dữ liệu.
5	RSA	Rivest–Shamir–Adleman: là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa.
6	SHA-256	Secure Hash Algorithm: là giải thuật dùng để chuyển một đoạn dữ liệu nhất định thành một đoạn dữ liệu có chiều dài

		không đổi với xác suất khác biệt cao. SHA-256 (trả lại kết quả dài 256 bit)
7	Chữ ký điện tử	Thông tin đi kèm theo dữ liệu (văn bản, hình ảnh, video...) nhằm mục đích xác định người chủ của dữ liệu đó

II. Yêu cầu bảo mật

1. Kênh truyền SSL/HTTPS

SSL/HTTPS được áp dụng để truyền nhận dữ liệu giữa hệ thống của đối tác và Gotadi. Mục đích sử dụng SSL/HTTPS là giúp dữ liệu trao đổi giữa đối tác và Gotadi được mã hóa, khó bị đánh cắp và giả mạo

2. Header bảo mật và thông kê lưu lượng truyền

Tất cả các request từ phía đối tác gọi sang hệ thống của Gotadi phải chứa các Headers bên dưới để phục vụ các nghiệp vụ về bảo mật và thông kê số liệu của Gotadi:

- **apikey**
- **x-ibe-req-code**
- **x-ibe-req-sale-channel**
- **x-ibe-req-type**
- **x-ibe-req-name**

3. Xác thực mã truy cập

Đối tác/ Gotadi phải cung cấp đúng access_code do bên còn lại cấp cho. Mục đích để xác thực request hợp lệ.

4. Xác thực chữ ký điện tử

Khi Gotadi và đối tác gửi request/response phải luôn kèm theo chữ ký điện tử để yêu cầu xác thực.

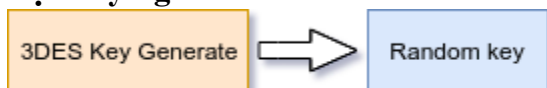
Chi tiết về cách thức tạo dữ liệu ký và ký dữ liệu sẽ được mô tả với mỗi request/response tại ([Mô tả chi tiết kết nối và trao đổi dữ liệu](#))

5. Mã hóa dữ liệu truyền

Dữ liệu truyền phải được mã hóa để bảo vệ thông tin giao dịch của khách hàng.

Mô tả thuật toán mã hóa:

- Tạo key ngẫu nhiên



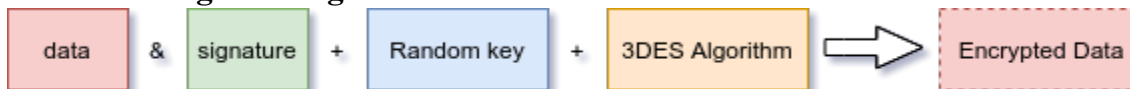
3DES Key generate là hàm tạo key ngẫu nhiên. Mỗi request và response sẽ phải ứng với một 3DES key riêng biệt.

- Ký chữ ký điện tử



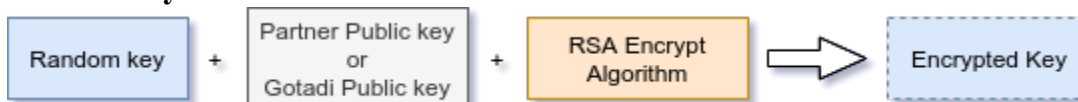
Bên gửi áp dụng thuật toán **RSA-SHA256** kết hợp với **Private key của chính mình** để ký chữ ký điện tử. Xác thực thông tin chính từ bên đối tác gửi sang và không bị thay đổi trên đường truyền.

- Mã hóa thông tin bằng 3DES



Data và chữ ký được mã hóa bằng thuật toán 3DES với key random đã được tạo trước đó. Cách thức tạo chuỗi mã hóa từ data và signature sẽ được mô tả trong bảng tham số của từng request tại [Mô tả chi tiết kết nối và trao đổi dữ liệu](#). (Lưu ý tùy vào API mà vị trí và thứ tự của signature trong chuỗi data sẽ khác nhau)

- Mã hóa key 3DES



Key ngẫu nhiên tạo ra sẽ được mã hóa bằng thuật toán mã hóa bất đối xứng RSA bằng **Public key của bên nhận**.

Mô tả thuật toán giải mã:

- Giải mã 3DES key



Bên nhận sử dụng **Private key của chính mình** để giải mã 3DES key nhận được.

- Giải mã dữ liệu



Bên nhận sử dụng 3DES key vừa giải mã được để giải mã nhận được data và chữ ký điện tử.

- **Xác thực chữ ký điện tử**



Bên nhận sử dụng **Public key của bên gửi** để xác thực chữ ký điện tử bằng thuật toán **RSA-SHA256**.

III. Quy trình kết nối

1. Quy trình kết nối

Là tiến trình đối tác kết nối với hệ thống của Gotadi gồm các bước sau:

Bước 1: Đối tác cung cấp các thông tin bao gồm:

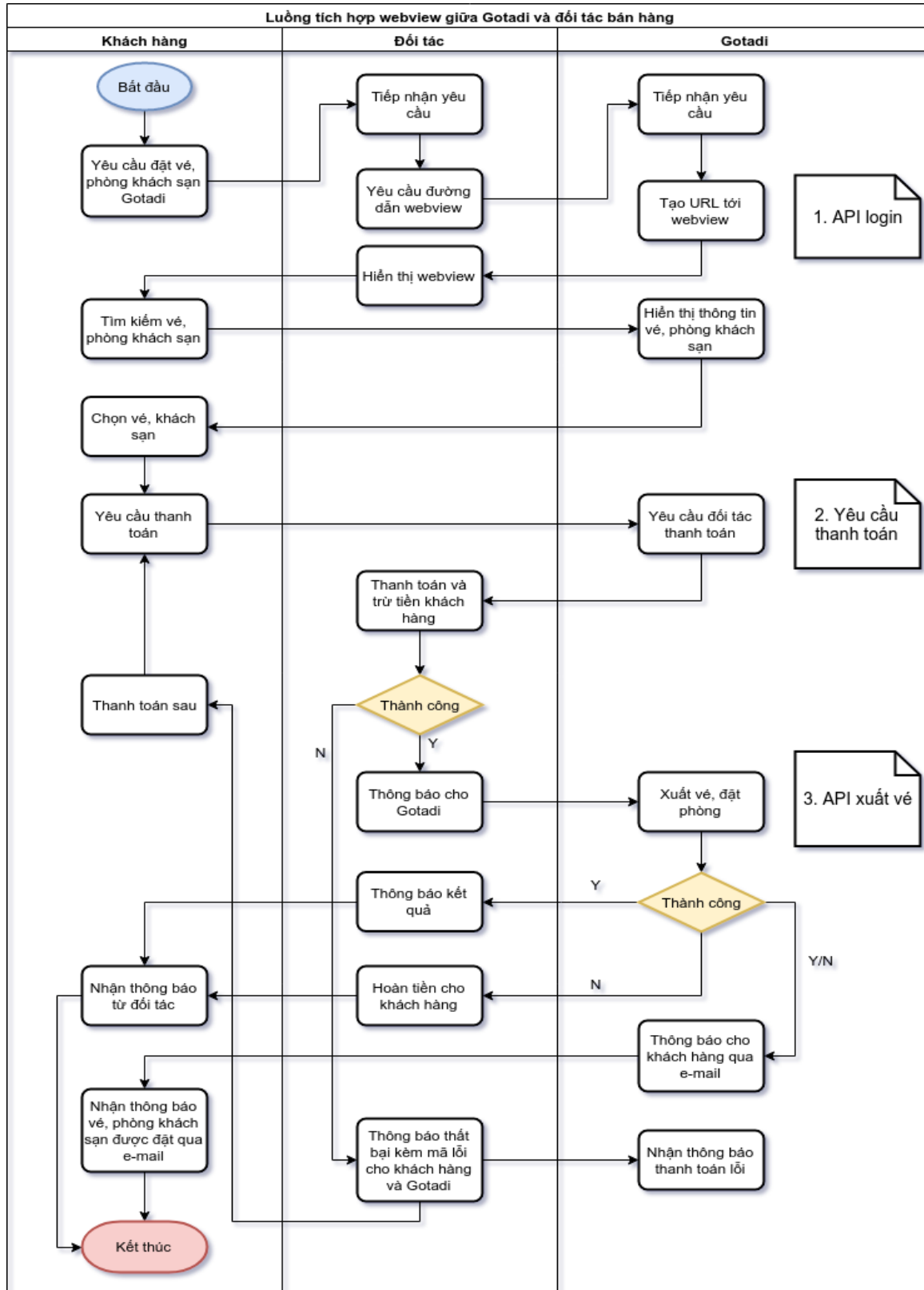
- Thông tin tạo tài khoản:
 - <**Thông tin tạo tài khoản Agency**>
- Thông tin kết nối:
 - Đường dẫn tới hệ thống của đối tác.
 - Public key của đối tác. (RSA key chiều dài tối thiểu **1024**)
 - Mã truy cập (Access code) của đối tác cấp cho Gotadi.

Bước 2: Sau khi nhận được thông tin của đối tác Gotadi sẽ tạo tài khoản và gửi trả các thông tin sau cho đối tác.

- Đường dẫn tới hệ thống của Gotadi.
- Public key của Gotadi. (RSA key chiều dài tối thiểu **1024**)
- Mã truy cập (Access code) của Gotadi cấp cho đối tác.
- Tham số truyền trên các request header:
 - apikey
 - x-ibe-req-code
 - x-ibe-req-sale-channel
 - x-ibe-req-type
 - x-ibe-req-name

Bước 3: Tiến hành kết nối và đối soát theo tài liệu này.

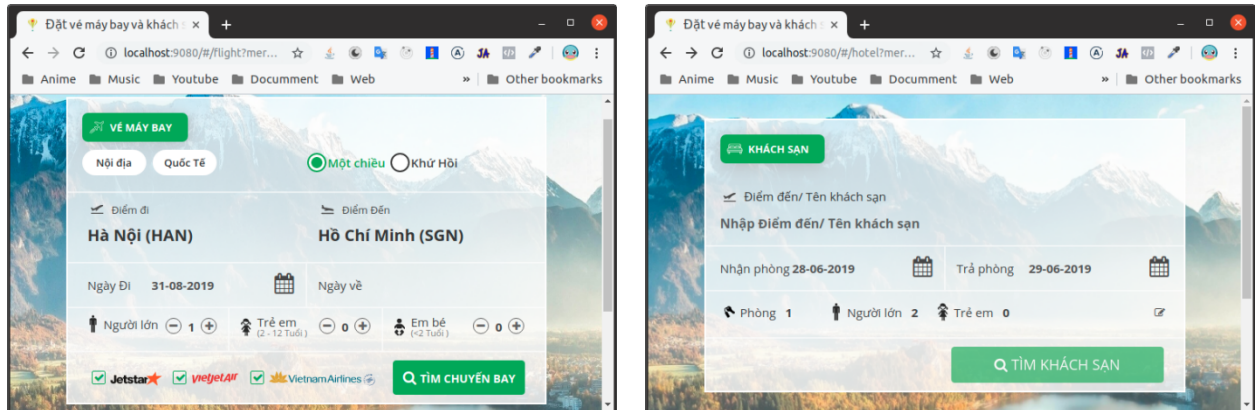
2. Mô tả luồng dữ liệu



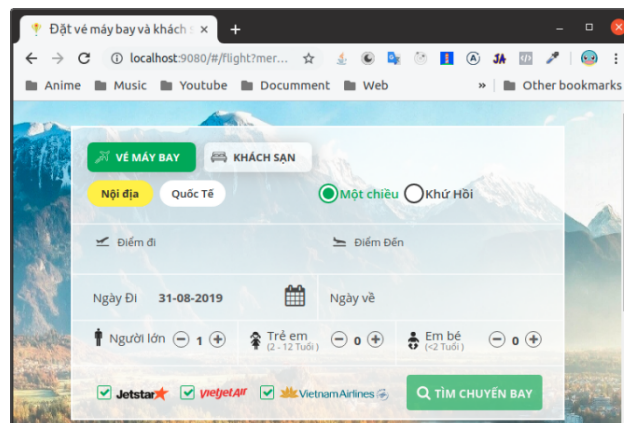
Bố cục giao diện webview

Gotadi cung cấp cho đối tác 2 loại bố cục (layout) bao gồm:

Đơn dịch vụ (single): Giao diện hiển thị đặt vé máy bay và khách sạn sẽ được tách riêng biệt.



Đa dịch vụ (dual): Giao diện hiển thị đặt vé máy bay và khách sạn sẽ hiển thị chung.



3. Mô tả chi tiết kết nối và trao đổi dữ liệu

a. Khởi tạo liên kết webview

- Mô tả: Gotadi cung cấp API cho phép đối tác yêu cầu đường dẫn tới webview của Gotadi.
- URL : https://<gotadi_gateway>/api/partner/login
- Phương thức : POST
- Content Type : application/json

Đặc tả dữ liệu đầu vào

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập Gotadi cấp cho đối tác.	M
user_id	String	Mã định danh khách hàng duy nhất của đối tác.	M
layout	String	Bao gồm các giá trị dual và single tương ứng với 2 loại layout Gotadi cung cấp cho đối tác.	M
product	String	Bao gồm các giá trị hotel và flight . Tương ứng với loại dịch vụ mà Gotadi cung cấp cho đối tác. Dùng để phân biệt dịch vụ hiển thị khi đối tác yêu cầu layout là single . Trong trường hợp layout dual dùng để mặc định focus một dịch vụ cụ thể.	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của đối tác. Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code user_id single_view product	M

Đặc tả mã hóa dữ liệu (Dữ liệu đối tác POST)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code user_id single_view product signature	M
key	String	Key giải mã data	M

Ví dụ về request Khởi tạo liên kết webview:

POST:	<code>https://<gotadi_gateway>/api/partner/login</code>
-------	---

JSON:	<pre>{ "data": "XXXXXXXXXXXX", "key": "XXXXXXXXXXXX" }</pre>
-------	--

Đặc tả dữ liệu đầu ra

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập đối tác cấp cho Gotadi.	M
error_code	String	Mã lỗi Gotadi trả về. Tham khảo Bảng chú thích mã lỗi	M
redirect_url	String	Đường dẫn tới webview của Gotadi	
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của Gotadi . Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code error_code redirect_url	M

Đặc tả mã hóa dữ liệu (Dữ liệu dạng JSON Gotadi trả về)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code error_code redirect_url signature	M
key	String	Key giải mã data	M

Ví dụ kết quả trả về từ Gotadi

```
{  
  "data": "XXXXXXXXXXXX",  
  "key": "XXXXXXXXXXXX"  
}
```

b. URL thanh toán

- Mô tả: Đối tác cung cấp API để Gotadi redirect yêu cầu thanh toán.
- URL : https://<partner_payment_gateway>/payment
Phương thức : **GET**
Content Type : **Request params**

Đặc tả dữ liệu đầu vào

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập đối tác cấp cho Gotadi.	M
booking_number	String	Mã định danh booking của Gotadi.	M
product_type	String	Mã sản phẩm của Gotadi. AIR: Vé máy bay. HOTEL: Khách sạn	M
total_amount	Double	Tổng số tiền thanh toán.	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của Gotadi . Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code booking_number product_type total_amount	M

Đặc tả mã hóa dữ liệu (Dữ liệu Gotadi redirect sang)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code booking_number product_type signature total_amount	M
key	String	Key giải mã data	M

Ví dụ về request yêu cầu thanh toán:

GET:	<a href="https://<partner_payment_gateway>/payment?data=XXXX&key=XXXXX">https://<partner_payment_gateway>/payment?data=XXXX&key=XXXXX
------	---

c. Yêu cầu xuất vé/phòng

- Mô tả: Gotadi cung cấp API cho phép đối tác yêu cầu xuất vé/phòng đã được đặt.
- URL : https://<gotadi_gateway>/api/partner/commit
Phương thức : **POST**
Content Type : **application/json**

Đặc tả dữ liệu đầu vào

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập Gotadi cấp cho đối tác.	M
booking_number	String	Mã định danh booking của Gotadi.	M
partner_trans_id	String	Mã định danh giao dịch của đối tác. Sử dụng để đối soát với hệ thống của Gotadi. Nếu đối tác không truyền mã giao dịch mặc định giá trị sẽ được gán bằng booking_number	O
product_type	String	Mã sản phẩm của Gotadi. AIR: Vé máy bay. HOTEL: Khách sạn	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của đối tác . Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code booking_number partner_trans_id product_type	M

Đặc tả mã hóa dữ liệu (Dữ liệu đối tác POST)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code booking_number partner_trans_id product_type signature	M
key	String	Key giải mã data	M

Ví dụ về request yêu cầu xuất vé/phòng:

POST:	https://<gotadi_gateway>/api/partner/commit
JSON:	<pre>{ "data": "XXXXXXXXXXXX", "key": "XXXXXXXXXXXX" }</pre>

Đặc tả dữ liệu đầu ra

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập đối tác cấp cho Gotadi.	M
booking_number	String	Mã định danh booking của Gotadi.	M
error_code	String	Mã lỗi Gotadi trả về. Tham khảo Bảng chú thích mã lỗi	M
product_type	String	Mã sản phẩm của Gotadi. AIR: Vé máy bay. HOTEL: Khách sạn	M
properties	JSON String	Thông tin chi tiết của đơn hàng. (Bảng chú thích thông tin chi tiết của vé máy bay , Bảng chú thích thông tin chi tiết của phòng khách sạn)	M
return_url	String	Trang hiển thị kết quả giao dịch của Gotadi. Nếu đối tác không xây dựng trang kết quả có thể sử dụng để hiển thị kết quả.	O
total_amount	Double	Tổng số tiền thanh toán.	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của Gotadi. Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code booking_number error_code product_type properties return_url total_amount	M

Đặc tả mã hóa dữ liệu (Dữ liệu dạng JSON Gotadi trả về)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code booking_number error_code product_type properties return_url signature total_amount	M
key	String	Key giải mã data	M

Ví dụ kết quả trả về từ Gotadi

<pre>{ "data": "XXXXXXXXXXXX", "key": "XXXXXXXXXXXX" }</pre>
--

d. Truy vấn kết quả giao dịch

- Mô tả: Gotadi cung cấp API cho phép đối tác yêu cầu truy vấn kết quả giao dịch trong trường hợp API [Yêu cầu xuất vé/phòng](#) bị lỗi timeout.
- URL : https://<gotadi_gateway>/api/partner/query-trans
 Phương thức : **POST**
 Content Type : **application/json**

Đặc tả dữ liệu đầu vào

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập Gotadi cấp cho đối tác.	M
booking_number	String	Mã định danh booking của Gotadi.	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của đối tác.	M

		Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code booking_number	
--	--	---	--

Đặc tả mã hóa dữ liệu (Dữ liệu đối tác POST)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code booking_number signature	M
key	String	Key giải mã data	M

Ví dụ về request Truy vấn kết quả giao dịch:

POST:	https://<gotadi_gateway>/api/partner/query-trans
JSON:	<pre>{ "data": "XXXXXXXXXXXX", "key": "XXXXXXXXXXXX" }</pre>

Đặc tả dữ liệu đầu ra

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
access_code	String	Mã truy cập đối tác cấp cho Gotadi.	M
booking_number	String	Mã định danh booking của Gotadi.	M
error_code	String	Mã lỗi Gotadi trả về. Tham khảo Bảng chú thích mã lỗi	M
product_type	String	Mã sản phẩm của Gotadi. AIR: Vé máy bay. HOTEL: Khách sạn	M
properties	JSON String	Thông tin chi tiết của đơn hàng. (Bảng chú thích thông tin chi tiết của vé máy)	M

		bay, Bảng chú thích thông tin chi tiết của phòng khách sạn)	
return_url	String	Trang hiển thị kết quả giao dịch của Gotadi. Nếu đối tác không xây dựng trang kết quả có thể sử dụng để hiển thị kết quả.	O
total_amount	Double	Tổng số tiền thanh toán.	M
signature	String	Chữ ký điện tử được tạo ra từ chuỗi dữ liệu ký và Private Key của Gotaid. Chuỗi dữ liệu ký được tạo ra theo thứ tự sau trước khi ký: access_code booking_number error_code product_type properties return_url total_amount	M

Đặc tả mã hóa dữ liệu (Dữ liệu dạng JSON Gotadi trả về)

Tham số	Kiểu dữ liệu	Mô tả	Yêu cầu
data	String	Chuỗi dữ liệu mã hóa được tạo ra từ chuỗi dữ liệu truyền và thuật toán mã hóa 3DES. Chuỗi dữ liệu được tạo theo thứ tự bên dưới trước khi mã hóa: access_code booking_number error_code product_type properties return_url signature total_amount	M
key	String	Key giải mã data	M

Ví dụ kết quả trả về từ Gotadi

<pre>{ "data": "XXXXXXXXXXXX", "key": "XXXXXXXXXXXX" }</pre>
--

IV. Phụ lục

1. Tạo chữ ký bằng RSA-SHA256

```
public static String signRSA(String data, String xmlPrivateKey) throws Exception {
    PrivateKey privKey = getPrivateKeyFromXML(xmlPrivateKey);
    Signature instance = Signature.getInstance("SHA256withRSA");
    instance.initSign(privKey);
    instance.update(data.getBytes("UTF-8"));
    byte[] signature = instance.sign();
    // convert Byte to Base64
    return Base64.encodeBase64String(signature);
}
```

2. Verify chữ ký bằng RSA-SHA256

```
public static boolean verifyRSA(String data, String signature, String xmlPublicKey) throws Exception {
    PublicKey pubKey = getPublicKeyFromXML(xmlPublicKey);
    Signature instance = Signature.getInstance("SHA256withRSA");
    instance.initVerify(pubKey);
    instance.update(data.getBytes("UTF-8"));
    return instance.verify(Base64.decodeBase64(signature));
}
```

3. Encrypt dữ liệu bằng RSA

```
public static String encryptRSA(byte[] byteData, String xmlPublicKey) throws Exception {
    Cipher cipher = createCipherEncrypt(xmlPublicKey);
    byte[] encrypted = cipher.doFinal(byteData);
    return Base64.encodeBase64URLSafeString(encrypted);
}
```

```
private static Cipher createCipherEncrypt(String xmlPublicKey)
    throws Exception {
    PublicKey publicKey = getPublicKeyFromXML(xmlPublicKey);
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.ENCRYPT_MODE, publicKey);
    return cipher;
}
```

4. Decrypt dữ liệu bằng RSA

```
public static byte[] decryptRSAToByte(String encrypted, String xmlPrivateKey) throws Exception {
    Cipher cipher = createCipherDecrypt(xmlPrivateKey);
    byte[] bts = Base64.decodeBase64(encrypted);
    byte[] decrypted = cipher.doFinal(bts);
    return decrypted;
}
```

```

private static Cipher createCipherDecrypt(String xmlPrivateKey)
    throws Exception {
    PrivateKey privateKey = getPrivateKeyFromXML(xmlPrivateKey);
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.DECRYPT_MODE, privateKey);
    return cipher;
}

```

5. Encrypt dữ liệu bằng TripleDes

```

public static String encryptTripleDes(String data, byte[] key) throws Exception {
    // Create and initialize the encryption engine
    Cipher cipher = Cipher.getInstance("DESede");
    SecretKeySpec keyspec = new SecretKeySpec(key, "DESede");
    cipher.init(Cipher.ENCRYPT_MODE, keyspec);
    byte[] encBytes = cipher.doFinal(data.getBytes("UTF-8"));
    return Base64.encodeBase64URLSafeString(encBytes);
}

```

6. Decrypt dữ liệu bằng TripleDes

```

public static String decryptTripleDes(String data, byte[] key)
    throws Exception {
    // Create and initialize the encryption engine
    Cipher cipher = Cipher.getInstance("DESede");
    SecretKeySpec keyspec = new SecretKeySpec(key,
        "DESede");
    cipher.init(Cipher.DECRYPT_MODE, keyspec);
    byte[] decBytes = cipher.doFinal(Base64.decodeBase64(data));
    return new String(decBytes, "UTF-8");
}

```

7. Bảng chú thích mã lỗi

Mã lỗi	Mô tả
00	Giao dịch thành công.
01	Vé/phòng đang chờ xử lý.
02	Xuất vé/phòng thất bại.
03	Xác thực link webview thất bại.
04	Chữ ký điện tử không hợp lệ.
05	Giải mã 3DES thất bại.
06	Mã xác thực (Access Code) không hợp lệ.
07	Dữ liệu sai định dạng.
08	Giao dịch đã được thanh toán.
99	Lỗi khác.

8. Bảng chú thích thông tin chi tiết của vé máy bay

Vé máy bay	
supplier_code	Mã hãng bay
supplier_name	Tên hãng bay
from_location_code	Mã sân bay khởi hành
from_location_name	Tên sân bay khởi hành
to_location_code	Mã sân bay đến
to_location_name	Tên sân bay đến
customer_name	Tên hành khách. (Họ/Tên)
booking_type	Loại vé nội địa hay quốc tế (DOME/INTE)
pnr	Mã đặt chỗ
round_type	Loại vé một chiều hay khứ hồi (OneWay, RoundTrip)

departure_date	Ngày khởi hành (định dạng ISO datetime: YYYY-MM-DDThh:mm:ssZ)
return_date	Ngày về (định dạng ISO datetime: YYYY-MM-DDThh:mm:ssZ)
total_infant	Tổng số trẻ sơ sinh
total_child	Tổng số trẻ em
total_adult	Tổng số người lớn

9. Bảng chú thích thông tin chi tiết của phòng khách sạn

Phòng khách sạn	
hotel_name	Tên khách sạn
customer_name	Tên khách hàng (Họ/Tên)
departure_date	Ngày khởi hành (định dạng ISO datetime: YYYY-MM-DDThh:mm:ssZ)
return_date	Ngày về (định dạng ISO datetime: YYYY-MM-DDThh:mm:ssZ)