Kigo

DEVELOPMENT INTRODUCTION

API revision

1

Document revision

4

Contact

api@kigo.net

1 Revision history

API Rev	Document		Date	Changes
	Rev	Status		
1	1			
1	2			Clarifications added.
1	3			Update on the diffs response and usage. Clarification for the booking process added. Requirements for Google added.
1	4			Requirements for Google erased.

2 Contents

Table of Contents

Revision history....2
Contents....3
Glossary....4
Preface....5
What you need to integrate your Kigo Solution with an external system using the Kigo API....5
Reporting issues....5
Copyright notice....5
Working with the API....6
Getting the catalog....6
Getting the content information....6
Getting the pricing structure....6
Getting the LOS pricing....6
Getting the availability....6
Keeping the information updated....6
Bookings management....7
Booking process....7
Language....8
Payment process....8
Redirection after booking....9
Booking details....9
Cancellations....9

3 Glossary

UNIT	The minimum independent (with its own key) element bookable.
Listing	An advertiser of one or more units that have the same characteristics: location, size, capacity, amenities and pricing.
Content information	The information of any Listing that is less susceptible to change, such the name, description, photos, number of beds or amenities, i.e.
Pricing structure	The information used to calculate how it costs to book any Listing: daily rates, fees, taxes, discounts. Example: Building 1 has 4 units (Units 1 to 4) each one of them with 2 beds and 2 baths. Building 1 has also 4 units (Units 5 to 8) each one of them with 1 bed and 1 bath. Building 2 has 3 units (Units 9 to 11) each one of them with 2 beds and 2 baths. Building 2 has also 2 units (Units 12 to 13) each one of them with 1 bed and 1 bath. Listing 1 contains: Unit 1, Unit 2, Unit 3, Unit 4, Unit 9, Unit 10 and Unit 11. Listing 2 contains: Unit 5, Unit 6, Unit 7, Unit 8, Unit 12 and Unit 13.

4 Preface

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

The API consists on a series of methods invoked by making HTTPS requests on the Kigo Rental Agency REST API servers. These methods can pull information from the Kigo system and can also create and cancel reservations.

It uses the OAuth 2.0 protocol for authentication and authorization.

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

In order to get the authorization, you'll need:

Token URL: https://auth.kigo.net/connect/token

Client ID: it will be provided by Kigo

Client Secret: it will be provided by Kigo

Scope: cm.api

Once you have the token, you must include it in the Header of the Listing API and Kigo Checkout API calls, "Authorization": "Bearer {access_token}".

4.2 Reporting issues

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

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 APIs and generic best approaches.

4.3 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.

5 Working with the API

We suggest the following order of operations/API calls:

5.1 Getting the catalog

You will need to pull all the information from the Listings to your own system and store it there in your own database.

You can get it the catalog by making a call to the endpoint **/channels/v2/listings.** passing each of these ids as **ccId**, so you'll get the list of all the Listings shared with your Kigo account.

5.2 Getting the content information

Once you have the list of Listings Ids (obtained from the first call to of **/listings**) it is time to pull the information of all those Listings calling to the endpoint **/listings/{id}** for each one to import the Content information.

5.3 Getting the pricing structure

The endpoint **/listings/{id}** will also provide the pricing structure for each Listing. Although you can use this information to show the breakdown of the rates, fees, taxes or discounts, **it should not be used to calculate the final price of a booking.**

5.4 Getting the LOS pricing

The endpoint **/listings/{id}/lospricing** will provide the price per day and number of guests of a Listing for the requested period. You can store the information provided in your side to use it later for searching by price, i.e.

5.5 Getting the availability

The endpoint **/listings/{id}/availability** will also provide the calendar for any given Listing within a range of specific dates. The data provided by it combined with some of the information given by **/listings/{id}**, such as the blocked out nights or the maximum stay allowed, i.e.

5.6 Keeping the information updated

You need to make calls regularly to the endpoint **/listings/diff** to get the list of the Listings that have changed their content, pricing, availability and/or status information since the last time you called.

Then, you should call **/listings/{id}, /listings/{id}/lospricing** and **/listings/{id}/availability** to get the changes and update the information on your side.

If a change of the Listing **status** has been indicated, you need to take it out of the catalog, any attempt to retrieve information regarding that listing will fail.

Note: the required parameter **modifiedAfter** is limited to 7 days in the past. Since you must make the calls with a higher frequency than that, this limitation shouldn't cause any issue.

5.7 Bookings management

5.7.1 Booking process

In order to create a booking, you need to make a call to the Kigo Checkout API, get the breakdown price and the URL needed to complete the booking and provide them to the potential guest from your property page.

Example:

This is a typical setup for a property page, every time the potential guest changes the dates or the number of guests, a call to the Kigo Checkout API is made, the page shows the breakdown provided by the API and set the “BOOK NOW” button to redirect the visitor to the Kigo Checkout page properly, using the URL provided by the API:

Here the visitors can confirm the reservation, indicating their personal information and the data needed for the payment.

Please, take a look at the “KIGO Checkout for KIGO NCM.pdf” document to find out how to work with Kigo Checkout.

5.7.2 Language

If you want to set the language of the Kigo Checkout webpage, because you have a multi-language site, i.e., you can do it through the param “culture” to any of the accepted values.

5.7.3 Payment process

You can set how the payment is going to be processed through the “checkouttype” field. If you indicate that is going to be “delegated”, the payment information won’t (credit card details or Paypal account) be requested, a held, unpaid booking will be created so you’ll need to process the payment and confirm the booking through Kigo Pro.

Note: **In order to be able to use the “delegated” option properly, your Kigo account will require a special configuration, please request it to our API team by email.**

5.7.4 Redirection after booking

You can set the page to which the guest will be redirected after the booking process through the “successurl”, “failureurl” or “cancellationurl” params.

Note: Take in account that Kigo won’t develop any of these pages as they are part of your integration.

5.7.5 Booking details

You can use the booking id to verify the booking has being created properly and get its information through the endpoint **/bookings/{id}.**

5.7.6 Cancellations

If you need to cancel a booking, you can do it through the endpoint **bookings/{id}/cancel**