



FlipKey Supplier API

Version 1.0 (March 28, 2012)

Intake API

[Change Log](#)

[API Format](#)

[Authentication](#)

[API Request Formats](#)

[API Response Formats](#)

[Data Formats](#)

[API Functions](#)

[1. Owner Data](#)

[1.1. New Owner - POST](#)

[1.2. Update Owner - PUT/PATCH](#)

[1.3. View Owner\(s\) - GET](#)

[2. Property Data](#)

[2.1. New Property - POST](#)

[2.2 Update Property - PUT/PATCH](#)

[2.3 Delete Property - DELETE](#)

[2.4. View Property\(ies\) - GET](#)

[3. Property Availability](#)

[3.1. Set All Booked Dates - PUT](#)

[3.2. Add Booked Dates - PATCH](#)

[3.3. Remove Booked Dates - DELETE](#)

[4. Reservations](#)

[4.1. Add Reservations - PUT](#)

[5. Lead Export](#)

[5.1. Leads - GET](#)

[Examples](#)

[Owners](#)

[Properties](#)

[Availability](#)

[Reservations](#)

Change Log

| | |
|-----------|--|
| 5/7/2012 | Updated requirements for address data, added verification_email |
| | Added note regarding no HTML in any description fields |
| 5/22/2012 | Update requirements for owner address data, plan to update GET to show warnings where an owner can not be verified without specific data |

API Format

The FlipKey API service can be used by a https call and any required POST data. All request and response data will be in JSON format. For each resource (owner, property, etc), a set of GET, POST, PUT, PATCH, DELETE calls can be used to control data. This documentation will define which functions are allowed for each resource and the format of the request and response data.

Request headers should be set with the following:

Content-Type: application/json

All requests will be sent over HTTPS for security.

Authentication

Each partner will be given a username and API key to access the FlipKey API. Using these credentials with basic authentication, the client can access the API.

API Request Formats

Each request will follow API Rest convention:

Fetching Data:

This will be done by sending a GET request using a URL like: <https://ws.flipkey.com/api/v1/property/12345/>

Creating property or owner:

This will be done by sending a POST request using a URL like:
<https://ws.flipkey.com/api/v1/property/>

Updating data:

This will be done by sending a PATCH request using a URL like:

<https://ws.flipkey.com/api/v1/property/12345/>

Replace data:

This will be done by sending a PUT request using a URL like:

<https://ws.flipkey.com/api/v1/property/12345/>

Deleting data:

This will be done by sending a DELETE request using a URL like:

<https://ws.flipkey.com/api/v1/property/12345/>

API Response Formats

| Status Code | Description |
|-----------------------------|--|
| 200 (OK) | GET request was successful. |
| 201 (CREATED) | New object created in response to POST request. Header will include a "Location" URI of the object. The object key can be parsed from the end of this URI or will be the "key" value when following the URI. |
| 202 (ACCEPTED) | PATCH (update) request was successful. |
| 204 (NO CONTENT) | PUT (replace) request was successful. |
| 400 (BAD REQUEST) | <p>The request could not be parsed due to one of the following</p> <ul style="list-style-type: none">• invalid request format• missing required data• data failed validation <p>If required data is missing or data fails validation, the response will be formatted as the request with information about why the request failed.</p> |
| 401 (UNAUTHORIZED) | Not authorized to make request. Username and password may be incorrect or may not have access to the API request. |
| 500 (INTERNAL SERVER ERROR) | This can be caused by an invalid API request or by an error on the server. |

Data Formats

These are the common formats that will be used in the FlipKey API.

| Member | Description |
|----------------|---|
| string | Any string. Restrictions on values and length requirements will be noted. |
| integer | A whole number. |
| decimal | A decimal number. |
| date | Formatted to conform to the extended form of ISO 8601 - eg. 2012-06-20 http://en.wikipedia.org/wiki/ISO_8601#Calendar_dates |
| boolean | Value of <i>true</i> or <i>false</i> . |
| object | A JSON array of data. |

API Functions

1. Owner - add and update owner data
2. Property- add, update, delete properties
3. Availability - add property booked dates
4. Reservations - add reservation data for review collection
5. Leads - export leads

1. Owner Data

Add and update owner data.

Function URL: <https://ws.flipkey.com/api/v1/owner/>

Allowed functions: GET, POST, PATCH, PUT

1.1. New Owner - POST

Add a new owner.

1.1.1 Input

This method requires a POST request with the following data. (Sample request date in Examples section).

| Member | Data type | Description | Required |
|------------|-----------|-------------------------------|----------|
| remote_key | string | Partner unique key for owner. | Yes |
| owner | object | Container for full owner data | Yes |

The **owner** object is defined as follows

| Member | Data type | Description | Required |
|------------|-----------|---|----------|
| email | string | Owner's email address (will be used for login). | Yes |
| password | string | Owner's password to be used on flipkey.com - this will only be used for owner creation and will not be included in a GET request. | Yes* |
| first_name | string | Owner's first name. | Yes |
| last_name | string | Owner's last name. | Yes |
| street_1 | string | Street address | Yes (1) |

| | | | |
|--------------------|---------------------------|---|---------|
| street_2 | string | Street address, second row | No |
| city | string | City | Yes (1) |
| state | string | State/Province | Yes (1) |
| country | string | Country (ISO-3166-1 alpha-2 format, 2 letters) | Yes (1) |
| zip | string | Zip code | No |
| website | string | Owner's website. | No |
| phone_number | string | Owner's phone number. | Yes |
| phone_country | string | Country of phone number - ISO-3166-1 alpha-2 format, 2 letters. Used for formatting phone number display | Desired |
| contact_preference | string ("email", "phone") | Contact preferences for owner. Email indicates email only, Phone indicates email and phone. (default: phone) | Desired |
| verification_email | string | A billing email for the owner, will not be published on www.flipkey.com , but used to verify owner data. | |
| ip_address | array of string | List of IP addresses of owner. | No* |

* password and ip_address only used for owner creation and will be ignored in subsequent updates.

(1) This data is required to verify the owner before the initial property is loaded. You can perform a PUT after the fact and not supply these fields, so its not a hard requirement for now, but they must be supplied before activation. Exceptions are made for city and state/province where they are not applicable, but most countries contain these entities.

1.1.2 Response

A 201 (CREATED) response will be returned when a new owner has successfully been added. The "Location" header will be a URI in the format: https://ws.flipkey.com/api/v1/owner/{owner_key}. The owner_key will be needed to add new properties to the owner or update the owner data. This key can be parsed from the URI or retrieved from the response data to a GET request to the URI.

1.2. Update Owner - PUT/PATCH

Update owner data. The URL will use the function URL with the owner_key (eg. <https://ws.flipkey.com/api/v1/owner/12345/>)

1.2.1 Input

This method requires a PUT or PATCH request with the following data.

| Member | Data type | Description | Required |
|--------|-----------|-------------------------------------|----------|
| owner | object | Container of owner data (see 1.1.1) | Yes |

1.2.2 Response

A 202/204 response to a successful request.

1.3. View Owner(s) - GET

Retrieve current owner data.

1.3.1 Input

This method requires a GET request. Omitting the owner_key will return all owners for partner. Including the owner_key (ie. <https://ws.flipkey.com/api/v1/owner/12345/>), will return only a single owner.

1.3.2 Response

If all owners are requested, an array of owner objects will be returned, indexed by 'owners'. If a single owner is requested, only the single owner object will be returned. The owner object is defined in 1.1.1.

2. Property Data

Add, update, remove properties.

Function URL: <https://ws.flipkey.com/api/v1/property/>
Allowed functions: GET, POST, PATCH, DELETE

2.1. New Property - POST

Add a new property

2.1.1 Input

This method requires a POST request with the following data. (Sample request date in Examples section).

| Member | Data type | Description | Required |
|------------|-----------|----------------------------------|----------|
| owner_key | integer | User ID | Yes |
| remote_key | string | Partner unique key for property. | Yes |
| property | object | Container for full property data | Yes |

The **property_data** is defined as follows

| Member | Data type | Description | Required |
|--------|-----------|-------------|----------|
|--------|-----------|-------------|----------|

| | | | |
|--------------|-----------------|---|--------|
| details | object | Property specific details | Yes |
| address | object | Property address data | Yes |
| amenities | array of string | List of amenities | No |
| bathrooms | array of object | List of bathroom descriptions | No |
| bedrooms | array of object | List of bedroom descriptions | No |
| descriptions | array of object | List of property, rate, location descriptions | Yes * |
| fees | array of object | List of property fees | No |
| photos | array of object | List of property photos | Yes ** |
| rates | array of object | List of property rates | Yes |
| videos | array of object | List of property videos | No |

*Property Required

**Minimum 5 photos required.

details

| Member | Data type | Description | Required |
|-----------------|---------------------------|-----------------------------------|----------|
| name | string | Property display name | Yes |
| occupancy | integer | Number of persons property sleeps | Yes |
| bedrooms | integer | Number of bedrooms in property | Yes |
| bathrooms | integer | Number of bathrooms in property | Yes |
| unit_size | integer | Size of unit (in feet or meters) | No |
| unit_size_units | string ("feet", "meters") | Units for unit_size | No |
| smoking | boolean | Smoking allowed in property | No |
| pet | boolean | Pets allowed in property | No |
| children | boolean | Property is child friendly | No |
| wheelchair | boolean | Property is wheelchair accessible | No |
| elderly | boolean | Property is elderly accessible | No |

| | | | |
|---------------|--|--|----|
| property_type | string ("house", "condo", "cabin", "villa", "resort", "b&b", "chalet", "castle", "yacht", "cottage", "houseboat", "estate", "farmhouse", "other") | Type of property, defaults to house. | No |
| check_in | string | Time property is available for check in. This is required to be in 24-hour clock time, with the format "hh:mm" (eg. 15:30). | No |
| check_out | string | Time property is available for check out. This is required to be in 24-hour clock time, with the format "hh:mm" (eg. 11:30). | No |

address

| Member | Data type | Description | Required |
|------------|-----------|---|-------------|
| address1 | string | Street address line 1 | Yes(1) |
| address2 | string | Street address line 2 | No |
| city | string | City | Yes |
| state | string | State | Desired (2) |
| country | string | Country | Yes |
| zip | string | Zip code | No |
| latitude | decimal | Latitude of property | Yes (1) |
| longitude | decimal | Longitude of property | Yes (1) |
| show_exact | boolean | Indicates if exact location is displayed in listing (default: true) | No |

1) Either the address1 or latitude and longitude are required. If coordinates are included but are not the exact location of property, show_exact address should be set to False.

2) Certain areas do not have state/province or postal codes, so these can be omitted, but the goal is to get as much address data as possible without doing a reverse lookup from latitude/longitude.

bathrooms

| Member | Data type | Description | Required |
|-------------|--|-------------------------|----------|
| type | string ("full", "half", "shower", "other") | Type of bathroom | Yes |
| description | string | Description of bathroom | No |

bedrooms

| Member | Data type | Description | Required |
|-------------|-----------|---|----------|
| king | integer | Number of king beds in bedroom | No * |
| queen | integer | Number of queen beds in bedroom | No * |
| standard | integer | Number of standard/full beds in bedroom | No * |
| twin | integer | Number of twin beds in bedroom | No * |
| bunk | integer | Number of bunk beds in bedroom | No * |
| other | integer | Number of beds of other type in bedroom | No * |
| description | string | Description of bedroom | No |

* Must specify at least one of king, queen, standard, twin, bunk, or other.

descriptions

| Member | Data type | Description | Required |
|----------|-----------|---|----------|
| property | string | Description of property | Yes * |
| rates | string | Additional information about rates and fees | No * |
| location | string | Additional information about location of property | No * |

All HTML must be stripped from posted data. You can use newlines to represent `
` and 2 newlines to represent `<p>data</p>`

fees

| Member | Data type | Description | Required |
|-------------|---|--|----------|
| type | string ("tax", "fee", "per-person", "security-deposit", "other") | Type of fee | Yes |
| name | string | Name of fee | Yes |
| description | string | Description of fee | No |
| units | string ("percent", "flat", "day", "week", "month", "person") | Unit application of fee | Yes |
| cost | decimal | Fee value | Yes |
| optional | boolean | Indicates if fee is optional (default: true) | No |

photos

| Member | Data type | Description | Required |
|-------------|---|--|----------|
| url | string | URL of photo (http://...) | Yes |
| description | string | Caption of photo | No |
| type | string ("bathroom", "bedroom", "diningroom", "exterior", "kitchen", "livingroom", "area", "other") | Type of photo content, no type is set. | No |

rates

| Member | Data type | Description | Required |
|------------|-----------|------------------------------|----------|
| start_date | date | Starting date of rate range. | Yes |
| end_date | date | Ending date of rate range. | Yes |

| | | | |
|--------------------|---------|--|------|
| label | string | Description of rate range | No |
| weeknight_min_rate | decimal | Minimum weeknight nightly rate | No * |
| weeknight_max_rate | decimal | Maximum weeknight nightly rate | No * |
| weekend_min_rate | decimal | Minimum weekend nightly rate | No * |
| weekend_max_rate | decimal | Maximum weekend nightly rate | No * |
| week_min_rate | decimal | Minimum weekly rate | No * |
| week_max_rate | decimal | Maximum weekly rate | No * |
| month_min_rate | decimal | Minimum monthly rate | No * |
| month_max_rate | decimal | Maximum monthly rate | No * |
| minimum_length | integer | Minimum nights required for booking in rate range. (Defaults to 1 night) | No |

* Requires at least one rate be provided.

videos

| Member | Data type | Description | Required |
|-------------|--------------------|---|----------|
| source | string ("youtube") | Video hosting source (only YouTube currently allowed) | Yes |
| url | string | Embed URL to video | Yes |
| description | string | Description of video | No |

2.1.2 Response

A 201 (CREATED) response will be returned when a new owner has successfully been added. The "Location" header will be a URI in the format: `http://ws.flipkey.com/api/v1/property/{property_key}/`. The `property_key` will be needed to add update the property data. This key can be parsed from the URI or retrieved from the response data to a GET request to the URI.

2.2 Update Property - PUT/PATCH

Update property data. The URL will use the function URL with the `property_key` (eg. `https://ws.flipkey.com/api/v1/property/12345/`)

2.2.1 Input

This method requires a PUT or PATCH request with the following data.

| Member | Data type | Description | Required |
|----------|-----------|--|----------|
| property | object | Container of property data (see 2.1.1) | Yes* |

| | | | |
|---------|---------|--|----|
| display | boolean | Set property to be displayed or hidden on the site. (true: property will be displayed, false: property will be hidden) | No |
|---------|---------|--|----|

* property not required if changing display option of property

2.2.2 Response

A 202/204 response to a successful request.

2.3 Delete Property - DELETE

Remove property from account. The URL will use the function URL with the property_key (eg. <https://ws.flipkey.com/api/v1/property/12345/>)

2.3.1 Input

This method requires a DELETE request.

2.3.2 Response

A 204 response will indicate the property was successfully deleted.

2.4. View Property(ies) - GET

Retrieve current property data.

2.4.1 Input

This method requires a GET request. Omitting the property_key will return all properties for partner. Including the property_key (ie. <https://ws.flipkey.com/api/v1/property/12345/>), will return only a single property.

2.4.2 Response

If all properties are requested, an array of property objects will be returned, indexed by 'properties'. If a single property is requested, only the single property object will be returned. The property object is defined in 2.1.1.

3. Property Availability

Set dates as booked.

Function URL: <https://ws.flipkey.com/api/v1/availability/>

Allowed functions: GET, PUT, PATCH

3.1. Set All Booked Dates - PUT

Set all booked dates for property - replaces current availability. The URL will use the function URL with the property_key (eg. <https://ws.flipkey.com/api/v1/availability/12345/>)

3.1.1 Input

This method requires a PUT request with the following data. (Sample request date in Examples section).

| Member | Data type | Description | Required |
|--------|------------------|---|----------|
| dates | array of objects | List of start and end dates indicating dates that are booked. | Yes |

dates - all dates between start_date and end_date (inclusive) will be marked as booked.

| Member | Data type | Description | Required |
|------------|-----------|---|----------|
| start_date | date | Start date of booked range | Yes |
| end_date | date | End date of booked range (this date will be marked as booked) | Yes |

3.1.2 Response

A 204 response to a successful request.

3.2. Add Booked Dates - PATCH

Set range of dates as booked. All other dates will remain unchanged. The URL will use the function URL with the property_key (eg. <https://ws.flipkey.com/api/v1/availability/12345/>)

3.2.1 Input

This method requires a PATCH request with the same data as required in 3.1.1.

3.2.2 Response

A 202 response to a successful request.

3.3. Remove Booked Dates - DELETE

Remove range of booked dates. The URL will use the function URL with the property_key (eg. <https://ws.flipkey.com/api/v1/availability/12345/>)

3.3.1 Input

This method requires a DELETE request with the same data as required in 3.1.1.

3.3.2 Response

A 204 response will indicate the dates were successfully deleted.

4. Reservations

Add reservation data for review collection.

Function URL: <https://ws.flipkey.com/api/v1/reservations/>

Allowed functions: GET, PATCH

4.1. Add Reservations - PUT

Add new reservations. The URL will use the function URL with the property_key (eg. <https://ws.flipkey.com/api/v1/reservations/12345/>)

4.1.1 Input

This method requires a PATCH request with the following data. (Sample request date in Examples section).

| Member | Data type | Description | Required |
|--------------|------------------|---------------------------|----------|
| reservations | array of objects | List of reservation data. | Yes |

reservations - all reservations that have checked out that need a review request.

| Member | Data type | Description | Required |
|-----------------|-----------|---------------------------|----------|
| email | string | Email of guest | Yes |
| start_date | date | Start date of reservation | Yes |
| end_date | date | End date of reservation | Yes |
| first_name | string | First name of guest | No |
| last_name | string | Last name of guest | No |
| reservation_key | string | Unique reservation key | No |

4.1.2 Response

A 202 response to a successful request.

5. Lead Export

Export of lead data and summary.

Function URL: https://ws.flipkey.com/api/v1/leads/{owner_key}/

Allowed functions: GET

5.1. Leads - GET

Export all leads for an owner.

5.1.1 Input

This method requires a GET request. Leads can be filtered by the following options. The request URL with filtering options will be structured like: https://ws.flipkey.com/api/v1/leads/{owner_key}?property_key=12345&start_date=2012-01-01&end_date=2012-03-01

| Result | Type | Description |
|--------|------|-------------|
|--------|------|-------------|

| | | |
|--------------|---------|---|
| property_key | integer | FlipKey unique identifier |
| start_date | date | Filter only leads received on or after this date |
| end_date | date | Filter only leads received on or before this date |

5.1.2 Response

A 200 response to a successful request with the following data.

| Result | Type | Description |
|--------|------------------|--|
| leads | array of objects | List of leads based on filter options. |

leads

| Result | Type | Description |
|--------------|---------|---------------------------|
| property_key | integer | FlipKey unique identifier |
| name | string | Name of guest |
| email | string | Email of guest |
| phone_number | string | Phone number of guest |
| check_in | date | Requested check in date |
| check_out | date | Requested check out date |
| message | string | Message from guest |
| requested | date | Date guest entered lead |

Examples

Owners

Full Example

```
{
  "remote_key": "1234",
  "owner": {
    "email": "test@test.com",
    "password": "testpass",
    "first_name": "Bob",
    "last_name": "McGee",
    "street_1": "123 Test Lane",
    "city": "Testertown",
    "state": "Missouri",
    "country": "US",
    "zip": "12345",
    "website": "www.test.com",
    "phone_number": "555-555-1234",
    "phone_country": "US",
    "contact_preference": "phone",
    "ip_address": ["255.1.1.1", "201.12.15.24"]
  }
}
```

Minimal Example

```
{
  "remote_key": "1234",
  "owner": {
    "email": "test@test.com",
    "password": "testpasswd",
    "first_name": "Bob",
    "last_name": "McGee",
    "phone_number": "555-555-1234",
  }
}
```

Properties

Full Example

```
{
  "owner_id": 515322,
  "remote_id": "test_prop",
  "property": {
    "details": {
      "name": "Test Property",
      "occupancy": 6,
    }
  }
}
```

```

    "bedrooms":3,
    "bathrooms":42,
    "unit_size":3000,
    "unit_size_units":"feet",
    "smoking":false,
    "pet":true,
    "children":false,
    "wheelchair":true,
    "elderly":false,
    "property_type":"castle",
    "check_in":"15:00",
    "check_out":"10:00"
  },
  "address":{
    "address1":"179 Lincoln St.",
    "address2":"Suite 405",
    "city":"Boston",
    "state":"MA",
    "country":"US",
    "zip":"02111",
    "latitude":42.350512,
    "longitude":-71.058889
  },
  "amenities":["TV", "Couch", "Fruit Snacks"],

  "bathrooms":[{"type":"full",
    "description":"Full bathroom description here."},
    {"type":"half",
    "description":"Half bathroom description here."}
  ],
  "bedrooms":[{"king":2,
    "queen":0,
    "standard":0,
    "twin":0,
    "bunk":0,
    "other":0,
    "description":"Bedroom description here. You can set bed types to zero if it's
easier."},
    {"queen":1,
    "twin":1,
    "description":"Bedroom description here. You can list more than one bed type."},
    {"bunk":1,
    "description":"Bedroom description here. You don't need to list types you aren't
using."}
  ],
  "descriptions":{
    "property":"This property and it's sheer number of bathrooms will shock and amaze
you.",
    "rates":"Additional rate information here.",
    "location":"Information about the location of the property here."
  },

```

```

    "fees":[{"type":"tax",
      "name":"Rental Tax",
      "description":"Description of the fee here.",
      "units":"percent",
      "cost":10,
      "optional":false}
    ],
    "photos":[{"url":"www.example.com/photo1.jpg",
      "description":"Description text goes here.",
      "type":"exterior"
    },
    {"url":"www.example.com/photo2.jpg",
      "description":"Description text goes here.",
      "type":"bedroom"
    },
    {"url":"www.example.com/photo3.jpg",
      "description":"Description text goes here.",
      "type":"bathroom"
    },
    {"url":"www.example.com/photo4.jpg",
      "description":"Description text goes here.",
      "type":"kitchen"
    },
    {"url":"www.example.com/photo5.jpg",
      "description":"Description text goes here.",
      "type":"area"
    }
  ],
  "rates":[{"start_date":"2012-05-01",
    "end_date":"2012-10-31",
    "label":"High Season",
    "week_min_rate":4242.42,
    "week_max_rate":4242.42,
    "minimum_length":7},
    {"start_date":"2012-11-01",
    "end_date":"2013-04-30",
    "label":"Low Season",
    "weeknight_min_rate":130.00,
    "weeknight_max_rate":130.00,
    "minimum_length":2}
  ],
  "videos":[{"source":"youtube",
    "url":"http://www.youtube.com/watch?v=XXXXXXX",
    "description":"Video description."}
  ]
}
}

```

Minimal Example

The more data supplied for a property the better, refer to the required table in 2.1.1. Here you have more of an option of omitting an entire block such as videos, or sending an empty list. But as you supply key data (rates, photos etc) as described above the required attributes should be followed.

Availability

```
{"dates": [  
  {"start_date": "2012-03-01",  
    "end_date": "2012-03-15"  
  },  
  {"start_date": "2012-03-22",  
    "end_date": "2012-03-25"  
  },  
  {"start_date": "2012-04-08",  
    "end_date": "2012-04-15"  
  }  
]
```

Reservations

```
{"reservations": [  
  {"email": "guest_1@test.com",  
    "start_date": "2012-03-01",  
    "end_date": "2012-03-15",  
    "first_name": "Jon",  
    "last_name": "Doe",  
    "reservation_key": "ABC123"  
  }  
]
```