

Kigo

CHANNELS API - V1

API revision

4

Document revision

14

Contact

api@kigo.net

1 Revision history

API Rev	Document		Date	Changes
	Rev	Status		
1	1	DRAFT	22 sep 2014	First implementation DRAFT, based on Kigo Rental Agency REST API - V1 Revision 14 (Documentation Revision 20).
1	2	ERRATA	10 feb 2015	Adding www prefix to API host. Commenting the values for PROP_INSTANT_BOOK. Commenting changes for PROP_BEDS and PROP_BED_TYPES
2	3		18 feb 2015	Adding methods diffPropertyAvailability, listPropertyAvailability and listPropertyAvailabilityBulk. Deprecating diffPropertyCalendarReservations. Reintroducing listPropertyCalendarReservations .
2	4		03 mar 2015	Photos read through readPropertyPhotoFile are no longer limited to JPEG format. Maximum length of PHOTO_ID string changed to 1024.
2	5		27 may 2015	In this version of the API, the PRICING_LIST member of the response of diffPropertyPricingSetup will always return null.
3	6		19 feb 2016	In this revision of the API, a new method diffPropertyContent has been added, a new object MIN_STAY has been added, the PROP_INFO and FEES objects have been expanded and the RES_GUEST object is mandatory for createConfirmedReservation and should be well informed if it's included on updateReservation.
3	7		12 abr 2016	In this revision of the API, the usage of the readPropertyPricingSetup and computePricing methods has been revised. The listProperties2 method has been updated to provide you ONLY active properties instead of everything.
3	8		25 jul 2016	In this revision, the response of createConfirmedReservation and updateReservation methods have been expanded, adding error messages to clarify why the operations have failed.
3	9		19 ago 2016	In this revision, the response of the computePricing and the computePricingBulk methods have been expanded, adding the amount for the optional fees. Also, the PROP_INFO object has been expanded.
4	10		18 apr 2017	Reservation status (RES_STATUS) CANCELED has change to CANCELED_CONFIRMED. Commenting changes for PROP_TOILETS and PROP_ELEVATOR. The FEES object has been improved. DISCOUNTS member of PRICING object has been deprecated and is returning an empty array.

API Rev	Document		Date	Changes
	Rev	Status		
4	11		18 jun 2017	The FEES object has been improved. New method, readPropertyPricingSetup2, added. New objects, TAXES and DEPOSITS added.
4	12		24 jul 2017	The FEES, TAXES and DEPOSITS objects have been clarified.
4	13		01 sep 2017	PROP_INFO has been updated: PROP_BEDROOMS, PROP_BEDS, PROP_BED_TYPES are no longer deprecated. PROP_UNITS object has been added to the Property information.
4	14		20 oct 2017	Field CONDITION_VALUES_UNIT added to clarify the application of MIN_CONDITION_VALUE and MAX_CONDITION_VALUE. TURN_DAY_RULES array has been added.

2 Incompatibilities with the legacy RENTAL AGENCY REST API – V1 (Rev. 14, Doc Rev. 20)

Item	Explanation
Host and base path	The previous host <code>http://app.kigo.net</code> becomes <code>https://www.kigoapis.com</code> The previous base path <code>/api/ra/v1/</code> becomes <code>/channels/v1/</code>
Authentication	The previous API used Basic HTTP Authentication. This new API uses a <code>subscription-key</code> query string parameter instead.
HTTP Codes	The 409 and 420 codes are not returned anymore in case of call rate limit being exceeded. 429 is returned instead (as already suggested in previous revisions). 404 is returned in case wrong HTTP method (The calls should always use the <code>POST</code> method) instead of 405.
<code>computePricing</code> <code>computePricingBulk</code> methods	The <code>RES_CREATE</code> parameter is deprecated and being completely ignored in this new implementation. It defaults to today's date.
<code>readReservation</code> <code>createConfirmedReservation</code> <code>updateReservation</code> <code>computePricing</code> methods	<code>RES_N_BABIES</code> input is deprecated and will be ignored. The output always defaults to 0.
<code>readPropertyPricingSetup</code> method	A same <code>FEE_TYPE_ID</code> might appear multiple times.
UDPA and UDRA	User-defined property attributes (<code>PROP_UDPA</code>) and user-defined reservations attributes (<code>RES_UDRA</code>) are now deprecated. Their input will be ignored and the output will always be empty arrays.
Removed methods	The following methods were removed: <code>listOwners</code> , <code>readOwner</code> , <code>addOwner</code> , <code>updateOwner</code> , <code>uploadPropertyPhotoFile</code> , <code>addProperty</code> , <code>updateProperty</code> , <code>updatePropertyPricingSetup</code> , <code>listUserDefinedPropertyAttributes</code> , <code>listUserDefinedReservationAttributes</code> and <code>listBookingSources</code> .
Multi-unit property availability	The following methods have been introduced to support multi-unit property availability: <code>diffPropertyAvailability</code> , <code>listPropertyAvailability</code> and <code>listPropertyAvailabilityBulk</code> . <code>diffPropertyCalendarReservations</code> became deprecated, as it is no longer suited for building availability calendars. The previously deprecated <code>listPropertyCalendarReservations</code> was reintroduced.

Item	Explanation
PROP_INSTANT_BOOK	This member returned by <code>readProperty2</code> has the opposite behaviour than in Kigo 1 API. In this implementation of the API, <code>TRUE</code> means that the property can be booked instantly, while <code>FALSE</code> means that it should not.
Photos	Photos read through <code>readPropertyPhotoFile</code> are no longer limited to JPEG format. Maximum length of <code>PHOTO_ID</code> string changed to 1024.
<code>diffPropertyPricingSetup</code>	The <code>PRICING_LIST</code> member of the response will always return <code>null</code> .

3 Contents

1	Revision history	2
2	Incompatibilities with the legacy	
	RENTAL AGENCY REST API – V1 (Rev. 14, Doc Rev. 20)	4
3	Contents	6
4	Glossary	8
5	Preface	9
	5.1 Intended audience	9
	5.2 Reporting issues	9
	5.3 Copyright notice	9
6	API version and revision numbers	10
	6.1 API version number	10
	6.2 API revision number	10
7	Implementation details	11
	7.1 Communication protocol	11
	7.2 Authentication	11
	7.3 Rate limiting policy	11
	7.4 Bandwidth limiting policy	11
	7.5 Character encoding	11
	7.6 Data encoding	11
	7.7 Data types	12
	7.7.1 JSON native data types	12
	7.7.2 Extended data types	12
8	Invoking API methods	15
	8.1 HTTP request	15
	8.2 HTTP reply	15
	8.2.1 Status code	16
	8.2.2 Content body	16
	8.2.3 Standard method reply object	16
9	diff methods family	18
	9.1 How does it work?	18
	9.2 Examples	18
	9.2.1 Initial invocation and a first diff	18
	9.2.2 Recovering from an error	19
10	API methods reference	20
	10.1 Miscellaneous	20
	10.1.1 Ping	20
	10.2 Managing properties	21
	10.2.1 listProperties2	21
	10.2.2 readProperty2	21
	10.2.3 readPropertyPhotoFile	22
	10.2.4 diffPropertyContent	23
	10.3 Managing property pricing	24
	10.3.1 readPropertyPricingSetup	24
	10.3.2 readPropertyPricingSetup2	25
	10.3.3 diffPropertyPricingSetup	27
	10.4 Managing property availability and reservations	28
	10.4.1 diffPropertyAvailability	28
	10.4.2 listPropertyAvailability	29
	10.4.3 listPropertyAvailabilityBulk	30
	10.4.4 listPropertyCalendarReservations	31
	10.4.5 readReservation	34
	10.4.6 createConfirmedReservation	38

10.4.7	updateReservation	41
10.4.8	cancelReservation	43
10.5	Real-time pricing computation	44
10.5.1	computePricing	44
10.5.2	computePricingBulk	47
10.6	Account and system inventory	48
10.6.1	listKigoCountries	48
10.6.2	listKigoPropertyTypes	49
10.6.3	listKigoPropertyBedTypes	49
10.6.4	listKigoPropertyAmenities	50
10.6.5	listKigoPropertyActivities	51
10.6.6	listKigoFeeTypes	52
10.7	Deprecated API methods	52
10.7.1	diffPropertyCalendarReservations	52
10.7.2	listCountries	53
10.7.3	listProperties	53
10.7.4	readProperty	53
11	Kigo data dictionary	54
11.1	Reservation status (RES_STATUS)	54
11.2	Online booking transaction state (OB_STATE)	54
11.3	Country ISO 3166-1 alpha-2 codes	54
11.4	Property provider	55
11.4.1	PROP_PROVIDER	55
11.5	Property type	55
11.6	Property bed type	55
11.7	Property amenity	55
11.8	Property activity	55
11.9	Property information	56
11.9.1	PROP_INFO	56
11.9.2	PROP_UNITS	59
11.9.3	PROP_RATE	59
11.9.4	PROP_PHOTOS	60
11.10	Property pricing	60
11.10.1	RENT	61
11.10.2	FEES	66
11.10.3	TAXES	69
11.10.4	DISCOUNTS	71
11.10.5	DEPOSIT	73
11.10.6	DEPOSITS	74
11.10.7	MIN_STAY	76
11.10.8	TURN_DAY_RULES	77
11.10.9	AVAILABILITY_RULES	78
12	References	79
12.1	ISO 3166	79
12.2	JSON	79
12.3	base64	79
12.4	Latitude and longitude WGS 84 decimal degree notation	79

4 Glossary

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

5 Preface

The Kigo CHANNELS API - V1 provides automated access from remote computer systems to a subset 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 the Kigo database into your application or website.

5.1 Intended audience

This documentation is aimed at the IT department of Kigo customers and partners building applications or websites that need to interact with the Kigo application through the CHANNELS API - V1.

5.2 Reporting issues

Please, send any API related questions and issues to api@kigo.net.

5.3 Copyright notice

Kigo exclusively owns the intellectual property in this documentation. You acknowledge that you must not perform any act which infringes the copyright or any other intellectual property rights of Kigo.

6 API version and revision numbers

This is the documentation for the CHANNELS API - V1 version 3, revision 4.

6.1 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.

6.2 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 usually maintained.

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

7 Implementation details

7.1 Communication protocol

The Kigo CHANNELS API - V1 is an API based on the HTTP protocol, JSON data encoding and Unicode character encoding, over the secure HTTPS transport.

7.2 Authentication

When invoking API methods, clients must authenticate using the received `subscription-key` query string parameter.

7.3 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 § 8.2.1 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.

7.4 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 § 8.2.1 Status code) during a couple of seconds, and until the bandwidth usage within the defined period goes under the limit.

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 a huge amount of data) in order not to trigger the bandwidth limiting policy.

7.5 Character encoding

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

7.6 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 § 7.7.1 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/>.

7.7 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.

7.7.1 JSON native data types

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

7.7.2 Extended data types

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

Type	Description
int	Positive integer value represented as <code>number</code> . Range: 0 - 2147483647.
string(X)	string having exactly X characters.
string(X, Y)	string having X up to Y characters (both inclusive).
date	<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.
time_hh	<code>string(5)</code> containing half-hour (minutes 0 and 30 only) duration or daytime representation in 24-hour "HH:MM" format. Examples of valid <code>time_hh</code> values: <ul style="list-style-type: none">• "00:00"• "09:30"• "21:00"• "48:00"

amount	<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"> • optional minus sign, followed by • one or more digits (if there is more than one digit, then the first one must be different from zero), followed by • the decimal point, followed by • exactly two digits. <p>Examples of valid <code>amount</code> values:</p> <ul style="list-style-type: none"> • <code>"0.00"</code> • <code>"-3.00"</code> • <code>"10.49"</code> 									
decimal_degree	<p>Decimal degrees used to express latitude and longitude geographic coordinates (see 12.4 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"> • optional minus sign, followed by • one to three digits (if there is more than one digit, then the first one must be different from zero), followed by • the decimal point, followed by • exactly six digits. <p>Examples of valid <code>decimal_degree</code> values:</p> <ul style="list-style-type: none"> • <code>"0.000000"</code> • <code>"-80.798512"</code> • <code>"178.120000"</code> 									
latlng	<p>Latitude and longitude geographic coordinates (see 12.4 Latitude and longitude WGS 84 decimal degree notation), expressed in decimal degrees.</p> <p>It is represented by an <code>object</code> holding the following members:</p> <table border="1" data-bbox="386 1568 1439 1765"> <thead> <tr> <th>Member</th> <th>Type</th> <th>Range</th> </tr> </thead> <tbody> <tr> <td>LATITUDE</td> <td><code>decimal_degree</code></td> <td><code>"-90.000000"</code> to <code>"90.000000"</code> (both inclusive)</td> </tr> <tr> <td>LONGITUDE</td> <td><code>decimal_degree</code></td> <td><code>"-179.999999"</code> to <code>"180.000000"</code> (both inclusive)</td> </tr> </tbody> </table>	Member	Type	Range	LATITUDE	<code>decimal_degree</code>	<code>"-90.000000"</code> to <code>"90.000000"</code> (both inclusive)	LONGITUDE	<code>decimal_degree</code>	<code>"-179.999999"</code> to <code>"180.000000"</code> (both inclusive)
Member	Type	Range								
LATITUDE	<code>decimal_degree</code>	<code>"-90.000000"</code> to <code>"90.000000"</code> (both inclusive)								
LONGITUDE	<code>decimal_degree</code>	<code>"-179.999999"</code> to <code>"180.000000"</code> (both inclusive)								

<code>time_interval</code>	<p>object containing exactly two members, <code>UNIT</code> (string) and <code>NUMBER</code> (int).</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">• 1 - 27 for the <code>"NIGHT"</code> <code>UNIT</code>• 1 - 11 for the <code>"MONTH"</code> <code>UNIT</code>• 1 - 5 for the <code>"YEAR"</code> <code>UNIT</code>
----------------------------	--

8 Invoking API methods

API methods are invoked by making an HTTPS POST request on the Kigo CHANNELS API - V1 servers.

8.1 HTTP request

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

<https://www.kigoapis.com/channels/v1/method?subscription-key=xxxxxx>,

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

Authentication keys 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 /channels/v1/ping?subscription-key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

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

Note that the raw post data size may not exceed 16 megabytes.

8.2 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, ...).

8.2.1 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.
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).
429	The API rate limiting policy and/or bandwidth limiting policy were triggered (see § 7.3 Rate limiting policy and § 7.4 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.

8.2.2 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.

8.2.3 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
API_VERSION	int	The API version number.
API_REVISION	int	The API revision number.
API_METHOD	string(1, 50)	The name of the invoked API method.
API_CALL_ID	string(1, 80)	A unique API call (invocation) number. May be used for troubleshooting.
API_RESULT_CODE	string(1, 20)	Result code returned by API method. See API method documentation for further information.
API_RESULT_TEXT	string(0, 1024)	Human readable message related to <code>API_RESULT_CODE</code> , not intended to be parsed and understood by software. May be used for troubleshooting.
API_REPLY		Additional data returned by the API method. In many situations, this value is not-null if <code>API_RESULT_CODE</code> is set to <code>E_OK</code> , and is null otherwise. See API method documentation for further information.
API_DEPRECATED (optional)	string(0, 65535)	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"      : 3,
  "API_REVISION"     : 4,
  "API_METHOD"       : "cancelReservation",
  "API_CALL_ID"      : "512e84d998f2a06df9ddc10fd7b6d3aa",
  "API_RESULT_CODE"  : "E_NOSUCH",
  "API_RESULT_TEXT"  : "The reservation was not found.",
  "API_REPLY"        : null
}
```

9 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.

9.1 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.

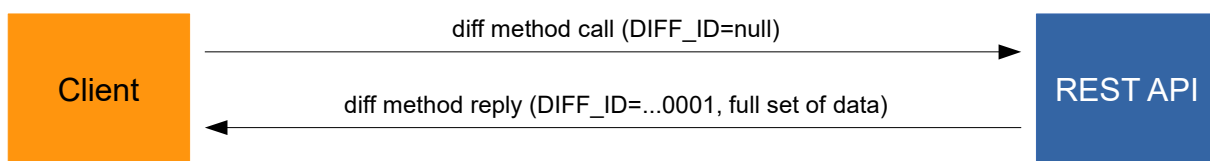
When an invalid or expired `DIFF_ID` is provided to an API method, the full list of items is returned in the result, as if a `null` `DIFF_ID` was provided.

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

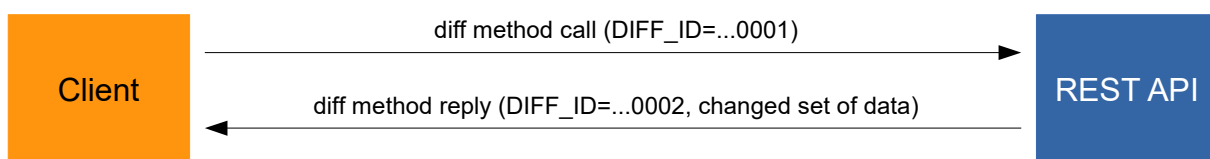
9.2 Examples

9.2.1 Initial invocation and a first diff

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

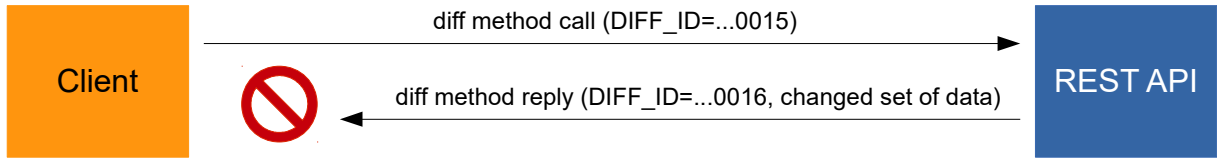


The client makes a new call using the previously obtained `DIFF_ID`.

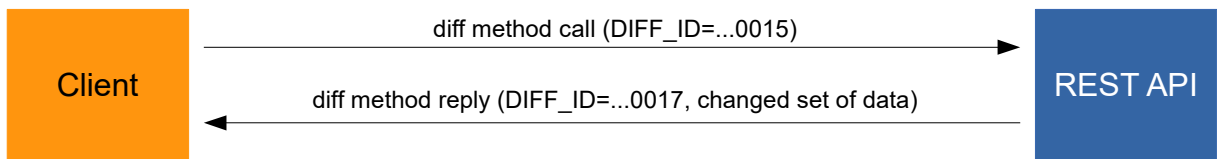


9.2.2 Recovering from an error

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)`.



10 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.

10.1 Miscellaneous

10.1.1 Ping

10.1.1.1 Usage

This is a very simple API method that may be used for testing purpose during development process, or for monitoring the Kigo CHANNELS API - V1 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.

10.1.1.2 Input

This method accepts data of any type.

10.1.1.3 Result and reply

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

10.1.1.4 Examples

Sample ping request

HTTP request:

```
POST /channels/v1/ping?subscription-key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
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"      : 3,
  "API_REVISION"     : 4,
  "API_METHOD"       : "ping",
  "API_CALL_ID"      : "f6b22b8e3ab28dbea873ed437c95351d",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        : "Are you there?"
}
```

10.2 Managing properties

Properties in Kigo are involved in almost every transaction, thus it is a requirement for all API users to manage Kigo properties to some extent. The main use cases are covered by the API, from mapping properties to synchronizing property information between Kigo and third-party systems.

10.2.1 listProperties2

10.2.1.1 Usage

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

10.2.1.2 Input

This method requires input data to be `null`.

10.2.1.3 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` holding the following members:

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>PROP_PROVIDER</code>	<code>object</code> or <code>null</code>	The current property provider information <code>object</code> (see § 11.4.1 <code>PROP_PROVIDER</code>), or <code>null</code> if the property is not currently being provided (not active). See § 11.4 Property provider for information on property providers.
<code>PROP_NAME</code>	<code>string(1, 100)</code>	The property name.

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).

10.2.2 readProperty2

10.2.2.1 Usage

This method returns property information.

10.2.2.2 Input

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

Member	Data type	Description
PROP_ID	int	The Kigo property id.

10.2.2.3 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 property information.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.
E_NOSUCH	The requested property was not found. <code>API_REPLY</code> is null.

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

Member	Data type	Description
PROP_ID	int	The Kigo property id.
PROP_PROVIDER	object or null	The current property provider information object (see § 11.4.1 PROP_PROVIDER), or null if the property is not currently being provided (not active). 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_INFO	object	General property information. See § 11.9.1 PROP_INFO.
PROP_UNITS	object	Property units information. See § 11.9.2 PROP_UNITS.
PROP_RATE	object	Property rate information. See § 11.9.3 PROP_RATE.
PROP_PHOTOS	array of object	List of property JPEG photos. See § 11.9.4 PROP_PHOTOS.
PROP_UDPA (deprecated in revision 1)	array	Deprecated in revision 1. An empty array is returned.

10.2.3 readPropertyPhotoFile

10.2.3.1 Usage

This method returns the binary content of a property photo file. You may invoke this method to retrieve property photos returned by a previous `readProperty2` 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.

10.2.3.2 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, 1024)	Property photo identifier obtained in the PROP_PHOTOS member of the readProperty2 method reply.

10.2.3.3 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 string that represents a base64 (RFC 4648) encoded photo file content.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.
E_NOSUCH	The specified property or photo was not found. <code>API_REPLY</code> is null.

10.2.4 diffPropertyContent

This method is a member of the diff methods family.

10.2.4.1 Usage

The purpose of this method is to retrieve the list of properties that had their contents changed since the last call to the method.

Recommended poll time: 6 hours.

10.2.4.2 Input

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

Member	Data type	Description
DIFF_ID	string(64) or null	See § 9 diff methods family.

10.2.4.3 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 content 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 properties ids:

Member	Data type	Description
DIFF_ID	string(64)	See § 9 diff methods family.
PROP_ID	array of int	List of properties with new or modified content. The list may contain properties that are no longer available from your account but that were returned by a previous diffPropertyContent call.

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).

10.3 Managing property pricing

Prices are set up at the property level or in other words every property in Kigo has its own pricing settings. Due to the Kigo's complete system with different factors such as rate calculation rules, booking settings or the Kigo Revenue Management, all of them have a direct impact on the pricing information and can often change it. Thus the API can only provide the closest approximation of the property pricing information at any time it's called. To add, in most of the cases the closest approximation yet refers to the exact price.

This closest approximation should be used just as a reference of the property prices information. The API will provide you this information via the methods below readPropertyPricingSetup or readPropertyPricingSetup2.

10.3.1 readPropertyPricingSetup

10.3.1.1 Usage

This method provides only the closest approximation of the pricing setup for the specified property, **but will not calculate the final price**. To get the final price, the API methods computePricing or computePricingBulk must be invoked.

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

See also: § 10.3.3 diffPropertyPricingSetup.

10.3.1.2 Input

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

Member	Data type	Description
PROP_ID	int	The Kigo property id.

10.3.1.3 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 an approximation of the property pricing setup.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.

E_NOSUCH	The property was not found, or is unavailable (i.e. was deactivated). API_REPLY is null.
----------	--

API_REPLY on E_OK result

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

Member	Data type	Description
PRICING	object or null	Holds an approximation of the property pricing setup, or null if the pricing is not configured 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).

The PRICING object holds the following members:

Member	Data type	Description
CURRENCY	string(3)	Pricing setup currency ISO 4217 code. This is the currency of the Kigo account the property pricing setup is being read from.
RENT	object	See § 11.10.1 RENT.
FEES	object	See § 11.10.2 FEES.
DISCOUNTS	object deprecated, it will be void	See § 11.10.4 DISCOUNTS.
DEPOSIT	object or null	See § 11.10.5 DEPOSIT. null If no deposit is configured.
DEPOSITS_MAY_APPLY	bool	It indicates that there are deposits that can be applied to the price.
DEPOSITS	array of object	See § 11.10.6 DEPOSITS. Empty array if no deposit are configured.
MIN_STAY	object	See § 11.10.7 MIN_STAY
AVAILABILITY_RULES	object	See § 11.10.9 AVAILABILITY_RULES

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).

10.3.2 readPropertyPricingSetup2

10.3.2.1 Usage

This method provides only the closest approximation of the pricing setup for the specified property, **but will not calculate the final price**. To get the final price, the API methods computePricing or computePricingBulk must be invoked.

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

See also: § 10.3.3 diffPropertyPricingSetup.

10.3.2.2 Input

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

Member	Data type	Description
PROP_ID	int	The Kigo property id.

10.3.2.3 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> holds an approximation of the property pricing setup.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.
E_NOSUCH	The property was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is null.

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 an approximation of the property pricing setup, or null if the pricing is not configured 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).

The `PRICING` `object` holds the following members:

Member	Data type	Description
CURRENCY	string(3)	Pricing setup currency ISO 4217 code. This is the currency of the Kigo account the property pricing setup is being read from.
RENT	<code>object</code>	See § 11.10.1 RENT.
FEES	<code>object</code>	See § 11.10.2 FEES.
TAXES	<code>object</code>	See § 11.10.3 TAXES.
DISCOUNTS	<code>object</code> deprecated, it will be void	See § 11.10.4 DISCOUNTS.

DEPOSITS_MAY_APPLY	bool	It indicates that there are deposits that can be applied to the price.
DEPOSITS	array of object	See § 11.10.6 DEPOSITS. Empty array if no deposit are configured.
MIN_STAY	object	See § 11.10.7 MIN_STAY
TURN_DAY_RULES	object	See § 11.10.8 TURN_DAY_RULES
AVAILABILITY_RULES	object	See § 11.10.9 AVAILABILITY_RULES

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).

10.3.3 diffPropertyPricingSetup

This method is a member of the diff methods family.

10.3.3.1 Usage

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

Recommended poll time: every 30 minutes to 24 hours.

10.3.3.2 Input

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

Member	Data type	Description
DIFF_ID	string(64) or null	See § 9 diff methods family.

10.3.3.3 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 § 9 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 diffPropertyPricingSetup call.
PRICING_LIST	null	In this version of the Channels API this member is always null

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).

10.4 Managing property availability and reservations

The following methods provide the ability to efficiently retrieve property availability calendars and to manage reservations (read, create, update, cancel).

Methods related to availability are capable of dealing with multi-unit properties.

10.4.1 diffPropertyAvailability

This method is a member of the diff methods family.

10.4.1.1 Usage

The purpose of this method is to retrieve the list of properties that had their availability calendars changed since a last call to the method.

Call(s) to listPropertyAvailability or listPropertyAvailabilityBulk should follow, to retrieve up-to-date availability calendars for all the identifiers retrieved in the PROP_ID array.

Recommended poll time: every 2 minutes to 4 hours.

10.4.1.2 Input

The method expects an object holding the following member:

Member	Data type	Description
DIFF_ID	string(64) or null	See § 9 diff methods family. A null value results in a PROP_ID array holding all property identifiers in the account.

10.4.1.3 Result and reply

The method may return the following API_RESULT_CODE values:

Result	Description
E_OK	The call was successful, API_REPLY contains a new DIFF_ID and a list of PROP_ID.
E_INPUT	Invalid input data. API_REPLY is null.

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

Member	Data type	Description
DIFF_ID	string(64)	See § 9 diff methods family.
PROP_ID	array of int	Unsorted list of PROP_ID.

10.4.2 listPropertyAvailability

10.4.2.1 Usage

This method retrieves the availability calendar of a property for a specific range of dates.

10.4.2.2 Input

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

Member	Data type	Description
PROP_ID	int	The Kigo property id.
LIST_START_DATE	date	The requested period start date. This date can be up to 60 months in the future.
LIST_END_DATE	date	The requested period end date. This date can be up to 24 months (2 years) later than LIST_START_DATE.

10.4.2.3 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 the availability of the property.
E_INPUT	Invalid input data. <code>API_REPLY</code> is null.
E_NOSUCH	The requested property was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is null.

On `E_OK` result, `API_REPLY` holds an `object` containing:

Member	Data type	Description
TOTAL_PROPERTY_UNITS	int	Total number of units of this property.
AVAILABILITY	array of object	Available units and maximum length of stay for each date of the queried range.

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

Member	Data type	Description
DATE	date	A specific date in the range from <code>LIST_START_DATE</code> to <code>LIST_END_DATE</code> . The dates are in ascendant order.

AVAILABLE_UNITS	int	The units available on the day specified by DATE. The value can range from 0 to TOTAL_PROPERTY_UNITS.
MAX_LOS	int	The maximum length in days for which a reservation starting on the DATE can be created (the continuous availability of a single unit from that DATE).

10.4.2.4 Examples

List the availability

HTTP request:

```
POST /channels/v1/listPropertyAvailability HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

{
  "PROP_ID"      : 17784,
  "LIST_START_DATE" : "2015-03-21",
  "LIST_END_DATE"  : "2015-03-23"
}
```

HTTP reply:

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

{
  "API_VERSION"      : 3,
  "API_REVISION"     : 4,
  "API_METHOD"       : "listPropertyAvailability",
  "API_CALL_ID"      : "c8911dee7ab80d21565732876f7f3cea",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "",
  "API_REPLY"        :
  {
    "TOTAL_PROPERTY_UNITS" : 2,
    "AVAILABILITY"        :
    [
      {
        "DATE"           : "2015-03-21",
        "AVAILABLE_UNITS" : 2,
        "MAX_LOS"        : 10
      },
      {
        "DATE"           : "2015-03-22",
        "AVAILABLE_UNITS" : 2,
        "MAX_LOS"        : 9
      }
    ]
  }
}
```

10.4.3 listPropertyAvailabilityBulk

10.4.3.1 Usage

This method enables the client system to retrieve the availability calendar for up to 10 properties at once, for a specific date range.

10.4.3.2 Input

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

Member	Data type	Description
<code>PROP_ID</code>	array of <code>int</code>	An <code>array</code> holding from 1 up to 10 Kigo property IDs.
<code>LIST_START_DATE</code>	<code>date</code>	The requested period start date. This date can be up to 60 months in the future.
<code>LIST_END_DATE</code>	<code>date</code>	The requested period end date. This date can be up to 24 months (2 years) later than <code>LIST_START_DATE</code> .

10.4.3.3 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 the list of property availability objects.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is <code>null</code> .
<code>E_NOSUCH</code>	One or more of the requested properties was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is <code>null</code> .

On `E_OK` result, `API_REPLY` holds an `array` having one element per `PROP_ID` requested in the input.

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

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>REPLY</code>	<code>object</code>	Contains the total number of units of this property and the <code>array</code> with the availability, identical the to <code>API_REPLY</code> <code>object</code> defined in § 10.4.2.3.

10.4.4 listPropertyCalendarReservations

10.4.4.1 Usage

This method returns the list of all `CONFIRMED` and `HOLD` property reservations that match a specified time period, including reservations that belong to other agencies or to the property owner.

10.4.4.2 Input

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

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>LIST_START_DATE</code>	<code>date</code>	The requested time period start date.
<code>LIST_END_DATE</code>	<code>date</code>	The requested time period end date.

The following restrictions apply:

- The minimum allowed `LIST_START_DATE` is "2000-01-01"
- The maximum allowed `LIST_START_DATE` is the date of the day plus 60 months (5 years)
- The requested period may not cover more than 24 months (2 years)

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

Note that this method will also list reservations that end (check-out) on `LIST_START_DATE` and reservations that start (check-in) on `LIST_END_DATE`.

10.4.4.3 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 the list of matched reservations, if any.
<code>E_INPUT</code>	Invalid input data. <code>API_REPLY</code> is null.
<code>E_NOSUCH</code>	The requested property was not found, or is unavailable (i.e. was deactivated). <code>API_REPLY</code> is null.

On `E_OK` result, `API_REPLY` holds an array of reservations ordered by the check-in date. Each reservation is represented by an object with following members:

Member	Data type	Description
<code>RES_ID</code>	int	The reservation id.
<code>RES_STATUS</code>	string	The reservation status (<code>CONFIRMED</code> or <code>HOLD</code>).
<code>RES_IS_FOR</code>	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>).
<code>RES_CHECK_IN</code>	date	The check-in date.
<code>RES_CHECK_OUT</code>	date	The check-out date.

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

10.4.4.4 Examples

List of reservation

HTTP request:

```
POST /channels/v1/listPropertyCalendarReservations HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

{
  "PROP_ID" : 17784,
  "LIST_START_DATE" : "2011-07-21",
```



```
"LIST_END_DATE" : "2011-09-30"
}
```

HTTP reply:

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

```
{
  "API_VERSION" : 3,
  "API_REVISION" : 4,
  "API_METHOD" : "listPropertyCalendarReservations",
  "API_CALL_ID" : "b8911dee7ab80d20565732876f7f3ced",
  "API_RESULT_CODE" : "E_OK",
  "API_RESULT_TEXT" : "",
  "API_REPLY" :
  [
    {
      "RES_ID" : 24985,
      "RES_IS_FOR" : true,
      "RES_CHECK_IN" : "2011-07-12",
      "RES_CHECK_OUT" : "2011-08-02"
    },
    {
      "RES_ID" : 11487,
      "RES_IS_FOR" : false,
      "RES_CHECK_IN" : "2011-08-14",
      "RES_CHECK_OUT" : "2011-08-12"
    }
  ]
}
```

Invalid time period

HTTP request:

```
POST /channels/v1/listPropertyCalendarReservations?subscription-
key=4nHm479ZZ267B99235GhN273c7mI4PHw
HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

{
  "PROP_ID"      :      17784,
  "LIST_START_DATE" :      "2007-07-21",
  "LIST_END_DATE"  :      "2014-09-30"
}
```

HTTP reply:

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

{
  "API_VERSION"      :      3,
  "API_REVISION"     :      4,
  "API_METHOD"       :      "listPropertyCalendarReservations",
  "API_CALL_ID"      :      "b5bc669a8d9dbb68c7bc4a1af1deabd3",
  "API_RESULT_CODE"  :      "E_INPUT",
  "API_RESULT_TEXT"  :      "The requested range exceeds 24 months",
  "API_REPLY"        :      null
}
```

10.4.5 readReservation

10.4.5.1 Usage

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

10.4.5.2 Input

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

Member	Data type	Description
RES_ID	int	The Kigo reservation id.

10.4.5.3 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 null.
E_NOSUCH	The requested reservation was not found, or does not belong to your Rental Agency Kigo account. <code>API_REPLY</code> is null.

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> (deprecated in revision 1)	<code>int</code>	The number of baby guests. Value is forced to 0.
<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)</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.
<code>BOOKING_SOURCE_ID</code>	<code>int</code> or <code>null</code>	The reservation booking source id (see § Error: No se encuentra la fuente de referencia Error: No se encuentra la fuente de referencia) 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</code> or <code>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> (deprecated in revision 1)	<code>array</code>	Deprecated in revision 1. 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
RENTDOWN_DUE	amount	Rent down payment due amount.
RENTDOWN_DUE_DATE	date or null	Rent down payment due date (date if specified, null otherwise)
RENTDOWN_PAID	amount	Rent down payment paid amount.
RENTDOWN_PAID_DATE	date or null	Rent down payment paid date (date if specified, null otherwise)
RENTDOWN_METHOD	string(0, 100)	Rent down payment method.
RENTREMAINING_DUE	amount	Rent remaining payment information.
RENTREMAINING_DUE_DATE	date or null	
RENTREMAINING_PAID	amount	
RENTREMAINING_PAID_DATE	date or null	
RENTREMAINING_METHOD	string(0, 100)	
DEPOSIT_DUE	amount	Security deposit information.
DEPOSIT_DUE_DATE	date or null	
DEPOSIT_PAID	amount	
DEPOSIT_PAID_DATE	date or null	
DEPOSIT_METHOD	string(0, 100)	
OTHER_DUE	amount	Other fees/deductions information.
OTHER_DUE_DATE	date or null	
OTHER_PAID	amount	
OTHER_PAID_DATE	date or null	
OTHER_METHOD	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).

10.4.5.4 Examples

Successful reservation read, with online booking information

HTTP request:

```
POST /channels/v1/readReservation?subscription-key=4nHm479ZZ267B99235GhN273c7mI4PHw
HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

{
  "RES_ID"      :      47882
}
```

HTTP reply:

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

{
  "API_VERSION"      :      3,
  "API_REVISION"     :      4,
  "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",
    "BOOKING_SOURCE_ID" :      null,
    "PMT_G2RÄ"        :
    {
      "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"
    }
}
}

```

10.4.6 createConfirmedReservation

10.4.6.1 Usage

This method creates a new, `PENDING CONFIRMATION` property reservation that will block the calendar.

Please note that the method does not:

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

10.4.6.2 Input

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

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>RES_CHECK_IN</code>	<code>date</code>	The reservation check-in date.
<code>RES_CHECK_OUT</code>	<code>date</code>	The reservation check-out date.
<code>RES_N_ADULTS</code>	<code>int</code>	The number of guests (adults), from 1 to 30.
<code>RES_N_CHILDREN</code>	<code>int</code>	The number of guests (children), from 0 to 29.
<code>RES_N_BABIES</code> (optional) (deprecated in revision 1)	<code>int</code>	The number of guests (babies), value is forced to 0.
<code>RES_GUEST</code> (mandatory since revision 3)	<code>object</code>	The reservation guest information.
<code>RES_COMMENT</code>	<code>string(0, 65535)</code>	The reservation comments.

<code>RES_COMMENT_GUEST</code> (optional)	<code>string(0, 65535)</code>	The guest reservation comment. This object member is optional (may be omitted).
<code>RES_UDRA</code> (optional) (deprecated in revision 1)	<code>array</code>	Deprecated in revision 1. It will be ignored.

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.

The `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(1, 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 string), it must be RFC 2822 compliant (addr-spec specification, § 3.4.1). However, the string length may not exceed 200 characters. If it's omitted, then the <code>RES_GUEST_PHONE</code> member must be specified.
<code>RES_GUEST_PHONE</code>	<code>string(0, 30)</code>	The guest phone number. If it's omitted, then the <code>RES_GUEST_EMAIL</code> member must be specified.
<code>RES_GUEST_COUNTRY</code>	<code>string(2)</code> or <code>null</code> if unspecified (deprecated, use <code>ZZ</code> value instead).	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 (deprecated, use <code>ZZ</code> value instead).

10.4.6.3 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> holds an object that contains a member called <code>RES_ID</code> , the id of the newly created reservation.
<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_CONFLICT</code>	Could not create reservation

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.

On `E_CONFLICT` result, `API_RESULT_TEXT` will contain all the error messages that warn of the reasons why the booking couldn't be created.

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).

10.4.6.4 Examples

Successful reservation create

HTTP request:

```
POST /channels/v1/createConfirmedReservation?subscription-
key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
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_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"         : []
}
```

HTTP reply:

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

{
  "API_VERSION"      : 3,
  "API_REVISION"     : 4,
  "API_METHOD"       : "createConfirmedReservation",
  "API_CALL_ID"      : "3bc3542f7130e6eca6b3b5dbd62e8030",
  "API_RESULT_CODE"  : "E_OK",
  "API_RESULT_TEXT"  : "The reservation was successfully created.",
  "API_REPLY"        :
  {
    "RES_ID"         : 24985
  }
}
```


Calendar conflict

HTTP request:

```
POST /channels/v1/createConfirmedReservation?subscription-
key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
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_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" : ""
}
```

HTTP reply:

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

{
  "API_VERSION" : 3,
  "API_REVISION" : 4,
  "API_METHOD" : "createConfirmedReservation",
  "API_CALL_ID" : "933486b2204dcd6c29640cee713ee3f3",
  "API_RESULT_CODE" : "E_CONFLICT",
  "API_RESULT_TEXT" : "This Property has a minimum stay of 4
nights.\nThis Property can only accommodate 2 people.\nUnfortunately, this unit is
not available for your selected period. Please try another unit.",
  "API_REPLY" : null
}
```

10.4.7 updateReservation

10.4.7.1 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.

10.4.7.2 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 (optional) (deprecated in revision 1)	int	The new number of baby guests. Value is forced to 0.
RES_GUEST	object or null	The reservation guest information. It's not mandatory for this method, but if it is included, it should have the same format as the RES_GUEST object explained in the createConfirmedReservation section.
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) (deprecated in revision 1)	array	Deprecated in revision 1, the array is ignored.

Same restrictions on RES_CHECK_IN and RES_CHECK_OUT as for createConfirmedReservation apply.

10.4.7.3 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.
E_NOSUCH	The reservation was not found, or does not belong to your Rental Agency Kigo account.
E_CONFLICT	Could not update reservation
E_DEACTIVATED	Could not update reservation due to the property no longer being available.

API_REPLY is always set to null.

On E_CONFLICT result, API_RESULT_TEXT will contain all the error messages that warn of the reasons why the booking couldn't be updated.

10.4.7.4 Examples

Successful reservation update

HTTP request:

```
POST /channels/v1/updateReservation?subscription-
key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
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_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"     : 3,
  "API_REVISION"    : 4,
  "API_METHOD"      : "updateReservation",
  "API_CALL_ID"     : "736adb6ff63c85a1e2d4313e61f3dc59",
  "API_RESULT_CODE" : "E_OK",
  "API_RESULT_TEXT" : "The reservation was successfully updated.",
  "API_REPLY"       : null
}
```

10.4.8 cancelReservation

10.4.8.1 Usage

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

10.4.8.2 Input

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

Member	Data type	Description
RES_ID	int	The Kigo reservation id.

10.4.8.3 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.

10.4.8.4 Examples

Attempt to cancel an already canceled reservation

HTTP request:

```
POST /channels/v1/cancelReservation?subscription-
key=4nHm479ZZ267B99235GhN273c7mI4PHw HTTP/1.0
Host: www.kigoapis.com
Content-Type: application/json

{
  "RES_ID"      :      31402,
}
```

HTTP reply:

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

{
  "API_VERSION"      :      3,
  "API_REVISION"     :      4,
  "API_METHOD"       :      "cancelReservation",
  "API_CALL_ID"      :      "9174ebc197b2cf9349554aa58269a076",
  "API_RESULT_CODE"  :      "E_ALREADY",
  "API_RESULT_TEXT"  :      "",
  "API_REPLY"        :      null
}
```

10.5 Real-time pricing computation

This API also offers to compute real-time prices on demand. In this scenario you will send a set of dates and Kigo will calculate the price for you.

10.5.1 computePricing

10.5.1.1 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.

It is mandatory be used to get the final price of every possible booking.

Please note that the method does NOT:

- enforce the property minimum and maximum stay times, but it will advise if the booking dates provided should break a min stay rule,
- enforce the property maximum number of guests but it will advise if the property could allow the number of guests provided.

10.5.1.2 Input

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

Member	Data type	Description
<code>PROP_ID</code>	<code>int</code>	The Kigo property id.
<code>RES_CREATE</code> (optional) (deprecated in revision 1)	<code>date</code>	This member was deprecated in this new API, and now the method only uses the current date as the creation date of the quote.
<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>	Number of guests (adults), from 1 to 30.
<code>RES_N_CHILDREN</code>	<code>int</code>	Number of guests (children), from 0 to 29.
<code>RES_N_BABIES</code> (deprecated in revision 1)	<code>int</code>	Number of guests (babies), value is forced to 0.

The following restrictions apply:

- Same restrictions on `RES_CHECK_IN` and `RES_CHECK_OUT` as in `createConfirmedReservation`

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

10.5.1.3 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.

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 previous API)	amount	This member was deprecated and renamed to <code>RENT_GUEST_CHARGES_AMOUNT</code> in previous API. Please refer to <code>RENT_GUEST_CHARGES_AMOUNT</code> instead.
<code>RENT_GUEST_CHARGES_AMOUNT</code> (deprecated in previous API)	amount	Extra guest charges amount. Due to a pricing algorithm change as the previous version 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</code> + <code>RENT_FEES_AMOUNT</code> + <code>RENT_DISCOUNTS_AMOUNT</code>
<code>RENT_DOWN_PMT_AMOUNT</code> (deprecated in previous API)	amount	Rent down payment amount. Deprecated in previous API. The value is now always equal to the value of <code>RENT_AMOUNT</code> .
<code>RENT_DOWN_PMT_DUE_DATE</code> (deprecated in previous API)	date or null	Rent down payment due date (null if there is no down payment). Deprecated in previous API.
<code>RENT_REM_PMT_AMOUNT</code> (deprecated in previous API)	amount	Rent remaining payment amount. Deprecated in previous API. The value is always "0.00".
<code>RENT_REM_PMT_DUE_DATE</code> (deprecated in previous API)	date or null	Rent remaining payment due date (null if there is no remaining payment). Deprecated in previous API. The value is always null.
<code>FEES_AMOUNT</code>	amount	Fees amount.
<code>OPTIONAL_FEES_AMOUNT</code>	amount	Optional fees amount.
<code>FEES_DUE_DATE</code> (deprecated in previous API)	date or null	Fees due date (null if there are no fees). Deprecated in previous API.
<code>TOTAL_AMOUNT</code>	amount	<code>RENT_AMOUNT</code> + <code>FEES_AMOUNT</code>
<code>DEPOSIT_AMOUNT</code>	amount	Security deposit amount.
<code>DEPOSIT_DUE_DATE</code> (deprecated in previous API)	date or null	Security deposit due date (null if there is no security deposit). Deprecated in previous 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).

10.5.2 computePricingBulk

10.5.2.1 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.

10.5.2.2 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> (optional) (deprecated in revision 1)	date	This member was deprecated in this new API, and now the method only uses the current date as the creation date of the quote.
<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> (deprecated in revision 1)	int	Number of guests (babies), value is forced to 0.

Same restrictions as for computePricing apply.

10.5.2.3 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.

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:

		<p>E_OK - The computation was successful, REPLY holds computation result details as described in API_REPLY on E_OK result.</p> <p>E_NOSUCH - The property was not found, or is unavailable (i.e. was deactivated). REPLY is null.</p> <p>E_EMPTY - The property pricing information is unavailable or the property pricing calculation is disabled. REPLY is null.</p> <p>E_LIMIT - The property pricing information is unavailable for the specified check-in/check-out dates. REPLY is null.</p> <p>E_CONFLICT - The property pricing information is unavailable due to a constraint. .</p>
REPLY	null or object	The value of this member depends on RESULT_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).

10.6 Account and system inventory

The methods covered in this section allow API users to access Kigo's data dictionary. This will be particularly useful to synchronize property information. All the following methods are read-only.

10.6.1 listKigoCountries

10.6.1.1 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.

10.6.1.2 Input

This method requires input data to be **null**.

10.6.1.3 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. API_REPLY is null .

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).

10.6.2 listKigoPropertyTypes

10.6.2.1 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.

10.6.2.2 Input

This method required input data to be `null`.

10.6.2.3 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).

10.6.3 listKigoPropertyBedTypes

10.6.3.1 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.

10.6.3.2 Input

This method requires input data to be `null`.

10.6.3.3 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 null.

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).

10.6.4 listKigoPropertyAmenities

10.6.4.1 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.

10.6.4.2 Input

This method requires input data to be `null`.

10.6.4.3 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 null.

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.

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

10.6.5 listKigoPropertyActivities

10.6.5.1 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.

10.6.5.2 Input

This method requires input data to be `null`.

10.6.5.3 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. <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
ACTIVITY_CATEGORY	array	List of activity categories, in undefined order.
ACTIVITY	array	List of activities, in undefined order.

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

Member	Data type	Description
ACTIVITY_CATEGORY_ID	int	The unique activity category identifier.
ACTIVITY_CATEGORY_LABEL	string(1, 50)	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
--------	-----------	-------------

ACTIVITY_ID	int	The unique activity identifier.
ACTIVITY_CATEGORY_ID	int	Identifier of the ACTIVITY_CATEGORY the activity belongs to.
ACTIVITY_LABEL	string(1, 50)	The activity label, such as used in Kigo application.

10.6.6 listKigoFeeTypes

10.6.6.1 Usage

This method returns the list of fee types (see § 11.10 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.

10.6.6.2 Input

This method requires input data to be `null`.

10.6.6.3 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. API_REPLY is null.

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
FEE_TYPE_ID	int	The type identifier of the fee.
FEE_TYPE_LABEL	string(1, 50)	The fee type label, such as used in Kigo application.
FEE_TYPE_INCLUDE_IN_RENT	bool	Specified whether the fees of this type may be included in rent (see § 11.10.2 FEES).
FEE_TYPE_UNITS	array of string	List of supported units for this type (see § 11.10.2 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).

10.7 Deprecated API methods

10.7.1 diffPropertyCalendarReservations

This method was deprecated due to its incompatibility with the new multi-unit capabilities of the system. Please use the methods diffPropertyAvailability, listPropertyAvailability, listPropertyAvailabilityBulk and listPropertyCalendarReservations instead.

10.7.2 listCountries

This method was renamed to listKigoCountries. Please use listKigoCountries instead.

10.7.3 listProperties

This method was obsoleted by listProperties2. Implementations that use this method should be upgraded to use the listProperties2 method instead.

10.7.4 readProperty

This method was obsoleted by readProperty2. Implementations that use this method should be upgraded to use the readProperty2 method 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_CONFIRMED</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 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.

Properties not being actively provided by a provider (inactive properties) may not have new reservations created, nor existing reservations modified through this API.

11.4.1 PROP_PROVIDER

The `PROP_PROVIDER` object is being used in `listProperties2` and `readProperty2` methods to represent the current property provider. It holds the following members:

Member	Data type	Description
<code>PROVIDER_TYPE</code>	string	"RA", as the property is being provided by a rental agency account.
<code>OWNER_ID</code> (deprecated in revision 1)	null	Deprecated in revision 1.
<code>RA_ID</code> (if <code>PROVIDER_TYPE</code> is "RA")	int	The provider Kigo rental agency identifier.

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) when reading the property provider information.

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 information

The Kigo application and the Kigo API (starting from revision 13) allow rental agencies to manage the property information. The Kigo API provides the following methods for managing the property information: `listProperties2`, `readProperty2`, `readPropertyPhotoFile`.

The property information is split into the below objects. Not all of them are always available, please refer to the individual API methods documentation.

11.9.1 PROP_INFO

The `PROP_INFO` object holds the following members:

Member	Data type	Description
<code>PROP_NAME</code>	<code>string(1, 100)</code>	The property name.
<code>PROP_STREETNO</code>	<code>string(0, 8)</code>	The property address street number.
<code>PROP_ADDR1</code>	<code>string(0, 60)</code>	The property address line 1 (street).
<code>PROP_ADDR2</code>	<code>string(0, 35)</code>	The property address line 2.
<code>PROP_ADDR3</code>	<code>string(0, 35)</code>	The property address line 3.
<code>PROP_APTNO</code>	<code>string(0, 10)</code>	Apartment number.
<code>PROP_POSTCODE</code>	<code>string(0, 8)</code>	Postal code.
<code>PROP_CITY</code>	<code>string(0, 26)</code>	City.
<code>PROP_REGION</code>	<code>string(0, 35)</code>	Region.
<code>PROP_COUNTRY</code>	<code>string(2)</code>	The property country ISO 3166-1-alpha-2 code (see § 11.3 Country ISO 3166-1 alpha-2 codes for more information).
<code>PROP_LATLNG</code>	<code>latlng</code> or <code>null</code>	Longitude and latitude geographic coordinates in WGS 84 coordinates system, or <code>null</code> if unspecified.
<code>PROP_PHONE</code>	<code>string(0, 30)</code>	Property phone number.
<code>PROP_AXSCODE</code>	<code>string(0, 65535)</code>	Access codes.
<code>PROP_BEDROOMS</code>	<code>int</code> or <code>null</code>	Number of bedrooms (<code>int</code> , range 0 - 20 inclusive), or <code>null</code> if unspecified.
<code>PROP_BEDS</code>	<code>int</code> or <code>null</code>	Number of beds (<code>int</code> , range 1 - 20 inclusive), or <code>null</code> 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 is empty if PROP_BEDS is null (unspecified number of beds) or equal to zero. See § 11.6 Property bed type for information on bed types, and § 10.6.3 listKigoPropertyBedTypes for retrieving the list of bed types defined in Kigo application.
PROP_BATHROOMS	int or null	Number of full bathrooms (int, range 0 - 30 inclusive), or null if unspecified.
PROP_TOILETS	int or null deprecated, it will be null	Number of toilets (int, range 0 - 30 inclusive) 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 § 10.6.2 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 have int (range 1 - 9999 inclusive) and string values respectively (the property size is specified), or both are null (property size is unspecified).
PROP_SIZE_UNIT	"SQMETER" (string), "SQFEET" (string) or null	
PROP_MAXGUESTS	int	Maximum allowed number of guests (total). Range: 1 - 30 inclusive.
PROP_MAXGUESTS_ADULTS	int	Maximum allowed number of guests (adults). Range: 1 - 30 inclusive, but may not exceed PROP_MAXGUESTS.
PROP_MAXGUESTS_CHILDREN	int	Maximum allowed number of guests (children). Range: 0 - 29 inclusive, but must be less than PROP_MAXGUESTS.
PROP_MAXGUESTS_BABIES	int	Maximum allowed number of guests (babies). Range: 0 - 29 inclusive, but must be less than PROP_MAXGUESTS.
PROP_FLOOR	string(0, 4)	Floor (European).
PROP_ELEVATOR (deprecated in version 4)	bool or null	Property has/doesn't have elevator (bool), or null if unspecified. Value is forced to null.
PROP_CIN_TIME	time_hh	Standard check-in and check-out

PROP_COUT_TIME	time_hh	times. Range: "00:00" - "23:30" inclusive. The check-in time may not be earlier than the check-out time.
PROP_CLEAN_TIME	time_hh	Average cleaning time. Range: "00:00" - "12:00" inclusive.
PROP_STAYTIME_MIN	time_interval	Minimum allowed stay time.
PROP_STAYTIME_MAX	time_interval	Maximum allowed stay time. Must be greater than PROP_STAYTIME_MIN.
PROP_SHORTDESCRIPTION	string(0, 65535)	Property short description.
PROP_SHORTDESCRIPTION_HTML	string(0, 65535)	Property short description in HTML format.
PROP_DESCRIPTION	string(0, 65535)	Property full description.
PROP_DESCRIPTION_HTML	string(0, 65535)	Property full description in HTML format.
PROP_AREADESCRIPTION	string(0, 65535)	Area description.
PROP_RENTAL_DETAILS	string(0, 65535)	Rental details.
PROP_INVENTORY	string(0, 65535)	Property inventory.
PROP_ARRIVAL_SHEET	string(0, 65535)	Property arrival sheet.
PROP_ARRIVAL_SHEET_HTML	string(0, 65535)	Property arrival sheet in HTML format.
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 § 10.6.4 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 § 10.6.5 listKigoPropertyActivities for retrieving the list of property activities defined in Kigo application.

PROP_NAME_ALL	object	The property name n all languages provided for that property, keyed by locale.
PROP_DESCRIPTION_ALL	object	Property's full description in all languages provided for that property, keyed by locale.
PROP_SHORTDESCRIPTION_ALL	object	Property's short description in all languages provided, for that property keyed by locale.
PROP_ARRIVAL_SHEET_ALL	object	Property's arrival sheet in all languages provided for that property, keyed by locale.
PROP_NAME_ALL_HTML	object	The property name n all languages provided for that property, keyed by locale, in HTML format.
PROP_DESCRIPTION_ALL_HTML	object	Property's full description in all languages provided for that property, keyed by locale, in HTML format.
PROP_SHORTDESCRIPTION_ALL_HTML	object	Property's short description in all languages provided, for that property keyed by locale, in HTML format.
PROP_ARRIVAL_SHEET_ALL_HTML	object	Property's arrival sheet in all languages provided for that property, keyed by locale, in HTML format.

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) when reading the property information.

11.9.2 PROP_UNITS

The PROP_UNITS member is an array that holds the ordered list of property units. Each property unit is represented by an object with following members:

Member	Data type	Description
UNIT_ID	int	The Kigo unit Id.
UNIT_NAME	string(1, 100)	The unit name.
UNIT_SUNRISE_DATE	date or null	The date the Unit is available to the market.
UNIT_SUNSET_DATE	date or null	The date the Unit was taken off the market.
UNIT_PERMIT_OR_TAXID	string(0, 255)	The unit Permit or Tax Id.

11.9.3 PROP_RATE

The PROP_RATE object holds the following members:

Member	Data type	Description
PROP_RATE_CURRENCY	string(3)	Currency ISO 4217 code.
PROP_RATE_NIGHTLY_FROM	amount or null	Property nightly, weekly and monthly display rates (amount, range "0.01" - "9999999999.99" inclusive), or null if unspecified. When a "to" rate is specified, the "from" rate must also be specified, and must be less then or equal to the "to" rate.
PROP_RATE_NIGHTLY_TO		
PROP_RATE_WEEKLY_FROM		
PROP_RATE_WEEKLY_TO		
PROP_RATE_MONTHLY_FROM		
PROP_RATE_MONTHLY_TO		

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) when reading the property rate information.

Please note these rates are not used in rent calculation. They are useful for display on Kigo websites when rent calculation is not available.

11.9.4 PROP_PHOTOS

The PROP_PHOTOS member 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, 1024)	Unique immutable property photo identifier.
PHOTO_PANORAMIC	bool	Photo information.
PHOTO_NAME	string(0, 50)	
PHOTO_COMMENTS	string(0, 255)	

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) when reading the property photos information.

11.10 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 CHANNELS API - V1 (as of revision 12).

The property pricing information is split into the below objects. Not all of them are always available, please refer to the individual API methods documentation.

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
CURRENCY	string(3)	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.
RENT	object	See § 11.10.1 RENT. Provided by readPropertyPricingSetup and diffPropertyPricingSetup methods.

FEES	object	See § 11.10.2 FEES. Provided by readPropertyPricingSetup.
TAXES	object	See § 11.10.2 FEES. Provided by readPropertyPricingSetup2.
DISCOUNTS (deprecated in revision 4)	object	See § 11.10.4 DISCOUNTS. Provided by readPropertyPricingSetup. It will contain empty values.
DEPOSIT	object or null	See § 11.10.5 DEPOSIT. Provided by readPropertyPricingSetup.
MIN_STAY	object	See § 11.10.7 MIN_STAY Provided by readPropertyPricingSetup.

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.

11.10.1 RENT

11.10.1.1 Definition

The `RENT` object holds the following members:

Member	Data type	Description
PERGUEST_CHARGE	object or null	Specified whether a per-guest charges apply (object) or not (null) to the amounts defined in the PERIODS.
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"> <code>"ADULTS"</code> - the number of guests will match only the number of adults <code>"ADULTS_CHILDREN"</code> - the number of guests will match the number of adults plus the number of children <code>"ADULTS_CHILDREN_BABIES"</code> - the number of guests will match the number of adults plus the number of children plus the number of babies
STANDARD	int	The standard occupancy. A single <code>WEEKLY_AMOUNTS</code> or

		NIGHTLY_AMOUNTS amount will apply to any number of guests between 1 and STANDARD. Allowed range: from 1 to 29.
MAX	int	The maximum occupancy. A single WEEKLY_AMOUNTS or NIGHTLY_AMOUNTS amount will apply to any number of guests that is equal to, or that exceeds MAX. Allowed range: from STANDARD+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) CHECK_OUT date must be equal to this period CHECK_IN date.
CHECK_OUT	date	The period check-out date. Must be greater than CHECK_IN date. The following period (if any) CHECK_IN date must be equal to this period CHECK_OUT date.
NAME	string(0, 50)	The period name.
STAY_MIN	time_interval	The minimum stay defined for this period. For a weekly-only period (WEEKLY value is true), the minimum stay may not be under 7 nights.
WEEKLY	bool	Specified whether the period is a weekly-only rental period (true) or not (false). A period may be weekly-only only if the period CHECK_IN and CHECK_OUT dates are on the same day of week.
WEEKLY_AMOUNTS	array of object	WEEKLY_AMOUNTS is available only if WEEKLY is true, while NIGHTLY_AMOUNTS is available only if WEEKLY is false.
NIGHTLY_AMOUNTS	array of object	The number of items in these arrays is determined by the product of items required by PERGUEST_CHARGE (see GUESTS_FROM below), of the stay ranges (only for WEEKLY = false, see STAY_FROM below) and of the night-of-week groups (only for WEEKLY = false, see WEEK_NIGHTS below).

Each WEEKLY_AMOUNTS item is an object holding the following members:

Member	Data type	Description
GUESTS_FROM	int	Represents the number of guests the AMOUNT applies to. If PERGUEST_CHARGE is null, then there is only one value, 1. If PERGUEST_CHARGE is not null, then each of the following values need to be provided in the

		<p>WEEKLY_AMOUNTS array:</p> <ul style="list-style-type: none"> • 1 - matches any number of guests from 1 to STANDARD included. • From STANDARD+1 to MAX-1 - matches that number of guests • MAX - matches any number of guests that is equal or greater than MAX.
AMOUNT	amount	The weekly amount, in range "0.01" - "99999999.99".

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

Member	Data type	Description
GUESTS_FROM	int	Same definition as for the WEEKLY_AMOUNTS items.
WEEK_NIGHTS	array of int	array holding the nights of week in ISO 8601 format (1 for Monday, 7 for Sunday) the below AMOUNT applies to. There must be as many NIGHTLY_AMOUNTS 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" - "99999999.99".

11.10.1.2 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"
    }
  ]
}

```

11.10.2 FEES

11.10.2.1 Definition

The `FEES` object holds the following members:

Member	Data type	Description
FEES	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
FEE_TYPE_ID	int	The fee type id. The <code>FEES</code> list may contain multiple items of a same fee type. See § 10.6.6 <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		
LABEL	string	The name of the fee as it has been set in Kigo application.
MIN_CONDITION_VALUE	int	This is the minimum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests.
MAX_CONDITION_VALUE	int	This is the maximum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests. If it's number of nights and the guest stay more nights than the value entered here, then the fee will not be applicable for the additional nights.
CONDITION_VALUES_UNIT	string	This is the unit considered by the <code>MIN_CONDITION_VALUE</code> or <code>MAX_CONDITION_VALUE</code> , it could be "NIGHT" or "GUEST".
EFFECTIVE_DATE	date or null	This is the first date that the fee becomes available to be charged.

START_DATE	date or null	The first applicable check-in date for the fee to be charged.
END_DATE	date or null	The last check-in date that the fee can be charged.
TAXABLE	bool or null	Indicates if the taxes should be applied to this fee. Method readPropertyPricingSetup will return null if the fee type belongs to a Tax.
IS_VAT_TAX	bool or null	Indicates if the tax should be applied to the fees marked as "TAXABLE". Method readPropertyPricingSetup will return null if the fee type doesn't belong to a Tax.

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" - "99999999.99" range								
"AMOUNT_PER_NIGHT"	amount in the "0.01" - "99999999.99" 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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" 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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" range								
"PERCENT_RENT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_PER_NIGHT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_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>PERCENT_ADULT</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> <tr> <td>PERCENT_CHILD</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.		
Member	Data type								
PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.								
PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.								

"PERCENT_RENT_PER_GUEST"	object holding the following members:	
	Member	Data type
	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.
	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.

11.10.2.2 Examples

Example of the FEES object holding two fees.

```
{
  "FEES": [
    {
      "FEE_TYPE_ID" : 16,
      "INCLUDE_IN_RENT" : false,
      "UNIT" : "PERCENT_RENT_PER_NIGHT",
      "VALUE" : "0.1",
      "LABEL" : "Extra Person fee",
      "MIN_CONDITION_VALUE" : 5,
      "MAX_CONDITION_VALUE" : 11,
      "CONDITION_VALUES_UNIT" : "GUEST",
      "EFFECTIVE_DATE" : null,
      "START_DATE" : null,
      "END_DATE" : null,
      "TAXABLE" : null,
      "IS_VAT_TAX" : true
    },
    {
      "FEE_TYPE_ID" : 10,
      "INCLUDE_IN_RENT" : false,
      "UNIT" : "AMOUNT_PER_NIGHT_PER_GUEST",
      "VALUE" :
        {
          "AMOUNT_ADULT" : "1.45",
          "AMOUNT_CHILD" : "1.45",
          "AMOUNT_BABY" : "0.00"
        },
      "LABEL" : "Linen Service",
      "EFFECTIVE_DATE" : "2016-10-01",
      "START_DATE" : "2016-10-01",
      "END_DATE" : "2017-11-26",
      "TAXABLE" : true,
      "IS_VAT_TAX" : null
    }
  ]
}
```

11.10.3 TAXES

11.10.3.1 Definition

The `TAXES` object holds the following members:

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

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

Member	Data type	Description
<code>FEE_TYPE_ID</code>	int	The tax type id. The <code>TAXES</code> list may contain multiple items of a same tax type. See § 10.6.6 <code>listKigoFeeTypes</code> for retrieving the list of tax types defined in Kigo application.
<code>UNIT</code>	string	The unit and value (parameters, dependent on unit) used for computing the tax amount.
<code>VALUE</code>		
<code>LABEL</code>	string	The name of the tax as it has been set in Kigo application.
<code>MIN_CONDITION_VALUE</code>	int	This is the minimum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests.
<code>MAX_CONDITION_VALUE</code>	int	This is the maximum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests. If it's number of nights and the guest stay more nights than the value entered here, then the fee will not be applicable for the additional nights.
<code>CONDITION_VALUES_UNIT</code>	string	This is the unit considered by the <code>MIN_CONDITION_VALUE</code> or <code>MAX_CONDITION_VALUE</code> , it could be "NIGHT" or "GUEST".
<code>IS_VAT_TAX</code>	bool or null	Indicates if the tax should be applied to the fees marked as "TAXABLE".

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" - "99999999.99" range
"AMOUNT_PER_NIGHT"	amount in the "0.01" - "99999999.99" 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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" 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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" range								
"PERCENT_RENT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_PER_NIGHT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_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>PERCENT_ADULT</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> <tr> <td>PERCENT_CHILD</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.		
Member	Data type								
PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.								
PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_PER_GUEST"	object holding the following members: <table border="1"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>PERCENT_ADULT</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> <tr> <td>PERCENT_CHILD</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.		
Member	Data type								
PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.								
PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.								

11.10.3.2 Examples

Example of the TAXES object holding a tax.

```
{
  "TAXES": [
    {
      "FEE_TYPE_ID"      : 16,
      "INCLUDE_IN_RENT" : false,
      "UNIT"             : "PERCENT_RENT_PER_NIGHT",
      "VALUE"            : "0.1",
      "LABEL"            : "Lodging Tax",
    }
  ]
}
```

```

        "MIN_CONDITION_VALUE" : 7,
        "MAX_CONDITION_VALUE" : 14,
        "CONDITION_VALUES_UNIT": "NIGHT",
        "IS_VAT_TAX" : true
    }
]
}

```

11.10.4 DISCOUNTS

11.10.4.1 Definition

The `DISCOUNTS` object holds the following members:

Member	Data type	Description
<code>EARLY_BIRD</code> (deprecated in revision 4)	object or null	The early bird discount setup (object) or null if no early discount is set up. Value is forced to null.
<code>LAST_MINUTE</code> (deprecated in revision 4)	array of object	Last minute discounts setup (0 to 10 array items). Value is forced to null.
<code>SPECIAL</code> (deprecated in revision 4)	array of object	The special (custom) discounts setup (0 to 20 array items). Value is forced to null.

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.
<code>PERCENT</code>	amount	Percentage calculated to be applied directly, not divided by 100.

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

Member	Data type	Description
<code>DAYS</code>	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 DAYS must be unique within the LAST_MINUTE array.
<code>PERCENT</code>	amount	Percentage calculated to be applied directly, not divided by 100.

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

Member	Data type	Description
<code>NAME</code>	<code>string(1, 50)</code>	The discount name.
<code>PERCENT</code>	<code>amount</code>	Percentage calculated to be applied directly, not divided by 100.
<code>VALID_FROM</code>	<code>date or null</code>	The discount validity, based on reservation create date. Either both <code>VALID_FROM</code> and <code>VALID_TO</code> are date or both are null. They may be equal (one day discount validity).
<code>VALID_TO</code>		
<code>CHECK_IN</code>	<code>date</code>	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> .
<code>CHECK_OUT</code>		

11.10.4.2 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"
    }
  ]
}
```


11.10.5 DEPOSIT

11.10.5.1 Definition

The `DEPOSIT` object holds the following members:

Member	Data type	Description
<code>UNIT</code>	<code>string</code>	The unit and value (parameters, dependent on unit) used for computing the deposit amount.
<code>VALUE</code>		
<code>LABEL</code>	<code>string</code>	The name of the deposit fee as it has been set in Kigo application.

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																	
<code>"AMOUNT"</code>	amount in the <code>"0.01"</code> - <code>"99999999.99"</code> range																	
<code>"PERCENT_RENT"</code>	Percentage calculated to be applied directly, not divided by 100.																	
<code>"STAYLENGTH"</code>	array (minimum 2 and maximum 40 items) of object, where each item is holding the following members: <table border="1" data-bbox="443 992 1436 1249"> <thead> <tr> <th>Member</th> <th>Data type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>STAY_FROM</code></td> <td><code>time_interval</code></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><code>UNIT</code></td> <td><code>string</code></td> <td rowspan="2">The unit and value (parameters, dependent on unit) used for computing the fee amount.</td> </tr> <tr> <td><code>VALUE</code></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 1339 1436 1507"> <thead> <tr> <th><code>UNIT</code></th> <th><code>VALUE</code> data type</th> </tr> </thead> <tbody> <tr> <td><code>"AMOUNT"</code></td> <td>amount in the <code>"0.00"</code> - <code>"99999999.99"</code> range</td> </tr> <tr> <td><code>"PERCENT_RENT"</code></td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	Description	<code>STAY_FROM</code>	<code>time_interval</code>	The minimum stay duration the below <code>UNIT</code> and <code>VALUE</code> apply to. The "1 night" range is mandatory.	<code>UNIT</code>	<code>string</code>	The unit and value (parameters, dependent on unit) used for computing the fee amount.	<code>VALUE</code>		<code>UNIT</code>	<code>VALUE</code> data type	<code>"AMOUNT"</code>	amount in the <code>"0.00"</code> - <code>"99999999.99"</code> range	<code>"PERCENT_RENT"</code>	Percentage calculated to be applied directly, not divided by 100.
Member	Data type	Description																
<code>STAY_FROM</code>	<code>time_interval</code>	The minimum stay duration the below <code>UNIT</code> and <code>VALUE</code> apply to. The "1 night" range is mandatory.																
<code>UNIT</code>	<code>string</code>	The unit and value (parameters, dependent on unit) used for computing the fee amount.																
<code>VALUE</code>																		
<code>UNIT</code>	<code>VALUE</code> data type																	
<code>"AMOUNT"</code>	amount in the <code>"0.00"</code> - <code>"99999999.99"</code> range																	
<code>"PERCENT_RENT"</code>	Percentage calculated to be applied directly, not divided by 100.																	

11.10.5.2 Examples

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

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

11.10.6 DEPOSITS

11.10.6.1 Definition

Each item of the `DEPOSITS` array is an object holding the following members:

Member	Data type	Description
<code>UNIT</code>	string	The unit and value (parameters, dependent on unit) used for computing the deposit amount.
<code>VALUE</code>		
<code>LABEL</code>	string	The name of the deposit fee as it has been set in Kigo application.
<code>MIN_CONDITION_VALUE</code>	int	This is the minimum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests.
<code>MAX_CONDITION_VALUE</code>	int	This is the maximum value of the condition that it's required to apply the fee. The value could be number of nights or number of guests. If it's number of nights and the guest stay more nights than the value entered here, then the fee will not be applicable for the additional nights.
<code>CONDITION_VALUES_UNIT</code>	string	This is the unit considered by the <code>MIN_CONDITION_VALUE</code> or <code>MAX_CONDITION_VALUE</code> , it could be "NIGHT" or "GUEST".
<code>EFFECTIVE_DATE</code>	date or null	This is the first date that the deposit becomes available to be charged.
<code>START_DATE</code>	date or null	The first applicable check-in date for the deposit to be charged.
<code>END_DATE</code>	date or null	The last check-in date that the deposit can be charged.
<code>TAXABLE</code>	bool or null	Indicates if the taxes should be applied to this deposit.
<code>IS_VAT_TAX</code>	bool or null	Deprecated. Method <code>readPropertyPricingSetup</code> will return <code>null</code> . Method <code>readPropertyPricingSetup2</code> won't included it in the response.

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
<code>"AMOUNT"</code>	amount in the <code>"0.01"</code> - <code>"99999999.99"</code> range
<code>"AMOUNT_PER_NIGHT"</code>	amount in the <code>"0.01"</code> - <code>"99999999.99"</code> range
<code>"AMOUNT_PER_NIGHT_PER_GUEST"</code>	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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" 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" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_CHILD</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> <tr> <td>AMOUNT_BABY</td> <td>amount in the "0.00" - "99999.99" range</td> </tr> </tbody> </table>	Member	Data type	AMOUNT_ADULT	amount in the "0.01" - "99999.99" range	AMOUNT_CHILD	amount in the "0.00" - "99999.99" range	AMOUNT_BABY	amount in the "0.00" - "99999.99" range
Member	Data type								
AMOUNT_ADULT	amount in the "0.01" - "99999.99" range								
AMOUNT_CHILD	amount in the "0.00" - "99999.99" range								
AMOUNT_BABY	amount in the "0.00" - "99999.99" range								
"PERCENT_RENT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_PER_NIGHT"	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_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>PERCENT_ADULT</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> <tr> <td>PERCENT_CHILD</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.		
Member	Data type								
PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.								
PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.								
"PERCENT_RENT_PER_GUEST"	object holding the following members: <table border="1"> <thead> <tr> <th>Member</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td>PERCENT_ADULT</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> <tr> <td>PERCENT_CHILD</td> <td>Percentage calculated to be applied directly, not divided by 100.</td> </tr> </tbody> </table>	Member	Data type	PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.	PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.		
Member	Data type								
PERCENT_ADULT	Percentage calculated to be applied directly, not divided by 100.								
PERCENT_CHILD	Percentage calculated to be applied directly, not divided by 100.								

11.10.6.2 Examples

Example of the `DEPOSITS` object with a percent amount deposit.

```
"DEPOSITS": [
  {
    "UNIT": "PERCENT_RENT_PER_NIGHT_PER_GUEST",
    "VALUE": {
      "PERCENT_ADULT": 0.4,
      "PERCENT_CHILD": 0.1
    },
    "LABEL": "Security Deposit",
    "MIN_CONDITION_VALUE": 2,
  }
]
```

```

        "MAX_CONDITION_VALUE": 5,
        "CONDITION_VALUES_UNIT": "NIGHT",
        "EFFECTIVE_DATE": "2017-05-19",
        "START_DATE": "2017-05-19",
        "END_DATE": "2018-03-31",
        "TAXABLE": false
    }
]

```

11.10.7 MIN_STAY

11.10.7.1 Definition

The `MIN_STAY` object holds the following members:

Member	Data type	Description
<code>DEFAULT_VALUE</code>	int	The default minimum number of nights this property can be rented for.
<code>MIN_STAY_RULES</code>	array of object	List of the extra rules concerning the minimum stay of the property.

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

Member	Data type	Description
<code>LABEL</code>	string	The name of the rule as it has been set in Kigo application.
<code>MIN_STAY_VALUE</code>	int	The minimum number of nights this property can be rented for, if the rule is applied.
<code>DATE_FROM</code>	date or null	The period in which the rule should apply.
<code>DATE_TO</code>		

11.10.7.2 Examples

Example of the `MIN_STAY` object holding four minimum stay rules.

```

{
  "MIN_STAY": {
    "DEFAULT_VALUE": 5,
    "MIN_STAY_RULES": [
      {
        "LABEL": "to June 2016",
        "MIN_STAY_VALUE": 5,
        "DATE_FROM": null,
        "DATE_TO": "2016-06-29"
      },
      {
        "LABEL": "General Min Stay November 15 - March 16",
        "MIN_STAY_VALUE": 4,

```

```

        "DATE_FROM": "2015-10-31",
        "DATE_TO": "2016-03-30"
    },
    {
        "LABEL": "July - August 2016",
        "MIN_STAY_VALUE": 7,
        "DATE_FROM": "2016-06-30",
        "DATE_TO": "2016-08-30"
    },
    {
        "LABEL": "September 2016 on",
        "MIN_STAY_VALUE": 5,
        "DATE_FROM": "2016-08-31",
        "DATE_TO": null
    }
]
}

```

11.10.8 TURN_DAY_RULES

11.10.8.1 Definition

Each item of the `TURN_DAY_RULES` array is an object holding the following members:

Member	Data type	Description
<code>LABEL</code>	string	The name of the Turn day rule as it has been set in Kigo application.
<code>DATE_FROM</code>	date or null	The date from wich the rule should apply.
<code>DATE_TO</code>	date or null	The date until wich the rule should apply.
<code>CHECK_IN_SUNDAY</code>	bool	True if the check-in can be done in Sundays, false if not.
<code>CHECK_IN_MONDAY</code>	bool	True if the check-in can be done in Mondays, false if not.
<code>CHECK_IN_TUESDAY</code>	bool	True if the check-in can be done in Tuesdays, false if not.
<code>CHECK_IN_WEDNESDAY</code>	bool	True if the check-in can be done in Wednesdays, false if not.
<code>CHECK_IN_THURSDAY</code>	bool	True if the check-in can be done in Thursdays, false if not.

CHECK_IN_FRIDAY	bool	True if the check-in can be done in Fridays, false if not.
CHECK_IN_SATURDAY	bool	True if the check-in can be done in Saturdays, false if not.
CHECK_OUT_SUNDAY	bool	True if the check-out can be done in Sundays, false if not.
CHECK_OUT_MONDAY	bool	True if the check-out can be done in Mondays, false if not.
CHECK_OUT_TUESDAY	bool	True if the check-out can be done in Tuesdays, false if not.
CHECK_OUT_WEDNESDAY	bool	True if the check-out can be done in Wednesdays, false if not.
CHECK_OUT_THURSDAY	bool	True if the check-out can be done in Thursdays, false if not.
CHECK_OUT_FRIDAY	bool	True if the check-out can be done in Fridays, false if not.
CHECK_OUT_SATURDAY	bool	True if the check-out can be done in Saturdays, false if not.

11.10.9 AVAILABILITY_RULES

11.10.9.1 Definition

The `AVAILABILITY_RULES` object holds the following members:

Member	Data type	Description
ALLOW_BOOKINGS_X_DAYS_FROM_TODAY	int	The minimum number of days from which the property can be booked.
ALLOW_BOOKINGS_X_DAYS_IN_ADVANCE	int	The maximum number of days in advance that a guest will be able to make a booking.

12 References

12.1 ISO 3166

Official ISO website

http://www.iso.org/iso/country_codes.htm

Wikipedia ISO 3166-1 article

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

12.2 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>

12.3 base64

Base64 data encoding RFC

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

12.4 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

Wikipedia Decimal degrees notation article

http://en.wikipedia.org/wiki/Decimal_degrees