



# Kigo

## RENTAL AGENCY REST API - V1

### API revision

12

### Document revision

18

### Contact

Vedad KAJTAZ  
vedad@kigo.net

## 1 Revision history

API Rev	Document		Date	Changes
	Rev	Status		
1	1	DRAFT	04 May 2011	First implementation DRAFT.
2	2		28 Oct 2011	First production release. Added UDRA support.
3	3		08 Nov 2011	Added <code>RES_COMMENT_GUEST</code> to <code>createConfirmedReservation</code> , <code>updateReservation</code> and <code>readReservation</code> .
4	4	DRAFT	29 Nov 2011	Added <code>API_DEPRECATED</code> to the standard method reply object. Added Bandwidth limiting policy. Added <code>listProperties</code> , <code>readProperty</code> , <code>readPropertyPhotoFile</code> , <code>listKigoPropertyTypes</code> , <code>listKigoPropertyBedTypes</code> , <code>listKigoPropertyAmenities</code> and <code>listKigoPropertyActivities</code> methods. Renamed <code>listCountries</code> to <code>listKigoCountries</code> .
4	5		07 Dec 2011	Added <code>computePricing</code> and <code>listUserDefinedPropertyAttributes</code> methods. Added UDPA support to <code>readProperty</code> .
4	6	ERRATA	09 Dec 2011	Longitude and latitude information were missing in Reading a property example. <code>E_INPUT</code> result was undocumented in <code>cancelReservation</code> , <code>readReservation</code> , <code>readProperty</code> and <code>readPropertyPhotoFile</code> methods. Seconds were missing from the <code>time</code> extended data type. Restrictions on <code>RES_CREATE</code> input object member of the <code>computePricing</code> method were not documented.
5	7		19 Dec 2011	Rent fees member was added to the <code>API_REPLY</code> object of <code>computePricing</code> method.
5	8	ERRATA	21 Dec 2011	<code>CURRENCY</code> member was missing from <code>API_REPLY</code> object of <code>computePricing</code> method.
6	9		05 Jan 2012	Added new members to the <code>readProperty</code> method <code>API_REPLY</code> object.
7	10		25 Jan 2012	Added new members to the <code>listProperties</code> method <code>API_REPLY</code> object.
8	11		20 Feb 2012	<code>computePricing</code> method <code>API_REPLY</code> object changes.
9	12		30 Apr 2012	Added new members to the <code>listProperties</code> and <code>readProperty</code> methods <code>API_REPLY</code> object.
10	13		02 Oct 2012	Added <code>RES_GUEST</code> object to <code>createConfirmedReservation</code> and <code>updateReservation</code> .
10	14		24 Jan 2013	Removed references to the insecure <code>http://</code> transport.
11	15		20 March 2013	Added <code>computePricingBulk</code> method.



API Rev	Document		Date	Changes
	Rev	Status		
12	16	DRAFT	07 June 2013	Added diffPropertyCalendarReservations, readPropertyPricingSetup, diffPropertyPricingSetup, updatePropertyPricingSetup and listKigoFeeTypes methods. Deprecated listPropertyCalendarReservations. Deprecated/renamed several reply object members from readProperty, computePricing and computePricingBulk.
12	17		22 July 2013	Minor changes.
12	18	ERRATA	18 September 2013	Fixed <code>time_interval</code> documentation.

## 2 Contents

1	<a href="#">Revision history</a>	2
2	<a href="#">Contents</a>	4
3	<a href="#">Glossary</a>	6
4	<a href="#">Preface</a>	7
	4.1 <a href="#">Intended audience</a>	7
	4.2 <a href="#">Reporting issues</a>	7
5	<a href="#">API version and revision numbers</a>	8
	5.1 <a href="#">API version number</a>	8
	5.2 <a href="#">API revision number</a>	8
6	<a href="#">Implementation details</a>	9
	6.1 <a href="#">Communication protocol</a>	9
	6.2 <a href="#">Authentication</a>	9
	6.3 <a href="#">Source address filtering</a>	9
	6.4 <a href="#">Rate limiting policy</a>	9
	6.5 <a href="#">Bandwidth limiting policy</a>	9
	6.6 <a href="#">Character encoding</a>	10
	6.7 <a href="#">Data encoding</a>	10
	6.8 <a href="#">Data types</a>	10
7	<a href="#">Invoking API methods</a>	12
	7.1 <a href="#">HTTP request</a>	12
	7.2 <a href="#">HTTP reply</a>	12
8	<a href="#">diff methods family</a>	15
	8.1 <a href="#">How does it work?</a>	15
	8.2 <a href="#">Examples</a>	15
	8.3 <a href="#">Diff method-specific rate limiting policy</a>	16
9	<a href="#">API methods reference</a>	17
	9.1 <a href="#">ping</a>	17
	9.2 <a href="#">diffPropertyCalendarReservations</a>	18
	9.3 <a href="#">createConfirmedReservation</a>	19
	9.4 <a href="#">updateReservation</a>	22
	9.5 <a href="#">cancelReservation</a>	24
	9.6 <a href="#">readReservation</a>	25
	9.7 <a href="#">listUserDefinedReservationAttributes</a>	30
	9.8 <a href="#">readPropertyPricingSetup</a>	32
	9.9 <a href="#">diffPropertyPricingSetup</a>	32
	9.10 <a href="#">updatePropertyPricingSetup</a>	34
	9.11 <a href="#">computePricing</a>	34
	9.12 <a href="#">computePricingBulk</a>	37
	9.13 <a href="#">listProperties</a>	38
	9.14 <a href="#">listUserDefinedPropertyAttributes</a>	39
	9.15 <a href="#">readProperty</a>	41
	9.16 <a href="#">readPropertyPhotoFile</a>	47
	9.17 <a href="#">listKigoCountries</a>	47
	9.18 <a href="#">listKigoPropertyTypes</a>	48
	9.19 <a href="#">listKigoPropertyBedTypes</a>	48
	9.20 <a href="#">listKigoPropertyAmenities</a>	49
	9.21 <a href="#">listKigoPropertyActivities</a>	50
	9.22 <a href="#">listKigoFeeTypes</a>	51
10	<a href="#">Deprecated API methods</a>	52
	10.1 <a href="#">listPropertyCalendarReservations</a>	52
	10.2 <a href="#">listCountries</a>	52
11	<a href="#">Kigo data dictionary</a>	53

11.1	<u>Reservation status (RES_STATUS)</u> .....	53
11.2	<u>Online booking transaction state (OB_STATE)</u> .....	53
11.3	<u>Country ISO 3166-1 alpha-2 codes</u> .....	53
11.4	<u>Property provider</u> .....	54
11.5	<u>Property type</u> .....	54
11.6	<u>Property bed type</u> .....	54
11.7	<u>Property amenity</u> .....	54
11.8	<u>Property activity</u> .....	54
11.9	<u>Property pricing</u> .....	54
12	<u>Property pricing computation algorithm overview</u> .....	67
12.1	<u>Input parameters</u> .....	67
12.2	<u>Rounding</u> .....	67
12.3	<u>Computing base rent</u> .....	67
12.4	<u>Computing fees included-in-rent</u> .....	68
12.5	<u>Computing fees not-included-in-rent</u> .....	68
12.6	<u>Computing discounts amount</u> .....	68
12.7	<u>Computing the total rent amount</u> .....	69
12.8	<u>Computing the total due amount</u> .....	69
12.9	<u>Computing the deposit amount</u> .....	69
13	<u>References</u> .....	70
13.1	<u>ISO 3166</u> .....	70
13.2	<u>JSON</u> .....	70
13.3	<u>base64</u> .....	70
13.4	<u>Latitude and longitude WGS 84 decimal degree notation</u> .....	70

### 3 Glossary

<b>API</b>	Application Programming Interface. In this context, a library of available HTTP request/response messages to interact with an application.
<b>HTTP</b>	Hypertext Transfer Protocol (HTTP), the foundation of data communication for the World Wide Web, is a networking protocol for distributed, collaborative, hypermedia information systems.
<b>HTTPS</b>	Hypertext Transfer Protocol Secure, a combination of the HTTP and the SSL/TLS protocols to provide encrypted communication and secure identification over the network.
<b>ISO</b>	International Organization for Standardization, an international organization that promulgates worldwide proprietary industrial and commercial standards.
<b>JPEG</b>	Widely used compression algorithm for digital photography.
<b>JSON</b>	Javascript Object Notation, a simple machine-readable data-interchange format.
<b>REST</b>	Representational State Transfer. A client-server architectural pattern.
<b>RFC</b>	Request for Comments, a memorandum published by the Internet Engineering Task Force describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems.
<b>Unicode</b>	Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems.
<b>UTF-8</b>	Universal Character Set Transformation Format, a widely-used multibyte character encoding for Unicode.
<b>WGS 84</b>	Latest revision of the World Geodetic System, the reference coordinate system used by the Global Positioning System (GPS). All geographic coordinates in Kigo API are WGS 84 coordinates.
<b>XML</b>	Extensible Markup Language, a set of rules for encoding documents in machine-readable form.



## 4 Preface

The Kigo Rental Agency REST API provides automated access from remote computer systems to a set of the Kigo application features.

These features include creating, modifying and canceling reservations, querying Kigo availability calendars and retrieving information about Kigo properties.

For instance, this API may be used to:

- Keep your application or website calendars synchronized with Kigo's calendars
- Create reservations real-time
- Import property information and photos from Kigo database into your application or website.

### 4.1 Intended audience

This documentation is aimed at the IT department of Kigo customers building applications or websites that need to interact with the Kigo application through the Kigo Rental Agency REST API.

### 4.2 Reporting issues

Please visit our support center at <https://support.kigo.net> for any API related questions and issues.



## 5 API version and revision numbers

This is the documentation for Rental Agency REST API version 1, revision 12.

### 5.1 API version number

Backward compatibility from the technical point of view is guaranteed for all API revisions within the same API version number.

A change in the API version number implies that the new version is no longer backward compatible with the previous versions.

While we do our best to continue supporting old versions of API, it may be sometimes necessary, due to constant evolution of the application, to end the support for those versions, and ask our customers to upgrade their applications to use the latest version of the API.

### 5.2 API revision number

The API revision number changes each time the API is updated in such a way that the backward compatibility with previous revisions of the same API version is maintained.

New API methods may be added, and existing methods may become deprecated but still supported.



## 6 Implementation details

### 6.1 Communication protocol

The Kigo Rental Agency REST API is a REST-like implementation based on the HTTP protocol, JSON data encoding and Unicode character encoding, over the secure HTTPS transport.

The insecure HTTP transport was discontinued in January 2013.

### 6.2 Authentication

When invoking API methods, clients must authenticate using Kigo Rental Agency account username and an API specific password, which may differ from the account password.

The Kigo Rental Agency REST API implements HTTP Basic Authentication (RFC 2617).

### 6.3 Source address filtering

In addition to providing credentials, the security may be enforced by setting up source IP address filtering.

The filter may contain individual IP addresses (e.g., `213.186.44.105`) and sub-networks (e.g., `213.186.44.0/24`).

Please contact us for setting up IP address filter for your Rental Agency REST API access. Only IPv4 addresses are currently supported.

### 6.4 Rate limiting policy

All API methods are bound to the per-account rate limiting policy. The rate limiting policy helps protect Kigo servers from being flooded by requests from misbehaving third-party applications.

The rate limiting policy limits the number of allowed API method calls during a given amount of time. When the limit is exceeded, the Kigo API server enters a self-protection mode where it refuses to serve new requests (see § 7.2.1 Status code) during a couple of seconds, and until the number of requests per period goes below the limit.

The rate limiting policy is set up on per-customer basis, in such a way that it should never be hit under normal operating conditions. If you feel the limit is set too low, or if you anticipate a significant increase of API usage, please contact us.

### 6.5 Bandwidth limiting policy

All API methods are bound to the per-account bandwidth limiting policy. The bandwidth limiting policy helps protect Kigo servers from excessive bandwidth usage triggered by third-party applications.

The bandwidth limiting policy limits the amount of data the API methods may serve during a given amount of time. When the limit is exceeded, the Kigo API server enters a self-protection mode where it refuses to serve new requests (see § 7.2.1 Status code) during a couple of seconds, and until the bandwidth usage within the defined period goes under the limit.

While the diff methods (see 8 diff methods family) are, as all API methods, subject to the bandwidth limit policy, the output from these methods is not counted against bandwidth usage.

The bandwidth limiting policy is set up on a per-customer basis, in such a way that it should not be hit by carefully designed third-party application under normal operating conditions.

Third-party application need to throttle bandwidth-intensive API method calls (for instance,

readPropertyPhotoFile may return huge amount of data) in order not to trigger the bandwidth limiting policy.

## 6.6 Character encoding

The API uses UTF-8 character encoding, the most widely used multibyte character encoding for Unicode.

## 6.7 Data encoding

All exchanges are encoded in the JSON (JavaScript Object Notation) format (RFC 4627).

JSON is a simple machine-readable data-interchange format that provides light-weight serialization for structured data. Despite the acronym, JSON is language independent, and is widely implemented in modern programming languages.

JSON is able to directly represent the most general computer science data structures (see § 6.8.1 JSON native data types).

Compared to XML, JSON has a significantly lower footprint, is more readable and more easily parsed.

For more information about JSON, please refer to <http://www.json.org/>.

## 6.8 Data types

In this API, we benefit from all native data types defined by the JSON format, and we also define other commonly used data types. Here's the exhaustive list of data types referenced in this documentation.

### 6.8.1 JSON native data types

Type	Description
<code>string</code>	A sequence of zero or more UTF-8 characters.
<code>number</code>	An integer or floating point number.
<code>object</code>	An unordered collection of (unique key, value) pairs. Those unique keys are also referred to as "object members".
<code>array</code>	An ordered and zero-based indexed sequence of values of any JSON type.
<code>bool</code>	Boolean, may be either <code>true</code> or <code>false</code> .
<code>null</code>	The null value is a primitive value that represents the empty, null reference.

### 6.8.2 Extended data types

Other non-primitive data types are used throughout this documentation. These are data abstractions with more specific, predefined characteristics.

Type	Description
<code>int</code>	Positive integer value represented as <code>number</code> . Range: 0 – 2147483647.
<code>string(X)</code>	<code>string</code> having exactly X characters.
<code>string(X, Y)</code>	<code>string</code> having X up to Y characters (both inclusive).
<code>date</code>	<code>string(10)</code> containing date representation in "YYYY-MM-DD" format. The default range "2000-01-01" – "2029-12-31" may be further restricted by individual methods.
<code>time</code>	<code>string(8)</code> containing daytime representation in 24-hour "HH:MM:SS" format.

<code>amount</code>	<p>In order to avoid cross-platform compatibility issues related to handling of floating point numbers, amounts in Kigo API are represented as <code>string</code> rather than <code>number</code> data types.</p> <p>Therefore, an <code>amount</code> is a <code>string</code> containing representation of a decimal number with:</p> <ul style="list-style-type: none"> <li>• optional minus sign, followed by</li> <li>• one or more digits (if there is more than one digit, then the first one must be different from zero), followed by</li> <li>• the decimal point, followed by</li> <li>• exactly two digits.</li> </ul> <p>Examples of valid <code>amount</code> values:</p> <ul style="list-style-type: none"> <li>• <code>"0.00"</code></li> <li>• <code>"-3.00"</code></li> <li>• <code>"10.49"</code></li> </ul>
<code>decimal_degree</code>	<p>Decimal degrees express latitude and longitude geographic coordinates (see 13.4 Latitude and longitude WGS 84 decimal degree notation).</p> <p>In order to avoid cross-platform issues related to handling of floating point numbers, decimal degrees in Kigo API are represented as <code>string</code> rather than <code>number</code> data types.</p> <p>Therefore, a <code>decimal_degree</code> is a <code>string</code> containing representation of a decimal number with:</p> <ul style="list-style-type: none"> <li>• optional minus sign, followed by</li> <li>• one to three digits (if there is more than one digit, then the first one must be different from zero), followed by</li> <li>• the decimal point, followed by</li> <li>• exactly six digits.</li> </ul> <p>Examples of valid <code>decimal_degree</code> values:</p> <ul style="list-style-type: none"> <li>• <code>"0.000000"</code></li> <li>• <code>"-80.798512"</code></li> <li>• <code>"178.120000"</code></li> </ul> <p>The value range of <code>decimal_degree</code> is <code>"-90.000000"</code> to <code>"90.000000"</code> (both inclusive) for latitudes and <code>"-179.999999"</code> to <code>"180.000000"</code> (both inclusive) for longitudes.</p>
<code>time_interval</code>	<p>object containing exactly two members, <code>UNIT</code> (<code>string</code>) and <code>NUMBER</code> (<code>int</code>).</p> <p><code>UNIT</code> may hold one of the following values: <code>"NIGHT"</code>, <code>"MONTH"</code> or <code>"YEAR"</code>.</p> <p>The allowed ranges for <code>NUMBER</code> are:</p> <ul style="list-style-type: none"> <li>• 1 – 27 for <code>"NIGHT"</code> <code>UNIT</code></li> <li>• 1 – 11 for <code>"MONTH"</code> <code>UNIT</code></li> <li>• 1 – 5 for <code>"YEAR"</code> <code>UNIT</code></li> </ul>

## 7 Invoking API methods

API methods are invoked by making an HTTPS request on the Kigo Rental Agency REST API servers.

### 7.1 HTTP request

In order to invoke an API method, the client makes an HTTPS POST request on

<https://app.kigo.net/api/ra/v1/method>,

where `method` is a unique case-sensitive string that identifies the desired API method.

Basic HTTP authentication must be used in every request.

The request `Content-Type` header must be set to `application/json` and the post data (HTTP request body) must hold a JSON representation of input data.

Example of a valid HTTP request:

```
POST /api/ra/v1/ping HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXN1cm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

{
  "PING" : "PONG"
}
```

Note that the raw post data size may not exceed 1 megabyte.

### 7.2 HTTP reply

Each request results in an HTTP reply being returned to the client.

The HTTP reply contains regular HTTP reply headers, an HTTP status code, and an optional HTTP content body.

The HTTP status code must be examined in order to determine whether the API method was successfully invoked, or if a problem occurred (such as failed authentication, malformed message, ...).

### 7.2.1 Status code

Code	Description
200	Service was successfully invoked, process content body for further information (success/failure, ...).
301	The URL of the service has changed permanently. The client should transparently follow the redirection. Note that Kigo will inform all API users whenever the API method URL changes (by providing latest revision of this document), along with setting up this redirection.
400	Bad (malformed) HTTP request (according to HTTP and/or these API specifications).
401	Bad credentials (i.e. missing or invalid username and/or password).
403	Access refused (i.e. access from unauthorized IP address or network).
404	Unknown API method.
405	Method not allowed (POST method expected).
409	The API rate limiting policy and/or bandwidth limiting policy were triggered (see § 6.4 Rate limiting policy and § 6.5 Bandwidth limiting policy). Client should wait (throttle) before retrying. The 409 status code will be replaced by the 429 code in a future API revision. The client must expect and handle both 409 and 429 codes.
410	The requested version of the API is no longer available.
413	Request entity too large (i.e. raw post data size exceeds the maximum allowed size).
415	Unsupported media type (i.e. the received <code>Content-Type</code> is not <code>application/json</code> , or the content itself is not a valid JSON string).
420	A <code>diff</code> family API method rate limiting policy was triggered (see § 8.3 Diff method-specific rate limiting policy). Client should wait (throttle) before retrying.
429	The API rate limiting policy and/or bandwidth limiting policy were triggered (see § 6.4 Rate limiting policy and § 6.5 Bandwidth limiting policy). Client should wait (throttle) before retrying.
500	The server encountered an unexpected condition which prevented it from fulfilling the request. Client should wait and resubmit the request.
503	The server is currently unable to handle the request due to temporary overloading or maintenance of the server. Client should wait and resubmit the request.

Note that other RFC 2616 compliant HTTP status codes may be returned by the API server or HTTP proxies. Those codes must be interpreted as specified by RFC 2616.

### 7.2.2 Content body

The HTTP reply content body is status dependent.

On successful API method invocation (status 200), the server replies with the Standard method reply object.

Otherwise, the `Content-Type` header is set to `text/plain`, and the content body may contain a human readable message explaining the reason of the failure. The message is a hint for developers, and is not intended to be parsed and understood by software. Developers should however consider logging these messages for further review.

### 7.2.3 Standard method reply object

On successful API method invocation, the server returns 200 HTTP status code and fills the HTTP reply content body with a standard method reply object.

The reply `Content-Type` is set to `application/json`.

The reply object is an `object` holding the following members:

Reply object member	Data type	Description
<code>API_VERSION</code>	<code>int</code>	The API version number.
<code>API_REVISION</code>	<code>int</code>	The API revision number.
<code>API_METHOD</code>	<code>string(1, 50)</code>	The name of the invoked API method.
<code>API_CALL_ID</code>	<code>string(1, 80)</code>	A unique API call (invocation) number. May be used for troubleshooting.
<code>API_RESULT_CODE</code>	<code>string(1, 20)</code>	Result code returned by API method. See API method documentation for further information.
<code>API_RESULT_TEXT</code>	<code>string(0, 1024)</code>	Human readable message related to <code>API_RESULT_CODE</code> , not intended to be parsed and understood by software. May be used for troubleshooting.
<code>API_REPLY</code>		Additional data returned by the API method. In many situations, this value is <code>not-null</code> if <code>API_RESULT_CODE</code> is set to <code>E_OK</code> , and is <code>null</code> otherwise. See API method documentation for further information.
<code>API_DEPRECATED</code> (optional)	<code>string(0, 65535)</code>	This optional member is provided only if the invoked API method is deprecated by the current API revision, or if the API method was invoked with deprecated argument(s). When provided, it holds a human readable message, not intended to be parsed and understood by software. The software should however notify the software maintainers whenever it encounters this member.

The result (`API_RESULT_CODE`) and reply (`API_REPLY`) members being API method specific, please refer to each API method documentation for their possible values and meanings.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

Example of a successful API method invocation, HTTP 200 reply and a standard method reply object in content body:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json
Content-Length: 246

{
  "API_VERSION"      : 1,
  "API_REVISION"    : 12,
  "API_METHOD"      : "cancelReservation",
  "API_CALL_ID"     : "512e84d998f2a06df9ddc10fd7b6d3aa",
  "API_RESULT_CODE" : "E_NOSUCH",
  "API_RESULT_TEXT" : "No such reservation (#11425)",
  "API_REPLY"       : null
}
```

## 8 diff methods family

The `diff` family of methods is designed to provide a list of items (along with optional, method-specific data) that have changed between subsequent invocations of the method related to the same set of data.

They are an alternative to requesting and receiving the full high volume set of data, where the receiving system would have to figure out which items have actually changed since the last update.

The type of data and the nature of reported changes are specific to each method. Please refer to their respective documentation.

Client implementation must expect occasionally receiving items that haven't changed (false positives), or that have reverted back to their initial state after successive changes.

### 8.1 How does it work?

When invoking a `diff` method, the client provides a unique method-specific identifier `DIFF_ID` with a value obtained from a previous method call (or `null` if there was no previous call). Along with the list of changed items, the API server shall provide a new `DIFF_ID` value that may be used in a subsequent call to the method. The previous `DIFF_ID` may still remain available for a short amount of time, so that a clients that experienced a temporary issue (for instance, a network outage while receiving the reply, or a server outage while processing it) may attempt to resume the process (see examples below).

The `DIFF_ID` identifier is volatile by design. While the Kigo API attempts to keep track of two `DIFF_ID` identifiers during a reasonable amount of time, it does not offer any guarantee on their lifetime.

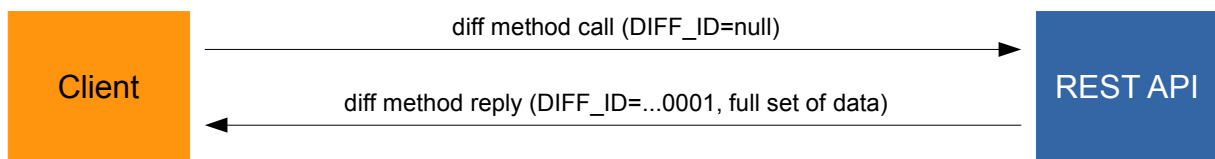
When an invalid or expired `DIFF_ID` is provided to an API method, the full list of items is returned in the result, as if a `null` `DIFF_ID` was provided.

A client implementation that honors the recommended method poll time should rarely or never hit an expired `DIFF_ID`.

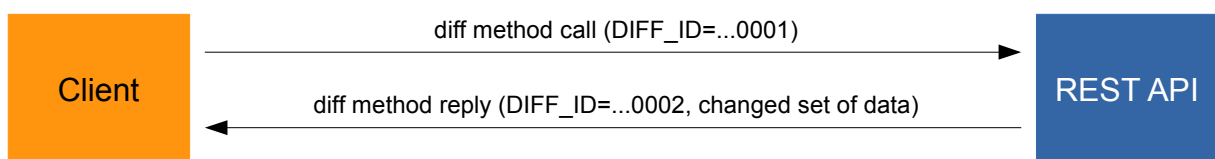
### 8.2 Examples

#### 8.2.1 Initial invocation and a first diff

The client does not have an `DIFF_ID` yet for the specific method and therefore sends a `null` `DIFF_ID`. The full set of data is returned.

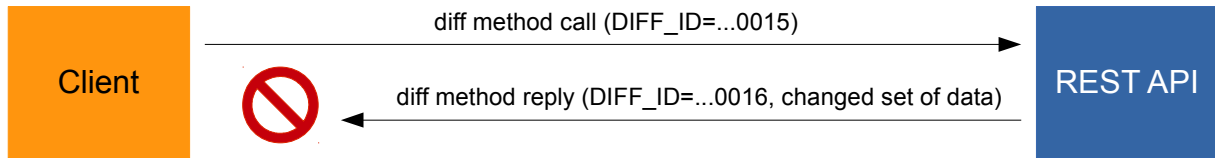


The client makes a new call using the previously obtained `DIFF_ID`.

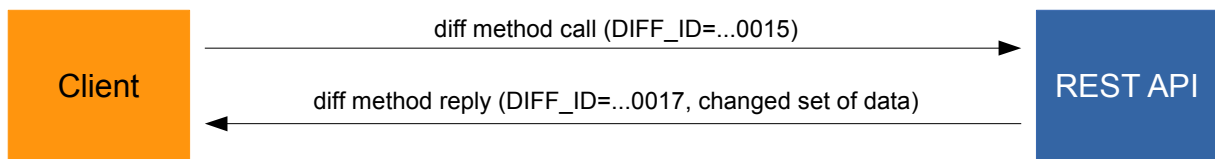


### 8.2.2 Recovering from an error

The client makes a call using the previously obtained `DIFF_ID (...0015)`, but fails to process the reply due to a network outage.



The clients successfully makes a new attempt with the same `DIFF_ID (...0015)`.



### 8.3 Diff method-specific rate limiting policy

Due to `diff` methods implementation being highly resource demanding, each `diff` method is subject to a method-specific rate limiting policy (see HTTP status code 420 in § 7.2.1 Status code).

The recommended poll time for each of the `diff` methods is outlined in their respective documentation. A client implementation that honors the recommended poll time should not trigger this policy.

However, the client application must gracefully handle situations where the limit is being hit.



## 9 API methods reference

This section contains a detailed reference of the API methods provided by this API version.

Most of the usage scenarios will require that your application implement only a subset of the API methods described in this chapter.

### 9.1 ping

#### 9.1.1 Usage

This is a very simple API method that may be used for testing purpose during development process, or for monitoring the Kigo REST API availability once your application is in production. It simply “echoes back” the data it receives in input.

We recommend that you start your development process by developing a call to this method, as it is the simplest possible method.

Like all other API methods, this method requires user authentication, and calls to this method do account for rate and bandwidth limit.

#### 9.1.2 Input

This method accepts data of any type.

#### 9.1.3 Result and reply

This method always returns `E_OK` result code, and `API_REPLY` is always set to data received in input.

#### 9.1.4 Examples

##### 9.1.4.1 Sample ping request

HTTP request:

```
POST /api/ra/v1/ping HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

"Are you there?"
```

HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json
Content-Length: 213

{
  "API_VERSION"       : 1,
  "API_REVISION"     : 12,
  "API_METHOD"       : "ping",
  "API_CALL_ID"      : "f6b22b8e3ab28dbea873ed437c95351d",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        : "Are you there?"
}
```

## 9.2 diffPropertyCalendarReservations

This method is a member of the diff methods family. It obsoletes and replaces the listPropertyCalendarReservations method.

### 9.2.1 Usage

This methods enables the client system to maintain a local copy of the property availability calendar by receiving information on new, updated and canceled reservations that impact the property availability calendar, from three months back from the current date, and up to the very last reservation in the property calendar.

The local calendar copy may then be used for displaying the availability calendar to your website visitors and running availability search queries.

Recommended poll time: every 2 minutes to 4 hours.

### 9.2.2 Input

The method expects an `object` holding the following member:

Member	Data type	Description
DIFF_ID	string(64) or null	See § 8 diff methods family.

### 9.2.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, <code>API_REPLY</code> contains a new <code>DIFF_ID</code> and a list of reservations.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.

On `E_OK` result, `API_REPLY` holds an `object` that contains a new `DIFF_ID` and a list of reservations:

Member	Data type	Description
DIFF_ID	string(64)	See § 8 diff methods family.
RES_LIST	array of object	Unsorted list of new, updated (status or dates) or canceled reservations.

Every `RES_LIST` array item is an `object` holding the following members:

Member	Data type	Description
RES_ID	int	The reservation id.
PROP_ID	int	The reservation property id.
RES_STATUS	string or null	The reservation status ( <code>CONFIRMED</code> or <code>HOLD</code> ) if the reservation has an impact on the property availability calendar, or <code>null</code> if the reservation was canceled or if it is related to a property your account has no longer access to.
RES_IS_FOR	bool	Indicates whether the reservation belongs to your Rental Agency Kigo account ( <code>true</code> ) or if it is an owner's or other agency's reservation ( <code>false</code> ).

RES_CHECK_IN	date or null	The check-in date, or <code>null</code> if the reservation was canceled or if it is related to a property your account has no longer access to.
RES_CHECK_OUT	date or null	The check-out date, or <code>null</code> if the reservation was canceled or if it is related to a property your account has no longer access to.

## 9.3 createConfirmedReservation

### 9.3.1 Usage

This method creates a new, `CONFIRMED` property reservation.

Please note that the method does not:

- enforce the property minimum and maximum stay times,
- enforce the property maximum number of guests.

### 9.3.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
PROP_ID	int	The Kigo property id.
RES_CHECK_IN	date	The reservation check-in date.
RES_CHECK_OUT	date	The reservation check-out date.
RES_N_ADULTS	int	The number of guests (adults), from 1 to 30.
RES_N_CHILDREN	int	The number of guests (children), from 0 to 29.
RES_N_BABIES	int	The number of guests (babies), from 0 to 29.
RES_GUEST (optional)	object	The reservation guest information.  This optional <code>object</code> member was introduced in API revision 10.
RES_COMMENT	string(0, 65535)	The reservation comments.
RES_COMMENT_GUEST (optional)	string(0, 65535)	The guest reservation comment.  This object member is optional (may be omitted).
RES_UDRA (optional)	array	The reservation UDRA (user-defined reservation attributes) values.  This object member is optional (may be omitted).  Values for all attributes are also optional. The <code>array</code> may hold only some (or none) of the attributes defined in your Kigo Rental Agency account.

The following restrictions apply:

- The maximum allowed `RES_CHECK_IN` is the date of the day plus 60 months (5 years)

- The reservation duration may not exceed 36 months (3 years)

`E_INPUT` result code is returned if not all of the above conditions are met.

Each `RES_UDRA` array item is expected to be an object with the following members:

Member	Data type	Description
<code>UDRA_ID</code>	<code>int</code>	The unique attribute identifier.  Nonexistent (e.g. deleted) attributes are silently ignored.  The list of attributes may be retrieved with the <code>listUserDefinedReservationAttributes</code> API call.
<code>UDRA_TEXT</code>	<code>string(0, 200)</code> for <code>SINGLE_LINE_TEXT</code>  <code>string(0, 65535)</code> for <code>MULTI_LINE_TEXT</code>	This type-specific member must be provided for <code>SINGLE_LINE_TEXT</code> and <code>MULTI_LINE_TEXT</code> attributes, and must not be provided for other attribute types.
<code>UDRA_CHOICE_ID</code>	<code>int</code> or <code>null</code>	This type-specific member must be provided for <code>SINGLE_CHOICE</code> attributes, and must not be provided for other attribute types.
<code>UDRA_CHOICES</code>	array of <code>int</code>	This type-specific member must be provided for <code>MULTI_UNORDERED_CHOICE</code> attributes, and must not be provided for other attribute types.

The optional `RES_GUEST` object holds the following members:

Member	Data type	Description
<code>RES_GUEST_FIRSTNAME</code>	<code>string(0, 50)</code>	The guest first name.
<code>RES_GUEST_LASTNAME</code>	<code>string(0, 50)</code>	The guest last name.
<code>RES_GUEST_EMAIL</code>	<code>string(0, 200)</code>	The guest email address. If not omitted (zero-length <code>string</code> ), it must be RFC 2822 compliant ( <code>addr-spec</code> specification, § 3.4.1). However, the <code>string</code> length may not exceed 200 characters.
<code>RES_GUEST_PHONE</code>	<code>string(0, 30)</code>	The guest phone number.
<code>RES_GUEST_COUNTRY</code>	<code>string(2)</code> or <code>null</code>	The guest country ISO 3166-1-alpha-2 code (see § 11.3 Country ISO 3166-1 alpha-2 codes for more information), or <code>null</code> if unspecified.

### 9.3.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, API_REPLY holds an object that contains a member called RES_ID, the id of the newly created reservation.
E_INPUT	Invalid input data. API_REPLY is null.
E_NOSUCH	The property was not found, or is unavailable (i.e. was deactivated). API_REPLY is null.
E_CONFLICT	Could not create reservation due to calendar conflict. API_REPLY is null.

On E\_OK result, API\_REPLY holds an object that contains the id of the new reservation:

Member	Data type	Description
RES_ID	int	The reservation id.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

### 9.3.4 Examples

#### 9.3.4.1 Successful reservation create

HTTP request:

```
POST /api/ra/v1/createConfirmedReservation HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXN1cm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

{
  "PROP_ID" : 1434,
  "RES_CHECK_IN" : "2011-07-21",
  "RES_CHECK_OUT" : "2011-07-26",
  "RES_N_ADULTS" : 2,
  "RES_N_CHILDREN" : 1,
  "RES_N_BABIES" : 0,
  "RES_GUEST" :
  {
    "RES_GUEST_FIRSTNAME" : "Robert",
    "RES_GUEST_LASTNAME" : "Roquefort",
    "RES_GUEST_EMAIL" : "robert@yahoo.co.uk",
    "RES_GUEST_PHONE" : "",
    "RES_GUEST_COUNTRY" : "GB"
  },
  "RES_COMMENT" : "",
  "RES_COMMENT_GUEST" : "",
  "RES_UDRA" :
  [
    {
      "UDRA_ID" : 161,
      "UDRA_CHOICE_ID" : 199
    },
    {
      "UDRA_ID" : 162,
      "UDRA_TEXT" : "John Smith told me about you!"
    }
  ]
}
```

#### HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION"      : 1,
  "API_REVISION"     : 12,
  "API_METHOD"       : "createConfirmedReservation",
  "API_CALL_ID"      : "3bc3542f7130e6eca6b3b5dbd62e8030",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        : {
                        "RES_ID"      : 24985
                      }
}
```

#### 9.3.4.2 Calendar conflict

#### HTTP request:

```
POST /api/ra/v1/createConfirmedReservation HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

{
  "PROP_ID"          : 1434,
  "RES_CHECK_IN"     : "2011-07-21",
  "RES_CHECK_OUT"    : "2011-07-26",
  "RES_N_ADULTS"     : 2,
  "RES_N_CHILDREN"   : 1,
  "RES_N_BABIES"     : 0,
  "RES_COMMENT"      : ""
}
```

#### HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION"      : 1,
  "API_REVISION"     : 12,
  "API_METHOD"       : "createConfirmedReservation",
  "API_CALL_ID"      : "933486b2204dcd6c29640cee713ee3f3",
  "API_RESULT_CODE"  : "E_CONFLICT",
  "API_RESULT_TEXT"  : "Dates not available",
  "API_REPLY"        : null
}
```

## 9.4 updateReservation

### 9.4.1 Usage

This method updates an existing property reservation. Only reservations that belong to your Rental Agency Kigo account may be updated.

Please note that the method does not:

- enforce the property minimum and maximum stay times,

- enforce the property maximum number of guests.

### 9.4.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
RES_ID	int	The Kigo reservation id to update.
RES_CHECK_IN	date	The new reservation check-in date.
RES_CHECK_OUT	date	The new reservation check-out date.
RES_N_ADULTS	int	The new number of adult guests, from 1 to 30.
RES_N_CHILDREN	int	The new number of children guests, from 0 to 29.
RES_N_BABIES	int	The new number of baby guests, from 0 to 29.
RES_GUEST (optional)	object	The reservation guest information.  This object member is optional (may be omitted).  Please refer to <code>createConfirmedReservation</code> documentation for information on RES_GUEST member format.
RES_COMMENT	string(0, 65535)	The new reservation comments.
RES_COMMENT_GUEST	string(0, 65535)	The guest reservation comment.  This object member is optional (may be omitted).
RES_UDRA (optional)	array	The reservation UDRA (user-defined reservation attributes) values.  This object member is optional (may be omitted).  Values for all attributes are also optional - the array may hold only some (or none) of the attributes defined in your Kigo Rental Agency account.  Values for attributes that are not provided in RES_UDRA array are not modified.  Please refer to <code>createConfirmedReservation</code> documentation for information on RES_UDRA member format.

Same restrictions on RES\_CHECK\_IN and RES\_CHECK\_OUT as for `createConfirmedReservation` apply.

### 9.4.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data.
E_NOSUCH	The reservation was not found, or does not belong to your Rental Agency Kigo account.
E_CONFLICT	Could not update reservation due to calendar conflict.
E_DEACTIVATED	Could not update reservation due to the property no longer being available.

API\_REPLY is always set to null.

## 9.4.4 Examples

### 9.4.4.1 Successful reservation update

HTTP request:

```
POST /api/ra/v1/updateReservation HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXN1cm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

{
  "RES_ID" : 31402,
  "RES_CHECK_IN" : "2011-07-21",
  "RES_CHECK_OUT" : "2011-07-25",
  "RES_N_ADULTS" : 2,
  "RES_N_CHILDREN" : 1,
  "RES_N_BABIES" : 0,
  "RES_COMMENT" : "Check-out 1 day earlier than initially planned"
}
```

HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION" : 1,
  "API_REVISION" : 12,
  "API_METHOD" : "updateReservation",
  "API_CALL_ID" : "736adb6ff63c85a1e2d4313e61f3dc59",
  "API_RESULT_CODE" : "E_OK",
  "API_RESULT_TEXT" : "",
  "API_REPLY" : null
}
```

## 9.5 cancelReservation

### 9.5.1 Usage

This method cancels a property reservation. Only reservations that belong to your Rental Agency Kigo account may be canceled.



### 9.5.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
RES_ID	int	The Kigo reservation id.

### 9.5.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data.
E_NOSUCH	The reservation was not found, or does not belong to your Rental Agency Kigo account.
E_ALREADY	The reservation is already canceled.

`API_REPLY` is always set to `null`.

### 9.5.4 Examples

#### 9.5.4.1 Attempt to cancel an already canceled reservation

HTTP request:

```
POST /api/ra/v1/cancelReservation HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcnQ=
Content-Type: application/json

{
  "RES_ID" : 31402,
}
```

HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION" : 1,
  "API_REVISION" : 12,
  "API_METHOD" : "cancelReservation",
  "API_CALL_ID" : "9174ebc197b2cf9349554aa58269a076",
  "API_RESULT_CODE" : "E_ALREADY",
  "API_RESULT_TEXT" : "",
  "API_REPLY" : null
}
```

## 9.6 readReservation

### 9.6.1 Usage

This method returns the details of a property reservation. Only reservations that belong to your Rental Agency Kigo account may be read.

### 9.6.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
<code>RES_ID</code>	<code>int</code>	The Kigo reservation id.

### 9.6.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful, <code>API_REPLY</code> contains reservation details.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .
<code>E_NOSUCH</code>	The requested reservation was not found, or does not belong to your Rental Agency Kigo account. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `object` with the reservation details. It contains the following members:

Member	Data type	Description
<code>RES_ID</code>	<code>int</code>	The reservation id.
<code>PROP_ID</code>	<code>int</code>	The property id.
<code>RES_STATUS</code>	<code>string</code>	The reservation status (see § 11.1 Reservation status ( <code>RES_STATUS</code> ) for list of possible statuses).
<code>RES_CHECK_IN</code>	<code>date</code>	The check-in date.
<code>RES_CHECK_OUT</code>	<code>date</code>	The check-out date.
<code>RES_N_ADULTS</code>	<code>int</code>	The number of adult guests.
<code>RES_N_CHILDREN</code>	<code>int</code>	The number of children guests.
<code>RES_N_BABIES</code>	<code>int</code>	The number of baby guests.
<code>RES_COMMENT</code>	<code>string(0, 65535)</code>	The reservation comments.
<code>RES_COMMENT_GUEST</code>	<code>string(0, 65535)</code>	The guest reservation comment.
<code>RES_GUEST_FIRSTNAME</code>	<code>string(0, 50)</code>	The guest first name.
<code>RES_GUEST_LASTNAME</code>	<code>string(0, 50)</code>	The guest last name.
<code>RES_GUEST_EMAIL</code>	<code>string(0, 200)</code>	The guest email address.
<code>RES_GUEST_PHONE</code>	<code>string(0, 30)</code>	The guest phone number. May contain any alphanumeric characters.
<code>RES_GUEST_COUNTRY</code>	<code>string(2) or null</code>	The guest country ISO 3166-1-alpha-2 code (see § 11.3 Country ISO 3166-1 alpha-2 codes for more information), or <code>null</code> if unspecified.
<code>PMT_G2RA</code>	<code>object</code>	Guest to Rental Agency payment information (see details below).
<code>OB</code>	<code>object or null</code>	Online booking information <code>object</code> , if any. <code>null</code> if no online booking occurred for the reservation.
<code>RES_UDRA</code>	<code>array</code>	The complete list of UDRA (user-defined reservation attributes) values for the reservation.



Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

The `PMT_G2RA` member of the `API_REPLY` object holds the following members:

Member	Data type	Description
<code>RENTDOWN_DUE</code>	amount	Rent down payment due amount.
<code>RENTDOWN_DUE_DATE</code>	date or null	Rent down payment due date (date if specified, null otherwise)
<code>RENTDOWN_PAID</code>	amount	Rent down payment paid amount.
<code>RENTDOWN_PAID_DATE</code>	date or null	Rent down payment paid date (date if specified, null otherwise)
<code>RENTDOWN_METHOD</code>	string(0, 100)	Rent down payment method.
<code>RENTREMAINING_DUE</code>	amount	Rent remaining payment information.
<code>RENTREMAINING_DUE_DATE</code>	date or null	
<code>RENTREMAINING_PAID</code>	amount	
<code>RENTREMAINING_PAID_DATE</code>	date or null	
<code>RENTREMAINING_METHOD</code>	string(0, 100)	
<code>DEPOSIT_DUE</code>	amount	Security deposit information.
<code>DEPOSIT_DUE_DATE</code>	date or null	
<code>DEPOSIT_PAID</code>	amount	
<code>DEPOSIT_PAID_DATE</code>	date or null	
<code>DEPOSIT_METHOD</code>	string(0, 100)	
<code>OTHER_DUE</code>	amount	Other fees/deductions information.
<code>OTHER_DUE_DATE</code>	date or null	
<code>OTHER_PAID</code>	amount	
<code>OTHER_PAID_DATE</code>	date or null	
<code>OTHER_METHOD</code>	string(0, 100)	

The `OB` member of the `API_REPLY` object, if not-null, holds the following members:

Member	Data type	Description
OB_STATE	string(1, 16)	The online booking/payment state (see § 11.2 Online booking transaction state (OB_STATE) for the list of possible values and their meaning).
OB_STATE_DATE	date	The date of the last OB_STATE change (i.e. the payment date).
OB_AMOUNT	amount	Payment amount.
OB_CURRENCY	string(3)	Payment currency ISO 4217 code.
OB_ENGINE_CODE	string(1, 16)	The code (name) of the online payment engine used to make the payment. Support for new payment engines may be added to Kigo at any time, therefore no exhaustive list of payment engines may be provided in this documentation.
OB_REF	string(0, 200)	Payment reference, provided by the online payment engine.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

The RES\_UDRA member of the API\_REPLY object is an array that holds objects with the following members:

Member	Data type	Description
UDRA_ID	int	The unique attribute identifier.
UDRA_TEXT	string(0, 200) for SINGLE_LINE_TEXT  string(0, 65535) for MULTI_LINE_TEXT	Available only for attributes with UDRA_TYPE of SINGLE_LINE_TEXT or MULTI_LINE_TEXT.  The attribute name and type may be retrieved with the listUserDefinedReservationAttributes API call.
UDRA_CHOICE_ID	int or null	The selected attribute choice identifier, or null if no choice was selected.  Available only for attributes with UDRA_TYPE of SINGLE_CHOICE.  The attribute name, type and the list of attribute choices may be retrieved with the listUserDefinedReservationAttributes API call.
UDRA_CHOICES	array of int	The list of selected attribute choices (may be empty).  Available only for attributes with UDRA_TYPE of MULTI_UNORDERED_CHOICE.  The attribute name, type and the list of attribute choices may be retrieved with the listUserDefinedReservationAttributes API call.

## 9.6.4 Examples

### 9.6.4.1 Successful reservation read, with online booking information

HTTP request:

```
POST /api/ra/v1/readReservation HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json
```

```
{
  "RES_ID" : 47882
}
```

#### HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json
```

```
{
  "API_VERSION" : 1,
  "API_REVISION" : 12,
  "API_METHOD" : "readReservation",
  "API_CALL_ID" : "e78b5b3a50255f6573a5815f2c24c093",
  "API_RESULT_CODE" : "E_OK",
  "API_RESULT_TEXT" : "",
  "API_REPLY" :
  {
    "RES_ID" : 47882,
    "PROP_ID" : 1434,
    "RES_STATUS" : "CONFIRMED",
    "RES_CHECK_IN" : "2011-08-14",
    "RES_CHECK_OUT" : "2011-09-07",
    "RES_N_ADULTS" : 1,
    "RES_N_CHILDREN" : 0,
    "RES_N_BABIES" : 0,
    "RES_COMMENT" : "",
    "RES_COMMENT_GUEST" : "",
    "RES_GUEST_FIRSTNAME" : "Robert",
    "RES_GUEST_LASTNAME" : "Roquefort",
    "RES_GUEST_EMAIL" : "robert@yahoo.co.uk",
    "RES_GUEST_PHONE" : "",
    "RES_GUEST_COUNTRY" : "GB",
    "PMT_G2RA" :
    {
      "RENTDOWN_DUE" : "1200.00",
      "RENTDOWN_DUE_DATE" : "2011-08-01",
      "RENTDOWN_PAID" : "1200.00",
      "RENTDOWN_PAID_DATE" : "2011-08-04",
      "RENTDOWN_METHOD" : "PayPal",
      "RENTREMAINING_DUE" : "2400.00",
      "RENTREMAINING_DUE_DATE" : "2011-08-14",
      "RENTREMAINING_PAID" : "0.00",
      "RENTREMAINING_PAID_DATE" : null,
      "RENTREMAINING_METHOD" : "",
      "DEPOSIT_DUE" : "0.00",
      "DEPOSIT_DUE_DATE" : null,
      "DEPOSIT_PAID" : "0.00",
      "DEPOSIT_PAID_DATE" : null,
      "DEPOSIT_METHOD" : "",
      "OTHER_DUE" : "0.00",
      "OTHER_DUE_DATE" : null,
      "OTHER_PAID" : "0.00",
      "OTHER_PAID_DATE" : null,
      "OTHER_METHOD" : ""
    },
    "OB" :
    {
      "OB_STATE" : "PAID",
```

```

        "OB_STATE_DATE"      :      "2011-08-01",
        "OB_AMOUNT"         :      "1200.00",
        "OB_CURRENCY"       :      "EUR",
        "OB_ENGINE_CODE"    :      "PAYPAL",
        "OB_REF"            :      "2FZ83282T90847819"
    },
    "RES_UDRA"              :
    [
        {
            "UDRA_ID"        :      161,
            "UDRA_CHOICE_ID" :      199
        },
        {
            "UDRA_ID"        :      162,
            "UDRA_TEXT"      :      "John Smith told me about you!"
        }
    ]
}

```

## 9.7 listUserDefinedReservationAttributes

### 9.7.1 Usage

This method returns the full list of user-defined reservation attributes defined in your Kigo Rental Agency account.

Implementing this method is required if your application needs to exchange user-defined reservation attribute values in createConfirmedReservation, updateReservation and readReservation API methods. Otherwise, implementing this method is unnecessary.

In order to keep your application's list of attribute and choices identifiers in sync with the Kigo application, this method should be invoked every time the user-defined reservation attributes or attribute choices are modified in the Kigo application.

### 9.7.2 Input

This method requires input data to be `null`.

### 9.7.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of user-defined reservation attributes. Each attribute is represented by an `object` with the following members:

Member	Data type	Description
UDRA_ID	int	The unique attribute identifier.
UDRA_NAME	string(1, 100)	The attribute name.
UDRA_TYPE	string(1, 22)	The attribute type. The following types are available: SINGLE_LINE_TEXT, MULTI_LINE_TEXT, SINGLE_CHOICE and MULTI_UNORDERED_CHOICE.
UDRA_CHOICES	array	This member holds the list of available choices.  It is available only for attributes of type SINGLE_CHOICE and MULTI_UNORDERED_CHOICE. The format is described below.

When available, every UDRA\_CHOICES array item is an object holding the following members:

Member	Data type	Description
UDRA_CHOICE_ID	int	The unique choice identifier.
UDRA_CHOICE_LABEL	string(1, 100)	The choice label.

## 9.7.4 Examples

### 9.7.4.1 Successful call

HTTP request:

```
POST /api/ra/v1/listUserDefinedReservationAttributes HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmq=
Content-Type: application/json

null
```

HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION"      : 1,
  "API_REVISION"     : 12,
  "API_METHOD"       : "listUserDefinedReservationAttributes",
  "API_CALL_ID"      : "b817db2b64c003f2dc1926cd08277830",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        :
  [
    {
      "UDRA_ID"       : 161,
      "UDRA_NAME"     : "Reason for stay",
      "UDRA_TYPE"     : "SINGLE_CHOICE",
      "CHOICES"       :
      [
        {
          "UDRA_CHOICE_ID" : 198,
          "UDRA_CHOICE_LABEL" : "Personal"
        },
        {
          "UDRA_CHOICE_ID" : 199,
          "UDRA_CHOICE_LABEL" : "Business"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  {
    "UDRA_ID"      : 162,
    "UDRA_NAME"   : "Where did you hear about us?",
    "UDRA_TYPE"   : "MULTI_LINE_TEXT"
  }
]
}

```

## 9.8 readPropertyPricingSetup

### 9.8.1 Usage

This method provides the pricing setup (rent computation ingredients) for the specified property. The pricing setup may then be used for remote batch or real-time price computations, as an alternative or a complement to invoking the computePricing or computePricingBulk API methods in real-time.

For properties provided by an agency, the pricing setup from that agency (including the currency setup) shall be used. The rent calculation adjustment and payment schedule defined in your own account for those properties are discarded by this method.

See also: § 9.9 diffPropertyPricingSetup, § 12 Property pricing computation algorithm overview.

### 9.8.2 Input

The method expects an `object` holding the following member:

Member	Data type	Description
PROP_ID	int	The Kigo property id.

### 9.8.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, <code>API_REPLY</code> holds the property pricing setup.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .
E_NOSUCH	The property was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is <code>null</code> .

#### 9.8.3.1 API\_REPLY on E\_OK result

On `E_OK` result, `API_REPLY` holds an `object` with the following members:

Member	Data type	Description
PRICING	<code>object</code> or <code>null</code>	Holds the property pricing setup ( <code>object</code> , see § 11.9 Property pricing) or <code>null</code> if the pricing is not set-up or not enabled for the requested property.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.9 diffPropertyPricingSetup

This method is a member of the diff methods family.



### 9.9.1 Usage

This methods enables the client system to maintain a local copy of the properties pricing setup by receiving information on new and modified pricing setups.

Recommended poll time: every 30 minutes to 24 hours.

### 9.9.2 Input

The method expects an `object` holding the following member:

Member	Data type	Description
DIFF_ID	string(64) or null	See § 8 diff methods family.

### 9.9.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, <code>API_REPLY</code> contains a new <code>DIFF_ID</code> and a list of properties that had their pricing setup updated.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.

On `E_OK` result, `API_REPLY` holds an `object` that contains a new `DIFF_ID` and a list of pricing setups:

Member	Data type	Description
DIFF_ID	string(64)	See § 8 diff methods family.
PROP_ID	array of int	List of properties with new, modified or disabled pricing setups. The list may contain properties that are no longer available from your account but that were returned by a previous <code>diffPropertyPricingSetup</code> call.
PRICING_LIST	array of object, or null	Depending on the number and nature of the modified pricing setups, the reply may contain only the list of properties (the above <code>PROP_ID</code> array), or also the full information on the related pricing setup.  If only the list of properties is provided, the <code>PRICING_LIST</code> member shall be null, and your application will need to invoke the <code>readPropertyPricingSetup</code> method for each of the properties listed in <code>PROP_ID</code> .  Otherwise, the <code>PRICING_LIST</code> member will hold the full pricing information, as described below.

When provided, each `PRICING_LIST` item is an `object` holding the following members:

Member	Data type	Description
PROP_ID	int	The property id.
PRICING	object or null	Holds the property pricing setup ( <code>object</code> , see § 11.9 Property pricing) or null if the pricing is not set-up, not enabled, or if the property is no longer available from your account.

## 9.10 updatePropertyPricingSetup

### 9.10.1 Usage

This methods partially updates the property pricing setup. Only the pricing for the properties provided by an owner may be updated using this method. Only the pricing for the properties that already have the pricing setup and calculations enabled may be updated.

### 9.10.2 Input

The method expects an object holding the following members:

Member	Data type	Description
PROP_ID	int	The property Id.
PRICING	object	The property pricing setup, defined in § 11.9 Property pricing. Only the RENT member is currently supported. Other members must not be provided. All amounts are to be provided in your Kigo account currency.

### 9.10.3 Result and reply

The method may return the following API\_RESULT\_CODE values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data.
E_NOSUCH	The property was not found, is unavailable (i.e. was deactivated), or was not shared by an owner.
E_LIMIT	The property pricing setup is not enabled and therefore could not be updated.

API\_REPLY is always set to null.

## 9.11 computePricing

### 9.11.1 Usage

This method provides real-time rent, fees and deposit amounts and due dates computation, based on property pricing setup in your, or the property provider's, Kigo account.

Please note that the method does not:

- verify the availability of the property for the specified dates,
- enforce the property minimum and maximum stay times,
- enforce the property maximum number of guests.

For properties provided by an agency, the pricing setup from that agency (including the currency setup) shall be used. The rent calculation adjustment and payment schedule defined in your own account for those properties are discarded by this method.

### 9.11.2 Input

The method expects an object holding the following members:

Member	Data type	Description
PROP_ID	int	The Kigo property id.
RES_CREATE	date	The create date (of the reservation). Rent discounts (e.g. early bird and last minute discounts) computation depends on this date.
RES_CHECK_IN	date	The check-in date.
RES_CHECK_OUT	date	The check-out date.
RES_N_ADULTS	int	Number of guests (adults), from 1 to 30.
RES_N_CHILDREN	int	Number of guests (children), from 0 to 29.
RES_N_BABIES	int	Number of guests (babies), from 0 to 29.

The following restrictions apply:

- Same restrictions on `RES_CHECK_IN` and `RES_CHECK_OUT` as in `createConfirmedReservation`
- The maximum allowed `RES_CREATE` is the date of the day plus 60 months (5 years)

`E_INPUT` result code is returned if not all of the above conditions are met.

However, there is no relationship restriction between `RES_CREATE` and `RES_CHECK_IN` / `RES_CHECK_OUT`.

### 9.11.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful, <code>API_REPLY</code> holds computation result details.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is null.
<code>E_NOSUCH</code>	The property was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is null.
<code>E_EMPTY</code>	The property pricing information is unavailable or the property pricing calculation is disabled. <code>API_REPLY</code> is null.
<code>E_LIMIT</code>	The property pricing information is unavailable for the specified check-in/check-out dates. <code>API_REPLY</code> is null.
<code>E_CONFLICT</code>	The property pricing information is unavailable due to a constraint. <code>API_REPLY</code> holds additional information.

#### 9.11.3.1 API\_REPLY on E\_OK result

On `E_OK` result, `API_REPLY` holds an `object` with the computation result details. It contains the following members:

Member	Data type	Description
<code>CURRENCY</code>	string(3)	Currency ISO 4217 code.
<code>RENT_BASE_AMOUNT</code>	amount	Rent base amount (before applying rent fees and discounts).
<code>EXTRA_GUEST_CHARGES_AMOUNT</code> (deprecated in revision 5)	amount	This member was deprecated and renamed to <code>RENT_GUEST_CHARGES_AMOUNT</code> in revision 5 of the API. Please refer to <code>RENT_GUEST_CHARGES_AMOUNT</code> instead.

<del>RENT_GUEST_CHARGES_AMOUNT</del> (deprecated in revision 8)	amount	Extra guest charges amount. Due to a pricing algorithm change as of the revision 8 of the API, this member is deprecated and it's value is always "0.00". Guest charges are now included in <code>RENT_BASE_AMOUNT</code> .
<code>RENT_FEES_AMOUNT</code>	amount	Rent fees (fees included in rent) amount.
<code>RENT_DISCOUNTS_AMOUNT</code>	amount	Rent discounts amount.
<code>RENT_AMOUNT</code>	amount	Total rent amount. Currently equals to: <code>RENT_BASE_AMOUNT + RENT_FEES_AMOUNT + RENT_DISCOUNTS_AMOUNT</code>
<del>RENT_DOWN_PMT_AMOUNT</del> (deprecated in revision 12)	amount	Rent down payment amount. Deprecated in revision 12. It's value is now always equal to the value of <code>RENT_AMOUNT</code> .
<del>RENT_DOWN_PMT_DUE_DATE</del> (deprecated in revision 12)	date or null	Rent down payment due date (null if there is no down payment). Deprecated in revision 12.
<del>RENT_REM_PMT_AMOUNT</del> (deprecated in revision 12)	amount	Rent remaining payment amount. Deprecated in revision 12. It's value is always "0.00".
<del>RENT_REM_PMT_DUE_DATE</del> (deprecated in revision 12)	date or null	Rent remaining payment due date (null if there is no remaining payment). Deprecated in revision 12. It's value is always null.
<code>FEES_AMOUNT</code>	amount	Fees amount.
<del>FEES_DUE_DATE</del> (deprecated in revision 12)	date or null	Fees due date (null if there are no fees). Deprecated in revision 12.
<code>TOTAL_AMOUNT</code>	amount	<code>RENT_AMOUNT + FEES_AMOUNT</code>
<code>DEPOSIT_AMOUNT</code>	amount	Security deposit amount.
<del>DEPOSIT_DUE_DATE</del> (deprecated in revision 12)	date or null	Security deposit due date (null if there is no security deposit). Deprecated in revision 12.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

### 9.11.3.2 API\_REPLY on E\_CONFLICT result

On `E_CONFLICT` result, `API_REPLY` holds an `object` with additional information. It contains the following members:

Member	Data type	Description
<code>CONSTRAINT_TYPE</code>	string(1, 255)	The type of the constraint that was met. Currently, there is a single constraint: "WEEKLY" (the specified check-in/check-out dates conflict with one or several weekly-only rental periods). However, new constraints may appear in further revisions of the API.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.12 computePricingBulk

### 9.12.1 Usage

This method is a bulk alternative to the computePricing method. It allows processing multiple properties at once, thus significantly decreasing the number of API calls.

### 9.12.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
<code>PROP_ID</code>	array of int	The list of unique Kigo property ids. The list must contain at least one and up to 20 elements.
<code>RES_CREATE</code>	date	The create date (of the reservation). Rent discounts (e.g. early bird and last minute discounts) computation depend on this date.
<code>RES_CHECK_IN</code>	date	The check-in date.
<code>RES_CHECK_OUT</code>	date	The check-out date.
<code>RES_N_ADULTS</code>	int	Number of guests (adults), from 1 to 30.
<code>RES_N_CHILDREN</code>	int	Number of guests (children), from 0 to 29.
<code>RES_N_BABIES</code>	int	Number of guests (babies), from 0 to 29.

Same restrictions as for computePricing apply.

### 9.12.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful, <code>API_REPLY</code> holds computation result details.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is null.

#### 9.12.3.1 API\_REPLY on E\_OK result

On `E_OK` result, `API_REPLY` holds an `array`, with one element for every `PROP_ID` requested in input.

Each element is an `object` holding the following members:

Member	Data type	Description
<code>PROP_ID</code>	int	The Kigo property id.
<code>RESULT_CODE</code>	string(1, 20)	One of the following values:  <code>E_OK</code> – The computation was successful, <code>REPLY</code> holds computation result details as described in 9.11.3.1 API_REPLY on E_OK result.  <code>E_NOSUCH</code> - The property was not found, or is unavailable (i.e. was deactivated). <code>REPLY</code> is null.  <code>E_EMPTY</code> - The property pricing information is unavailable or the property pricing calculation is disabled. <code>REPLY</code> is null.

		<p><code>E_LIMIT</code> - The property pricing information is unavailable for the specified check-in/check-out dates. <code>REPLY</code> is <code>null</code>.</p> <p><code>E_CONFLICT</code> - The property pricing information is unavailable due to a constraint. <code>REPLY</code> holds additional information as described in 9.11.3.2 <code>API_REPLY</code> on <code>E_CONFLICT</code> result.</p>
<code>REPLY</code>	<code>null</code> or <code>object</code>	The value of this member depends on <code>RESULT_CODE</code> as described above.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.13 listProperties

### 9.13.1 Usage

This method returns the list of properties currently active in the rental agency Kigo account. The method should be invoked periodically to keep your application's list of properties in sync with your Kigo account list of active properties.

Properties that are no longer active in your Kigo account may not have new reservations created, nor existing reservations modified, using this API.

### 9.13.2 Input

This method requires input data to be `null`.

### 9.13.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of properties, in undefined order. Each property is represented by an `object` with following members:

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>PROP_NAME</code>	<code>string(1, 100)</code>	The property name.
<code>PROP_PROVIDER_OWNER_ID</code>	<code>int</code> or <code>null</code>	The property provider owner Kigo id (if the property is provided by an owner account), or <code>null</code> . See § 11.4 Property provider for information on property providers.
<code>PROP_PROVIDER_RA_ID</code>	<code>int</code> or <code>null</code>	The property provider rental agency Kigo id (if the property is provided by a rental agency account), or <code>null</code> . See § 11.4 Property provider for information on property providers.
<code>PROP_INSTANT_BOOK</code>	<code>bool</code>	Indicates whether the instant booking is allowed without

		requirement to contact the owner first.
--	--	---

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.14 listUserDefinedPropertyAttributes

### 9.14.1 Usage

This method returns the full list of user-defined property attributes defined in your Kigo Rental Agency account.

Implementing this method is required if your application needs to read user-defined property attribute values in the readProperty API method. Otherwise, implementing this method is unnecessary.

In order to keep your application's list of attribute and choices identifiers in sync with the Kigo application, this method should be invoked every time the user-defined property attributes or attribute choices are modified in the Kigo application.

### 9.14.2 Input

This method requires input data to be `null`.

### 9.14.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of user-defined reservation attributes. Each attribute is represented by an `object` with the following members:

Member	Data type	Description
<code>UDPA_ID</code>	<code>int</code>	The unique attribute identifier.
<code>UDPA_NAME</code>	<code>string(1, 100)</code>	The attribute name.
<code>UDPA_TYPE</code>	<code>string(1, 22)</code>	The attribute type. The following types are available: <code>SINGLE_LINE_TEXT</code> , <code>MULTI_LINE_TEXT</code> , <code>SINGLE_CHOICE</code> and <code>MULTI_UNORDERED_CHOICE</code> .
<code>UDPA_CHOICES</code>	<code>array</code>	This member holds the list of available choices.  It is available only for attributes of type <code>SINGLE_CHOICE</code> and <code>MULTI_UNORDERED_CHOICE</code> . The format is described below.

When available, every `UDPA_CHOICES` `array` item is an `object` holding the following members:

Member	Data type	Description
<code>UDPA_CHOICE_ID</code>	<code>int</code>	The unique choice identifier.

UDPA_CHOICE_LABEL	string(1, 100)	The choice label.
-------------------	----------------	-------------------

## 9.14.4 Examples

### 9.14.4.1 Successful call

#### HTTP request:

```
POST /api/ra/v1/listUserDefinedPropertyAttributes HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

null
```

#### HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION"      : 1,
  "API_REVISION"     : 12,
  "API_METHOD"       : "listUserDefinedPropertyAttributes",
  "API_CALL_ID"      : "8a248375e61ed429bd815c876f1eab2a",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        :
  [
    {
      "UDPA_ID"       : 39,
      "UDPA_NAME"     : "Nearby attractions",
      "UDPA_TYPE"     : "MULTI_UNORDERED_CHOICE",
      "UDPA_CHOICES" :
      [
        {
          "UDPA_CHOICE_ID" : 129,
          "UDPA_CHOICE_LABEL" : "George Pompidou"
        },
        {
          "UDPA_CHOICE_ID" : 130,
          "UDPA_CHOICE_LABEL" : "Notre Dame"
        },
        {
          "UDPA_CHOICE_ID" : 131,
          "UDPA_CHOICE_LABEL" : "Place des Vosges"
        }
      ]
    },
    {
      "UDPA_ID"       : 1077,
      "UDPA_NAME"     : "Class",
      "UDPA_TYPE"     : "SINGLE_CHOICE",
      "UDPA_CHOICES" :
      [
        {
          "UDPA_CHOICE_ID" : 5564,
          "UDPA_CHOICE_LABEL" : "Economy"
        },
        {
          "UDPA_CHOICE_ID" : 5565,
          "UDPA_CHOICE_LABEL" : "Standard"
        }
      ]
    }
  ]
}
```



```

        {
            "UDPA_CHOICE_ID" : 5566,
            "UDPA_CHOICE_LABEL" : "Luxury"
        }
    ],
    {
        "UDPA_ID" : 25,
        "UDPA_NAME" : "Our reference",
        "UDPA_TYPE" : "SINGLE_LINE_TEXT"
    }
]
}

```

## 9.15 readProperty

### 9.15.1 Usage

This method returns active Kigo property information.

You may want to use this method if:

- your application doesn't have a property content management user-interface, and retrieves all property information from Kigo – you may want to invoke this method periodically to keep your application's property information in sync with the property information in the Kigo application,
- your application does have a property content management system and user-interface, but pre-initializes the property information with property information and photos from Kigo – you would invoke this method on each new Kigo property being added to your system.

### 9.15.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
PROP_ID	int	The Kigo property id.

### 9.15.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, <code>API_REPLY</code> contains reservation details.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .
E_NOSUCH	The requested property was not found, or is no longer active. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `object` with the property details. It contains the following members:

Member	Data type	Description
PROP_ID	int	The Kigo property id.
PROP_NAME	string(1, 100)	The property name.

PROP_PROVIDER_OWNER_ID	int or null	The property provider owner Kigo id (if the property is provided by an owner account), or null. See § 11.4 Property provider for information on property providers.
PROP_PROVIDER_RA_ID	int or null	The property provider rental agency Kigo id (if the property is provided by a rental agency account), or null. See § 11.4 Property provider for information on property providers.
PROP_INSTANT_BOOK	bool	Indicates whether the instant booking is allowed without requirement to contact the owner first.
PROP_STREETNO	string(0, 8)	Street number.
PROP_ADDR1	string(0, 60)	Street address – line 1 (street name).
PROP_ADDR2	string(0, 35)	Street address – line 2.
PROP_ADDR3	string(0, 35)	Street address – line 3.
PROP_APTNO	string(0, 10)	Apartment number.
PROP_POSTCODE	string(0, 8)	Postal code.
PROP_CITY	string(0, 26)	City.
PROP_REGION	string(0, 35)	Region.
PROP_COUNTRY	string(2)	The property country ISO 3166-1-alpha-2 code (see § 11.3 Country ISO 3166-1 alpha-2 codes for more information).
PROP_LATITUDE	decimal_degree or null	Longitude and latitude geographic coordinates in WGS 84 coordinates system. Either both members have decimal_degree values (known coordinates), or both are null (unspecified coordinates).
PROP_LONGITUDE		
PROP_PHONE	string(0, 30)	Property phone number.
PROP_AXSCODE	string(0, 65535)	Access codes.
PROP_BEDROOMS	int or null	Number of bedrooms (int), or null if unspecified.
PROP_BEDS	int or null	Number of beds (int), or null if unspecified.
PROP_BED_TYPES	array of int	A list that holds the bed type (BED_TYPE_ID from listKigoPropertyBedTypes reply) for each property bed. The list will be empty if PROP_BEDS is zero (no beds) or null (unspecified number of beds). See § 11.6 Property bed type for information on bed types, and § 9.19 listKigoPropertyBedTypes for retrieving the list of bed types defined in Kigo application.
PROP_BATHROOMS	int or null	Number of full bathrooms (int), or null if unspecified.
PROP_TOILETS	int or null	Number of toilets (int) or null, if unspecified.
PROP_TYPE_ID	int or null	The property type (int) or null if unspecified. See § 11.5 Property type for information on property types, and § 9.18 listKigoPropertyTypes for retrieving the list of property types defined in Kigo application.
PROP_SIZE	int or null	The property size. Either PROP_SIZE and

PROP_SIZE_UNIT	"SQMETER" (string), "SQFEET" (string) or null	PROP_SIZE_UNIT have int and string values respectively (the property size is specified), or both are null (property size is unspecified).
PROP_MAXGUESTS	int	Maximum allowed number of guests (total).
PROP_MAXGUESTS_ADULTS	int	Maximum allowed number of guests (adults).
PROP_MAXGUESTS_CHILDREN	int	Maximum allowed number of guests (children).
PROP_MAXGUESTS_BABIES	int	Maximum allowed number of guests (babies).
PROP_FLOOR	string(0, 4)	Floor (European).
PROP_ELEVATOR	bool or null	Property has/doesn't have elevator (bool), or null if unspecified.
PROP_CIN_TIME	time	Standard check-in time.
PROP_COUT_TIME	time	Standard check-out time.
PROP_STAY_MIN (deprecated in revision 12)	int	Minimum allowed stay time.
PROP_STAY_MIN_UNIT (deprecated in revision 12)	string holding "NIGHT", "MONTH" or "YEAR"	Deprecated by PROP_STAYTIME_MIN in revision 12.
PROP_STAYTIME_MIN	time_interval	Minimum allowed stay time.
PROP_STAY_MAX (deprecated in revision 12)	int	Maximum allowed stay time.
PROP_STAY_MAX_UNIT (deprecated in revision 12)	string holding "NIGHT", "MONTH" or "YEAR"	Deprecated by PROP_STAYTIME_MAX in revision 12.
PROP_STAYTIME_MAX	time_interval	Maximum allowed stay time.
PROP_SHORTDESCRIPTION	string(0, 65535)	Property short description.
PROP_DESCRIPTION	string(0, 65535)	Property full description.
PROP_AREADESCRIPTION	string(0, 65535)	Area description.
PROP_INVENTORY	string(0, 65535)	Property inventory.
PROP_ARRIVAL_SHEET	string(0, 65535)	Property arrival sheet.
PROP_AMENITIES	array of int	Unordered list of property amenities (AMENITY_ID from listKigoPropertyAmenities reply). See § 11.7 Property amenity for information on property amenities, and § 9.20 listKigoPropertyAmenities for retrieving the list of property amenities defined in Kigo application.
PROP_ACTIVITIES	array of int	Unordered list of property activities (ACTIVITY_ID from listKigoPropertyActivities reply). See § 11.8 Property activity for information on property activities, and § 9.21 listKigoPropertyActivities for retrieving the list of property activities defined in Kigo application.
PROP_RATE_NIGHTLY_FROM	amount or null	Property nightly, weekly and monthly rates (amount), or null if unspecified.
PROP_RATE_NIGHTLY_TO		
PROP_RATE_WEEKLY_FROM		

PROP_RATE_WEEKLY_TO		
PROP_RATE_MONTHLY_FROM		
PROP_RATE_MONTHLY_TO		
PROP_RATE_CURRENCY	string(3)	Currency ISO 4217 code.
PROP_PHOTOS	array of object	List of property JPEG photos.
PROP_UDPA	array	The complete list of UDPA (user-defined property attributes) values for the property.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

The PROP\_PHOTOS member of the API\_REPLY object is an array that holds the ordered list of property photos. Each property photo is represented by an object with following members:

Member	Data type	Description
PHOTO_ID	string(1, 200)	Unique immutable property photo identifier for use in readPropertyPhotoFile API method calls.
PHOTO_PANORAMIC	bool	Photo information.
PHOTO_NAME	string(0, 50)	
PHOTO_COMMENTS	string(0, 255)	
PHOTO_SIZE	int	The raw size of the photo file, in bytes.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

The PROP\_UDPA member of the API\_REPLY object is an array that holds objects with the following members:

Member	Data type	Description
UDPA_ID	int	The unique attribute identifier.
UDPA_TEXT	string(0, 200) for SINGLE_LINE_TEXT  string(0, 65535) for MULTI_LINE_TEXT	Available only for attributes with UDPA_TYPE of SINGLE_LINE_TEXT or MULTI_LINE_TEXT.  The attribute name and type may be retrieved with the listUserDefinedPropertyAttributes API call.
UDPA_CHOICE_ID	int or null	The selected attribute choice identifier, or null if no choice was selected.  Available only for attributes with UDPA_TYPE of SINGLE_CHOICE.  The attribute name, type and the list of attribute choices may be retrieved with the listUserDefinedPropertyAttributes API call.

<p>UDPA_CHOICES</p>	<p>array of int</p>	<p>The list of selected attribute choices (may be empty).</p> <p>Available only for attributes with UDPA_TYPE of MULTI_UNORDERED_CHOICE.</p> <p>The attribute name, type and the list of attribute choices may be retrieved with the listUserDefinedPropertyAttributes API call.</p>
---------------------	---------------------	--

## 9.15.4 Examples

### 9.15.4.1 Reading a property

HTTP request:

```
POST /api/ra/v1/readProperty HTTP/1.0
Host: app.kigo.net
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=
Content-Type: application/json

{
  "PROP_ID"      :      17784
}
```

HTTP reply:

```
HTTP/1.0 200 API Method successfully invoked
Content-Type: application/json

{
  "API_VERSION"      :      1,
  "API_REVISION"     :      12,
  "API_METHOD"       :      "readProperty",
  "API_CALL_ID"      :      "43b6e85eef11d381eacd3feb1f00ba10",
  "API_RESULT_CODE"  :      "E_OK",
  "API_RESULT_TEXT"  :      "",
  "API_REPLY"        :
  {
    "PROP_ID"        :      17784,
    "PROP_NAME"      :      "Champs Elysées",
    "PROP_PROVIDER_OWNER_ID" :      null,
    "PROP_PROVIDER_RA_ID" :      7445,
    "PROP_INSTANT_BOOK" :      true,
    "PROP_STREETNO"  :      "",
    "PROP_ADDR1"     :      "Avenue des Champs Elysées",
    "PROP_ADDR2"     :      "",
    "PROP_ADDR3"     :      "",
    "PROP_APTNO"     :      "",
    "PROP_POSTCODE"  :      "75008",
    "PROP_CITY"      :      "Paris",
    "PROP_REGION"    :      "",
    "PROP_COUNTRY"   :      "FR",
    "PROP_LATITUDE"  :      "48.869720",
    "PROP_LONGITUDE" :      "2.308290",
    "PROP_PHONE"     :      "",
    "PROP_AXSCODE"   :      "",
    "PROP_BEDROOMS"  :      3,
    "PROP_BEDS"      :      3,
    "PROP_BED_TYPES" :      [ 1, 1, 3 ],
    "PROP_BATHROOMS" :      3,
    "PROP_TOILETS"   :      3,
    "PROP_TYPE_ID"   :      14,
    "PROP_SIZE"      :      112,
  }
}
```

```

"PROP_SIZE_UNIT" : "SQMETER",
"PROP_MAXGUESTS" : 6,
"PROP_MAXGUESTS_ADULTS" : 6,
"PROP_MAXGUESTS_CHILDREN" : 5,
"PROP_MAXGUESTS_BABIES" : 2,
"PROP_FLOOR" : "4",
"PROP_ELEVATOR" : true,
"PROP_CIN_TIME" : "14:00",
"PROP_COUT_TIME" : "10:00"
"PROP_STAY_MIN" : 5,
"PROP_STAY_MIN_UNIT" : "NIGHT",
"PROP_STAY_MAX" : 2,
"PROP_STAY_MAX_UNIT" : "MONTH",
"PROP_SHORTDESCRIPTION" : "",
"PROP_DESCRIPTION" : "",
"PROP_AREADESCRIPTION" : "",
"PROP_INVENTORY" : "",
"PROP_ARRIVAL_SHEET" : "",
"PROP_AMENITIES" : [ 1, 2, 3, 11, 12, 13, 22 ],
"PROP_ACTIVITIES" : [ 23, 28 ],
"PROP_RATE_NIGHTLY_FROM" : "149.95",
"PROP_RATE_NIGHTLY_TO" : null,
"PROP_RATE_WEEKLY_FROM" : null,
"PROP_RATE_WEEKLY_TO" : null,
"PROP_RATE_MONTHLY_FROM" : null,
"PROP_RATE_MONTHLY_TO" : null,
"PROP_RATE_CURRENCY" : "USD",
"PROP_PHOTOS" :
[
  {
    "PHOTO_ID" : "781211e6d343db75",
    "PHOTO_PANORAMIC" : false,
    "PHOTO_NAME" : "Living room",
    "PHOTO_COMMENTS" : "",
    "PHOTO_SIZE" : 1860022
  },
  {
    "PHOTO_ID" : "bc30b4965d32b1dc",
    "PHOTO_PANORAMIC" : true,
    "PHOTO_NAME" : "View from balcony",
    "PHOTO_COMMENTS" : "Champs Elysées Avenue",
    "PHOTO_SIZE" : 1947614
  }
],
"PROP_UDPA" :
[
  {
    "UDPA_ID" : 25,
    "UDPA_TEXT" : "75001-B40"
  },
  {
    "UDPA_ID" : 39,
    "UDPA_CHOICES" : [ 129, 130 ]
  },
  {
    "UDPA_ID" : 1077,
    "UDPA_CHOICE_ID" : 5564
  }
]
}

```

## 9.16 readPropertyPhotoFile

### 9.16.1 Usage

This method returns the binary content of a property JPEG photo file. You may invoke this method to retrieve new property photos discovered in a prior readProperty call.

Property photo files are immutable, and need to be downloaded only once.

Care must be taken not to trigger the bandwidth limiting policy when retrieving multiple property photos, by throttling calls to this method.

### 9.16.2 Input

The method expects an `object` holding the following members:

Member	Data type	Description
PROP_ID	int	The Kigo property id.
PHOTO_ID	string(1, 200)	Property photo identifier obtained in readProperty method reply.

### 9.16.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful, <code>API_REPLY</code> holds a <code>string</code> that represents a base64 (RFC 4648) encoded JPEG photo file content.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .
E_NOSUCH	The specified property or photo was not found, or the property is no longer active. <code>API_REPLY</code> is <code>null</code> .

## 9.17 listKigoCountries

### 9.17.1 Usage

This method returns the list of countries used in Kigo application. You may want to invoke this method periodically to keep your application's list of countries in sync with the Kigo list.

### 9.17.2 Input

This method requires input data to be `null`.

### 9.17.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of countries, in undefined order. Each country is represented by an `object` with following members:

Member	Data type	Description
COUNTRY_ISO_3166_1_ALPHA_2	string(2)	The country ISO 3166-1 alpha-2 code.
COUNTRY_NAME	string(1, 50)	The country name, such as used in Kigo application.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.18 listKigoPropertyTypes

### 9.18.1 Usage

This method returns the list of property types used in Kigo application. You may want to invoke this method periodically to keep your application's list of property types in sync with the Kigo list.

### 9.18.2 Input

This method required input data to be `null`.

### 9.18.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
E_OK	The call was successful.
E_INPUT	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of property types, in undefined order. Each property type is represented by an `object` with following members:

Member	Data type	Description
PROP_TYPE_ID	int	The unique property type identifier.
PROP_TYPE_LABEL	string(1, 50)	The property type label, such as used in Kigo application.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.19 listKigoPropertyBedTypes

### 9.19.1 Usage

This method returns the list of property bed types used in Kigo application. You may want to invoke this method periodically to keep your application's list of bed types in sync with the Kigo list.

### 9.19.2 Input

This method requires input data to be `null`.



### 9.19.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of bed types, in undefined order. Each bed type is represented by an `object` with following members:

Member	Data type	Description
<code>BED_TYPE_ID</code>	<code>int</code>	The unique bed type identifier.
<code>BED_TYPE_LABEL</code>	<code>string(1, 50)</code>	The bed type label, such as used in Kigo application.

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 9.20 listKigoPropertyAmenities

### 9.20.1 Usage

This method returns the list of property amenities used in Kigo application. You may want to invoke this method periodically to keep your application's list of property amenities in sync with the Kigo list.

### 9.20.2 Input

This method requires input data to be `null`.

### 9.20.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `object` with the following members:

Member	Data type	Description
<code>AMENITY_CATEGORY</code>	<code>array</code>	List of amenity categories, in undefined order.
<code>AMENITY</code>	<code>array</code>	List of amenities, in undefined order.

Each member of the `AMENITY_CATEGORY` `array` is an `object` holding the following members:

Member	Data type	Description
<code>AMENITY_CATEGORY_ID</code>	<code>int</code>	The unique amenity category identifier.
<code>AMENITY_CATEGORY_LABEL</code>	<code>string(1, 50)</code>	The amenity category label, such as used in Kigo

		application.
--	--	--------------

Each member of the `AMENITY` array is an `object` holding the following members:

Member	Data type	Description
<code>AMENITY_ID</code>	<code>int</code>	The unique amenity identifier.
<code>AMENITY_CATEGORY_ID</code>	<code>int</code>	Identifier of the <code>AMENITY_CATEGORY</code> the amenity belongs to.
<code>AMENITY_LABEL</code>	<code>string(1, 50)</code>	The amenity label, such as used in Kigo application.

## 9.21 listKigoPropertyActivities

### 9.21.1 Usage

This method returns the list of property activities used in Kigo application. You may want to invoke this method periodically to keep your application's list of property activities in sync with the Kigo list.

### 9.21.2 Input

This method requires input data to be `null`.

### 9.21.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `object` with the following members:

Member	Data type	Description
<code>ACTIVITY_CATEGORY</code>	<code>array</code>	List of activity categories, in undefined order.
<code>ACTIVITY</code>	<code>array</code>	List of activities, in undefined order.

Each member of the `ACTIVITY_CATEGORY` array is an `object` holding the following members:

Member	Data type	Description
<code>ACTIVITY_CATEGORY_ID</code>	<code>int</code>	The unique activity category identifier.
<code>ACTIVITY_CATEGORY_LABEL</code>	<code>string(1, 50)</code>	The activity category label, such as used in Kigo application.

Each member of the `ACTIVITY` array is an `object` holding the following members:

Member	Data type	Description
<code>ACTIVITY_ID</code>	<code>int</code>	The unique activity identifier.
<code>ACTIVITY_CATEGORY_ID</code>	<code>int</code>	Identifier of the <code>ACTIVITY_CATEGORY</code> the activity belongs to.
<code>ACTIVITY_LABEL</code>	<code>string(1, 50)</code>	The activity label, such as used in Kigo application.

## 9.22 listKigoFeeTypes

### 9.22.1 Usage

This method returns the list of fee types (see § 11.9 Property pricing) used in Kigo application. You may want to invoke this method periodically to keep your application's list of fee types in sync with the Kigo list.

### 9.22.2 Input

This method requires input data to be `null`.

### 9.22.3 Result and reply

The method may return the following `API_RESULT_CODE` values:

Result	Description
<code>E_OK</code>	The call was successful.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` of fee types, in undefined order. Each fee type is represented by an `object` with following members:

Member	Data type	Description
<code>FEE_TYPE_ID</code>	<code>int</code>	The unique fee type identifier.
<code>FEE_TYPE_LABEL</code>	<code>string(1, 50)</code>	The fee type label, such as used in Kigo application.
<code>FEE_TYPE_INCLUDE_IN_RENT</code>	<code>bool</code>	Specified whether the fees of this type may be included in rent (see § 11.9.2 FEES).
<code>FEE_TYPE_UNITS</code>	<code>array of string</code>	List of supported units for this type (see § 11.9.2 FEES).

Note that other members may be added in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them).

## 10 Deprecated API methods

### 10.1 `listPropertyCalendarReservations`

This method was obsoleted by `diffPropertyCalendarReservations` in revision 12 of this API. Implementations that use this method should be upgraded to use the `diffPropertyCalendarReservations` method instead.

### 10.2 `listCountries`

This method was renamed to `listKigoCountries` in revision 4 of this API. Please use `listKigoCountries` instead.

## 11 Kigo data dictionary

### 11.1 Reservation status (`RES_STATUS`)

The table below lists all Kigo reservation statuses, along with their description.

While this version of API may only create `CONFIRMED` reservations, it may retrieve or manipulate reservations with any of the statuses below. Therefore, implementations must recognize and expect to deal with all of them.

Reservation status	Description
<code>CONFIRMED</code>	A normal property reservation.
<code>HOLD</code>	Hold dates – different from <code>CONFIRMED</code> reservations in that they may be set to expire (get canceled) automatically by the property provider.
<code>CANCELED</code>	A <code>CONFIRMED</code> reservation that was canceled.
<code>CANCELED_HOLD</code>	A <code>HOLD</code> that was canceled.

### 11.2 Online booking transaction state (`OB_STATE`)

The table below lists all online booking transaction states that may be reported in `OB_STATE` member of the `OB` object in `API_REPLY` of a successful `readReservation` method API call.

Transaction state	Description
<code>PAID</code>	Kigo was informed about the successful payment.
<code>ERROR</code>	While the transaction was already in <code>PAID</code> state, Kigo was informed about a payment problem by the online payment engine. There are many possible reasons for this to occur, although the most frequent are customer complaint and fraud protection system triggered by the payment engine. You should connect to the Kigo application and review the transaction logs for more information about the problem.

No other booking transaction states may be reported by this version of the API.

### 11.3 Country ISO 3166-1 alpha-2 codes

This API exchanges country information using the standard ISO 3166-1 alpha-2 codes.

Kigo uses a subset of “officially assigned” ISO 3166-1 alpha-2 codes, some of the “transitionally reserved” codes (e.g. countries that no longer exist may remain in the Kigo countries list for undefined amount of time), and the two “user-assigned” country codes:

- `AA` which identifies “Other” country choice in Kigo application,
- `ZZ` which identifies “Unknown” country choice in Kigo application.

Changes to the ISO 3166-1 and Kigo countries lists may occur at any time. Those changes do not imply API revision nor version number update.

You may keep your application in sync with the list of countries supported by Kigo by periodically invoking the `listKigoCountries` API method.

## 11.4 Property provider

Properties in a Kigo rental agency or portal account are provided by connected Kigo owner or rental agency accounts. When a single property is being offered through more than one connection (eg. by two rental agencies), the preferred one may be selected in the Kigo GUI.

## 11.5 Property type

The Kigo application maintains a custom list of property types. A property type is identified by a unique integer type identifier, referred to as `PROP_TYPE_ID`.

You may keep your application in sync with the list of property types supported by Kigo by periodically invoking the `listKigoPropertyTypes` API method.

## 11.6 Property bed type

The Kigo application maintains a custom list of bed types. A bed type is identified by a unique integer type identifier, referred to as `BED_TYPE_ID`.

You may keep your application in sync with the list of bed types supported by Kigo by periodically invoking the `listKigoPropertyBedTypes` API method.

## 11.7 Property amenity

The Kigo application maintains a custom list of property amenities and amenity categories.

You may keep your application in sync with the lists of amenities and amenity categories supported by Kigo by periodically invoking the `listKigoPropertyAmenities` API method.

## 11.8 Property activity

The Kigo application maintains a custom list of property activities and activity categories.

You may keep your application in sync with the lists of activities and activity categories supported by Kigo by periodically invoking the `listKigoPropertyActivities` API method.

## 11.9 Property pricing

The Kigo application allows users to read and update the property pricing setup, through the Kigo application, and to some extent through the Kigo REST API (as of revision 12).

The property pricing setup is stored into an `object` holding the following members. Not all of them are always available. Please refer to the individual API methods documentation.

Member	Data type	Description
<code>CURRENCY</code>	<code>string(3)</code>	Pricing setup currency ISO 4217 code. This is the currency of the Kigo account the property pricing setup is being read from. It may not be updated through the API.
<code>RENT</code>	<code>object</code>	See § 11.9.1 RENT. Provided by <code>readPropertyPricingSetup</code> and <code>diffPropertyPricingSetup</code> methods, and may be updated by the <code>updatePropertyPricingSetup</code> method.
<code>FEES</code>	<code>object</code>	See § 11.9.2 FEES. Provided by <code>readPropertyPricingSetup</code> and <code>diffPropertyPricingSetup</code> methods.
<code>DISCOUNTS</code>	<code>object</code>	See § 11.9.3 DISCOUNTS. Provided by <code>readPropertyPricingSetup</code> and <code>diffPropertyPricingSetup</code> methods.

DEPOSIT	object or null	See § 11.9.4 DEPOSIT. Provided by readPropertyPricingSetup and diffPropertyPricingSetup methods
---------	----------------	---

Note that other members may be added to any `object` documented in this chapter, in further revisions of the API. Your application must handle those new members gracefully (i.e. ignore them) when receiving the property pricing setup.

## 11.9.1 RENT

### 11.9.1.1 Definition

The `RENT` object holds the following members:

Member	Data type	Description
PERGUEST_CHARGE	object or null	Specified whether a per-guest charges apply ( <code>object</code> ) or not ( <code>null</code> ) to the amounts defined in the <code>PERIODS</code> .
PERIODS	array of object	List of the periods (minimum 1, maximum 50 periods).

The `PERGUEST_CHARGE` object holds the following members:

Member	Data type	Description
TYPE	string	This value defines how the numbers of reservation adults, children and babies are taken into account when the computation process needs to pick the appropriate weekly or nightly amount in the <code>WEEKLY_AMOUNTS</code> or <code>NIGHTLY_AMOUNTS</code> lists.  Possible values are: <ul style="list-style-type: none"> <li>"ADULTS" – the number of guests will match only the number of adults</li> <li>"ADULTS_CHILDREN" – the number of guests will match the number of adults plus the number of children</li> <li>"ADULTS_CHILDREN_BABIES" – the number of guests will match the number of adults plus the number of children plus the number of babies</li> </ul>
STANDARD	int	The standard occupancy. A single <code>WEEKLY_AMOUNTS</code> or <code>NIGHTLY_AMOUNTS</code> amount will apply to any number of guests between 1 and <code>STANDARD</code> . Allowed range: from 1 to 29.
MAX	int	The maximum occupancy. A single <code>WEEKLY_AMOUNTS</code> or <code>NIGHTLY_AMOUNTS</code> amount will apply to any number of guests that is equal to, or that exceeds <code>MAX</code> . Allowed range: from <code>STANDARD</code> +1 to 30.

Each `PERIODS` item is an `object` holding the following members:

Member	Data type	Description
CHECK_IN	date	The period check-in date. The preceding period (if any) <code>CHECK_OUT</code> date must be equal to this period <code>CHECK_IN</code> date.
CHECK_OUT	date	The period check-out date. Must be greater than <code>CHECK_IN</code> date. The following period (if any) <code>CHECK_IN</code> date must be equal to this

		period <code>CHECK_OUT</code> date.
<code>NAME</code>	<code>string(0, 50)</code>	The period name.
<code>STAY_MIN</code>	<code>time_interval</code>	The minimum stay defined for this period. For a weekly-only period ( <code>WEEKLY</code> value is <code>true</code> ), the minimum stay may not be under 7 nights.
<code>WEEKLY</code>	<code>bool</code>	Specified whether the period is a weekly-only rental period ( <code>true</code> ) or not ( <code>false</code> ). A period may be weekly-only only if the period <code>CHECK_IN</code> and <code>CHECK_OUT</code> dates are on the same day of week.
<code>WEEKLY_AMOUNTS</code>	<code>array of object</code>	<p><code>WEEKLY_AMOUNTS</code> is available only if <code>WEEKLY</code> is <code>true</code>, while <code>NIGHTLY_AMOUNTS</code> is available only if <code>WEEKLY</code> is <code>false</code>.</p> <p>The number of items in these arrays is determined by the product of items required by <code>PERGUEST_CHARGE</code> (see <code>GUESTS_FROM</code> below), of the stay ranges (only for <code>WEEKLY = false</code>, see <code>STAY_FROM</code> below) and of the night-of-week groups (only for <code>WEEKLY = false</code>, see <code>WEEK_NIGHTS</code> below).</p>
<code>NIGHTLY_AMOUNTS</code>	<code>array of object</code>	

Each `WEEKLY_AMOUNTS` item is an `object` holding the following members:

Member	Data type	Description
<code>GUESTS_FROM</code>	<code>int</code>	<p>Represents the number of guests the <code>AMOUNT</code> applies to.</p> <p>If <code>PERGUEST_CHARGE</code> is <code>null</code>, then there is only one value, 1.</p> <p>If <code>PERGUEST_CHARGE</code> is not <code>null</code>, then each of the following values need to be provided in the <code>WEEKLY_AMOUNTS</code> array:</p> <ul style="list-style-type: none"> <li>1 – matches any number of guests from 1 to <code>STANDARD</code> included.</li> <li>From <code>STANDARD+1</code> to <code>MAX-1</code> – matches that number of guests</li> <li><code>MAX</code> – matches any number of guests that is equal or greater than <code>MAX</code>.</li> </ul>
<code>AMOUNT</code>	<code>amount</code>	The weekly amount, in range "0.01" – "999999.99".

Each `NIGHTLY_AMOUNTS` item is an `object` holding the following members:

Member	Data type	Description
<code>GUESTS_FROM</code>	<code>int</code>	Same definition as for the <code>WEEKLY_AMOUNTS</code> items.
<code>WEEK_NIGHTS</code>	<code>array of int</code>	<code>array</code> holding the nights of week in ISO 8601 format (1 for Monday, 7 for Sunday) the below <code>AMOUNT</code> applies to. There must be as many <code>NIGHTLY_AMOUNTS</code> records as necessary in a period for all the week nights to be covered, regardless of the period duration (even if shorter than a week).



STAY_FROM	time_interval	The minimum stay duration the below AMOUNT applies to. The same stay ranges apply to all the periods (they may not differ from one period to another). The "1 night" range is mandatory.
AMOUNT	amount	The nightly amount for the GUESTS_FROM x WEEK_NIGHTS x STAY_FROM combination. Range: "0.01" – "999999.99".

### 11.9.1.2 Examples

Example of the RENT object with no PERGUEST\_CHARGE defined (so that there is always only one GUESTS\_FROM value of 1) and with only one length of stay range (the mandatory "1 night" range).

```
{
  "PERGUEST_CHARGE" : null,
  "PERIODS" : [
    {
      "CHECK_IN" : "2014-01-01",
      "CHECK_OUT" : "2014-03-01",
      "NAME" : "Winter 2014",
      "STAY_MIN" : { "UNIT" : "NIGHT", "NUMBER" : 3 },
      "WEEKLY" : false,
      "NIGHTLY_AMOUNTS" : [
        {
          "GUESTS_FROM" : 1,
          "WEEK_NIGHTS" : [ 1, 2, 3, 4, 5, 6, 7 ],
          "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
          "AMOUNT" : "100.00"
        }
      ]
    },
    {
      "CHECK_IN" : "2014-03-01",
      "CHECK_OUT" : "2014-05-31",
      "NAME" : "",
      "STAY_MIN" : { "UNIT" : "NIGHT", "NUMBER" : 7 },
      "WEEKLY" : true,
      "WEEKLY_AMOUNTS" : [
        {
          "GUESTS_FROM" : 1,
          "AMOUNT" : "650.00"
        }
      ]
    }
  ]
}
```

Example of the RENT object with PERGUEST\_CHARGE defined with standard occupancy of 2 guests (adults + children) and maximum occupancy of 4 guests. Moreover, a different rate is applied for the Saturdays and Sundays on the first period. The length of stay ranges are "1 to < 1 month" and "1 month or longer".

```
{
  "PERGUEST_CHARGE" : {
    "TYPE" : "ADULTS_CHILDREN",
    "STANDARD" : 2,
    "MAX" : 4
  },
  "PERIODS" : [
    {
      "CHECK_IN" : "2014-01-01",
      "CHECK_OUT" : "2014-03-01",

```

```

"NAME" : "Winter 2014",
"STAY_MIN" : { "UNIT" : "NIGHT", "NUMBER" : 3 },
"WEEKLY" : false,
"NIGHTLY_AMOUNTS" : [
  {
    "GUESTS_FROM": 1,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
    "AMOUNT" : "300.00"
  },
  {
    "GUESTS_FROM": 1,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
    "AMOUNT" : "200.00"
  },
  {
    "GUESTS_FROM": 3,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
    "AMOUNT" : "320.00"
  },
  {
    "GUESTS_FROM": 3,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
    "AMOUNT" : "220.00"
  },
  {
    "GUESTS_FROM": 4,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
    "AMOUNT" : "340.00"
  },
  {
    "GUESTS_FROM": 4,
    "WEEK_NIGHTS": [ 1, 2, 3, 4, 5 ],
    "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
    "AMOUNT" : "240.00"
  },
  {
    "GUESTS_FROM": 1,
    "WEEK_NIGHTS": [ 6, 7 ],
    "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
    "AMOUNT" : "310.00"
  },
  {
    "GUESTS_FROM": 1,
    "WEEK_NIGHTS": [ 6, 7 ],
    "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
    "AMOUNT" : "210.00"
  },
  {
    "GUESTS_FROM": 3,
    "WEEK_NIGHTS": [ 6, 7 ],
    "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
    "AMOUNT" : "330.00"
  },
  {
    "GUESTS_FROM": 3,
    "WEEK_NIGHTS": [ 6, 7 ],
    "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
    "AMOUNT" : "230.00"
  }
]

```

```

    },
    {
      "GUESTS_FROM": 4,
      "WEEK_NIGHTS": [ 6, 7 ],
      "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
      "AMOUNT"     : "350.00"
    },
    {
      "GUESTS_FROM": 4,
      "WEEK_NIGHTS": [ 6, 7 ],
      "STAY_FROM" : { "UNIT" : "MONTH", "NUMBER" : 1 },
      "AMOUNT"     : "250.00"
    }
  ]
},
{
  "CHECK_IN"      : "2014-03-01",
  "CHECK_OUT"     : "2014-05-31",
  "NAME"          : "",
  "STAY_MIN"      : {
    "UNIT"        : "NIGHT",
    "NUMBER"      : 7
  },
  "WEEKLY"        : true,
  "WEEKLY_AMOUNTS" : [
    {
      "GUESTS_FROM": 1,
      "AMOUNT"     : "2500.00"
    },
    {
      "GUESTS_FROM": 3,
      "AMOUNT"     : "2600.00"
    },
    {
      "GUESTS_FROM": 4,
      "AMOUNT"     : "2700.00"
    }
  ]
}
]
}

```

This above example matches the following setup in the Kigo application:

## Nightly / weekly rates

Enable night of week pricing

Enable per-guest charge [Change settings](#)

Applies to Adults + children for standard capacity of 2 guests and maximum capacity of 4 guests.

Rental period	Period name (optional) Minimum stay	Weekly rate	Nightly rates			
			Night of week	For	Length of stay	
		1 night to < 1 month			1 month or longer	Add

Add new period

Check-in: <input type="text" value="Wed, 01 Jan 2014"/> Check-out: <input type="text" value="Sat, 01 Mar 2014"/> <input type="checkbox"/> Weekly rental only <a href="#">Delete period</a>	Period name: <input type="text" value="Winter 2014"/> Minimum stay: <input type="text" value="3"/> <input type="text" value="Night"/>	<input checked="" type="radio"/> Mon <input checked="" type="radio"/> Tue <input checked="" type="radio"/> Wed <input checked="" type="radio"/> Thu <input checked="" type="radio"/> Fri <input type="radio"/> Sat <input type="radio"/> Sun	1 to 2 guests <input type="text" value="300"/> <input type="text" value="200"/> 3 guests <input type="text" value="320"/> <input type="text" value="220"/> 4 guests + <input type="text" value="340"/> <input type="text" value="240"/>
		<input type="radio"/> Mon <input type="radio"/> Tue <input type="radio"/> Wed <input type="radio"/> Thu <input type="radio"/> Fri <input checked="" type="radio"/> Sat <input checked="" type="radio"/> Sun	1 to 2 guests <input type="text" value="310"/> <input type="text" value="210"/> 3 guests <input type="text" value="330"/> <input type="text" value="230"/> 4 guests + <input type="text" value="350"/> <input type="text" value="250"/>

[Add a new night-of-week group](#)

Add new period

Check-in: <input type="text" value="Sat, 01 Mar 2014"/> Check-out: <input type="text" value="Sat, 31 May 2014"/> <input checked="" type="checkbox"/> Weekly rental only (Saturday - Saturday) <a href="#">Delete period</a>	Period name: <input type="text"/> Minimum stay: <input type="text" value="7"/> <input type="text" value="Night"/>	1 to 2 guests <input type="text" value="2500"/> 3 guests <input type="text" value="2600"/> 4 guests + <input type="text" value="2700"/>	
---	--	---	--

Add new period

## 11.9.2 FEES

### 11.9.2.1 Definition

The `FEES` object holds the following members:

Member	Data type	Description
<code>FEES</code>	array of object	List of the fees that apply to the property.

Each `FEES` item is an object holding the following members:

Member	Data type	Description
<code>FEE_TYPE_ID</code>	int	The fee type id. The <code>FEES</code> list may contain only one item of each fee type. See § 9.2.2 <code>listKigoFeeTypes</code> for retrieving the list of fee types

		defined in Kigo application.
INCLUDE_IN_RENT	bool	Indicates whether the computed fee amount should be included in the rent, or whether it should account against the fees total amount.
UNIT	string	The unit and value (parameters, dependent on unit) used for computing the fee amount.
VALUE		

The available `UNIT` and the matching data type of the `VALUE` member is defined in the following table:

<code>UNIT</code>	<code>VALUE</code> data type															
"AMOUNT"	amount in the "0.01" - "999999.99" range															
"AMOUNT_PER_NIGHT"	amount in the "0.01" - "999999.99" range															
"AMOUNT_PER_NIGHT_PER_GUEST"	object holding the following members: <table border="1" data-bbox="443 815 1437 1005"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>AMOUNT_ADULT</td> <td>amount in the "0.01" - "999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "999.99" range	AMOUNT_CHILD	amount in the "0.00" - "999.99" range	AMOUNT_BABY	amount in the "0.00" - "999.99" range							
Member	Data type															
AMOUNT_ADULT	amount in the "0.01" - "999.99" range															
AMOUNT_CHILD	amount in the "0.00" - "999.99" range															
AMOUNT_BABY	amount in the "0.00" - "999.99" range															
"AMOUNT_PER_GUEST"	object holding the following members: <table border="1" data-bbox="443 1084 1437 1274"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>AMOUNT_ADULT</td> <td>amount in the "0.01" - "999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "999.99" range	AMOUNT_CHILD	amount in the "0.00" - "999.99" range	AMOUNT_BABY	amount in the "0.00" - "999.99" range							
Member	Data type															
AMOUNT_ADULT	amount in the "0.01" - "999.99" range															
AMOUNT_CHILD	amount in the "0.00" - "999.99" range															
AMOUNT_BABY	amount in the "0.00" - "999.99" range															
"PERCENT_RENT"	amount in the "0.01" - "100.00" range															
"STAYLENGTH"	array (minimum 2 and maximum 40 items) of object, where each item is holding the following members: <table border="1" data-bbox="443 1429 1437 1675"> <thead> <tr> <th>Member</th> <th>Data type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>STAY_FROM</td> <td>time_interval</td> <td>The minimum stay duration the below <code>UNIT</code> and <code>VALUE</code> apply to. The "1 night" range is mandatory.</td> </tr> <tr> <td>UNIT</td> <td>string</td> <td rowspan="2">The unit and value (parameters, dependent on unit) used for computing the fee amount.</td> </tr> <tr> <td>VALUE</td> <td></td> </tr> </tbody> </table> <p>The available <code>UNIT</code> and the matching data type of the <code>VALUE</code> member is defined in the following table:</p> <table border="1" data-bbox="443 1771 1437 1868"> <thead> <tr> <th><code>UNIT</code></th> <th><code>VALUE</code> data type</th> </tr> </thead> <tbody> <tr> <td>"AMOUNT"</td> <td>amount in the "0.00" - "999999.99" range</td> </tr> </tbody> </table>	Member	Data type	Description	STAY_FROM	time_interval	The minimum stay duration the below <code>UNIT</code> and <code>VALUE</code> apply to. The "1 night" range is mandatory.	UNIT	string	The unit and value (parameters, dependent on unit) used for computing the fee amount.	VALUE		<code>UNIT</code>	<code>VALUE</code> data type	"AMOUNT"	amount in the "0.00" - "999999.99" range
Member	Data type	Description														
STAY_FROM	time_interval	The minimum stay duration the below <code>UNIT</code> and <code>VALUE</code> apply to. The "1 night" range is mandatory.														
UNIT	string	The unit and value (parameters, dependent on unit) used for computing the fee amount.														
VALUE																
<code>UNIT</code>	<code>VALUE</code> data type															
"AMOUNT"	amount in the "0.00" - "999999.99" range															

"AMOUNT_PER_NIGHT_PER_GUEST"	object holding the following members:								
	<table border="1"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>AMOUNT_ADULT</td> <td>amount in the "0.01" - "999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "999.99" range	AMOUNT_CHILD	amount in the "0.00" - "999.99" range	AMOUNT_BABY	amount in the "0.00" - "999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "999.99" range								
AMOUNT_BABY	amount in the "0.00" - "999.99" range								
"AMOUNT_PER_GUEST"	object holding the following members:								
	<table border="1"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>AMOUNT_ADULT</td> <td>amount in the "0.01" - "999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "999.99" range	AMOUNT_CHILD	amount in the "0.00" - "999.99" range	AMOUNT_BABY	amount in the "0.00" - "999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "999.99" range								
AMOUNT_BABY	amount in the "0.00" - "999.99" range								
"PERCENT_RENT"	amount in the "0.01" - "100.00" range								

### 11.9.2.2 Examples

Example of the FEES object holding two fees.

```
{
  "FEES": [
    {
      "FEE_TYPE_ID" : 4,
      "INCLUDE_IN_RENT" : true,
      "UNIT" : "PERCENT_RENT",
      "VALUE" : "0.25"
    },
    {
      "FEE_TYPE_ID" : 10,
      "INCLUDE_IN_RENT" : false,
      "UNIT" : "STAYLENGTH",
      "VALUE" : [
        {
          "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 1 },
          "UNIT" : "AMOUNT_PER_NIGHT_PER_GUEST",
          "VALUE" : {
            "AMOUNT_ADULT" : "1.45",
            "AMOUNT_CHILD" : "1.45",
            "AMOUNT_BABY" : "0.00"
          }
        }
      ]
    },
    {
      "STAY_FROM" : { "UNIT" : "NIGHT", "NUMBER" : 7 },
      "UNIT" : "AMOUNT",
      "VALUE" : "0.00"
    }
  ]
}
```

This above example matches the following setup in the Kigo application:

Fee	Include in rent ?														
<input type="checkbox"/> Booking fees	<input type="checkbox"/>	amount													
<input type="checkbox"/> Check-in fees (weekday)	<input type="checkbox"/>	amount													
<input type="checkbox"/> Check-in fees (weekend)	<input type="checkbox"/>	amount													
<input type="checkbox"/> Check-out fees (weekday)	<input type="checkbox"/>	amount													
<input type="checkbox"/> Check-out fees (weekend)	<input type="checkbox"/>	amount													
<input type="checkbox"/> Cleaning fees	<input type="checkbox"/>	amount													
<input type="checkbox"/> Linen pack	<input type="checkbox"/>	amount													
<input checked="" type="checkbox"/> Occupancy tax	<input checked="" type="checkbox"/>	per length of stay	<table border="1"> <tr> <td><b>1 night to 6 nights</b></td> <td><b>7 nights or longer</b></td> <td><b>Add</b></td> </tr> <tr> <td>amount per night, per guest</td> <td>amount</td> <td></td> </tr> <tr> <td>1.45    1.45    0</td> <td>0</td> <td></td> </tr> <tr> <td>per adult    per child    per baby</td> <td></td> <td></td> </tr> </table>	<b>1 night to 6 nights</b>	<b>7 nights or longer</b>	<b>Add</b>	amount per night, per guest	amount		1.45    1.45    0	0		per adult    per child    per baby		
<b>1 night to 6 nights</b>	<b>7 nights or longer</b>	<b>Add</b>													
amount per night, per guest	amount														
1.45    1.45    0	0														
per adult    per child    per baby															
<input type="checkbox"/> Utilities charges	<input type="checkbox"/>	amount													
<input type="checkbox"/> Damage waiver	<input type="checkbox"/>	amount													
<input type="checkbox"/> Damage insurance	<input type="checkbox"/>	amount													
<input checked="" type="checkbox"/> Cancellation insurance	<input checked="" type="checkbox"/>	% of rent	0.25%												
<input type="checkbox"/> Bank fees	<input type="checkbox"/>	amount													
<input type="checkbox"/> Credit card fees		amount													

### 11.9.3 DISCOUNTS

#### 11.9.3.1 Definition

The `DISCOUNTS` object holds the following members:

Member	Data type	Description
<code>EARLY_BIRD</code>	object or null	The early bird discount setup (object) or null if no early discount is set up.
<code>LAST_MINUTE</code>	array of object	Last minute discounts setup (0 to 10 array items).
<code>SPECIAL</code>	array of object	The special (custom) discounts setup (0 to 20 array items).

The `EARLY_BIRD` object holds the following members:

Member	Data type	Description
<code>DAYS</code>	int	Minimum number of days difference between the booking date and the check-in date for the discount to apply. Range: 1 – 999.

PERCENT	amount	Range: "0.01" - "99.99"
---------	--------	-------------------------

Each `LAST_MINUTE` item is an `object` holding the following members, sorted by `DAYS` member value:

Member	Data type	Description
DAYS	int	Maximum number of days difference between the booking date and the check-in date for the discount to apply. Range: 1 – 99. The value of <code>DAYS</code> must be unique within the <code>LAST_MINUTE</code> array.
PERCENT	amount	Range: "0.01" - "99.99"

Each `SPECIAL` item is an `object` holding the following members, sorted by `CHECK_IN`:

Member	Data type	Description
NAME	string(1, 50)	The discount name.
PERCENT	amount	Range: "0.01" - "99.99"
VALID_FROM	date or null	The discount validity, based on reservation create date. Either both <code>VALID_FROM</code> and <code>VALID_TO</code> are <code>date</code> or both are <code>null</code> . They may be equal (one day discount validity).
VALID_TO		
CHECK_IN	date	The discount period boundaries. <code>CHECK_OUT</code> must be greater than <code>CHECK_IN</code> . <code>CHECK_IN</code> must be greater or equal to a previous special discount <code>CHECK_OUT</code> .
CHECK_OUT		

### 11.9.3.2 Examples

Example of the `DISCOUNTS` `object` with the early bird discount only.

```
{
  "EARLY_BIRD" : {
    "DAYS" : 90,
    "PERCENT" : "10.00"
  },
  "LAST_MINUTE" : [],
  "SPECIAL" : []
}
```

Example of the `DISCOUNTS` `object` with three last minute and one special discount.

```
{
  "EARLY_BIRD" : null,
  "LAST_MINUTE" : [
    {
      "DAYS" : 1,
      "PERCENT" : "15.00"
    },
    {
      "DAYS" : 2,
      "PERCENT" : "10.00"
    },
    {

```



```

        "DAYS"      :      3,
        "PERCENT"   :      "5.00"
    },
    "SPECIAL"      :      [
        {
            "NAME"       :      "New year booked on August",
            "PERCENT"    :      "15.00",
            "VALID_FROM" :      "2013-08-01",
            "VALID_TO"   :      "2013-08-31",
            "CHECK_IN"   :      "2013-12-30",
            "CHECK_OUT"  :      "2014-01-03"
        }
    ]
}

```

### 11.9.4 DEPOSIT

#### 11.9.4.1 Definition

The `DEPOSIT` object holds the following members:

Member	Data type	Description
UNIT	string	The unit and value (parameters, dependent on unit) used for computing the deposit amount.
VALUE		

The available `UNIT` and the matching data type of the `VALUE` member is defined in the following table:

UNIT	VALUE data type																	
"AMOUNT"	amount in the "0.01" - "999999.99" range																	
"PERCENT_RENT"	amount in the "0.01" - "100.00" range																	
"STAYLENGTH"	array (minimum 2 and maximum 40 items) of object, where each item is holding the following members: <table border="1" data-bbox="445 1364 1437 1615"> <thead> <tr> <th>Member</th> <th>Data type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>STAY_FROM</td> <td>time_interval</td> <td>The minimum stay duration the below UNIT and VALUE apply to. The "1 night" range is mandatory.</td> </tr> <tr> <td>UNIT</td> <td>string</td> <td rowspan="2">The unit and value (parameters, dependent on unit) used for computing the fee amount.</td> </tr> <tr> <td>VALUE</td> <td></td> </tr> </tbody> </table> <p>The available <code>UNIT</code> and the matching data type of the <code>VALUE</code> member is defined in the following table:</p> <table border="1" data-bbox="445 1709 1437 1850"> <thead> <tr> <th>UNIT</th> <th>VALUE data type</th> </tr> </thead> <tbody> <tr> <td>"AMOUNT"</td> <td>amount in the "0.00" - "999999.99" range</td> </tr> <tr> <td>"PERCENT_RENT"</td> <td>amount in the "0.01" - "100.00" range</td> </tr> </tbody> </table>	Member	Data type	Description	STAY_FROM	time_interval	The minimum stay duration the below UNIT and VALUE apply to. The "1 night" range is mandatory.	UNIT	string	The unit and value (parameters, dependent on unit) used for computing the fee amount.	VALUE		UNIT	VALUE data type	"AMOUNT"	amount in the "0.00" - "999999.99" range	"PERCENT_RENT"	amount in the "0.01" - "100.00" range
Member	Data type	Description																
STAY_FROM	time_interval	The minimum stay duration the below UNIT and VALUE apply to. The "1 night" range is mandatory.																
UNIT	string	The unit and value (parameters, dependent on unit) used for computing the fee amount.																
VALUE																		
UNIT	VALUE data type																	
"AMOUNT"	amount in the "0.00" - "999999.99" range																	
"PERCENT_RENT"	amount in the "0.01" - "100.00" range																	

#### 11.9.4.2 Examples

Example of the `DEPOSIT` object with a fixed amount deposit.

```
{  
  "UNIT"      : "AMOUNT",  
  "VALUE"     : "100.00"  
}
```

## 12 Property pricing computation algorithm overview

This chapter briefly exposes the Kigo Property pricing computation algorithm, such as used:

- in the Kigo application
- on Kigo powered websites
- by the Kigo REST API `computePricing` and `computePricingBulk` methods.

If your application is making pricing computations based on the property pricing setup such as provided by the `readPropertyPricingSetup` and `diffPropertyPricingSetup` API methods, then you might want to conform to this computation algorithm overview.

Whether it is a requirement to comply to the same computation rules, and whether all components (discounts, fees, ...) need to be included, depends on your contractual relationship with your property providers.

Also, please note that differences exist between precisions in floating point implementation on different computer platforms. Those differences might, under rare circumstances, result in slightly different computation results.

### 12.1 Input parameters

The following parameters are required by the computation algorithm (in addition to the pricing setup outlined in § 11.9 Property pricing):

<code>CREATE_DATE</code>	The reservation create date.
<code>CHECK_IN</code>	The check-in date.
<code>CHECK_OUT</code>	The check-out date.
<code>GUEST_ADULTS</code>	Number of adults.
<code>GUEST_CHILDREN</code>	Number of children.
<code>GUEST_BABIES</code>	Number of babies.

### 12.2 Rounding

Unless otherwise specified, the multiplication and division results (for instance, computing an amount based on a percentage), that result in floating point numbers, are rounded to two decimal digits away from zero (for instance, "2.455125" is rounded to "2.46" while "2.4545" is rounded to "2.45").

### 12.3 Computing base rent

The base rent is computed by iterating through the stay nights, from the `CHECK_IN` date (included) to the `CHECK_OUT` date (excluded), finding out the matching period, and summing the amount by the nightly or weekly unit amounts. Note that the nightly unit amount may be length-of-stay, day-of-week and number-of-guests dependent, while the weekly unit amount may be number-of-guests dependent.

The Kigo computation implementation fails computing the rent if not all the stay nights match the defined pricing periods, or if the check-in or check-out do not match the turnover day of a weekly-only period.

The amount produced by this computation is equivalent to `RENT_BASE_AMOUNT` returned by the `computePricing` method.

During the above stay nights iteration, a nightly raw rent amount is remembered for each night, to be used as one of the components of the special discounts computation (see § 12.6.3 Special discounts). For weekly-

only periods, that amount is equal to the weekly unit amount divided by 7, not rounded.

## 12.4 Computing fees included-in-rent

The amount of fees included in rent is computed by iterating through the fees (in no particular order) having the `INCLUDE_IN_RENT` member set to `true`. The total amount obtained is equivalent to `RENT_FEES_AMOUNT` returned by the `computePricing` method.

A per-night fees-included-in-rent amount is remembered for being used as one of the components of the special discounts computation (see § 12.6.3 Special discounts). It is obtained by dividing the total amount obtained above by the number of stay nights. The division result is not rounded.

## 12.5 Computing fees not-included-in-rent

The amount of fees not included in rent is computed by iterating through the fees (in no particular order) having the `INCLUDE_IN_RENT` member set to `false`. The amount obtained is equivalent to `FEES_AMOUNT` returned by the `computePricing` method.

## 12.6 Computing discounts amount

### 12.6.1 Early bird discount

The “early bird” discount applies if it is if set-up (`EARLY_BIRDS` is not `null`) and if the number of days between the `CREATE_DATE` and the `CHECK_IN` date is greater or equal to the `DAYS` member of the `EARLY_BIRDS` object. The `PERCENT` is applied to the sum of `RENT_BASE_AMOUNT` and `RENT_FEES_AMOUNT`.

### 12.6.2 Last minute discount

The “last minute” discount applies if the number of days between the `CREATE_DATE` and the `CHECK_IN` date is less or equal to one of (the first one that matches) `DAYS` member of the `LAST_MINUTE` array. The `PERCENT` is applied to the sum of `RENT_BASE_AMOUNT` and `RENT_FEES_AMOUNT`.

### 12.6.3 Special discounts

For each special discount defined, the following processing is applied:

1. If the discount validity does not match (`VALID_FROM` and `VALID_TO` are not `null`, and `CREATE_DATE` is not within the period defined by `VALID_FROM` to `VALID_TO`, both included), then the discount is discarded
2. The discount amount is computed by iterating through the stay nights. For every stay nights that is within the period defined by `CHECK_IN` – `CHECK_OUT` (from discount settings) period, the `PERCENT` is applied (without rounding) to the sum of the previously computed nightly raw rent amount for that night (see § 12.3 Computing base rent) and of the previously computed per-night fees-included-in-rent amount (see § 12.4 Computing fees included-in-rent)
3. Amounts obtained in the previous step are summed and then rounded.

### 12.6.4 Total

All the three discounts (if applicable) are summed into a total discounts amount, equivalent to `RENT_DISCOUNTS_AMOUNT` returned by the `computePricing` method.



### 12.7 Computing the total rent amount

The total rent amount is obtained by summing the base rent amount, the fees-included-in-rent amount and the discounts total amount. It is equivalent to `RENT_AMOUNT` returned by the `computePricing` method.

### 12.8 Computing the total due amount

The total due amount is obtained by summing the total rent amount with the fees not-included-in-rent amount. It is equivalent to `TOTAL_AMOUNT` returned by the `computePricing` method.

### 12.9 Computing the deposit amount

The computed deposit is equivalent to `DEPOSIT_AMOUNT` returned by the `computePricing` method.

## 13 References

### 13.1 ISO 3166

Official ISO website

[http://www.iso.org/iso/country\\_codes.htm](http://www.iso.org/iso/country_codes.htm)

Wikipedia ISO 3166-1 article

[http://en.wikipedia.org/wiki/ISO\\_3166-1](http://en.wikipedia.org/wiki/ISO_3166-1)

Wikipedia ISO 3166-1 alpha-2 article

[http://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2)

### 13.2 JSON

Unofficial JSON resources website by the creator of JSON, Douglas Crockford

<http://www.json.org>

Wikipedia JSON article

<http://en.wikipedia.org/wiki/JSON>

JSON RFC

<http://tools.ietf.org/html/rfc4627>

### 13.3 base64

Base64 data encoding RFC

<http://tools.ietf.org/html/rfc4648>

### 13.4 Latitude and longitude WGS 84 decimal degree notation

Wikipedia Longitude and latitude articles

<http://en.wikipedia.org/wiki/Longitude>

<http://en.wikipedia.org/wiki/Latitude>

Wikipedia WGS 84 article

[http://en.wikipedia.org/wiki/World\\_Geodetic\\_System](http://en.wikipedia.org/wiki/World_Geodetic_System)

Wikipedia Decimal degrees notation article

[http://en.wikipedia.org/wiki/Decimal\\_degrees](http://en.wikipedia.org/wiki/Decimal_degrees)