Kigo Channels

API DEVELOPMENT INTRODUCTION

API revision
4
Document revision
3
Contact
api@kigo.net

Overview of the API

The Kigo Channels API is a REST-like implementation based on the HTTP protocol, JSON data encoding and Unicode character encoding, over the secure HTTPS transport. Be aware: **Kigo Channels API doesn't work with XML feeds**.

The API consists on a series of methods invoked by making HTTPS requests on the Kigo Rental Agency REST API servers. These methods are translated "under-the-hood" to the native system API (previously InstaManager API) and can pull information from the Kigo system (availability calendars and information about Kigo properties and reservations) and can also create, modify and cancel reservations.

In Kigo we use the word "properties" when we talk about apartments, hotel rooms, bungalows, villas, beach houses, etc... Every property can have one or more "units" or indistinguishable instances of that same property.

What you need to integrate your Kigo Solution with an external system using the Kigo API

If you request API access, you will get an API key. You will need to use this API key in all your API method calls.

Even if your portal intends to use the API for most of its day to day operations, the use of the Kigo back office could be necessary at some point. We advise that you familiarize yourself with the application.

Your company will need a competent IT department to integrate Kigo with your system using the API. **Be aware that Kigo IT and support departments will neither check, revise, debug nor correct your code**. They will only solve high level doubts and questions about what is possible and what isn't using Kigo API and generic best approaches.

If you have previously developed an API integration with the Kigo Legacy system, the integration with the new Kigo using the Channels API should present few challenges and be able to re-use most of your previous code.

This document is an introduction.
*It should be used together with the **Kigo Channels API specification.***

In case the information in this document (introduction) contradicts the information in the specification, consider the information in the specification to be correct.

*We suggest that your IT department programs the **ping** method first. It is a very basic, simple method that Kigo provides just to check if you are able to establish the connection and authenticate with your API account credentials.*

The next steps will assume you have properties in your Kigo Solution:

- When Kigo grants API access to your Solution, it shares several properties with it from the Kigo Data Sharing Solution (RA_ID #11375), so your developers can do tests on those properties.

- During the certification process, a couple of properties will be temporarily shared with your Solution. They will be unshared once the certification process is completed.

- You will receive properties from other Kigo agencies through the Kigo Channel Manager once your integration has been certified by Kigo team.

From now on, when this document refers to “your properties”, it means the properties in your Kigo Portal Solution shared with your company from another Kigo Agency Solution through the Kigo Channel Manager.

PULL PROPERTY INFORMATION, PHOTOS AND PRICING

You will need to pull all the information from properties in your Kigo Solution to your own system, and store it there in your own database especially for properties that other agencies have shared with you.

The API method **listProperties2** returns a list of all properties in your Kigo Solution.

The method **listProperties2** returns a list where you have the Kigo property ID and also the Kigo Solution ID of the agency sharing that property with your Solution.

Your Agency should use **listProperties2** regularly (at least once a day) saving the updated list to check if new properties have been shared (or unshared) with it from your partner agencies.

The method **diffPropertyContent** should be used to retrieve the list of properties in which "static" information (the "less dynamic", to be exact) has been changed.

Once you have the list of property Ids (obtained from the first call of **listProperties2 or** from the **diffPropertyContent** call or new properties), it is time to pull the information of all those properties using **readProperty2** for each one to import the "static" information.

Thanks to the Rental Agency ID provided together with the property description as the answer of **listProperties2** and **readProperty2**, your company will be able to assign this property to the appropriate account in your own system, if necessary. This is useful in case you want to differentiate the properties by customer account internally in your system, because of different commercial agreements for example.

You should use **readPropertyPricingSetup2** to retrieve the pricing and rules of minimum stay defined for each property. We strongly advise that you familiarize yourself with all the components of the pricing and rates in the Kigo back office, to have a better understanding of all the information retrieved by **readPropertyPricingSetup2.** Keep in mind that this information is intended to be used as reference for the listings, **it shouldn't be used to calculate the final price of a booking**.

In order to get the final price of a booking you must call the **computePricing** method**.** This way you can **guarantee** that the price offered will be exactly as if it was provided directly by Kigo.
The **computePricing** method will not return a complete breakdown of all the pricing components, and thus you won't be able to display a breakdown of that price. This method also will return an error if the price can't be retrieved for any reason, the property not being available, by example.

Of course, the pricing and rates information of the properties can change, so Kigo strongly suggests that you use the API method **diffPropertyPricingSetup** regularly (between every 30 minutes and 4 hours) to keep the pricing information in your system as much as correctly up-to-date as possible.

It is quite possible that your own system database scheme doesn't make it easy for you to save all the pricing information for properties coming from Kigo. Probably some compromises and simplifications need to be made. You should make that clear to agencies sharing properties with you through the Kigo Channel Manager.

CALENDAR SYNCHRONIZATION

Probably the information that needs to be more up-to-date between your external system and your Kigo account are the availability calendars of the properties. The availability calendars change constantly, so you will need to refresh the availability very often to avoid potential double-bookings.

The API method **diffPropertyAvailability** will return a list of IDs of all the properties whose availability has changed since the last time you invoked this method. As a member of the Diff family methods, the first time you invoke this method (with a DIFF_ID == null) it will obviously return a list with the IDs of ALL your properties. Every time you invoke the method it will also return a new DIFF_ID that you should use in your next diffPropertyAvailability call. If you do it this way, the output of the call will include only the properties that have had availability changes during the interval between both calls.

Please note that **Kigo will not push availability calendar changes to your system** through the API, it is your responsibility to retrieve those changes regularly. To keep your availability calendars up-to-date, Kigo recommends that you use **diffPropertyAvailability** often.

Once you have a list of IDs of properties whose availability has changed, you can start recovering the new availability using **listPropertyAvailability** for each of them for the range of dates you want.

Besides the available dates, you will probably also want to retrieve the information concerning the reservations. You can use **listPropertyCalendarReservations** to retrieve the list of reservation IDs for a specific property and a specific interval of dates.

Now that you have the Kigo ID for all the reservations, you can invoke the API method **readReservation** as many times as it is necessary to retrieve all the available information for every reservation. Please be also aware that you should only try to use **readReservation** to retrieve information of reservations belonging to your agency, otherwise the method will return E_NOSUCH. All the reservations created by your agency (in any property) or the reservations created in your properties (by any agency) “belong” to you and can be read.

If you want to modify the Kigo availability calendars, there is only one way to do it: creating reservations. The API method **createConfirmedReservation** does precisely that. Obviously you also can push changes to the Kigo availability calendars modifying and cancelling reservations (**updateReservation** and **cancelReservation**). Please, notice that you can only modify and delete reservations that belong to your own agency.

COMMON TECHNICAL ISSUES

HTTP response 429

If you get this response to your API call, it means you have exceeded the maximum number of API calls in a given period of time. As stated in the API specification, you should wait (throttle) before retrying. It is much more efficient (time wise) to make sure you never exceed the limit though. This can be easily achieved by implementing a waiting time between API calls.

E_INPUT result

This will be the most typical error you will get while developing your integration. The error code will not indicate what is wrong with your input, but you can be sure that something is.

Most typical INPUT errors include:

- Passing an ID or another integer as if it was a string (i.e. “between quotes”).
- Wrong JSON codification of the **null** value.
- Not passing a mandatory parameter. Some strings can be empty (“”) but are still mandatory in most methods.
- Parameter values that not respect limitations in relation to other parameter values. For example a check-out date that is not in the future in compared to the check-in date.
- Incorrect codification in JSON of Kigo extended data types (see section 7.7.2 of the specifications).

FAQ

I cannot see my portal commission anywhere in Kigo's rent calculation information
Portal commission is not managed within the Kigo rent calculation. It needs to be calculated and managed within your own system.

I can't find any method to create Hold dates.
Correct. Channels API doesn't support the creation of Hold dates, just Pending Confirmation reservations.

Can I send and receive inquiries using Kigo's API?
Unfortunately, at the present Channels API doesn't provide support for inquiries.

How will I know when my provider / reseller agencies cancels or modifies reservations on properties shared with me?
The cancellations and modifications are not pushed to your system. Kigo never pushes any information through the API. As you will be controlling your properties availability calendars using **diffPropertyAvailability**, you will be made aware of changes in availability on your next call to this method and you can then retrieve the specific changes using a combinations of the methods **listPropertyAvailability** and **listPropertyCalendarReservations**.

What will happen to agencies and properties IDs when you migrate them from Kigo Legacy to the new Kigo?
Kigo will take care of migrating all the agencies and properties from the Kigo Legacy platform to the new Kigo platform. Kigo will extend an API method (most probably **listProperties2**) so it return to you the ID that the properties had in the Legacy platform in addition to the ID they have in the new platform. This way, it will be very easy for you to identify those properties and re-map them in your own system as needed.

What is this "certification process" I keep reading about?
Once you have your integration with the new Kigo platform ready (or nearly ready) using our Channels API, you should contact our API team (api@kigo.net) and inform them.
Kigo certification is an informal process that involves our IT team checking the logs of your API integration and talking with your developers to make sure your integration adheres to certain standards that Kigo considers necessary for the portal to go live in our Channel Manager.

These are, for example but not limited to, intervals of availability calendar and price synchronization, correct static information synchronization and correct pricing. Our IT team will then suggest the necessary corrections and modifications for your integration.

This will go on until Kigo considers your integration is correct. After that point you can go "live" in the new Kigo, appear in the Channel Manager and start receiving properties from other agencies in the system.