MIDAS

API Documentation

https://mid.as/api

v2.60

API

...making your facilities work for you!

Table of Contents

API Usage Guide

Overview

The MIDAS API (Application Programming Interface) allows developers to interface directly with a MIDAS booking system from their own websites and applications.

Installing the API

The MIDAS API is an optional add-on for MIDAS v4.03 (or later), that can be purchased along with a MIDAS booking system, or added at a later stage.

- To purchase MIDAS and the API addon, please visit: https://mid.as/pricing
- To add the API addon to an existing MIDAS installation, please go to MIDAS Admin Options → Manage Addons → Available Addons → API Access.

Once purchased, the API will become available for one-click installation via MIDAS Admin Options → Manage Addons → Addons Ready To Install → API Access.

Once installed, various API settings are available via MIDAS Admin Options → Manage Addons → Installed Addons → API Access.

Making API calls

API calls to a MIDAS booking system can be made in a number of ways:

API calls via HTTP

To make an API call, your application will need to perform an HTTP POST request (HTTP GET requests can also be enabled - see API Settings below) to https://*your_midas_url*/api.pl.

Each request must include as a minimum:

1. Your unique API key.
2. An API command.
3. All associated required parameters (if applicable) for the particular API command being issued.

The API key itself may be passed either in a GET/POST parameter named "key", or via an "X-API-KEY" HTTP header (recommended).

Please refer to the API Command Reference for details of available API commands and associated parameters.

API calls from the Command Line

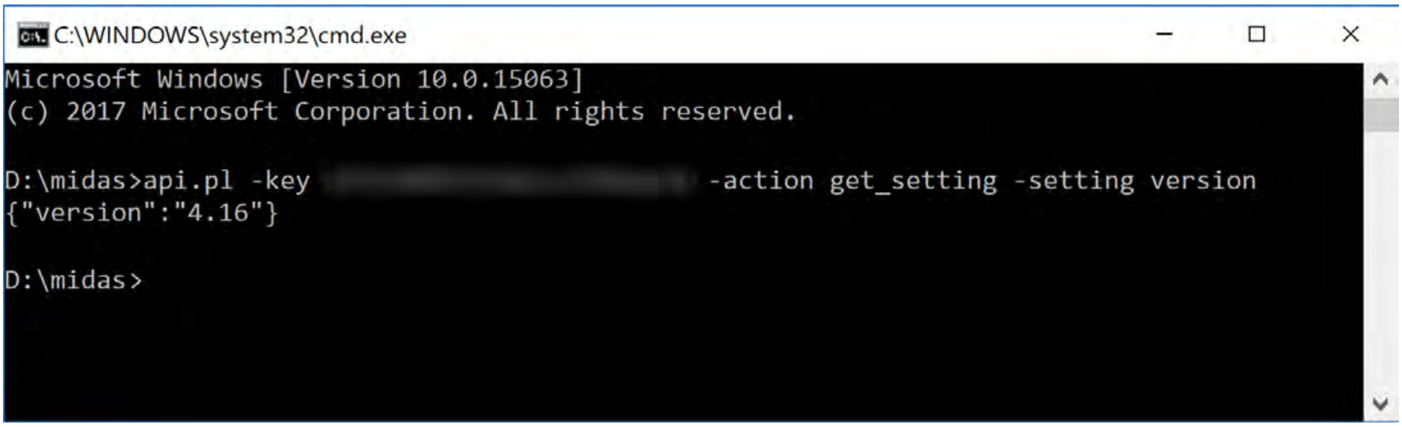
(Available in API v2.19+ and for self-hosted editions of MIDAS only)

In addition to making API calls via HTTP, the API also permits calls to be made directly from the command line.

API calls made in this way have the added benefit of being significantly faster (assuming the call is made from the command line on the same server as the MIDAS system), as they do not require an HTTP transaction to be conducted.

To make API calls from the command line, simply call "api.pl" (located within your server's MIDAS directory) and pass API parameters & values via command line switches.

For example, to retrieve the currently installed version of MIDAS from the Windows command line (cmd):



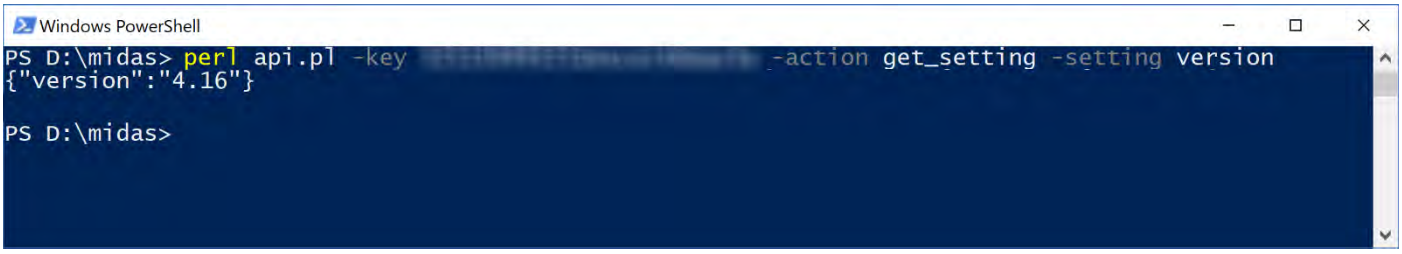
C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

D:\midas>api.pl -key [redacted] -action get_setting -setting version
{"version":"4.16"}

D:\midas>
```

...or from Windows PowerShell:



Windows PowerShell

```
PS D:\midas> perl api.pl -key [redacted] -action get_setting -setting version
{"version":"4.16"}

PS D:\midas>
```

API calls from POSTMAN

Postman is a third party API development tool, helping developers build, test, collaborate and integrate APIs.

If you use Postman, we have a MIDAS API collection available for import. For more information, please see our KB article: Using the MIDAS API with Postman.

Fair Usage Policy

We have a "fair usage" policy in place for API calls made to cloud-hosted MIDAS systems. For more information, please see: Usage Limits

API responses

JSON

By default, the MIDAS API returns JSON (JavaScript Object Notation) formatted data. Example responses are shown throughout the API Command Reference. Error responses are also returned in JSON format and denoted by the term "error".

JSONP

Optionally, the MIDAS API (from API v2.19 onwards) can return JSONP (JSON with Padding) data.

JSONP is a method commonly used to bypass the cross-domain policies in web browsers. Typically, modern browsers won't allow you to make AJAX (Asynchronous Javascript) requests from one domain to another domain perceived to be on a different server.

For example, if your MIDAS system is running on server A, and you wish to make client-side API calls through Javascript, you would typically only be able to make such AJAX requests from pages residing on server A itself. Attempting to initiate an AJAX request for server A from server B would be blocked by the user's browser.

JSON and JSONP behave differently on the client and the server. JSONP requests are not dispatched using the XMLHTTPRequest and the associated browser methods. Instead, a <script> tag is created, the source of which is set to the target URL. This script tag is then added to the DOM (normally inside the element).

JSONP support in the MIDAS API (which is disabled by default) can be enabled via the API settings screen.

⚠ Warning: Before enabling JSONP support, you should instead look to use CORS (Cross Origin Resource Sharing) wherever possible, as JSONP has inherent security risks as it injects Javascript code directly into your web pages

Consider a simple API call to retrieve the version number of MIDAS (passing the "`action`" parameter with a value of "`get_setting`", and a "`setting`" parameter with a value of "`version`"). The typical JSON response would be:

Example JSON Response:

```
{"version":"4.32"}
```

Now consider the same API call, with JSONP support enabled, and an additional "`callback`" parameter passed with a value of "`myfunction`"). The JSONP response would then be:

Example JSONP Response:

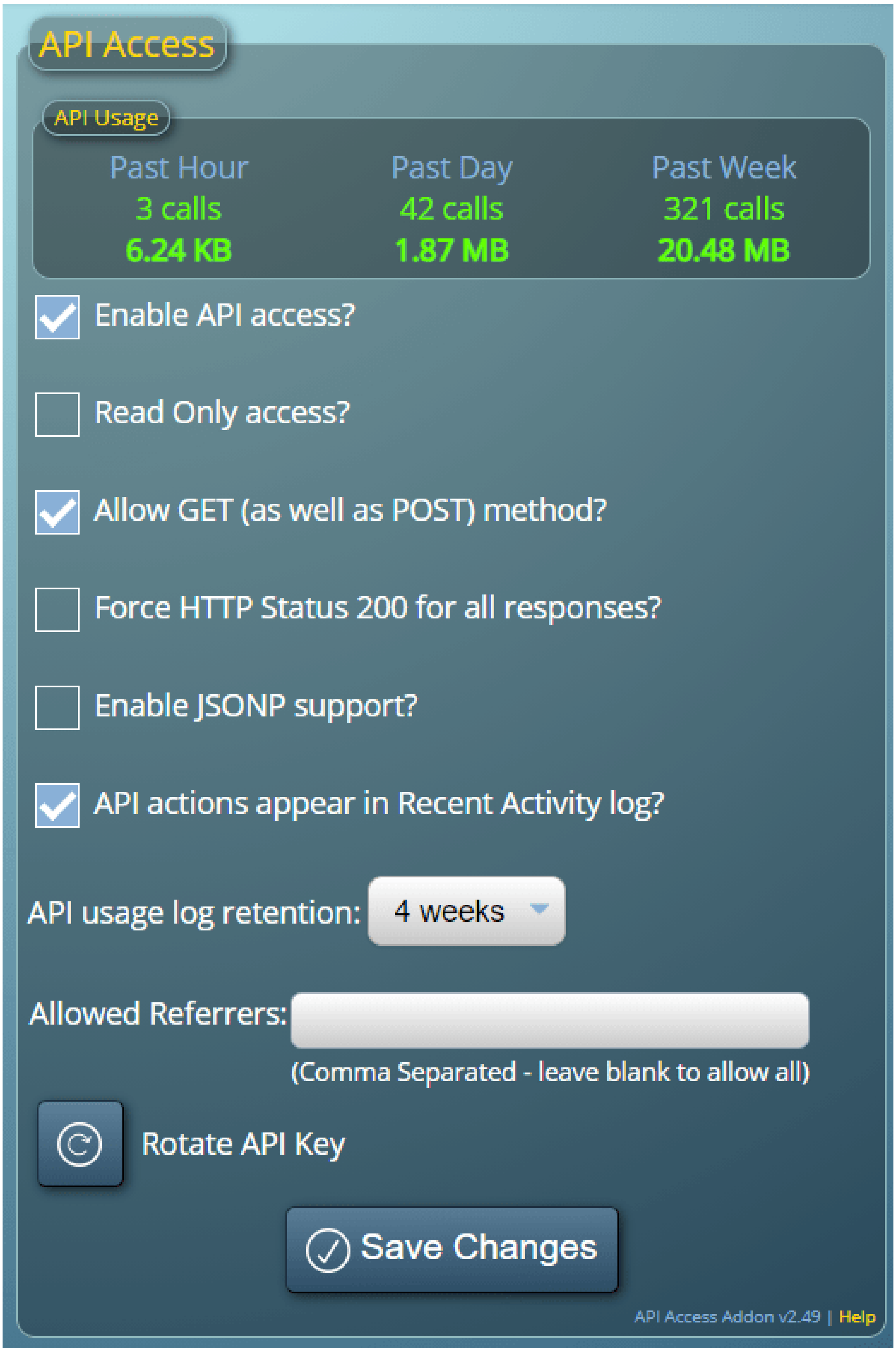
```
myfunction({"version":"4.32"})
```

The "`callback`" parameter must contain the name of an existing Javascript function on the calling page.

Upon receiving the JSONP response, the user's browser will execute the "`myfunction`" Javascript function, passing the JSON data `{"version":"4.32"}` to it accordingly.

API Settings

A number of API settings are available via MIDAS Admin Options → Manage Add-ons → API Access.



API Access

API Usage

Past Hour	Past Day	Past Week
3 calls	42 calls	321 calls
6.24 KB	1.87 MB	20.48 MB

☑ Enable API access?

□ Read Only access?

☑ Allow GET (as well as POST) method?

□ Force HTTP Status 200 for all responses?

□ Enable JSONP support?

☑ API actions appear in Recent Activity log?

API usage log retention: 4 weeks

Allowed Referrers:

(Comma Separated - leave blank to allow all)

Setting	Description
Enable API access?	Enables or disables API access. If disabled, any API calls will return: `{"error":"not_enabled"}`
Read Only access?	With this setting enabled, the API will operate in "read-only" mode and will be prevented from making any changes to your MIDAS. If you only intend to read data from MIDAS, and not write/make changes using the API, you should select this option.
Allow GET (as well as POST) method?	By default, only http POST requests to the API are permitted, and http GET requests are automatically rejected. Enabling this setting will permit either GET or POST http requests to be made to the API. See also: Should I use GET or POST when making API calls?
Force HTTP Status 200 for all responses?	If enabled, all API calls should return a standard http 200 (OK) status code. If disabled, the http status code returned by the API may vary. Successful API calls will return either a 200 (OK) or 201 (Created) status. 201 status codes are specifically for successful "`add_*`" and "`gen_invoice`" API calls. API calls which don't successfully complete will return a 4xx or 5xx status code. For more information, please refer to Appendix C - Error Handling.
Enable JSONP support?	Enables/Disables returning JSONP data in API responses.
API actions appear in Recent Activity log?	By default when a user performs an action in MIDAS (such as adding a booking), the action is recorded in the Recent Activity log. This setting controls whether actions performed through the API that result in changes to data within MIDAS should also be included in the Recent Activity log.
API usage log retention	Sets how long API usage logs should be retained for. (Only available to self-hosted editions of MIDAS)
Allowed Referrers	Allows calls to your API to be restricted from certain domains/IP addresses. If left blank, API calls will be allowed from any referrer. For calls from banned referrers, the API will return: `{"error":"referrer not allowed"}`
Rotate API Key	Clicking this button will generate and display a new API key, which can then be copied to the clipboard. WARNING: Generating a new API key will immediately invalidate your previous API key. Therefore, after generating a new key, you will need to update your API calls to use the newly generated key.

Global Parameters

The following parameters must be supplied with call made to the MIDAS API:

Required Parameters	Possible Values	Description
key	*<API key>*	Your unique API key. For security, the API will only respond to your unique key. Calls made to the API without a key, or with an invalid key, will not be processed and will return an "invalid api key"/401 Unauthorized error. You should not make your API key visible/available to anyone.
action	*<API command>*	The action you wish the API to perform. See the API Command Reference for a full list of supported commands.

If making API calls via HTTP, Your API "key" may alternatively be passed to the server in an "`X-API-KEY`" HTTP header, rather than as a GET/POST parameter

The following parameters are optional, but may also be included with any API call:

Optional Parameters	Possible Values	Description
l	*<language>*	By default, the API will use the "en-US" language pack where applicable. To optionally use a different language pack, its corresponding language code can be specified.
db	*<database>*	MIDAS allows you to optionally run multiple, independent, databases from the same interface. If multiple databases have been setup in your MIDAS, API calls will be made to whichever database is currently set as the "default". To make API calls to one of the other defined databases, its corresponding database code can be passed in the "`db`" parameter. To locate the correct database code to pass with the "`db`" parameter, open up your midas.dat settings file in a standard text editor and locate the <database> section within this file. This section contains details of all defined databases for your MIDAS. The database "code" to use is the current value of the "r" parameter for the relevant database.
logas	*<user id> or <email address>*	By default, all API calls are actioned in MIDAS under an internal "API" account. By setting the optional "`logas`" parameter to either the internal ID or the email address of an existing user account, API calls will instead be recorded as having been actioned under that particular user account.

runas	*<user id> or <email address>*	Similar to the "`logas`" parameter above, but the "`runas`" parameter abides by a specified user's permissions when making API calls. If both "`logas`" and "`runas`" parameters are supplied, "`runas`" will override "`logas`".
callback	*<Javascript function name>*	If JSONP support is enabled, the `callback` parameter can contain the name of an existing Javascript function to execute upon receiving an API response. The received API response will be passed to this function
epoch	1 | 0	For API calls that require start/end times, setting "`epoch`" to "`1`" will accept start/end values in epoch seconds (instead of the default YYYYMMDDHHMM format) For API calls that return a date/time, setting "`epoch`" to "`1`" will return date/time values in epoch seconds (instead of following the current MIDAS date/time format settings, for instance, "DD/MM/YYYY @ HH:MM"

What are epoch seconds?

Epoch seconds are the number of seconds that have elapsed since midnight Coordinated Universal Time (UTC) on 1 January 1970

Test the API

The MIDAS API has been enabled for testing purposes with our online public demo MIDAS system.

You can make test API calls to the demo through the following URL: `https://demo.mid.as/api.pl`

The Demo API key (which must be included with your API calls) is: **`797769685251f9i80MWOhwOC`**

NOTE: Only "read-only" calls may be made to this demo API

Usage Limits

NOTE: The following section on Usage Limits applies to cloud-hosted customers using the MIDAS API. It does not apply to self-hosted API customers.

In order to ensure service quality for all our cloud-hosted customers, we have a "fair usage" policy in place in relation to making API calls.

API call frequency limits

We employ rate-limiting on inbound traffic to our servers.

To avoid hitting these limits, please ensure that you limit the frequency of API calls to your cloud-hosted MIDAS system.

An excessively high frequency of API calls may result in your IP address being automatically blocked and/or your API access being suspended.

Guidance: Limit the frequency of API calls to an average of no more than one API call per second (and a **maximum of 720 API calls per hour**).

API bandwidth limits

The MIDAS API is capable of returning very large data sets.

Customers routinely making API calls which return extremely large data sets may result in their API access being suspended.

You should therefore carefully consider the parameters you pass along with your API calls to ensure that they restrict the data set being returned to the minimum amount of information you require.

Guidance: Craft your API calls to return the minimum size of data you require (and **a maximum of 480 MB of data per day**).

What happens if I exceed these limits?

If you exceed these limits, API access to your MIDAS system will be temporarily suspended, with subsequent API calls returning an error. API access will be automatically re-instated once API usage falls below these limits again.

For advice on reducing your API usage to avoid hitting these limits, please see our KB article: How to reduce API usage.

Can I monitor my API usage?

From API v4.28+, you can view your current API usage via MIDAS Admin Options → Manage Addons → API Access.

API usage may also be monitored directly through the API itself via the new "`util_api_usage`" call.

Can I increase these limits?

We believe the frequency/bandwidth limits outlined above are fair, and will not impact the vast majority of API users.

However, if you are regularly exceed these limits and are unable to refactor your code to reduce the frequency of API requests and/or the amount of data returned by your API calls, then please contact us. We may be able to increase your limits for an additional charge.

Alternatively, you may like to consider migrating to a self-hosted MIDAS system. API calls to self-hosted MIDAS systems do not have any usage limits imposed.

API Command Reference

add_booking

Adds a new booking.

Required Parameters	Possible Values	Description
Start	Valid date and time (Format: YYYYMMDDHHMM)	Start time
end	Valid date and time (Format: YYYYMMDDHHMM)	End time
venue	Venue ID | Venue Name	The unique numeric ID or the name of the venue the booking is for

Optional Parameters	Possible Values	Description
client	Client ID	The numeric ID of the existing client that the booking to be added is for
type	Booking Type ID | Booking Type Name	The numeric ID of the Booking Type, or the name of the Booking Type the booking should be assigned
attendees	Number	The number of booking attendees
notes	String	Notes about the booking
resources	Quantity & Resource ID |Resource Name	Resources to be added to the booking. Values should be passed as "`qty|resource (ID or name)`". For example, to add 2 Tables to a booking, pass "`2|Tables`". This parameter can be added multiple times to add further resources
accept_limited_resources	0 | 1	For bookings with Resources, this parameter determines whether those bookings should still be added if the quantity of resources requested isn't available. For example, if you add 10 Tables to a booking, yet only 5 Tables are available, an error will be returned and the booking won't be added. However, passing the `accept_limited_resources` parameter

		with a value of "`1`" will add the booking, but with reduced resource quantities (in the previous example, this would mean that only 5 Tables would be added to the booking, instead of the desired 10)
no_watches	0 | 1	Setting the "`no_watches`" parameter to "`1`" will suppress generation of Watch notifications for users who are monitoring new bookings, otherwise omitting this parameter (or setting it to "`0`") will allow Watch notifications to be generated accordingly
status	request | confirmed	If omitted or set to "`confirmed`", bookings will be added. If set to "`request`", the booking will be added as a provisional booking "request" only
<custom fields>	Custom Value	If you've defined custom booking fields in your MIDAS, you can optionally specify values for these fields for the booking you're adding. For example, if you have a custom field named "My Custom Field", you would pass the parameter "`my_custom_field`" together with the value you wish to set

Example Response

```
{
	"booking_added":"123"
}
```

Variables Returned (on success)	
booking_added	A unique booking ID corresponding the newly added booking

Variables Returned (on error)	

error	The reason why the booking couldn't be added. Typical examples include:
	`Unavailable` *(clashes with an existing booking)*
	`Unavailable - Blocked By Meeting Room 1`
	`45 Attendees will exceed the maximum capacity of Meeting Room 1 (15)`
	`Outside Operating Hours`
	`Venue unavailable from 1/1/2015`
	`Venue unavailable between 1/1/2015-1/2/2015`

```
                 Venue not available until 1/1/2015
                 Only x5 Tables Available
```

add_client

Adds a new client record.

Required Parameters	Possible Values	Description
client	String	Client name

Optional Parameters	Possible Values	Description
org	String	The client's Organization name
email	Email address	The client's Email address
address	String	The client's postal/mailing address
phone	Number	The client's telephone number
fax	Number	The client's fax number
mobile	Number	The client's mobile/cell number
notes	String	Notes about the client
tax_exempt	0 | 1	Specifies whether the client is exempt from being charged tax on their invoices. If this parameter is omitted a default value of "0" will be assumed (i.e. client' is not exempt from tax)
credit	Number	Applies an initial amount of credit to the client. Client credit is then automatically applied to their subsequent invoices and decreased accordingly. If this parameter is omitted no credit is applied

Example Response

```
{
      "client_added":"56"
}
```

Variables Returned	
client_added	A unique client ID corresponding the newly added client record

add_invoice

Creates a new invoice.

Required Parameters	Possible Values	Description
client	Client ID	The numeric ID of the existing client that the invoice being added for
item_*x*	String	The item to be invoiced. Increment *x* to each new item to appear on the invoice. For example, to create an invoice with two items, the first item to appear on the invoice should be passed as item_1, the second as item_2, and so forth
qty_*x*	Number	The quantity of the corresponding `item_`*x* item to be added
rate_*x*	Decimal	The rate (charge) (without currency symbol - i.e. 10.00) of the corresponding `item_`*x* item to be added
tax_*x*	Percentage	The tax rate to be charged against the corresponding `item_`*x* item to be added. If omitted, the global Payment Terms setting will be used

Optional Parameters	Possible Values	Description
type	regular | deposit | cancel | credit	The type of invoice to be created. A value of "`deposit`" denotes a Deposit Invoice, a value of "`cancel`" denotes a Cancellation Invoice, and a value of "`credit`" denotes a Credit Note. If omitted, a regular invoice will be created
notes	String	Additional notes to appear on the invoice
internal_notes	String	Additional internal notes associated with the invoice
term	Number	The number of days allowed for the client to pay from the date of invoice. If omitted, the global Default Tax Amount setting will be used
paid	Decimal	The amount paid against the invoice (without currency symbol - i.e. 10.00)
credit	Decimal	The amount of credit applied to the invoice (without currency symbol - i.e. 10.00)
refunded	Decimal	The amount refunded against the invoice (without currency symbol - i.e. 10.00)

Example Response

```
{
	"invoice_added":"MIDAS00075"
}
```

Variables Returned	
invoice_added	The Invoice number/reference of the newly created invoice

add_venue_ group

Adds a new venue group

Required Parameters	Possible Values	Description
group	*String*	Venue Group name

Optional Parameters	Possible Values	Description
venues	*<Venue IDs>*	Comma separated list of venue IDs to include in venue group

Example Response

```
{
      "venue_group_added":"4"
}
```

Variables Returned	
venue_group_added	A unique venue group ID corresponding the newly added venue group

add_resource

Adds a new resource to the system.

Required Parameters	Possible Values	Description
resource	String	The name of the new resource

Optional Parameters	Possible Values	Description
type	equipment | consumable | staffing | <custom category>	The type of resource to be added. If this parameter is omitted, the resource type will default to "`equipment`"
qty	Number	The available quantity of the new resource. If this parameter is omitted an "Unlimited" available quantity will be assumed
charge	Decimal	The charge (without currency symbol - i.e. 10.00) for the resource. This will be a per-hour charge where the type parameter is set to "`staffing`", otherwise this is a per-booking charge
tax	Percentage	The tax rate to be charged against the item to be added. If omitted, the global tax rate will be used
public	0 | 1	Sets whether the resource is publicly requestable (can appear on the Public Booking Request screen). If this parameter is omitted, the resource won't be publicly requestable
venues	Venue IDs	A comma-separated list of Venue ID's the resource will be available to. If this parameter is omitted, the resource will be available to all venues

Example Response

```
{
	"resource_added":"67"
}
```

Variables Returned	
resource_added	A unique resource ID corresponding the newly added resource

approve_request

Approves a booking request.

Required Parameters	Possible Values	Description
id	Booking ID	The numeric ID of the booking request to be approved

Optional Parameters	Possible Values	Description
silent	0 | 1	By default when a booking request is approved, the original requestor is sent an email notification informing them that their request has been approved. Setting "silent" to "1" will suppress the sending of this notification

Example Response

```
{
	"request_approved":"1"
}
```

Variables Returned	
request_approved	Returns "1" after a successful approval

authenticate_user

Authenticate against a MIDAS user account.

Note: "`authenticate_user`" calls respect the "Max Invalid Login Attempts" security setting.

Tip: For increased security, we strongly advise that "`authenticate_user`" API calls are only made using http POST methods (and not http GET methods)

Required Parameters	Possible Values	Description
email	Email Address	Email address of account to authenticate
pw	String	Corresponding password for account to authenticate

Optional Parameters	Possible Values	Description
bypass_fail_count	0 | 1	If set to 1, the API will bypass the previously failed login count checking & incrementing. If omitted, or explicitly set to 0, each failed authentication attempt via the API will count towards a failed login attempt against the user account. After too many failed login attempts, user accounts are automatically locked. This threshold may be set within the software via MIDAS Admin Options → Manage MIDAS → Security
include_locked	0 | 1	If set to 1 the API will be allowed to authenticate accounts even if user account is locked/suspended. If omitted, or explicitly set to 0, authentication will fail for user accounts which are currently locked/suspended

Example Response

```
{
	"authenticated":"1",
	"id":"4"
}
```

Variables Returned	
authenticated	Returns "1" if supplied email address and corresponding password are valid, otherwise returns "0"
id	If the returned "authenticated" response is "1", the "id" parameter contains the internal user ID of the authenticated account

del_booking

Deletes a single booking.

Required Parameters	Possible Values	Description
id	Booking ID	The unique numerical reference of the booking to be deleted (which can be determined from the "get_bookings" call)

Optional Parameters	Possible Values	Description
force	0 | 1	By default, when a booking is "deleted", it still persists in a "hidden" state (allowing it to be restored at a later stage). Passing the "force" parameter with a value of "1" will mean that the booking will be completely removed from the system
no_watches	0 | 1	Setting the "`no_watches`" parameter to "`1`" will supress generation of Watch notifications for users who are monitoring deleted bookings, otherwise omitting this parameter (or setting it to "`0`") will allow Watch notifications to be generated accordingly

Example Response

```
{
	"bookings_deleted":"1"
}
```

Variables Returned	
bookings_deleted	The number of bookings deleted

del_bookings

Deletes multiple bookings across a date range.

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Sets the start of the window in which bookings should be deleted
end	Valid date and time (Format: YYYYMMDDHHMM)	Sets the end of the window in which bookings should be deleted

Optional Parameters	Possible Values	Description
client	Client ID	Limits deleting bookings to a specific client
venue	*<Venue Name>*	Limits deleting of bookings to a specific venue
type	*<Booking Type>*	Limits deleting of bookings to a specific booking type

Any combination of the above optional parameters may be used to limit which bookings are deleted. For example, specifying both a client and a venue will only delete bookings which match BOTH criteria.

Optional Parameters	Possible Values	Description
force	0 | 1	By default, when a booking is "deleted", it still persists in a "hidden" state (allowing it to be restored at a later stage). Passing the "force" parameter with a value of "1" will mean that the booking will be completely removed from the system
no_watches	0 | 1	Setting the "no_watches" parameter to "1" will supress generation of Watch notifications for users who are monitoring deleted bookings, otherwise omitting this parameter (or setting it to "0") will allow Watch notifications to be generated accordingly

Example Response

```
{
      "bookings_deleted":"7"
}
```

Variables Returned

bookings_deleted	The number of bookings deleted

del_client

Deletes existing client records.

Required Parameters ***(at least 1 required)***	**Possible Values**	**Description**
id	Client ID	The unique numerical id of the client (which can be determined from the "get_client" call) Passing this parameter will override all other required parameters
client	String	Client name
org	String	Organization name
email	Email address	Email address
address	String	Postal/mailing address
phone	Number	Telephone number
fax	Number	Fax number
mobile	Number	Mobile/cell number
notes	String	Notes about the client

Optional Parameters	**Possible Values**	**Description**
retain_bookings	1 | 0	By default, when a client is deleted, all associated bookings for the client are also deleted. Passing the `retain_bookings` parameter with a value of "1" will mean that no bookings will be removed when a client is deleted

Example Response

```
{
      "clients_deleted":"1",
      "bookings_deleted":"24"
}
```

Variables Returned	
clients_deleted	The number of clients deleted
bookings_deleted	The number of bookings deleted

del_invoice

Deletes an existing invoice

Required Parameters	Possible Values	Description
invoice	Invoice ID	The number/reference of the invoice to delete

Example Response

```
{
      "invoice_deleted":"MIDAS00075"
}
```

Variables Returned	
invoice_deleted	The Invoice number/reference of the deleted invoice

del_resource

Deletes an existing resource, and removes it from any existing bookings

Required Parameters	Possible Values	Description
resource	Resource ID	The numeric ID of the existing resource to delete

Example Response

```
{
      "resource_deleted":"67",
      "bookings_affected":"4,8,15,16,23,42"
}
```

Variables Returned	
resource_deleted	The numeric ID of the deleted resource
bookings_affected	A comma-separated list of all the booking ID's from which the deleted resource has been removed

del_venue_ group

Deletes a venue group

Required Parameters *(only 1 required)*	**Possible Values**	**Description**
id	*<Venue Group ID>*	Venue Group ID
group	*String*	Venue Group name

Example Response

```
{
      "venue_group_deleted":"4"
}
```

Variables Returned	
venue_group_deleted	A unique venue group ID corresponding the deleted venue group

email_client

Send an email to an existing client.

Required Parameters	Possible Values	Description
client	Client ID	The numeric ID of the existing client to email
subject	String	The subject line of the email
body	Text	The main body of the email. HTML is allowed

Optional Parameters	Possible Values	Description
invoice	Invoice ID	The number/reference of an invoice to include within the email. Note: Only invoices for the specific client being emailed can be included
receipt	0 | 1	If a `"receipt"` parameter is included along with the `"invoice"` parameter, a receipt for specified invoice will be included
style	String | Text	This optional parameter can be used to include custom CSS styling rules within the email

Example Response

```
{
	"completed":"1"
}
```

Variables Returned

completed	Returns "1" once the API call has completed

gen_invoice

Automatically generate invoice(s).

To generate an invoice from an existing booking:

Required Parameters	Possible Values	Description
booking	Booking ID	The numeric ID of the existing booking for which to generate an invoice for

Example Response

```
{
      "invoice_created":"MIDAS00075"
}
```

Variables Returned	
invoice_created	The Invoice number/reference of the newly generated invoice

To generate a series of invoices across a date range:

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Sets the start of the window in which bookings should be included in generated invoices
end	Valid date and time (Format: YYYYMMDDHHMM)	Sets the end of the window in which bookings should be included in generated invoices

Optional Parameters	Possible Values	Description
client	Client ID	The numeric ID of the existing client who's bookings should be included on the generated invoice

Including the optional "client" parameter will limit invoice generation to a single invoice containing all bookings for the specified client which take place between the specified start/end values.

Omitting the optional "client" parameter will generate separate invoices for each client with bookings taking place between the specified start/end values.

Example Response

```
{
      "invoices_created":"MIDAS00076,MIDAS00077,MIDAS00078,MIDAS00079"
}
```

Variables Returned

invoices_created	A comma-separated list of all newly generated invoices

get_activity

Returns all MIDAS user activity between two times. The optional "`user`" parameter can be used to limit the activity returned to a specific user, and the optional "filter" parameter can be used to limit the results to a specific activity.

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Start time
end	Valid date and time (Format: YYYYMMDDHHMM)	End time

Optional Parameters	Possible Values	Description
user	*<User's Full Name>* | SYSTEM	Limit the returned activity to a specific user identified by *<User's Full Name>* or the keyword "`SYSTEM`". "`SYSTEM`" will return activity not assigned to a specific user (for example database backups, new booking requests received, etc). If this parameter is omitted, activity for all users (including "`SYSTEM`") will be returned
filter	*<activity code>*	Limit the returned activity to a specific action. See Appendix A for a list of activity codes

Example Response

```
[
      {
      "time":"5/4/2013 @ 07:53",
      "user":"SYSTEM",
      "action":"BKUP"
      },
      {
      "time":"5/4/2013 @ 07:26",
      "user":"Joe Bloggs",
      "action":"ADD",
      "data1":"12/5/2013 @ 22:00"
      }
]
```

Variables Returned	
time	Date/Time activity occurred
user	User who generated activity

action	Code of action performed. See Appendix A for a list of activity codes
data1	Returns additional data specific to action. See Appendix A for more information
data2	Returns additional data specific to action. See Appendix A for more information
data3	Returns additional data specific to action. See Appendix A for more information
data4	Returns additional data specific to action. See Appendix A for more information

get_availability

Checks whether a venue is available (free) for booking between two times

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Start time
end	Valid date and time (Format: YYYYMMDDHHMM)	End time
venue	*<Venue name>*	Venue

Optional Parameters	Possible Values	Description
attendees	*<number>*	The number of desired attendees. If specified, availability will also be dependent upon whether the number of attendees if less than the venue's capacity and/or maximum global occupancy levels.

Example Response

```
{
      "availability":"1"
}
```

Variables Returned	
availability	Current Venue Availability A value of "1" means the venue is "available" on the dates/times specified A value of "*Unavailable*" means the venue is not available on the dates/times specified (i.e. clashes with an existing booking) Any other values also mean "Unavailable" but will specify the reason for the unavailability i.e. "*This will exceed your maximum occupancy level*", or "*Venue unavailable from 6/5/2103*"

get_booking

Returns details of a specific booking.

Required Parameters	Possible Values	Description
id	Booking ID	The unique numerical reference of the booking

Example Response

```
{
        "id":"222",
        "start":"5/4/2013 @ 13:00",
        "end":"5/4/2013 @ 14:00",
        "venue":"Room 3",
        "client":"Some User (Some Org)",
        "type":"Community",
        "attendees":"30",
        "notes":"",
        "resources":[
                {
                "resource":"Laptop",
                "qty":"2"
                },
                {
                "resource":"Projector",
                "qty":"1"
                }
                ],
        "invoice":"MIDAS00001",
        "history":[
                {
                "action":"Added",
                "date":"2/4/2013 @ 08:23",
                "user":"Joe Bloggs"
                },
                {
                "action":"Modified",
                "date":"2/4/2013 @ 16:52",
                "user":"Joe Bloggs"
                }
                ],
        "status":"confirmed"
        }
}
```

Variables Returned	
id	Internal Booking ID
start	Start date/time of booking
end	End date/time of booking

venue	Venue
client	Client the booking is for
type	Booking Type
attendees	Estimated number of people attending booking
notes	Booking Notes
resources	Resources assigned to booking
resource	The name of the resource
qty	The quantity of the resource
invoice	Invoice in which the booking appears
history	Booking history
action	The action performed. This will be one of the following: "Booking Request Received", "Added", "Modified", "Deleted", "Restored", or "Invoice Created"
date	The date/time the action occurred
user	The user who performed the action
custom fields	Any custom booking fields setup in your MIDAS will also be returned
status	Denotes whether the booking is a "request", a "confirmed" booking, or whether it has been "deleted"

get_booking_type

Returns details of a specific booking type

Required Parameters	Possible Values	Description
id	Booking Type Name or ID	The name or numeric ID of the booking type

Example Response

```
[
      {
      "id":"1",
      "name":"Community",
      "color":"#FF0000",
      "discount":"",
      "tentative":1,
      "tentative_expire":"",
      "public":1
      }
]
```

Variables Returned	
id	The internal ID of the booking type
name	The name of the booking type
color	The hexadecimal color associated with the booking type
discount	The absolute numerical or percentage (%) discount associated with the booking type
tentative	Returns a `1` if the booking type has been set as tentative and set to expire a number of minutes after bookings of this type are added. Returns a `2` if the booking type has been set as tentative and set to expire a number of minutes before bookings of this type commence. Returns a `0` if the booking type is not currently set as being tentative.
tentative_expire	If the booking type is "tentative", this parameter denotes the expiry time in minutes
public	Returns a `1` if the booking type has been set as public, otherwise returns a `0`

get_booking_types

Returns all currently defined booking types and their corresponding internal IDs

Required Parameters
This API command has no additional required parameters

Example Response

```
[
	{
	"id":"1",
	"name":"Community"
	},
	{
	"id":"2",
	"name":"Internal"
	},
	{
	"id":"3",
	"name":"Maintenance"
	}
]
```

Variables Returned	
id	The internal ID of the booking type
name	The name of the booking type

get_bookings

Returns all bookings between two dates/times

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Sets the start of the window for which bookings should be returned
end	Valid date and time (Format: YYYYMMDDHHMM)	Sets the end of the window for which bookings should be returned

Optional Parameters	Possible Values	Description
client	*<Client ID>*	Limits returned bookings to a specific client, based on a Client ID (which may first be obtained via a "get_client" call)
filter_field	attendees | invoice | notes | *<Custom Booking Field>*	The internal name of a custom booking field against which to filter bookings based upon its corresponding "`filter_string`" value (see below). For example, if you have a custom booking field named "My Field", pass the value "`my_field`". This parameter will also accept values of "`attendees`", "`invoice`", or "`notes`" to match bookings against attendee numbers, invoice numbers, and booking notes
filter_string	*String*	Used in conjunction with "`filter_field`" above to return bookings which match a specified value of a custom booking field.
filter_type	contains | not_contains | begins_with | ends_with | greater_than | less_than | not_equals	Used in conjunction with "`filter_string`" above to control how matching bookings are determined. For example, if "`filter_field`" was set to "`my_field`", "`field_string`" was set to "`room`", and "`filter_type`" was set to "`begins_with`", bookings will be returned where the value of the custom "My Field" booking field stars with the phrase "room" - i.e. "room 1", "room 2", etc would be matched, whereas "meeting room" wouldn't be matched as it doesn't start with the phrase "room". "`greater_than`" and "`less_than`" values relate

		to numeric booking fields. If the "`filter_type`" parameter is omitted, an exact match with the "`filter_string`" will be enforced.
filter_casesensitive	0 | 1	Controls the case sensitivity of the "`filter_string`" parameter above. If set to "`1`" booking will only be returned which exact match the "`filter_string`". For example, if "`filter_string`" contained "room" and "`filter_casesensitivity`" was set to "`1`", only bookings matching "room" (and not "Room") would be returned. If the "`filter_casesensitive`" parameter is omitted, or explicitly set to "`0`", the case of the "`filter_string`" value will be ignored.
modified_before	*Valid date and time (Format: YYYYMMDDHHMM)*	Limits returned bookings to those which were last modified before a specified date/time
modified_since	*Valid date and time (Format: YYYYMMDDHHMM)*	Limits returned bookings to those which were last modified after a specified date/time
venue	*<Venue name>* | *<Venue ID>*	Limits returned bookings to a specific venue name, venue ID, or comma separated list of venue IDs
type	*<Booking Type>*	Limits returned bookings to a specific booking type
status	all | request | deleted	Controls which bookings are returned. "`request`" will only return booking requests. "`deleted`" will only return deleted bookings. "`all`" will return all bookings (including deleted bookings and booking requests). If this parameter is omitted only confirmed bookings will be returned

Example Response

```
[
        {
        "id":"222",
        "start":"5/4/2013 @ 13:00",
        "end":"5/4/2013 @ 14:00",
        "venue_id":"3",
        "venue":"Room 3",
        "client":"Some User (Some Org)",
        "type":"Community",
        "attendees":"30",
        "notes":"",
        "resources":[
           {
           "resource":"Laptop",
           "qty":"2"
           },
           {
           "resource":"Projector",
           "qty":"1"
           }
           ],
      "invoice":"MIDAS00001",
      "history":[
           {
           "action":"Added",
           "date":"2/4/2013 @ 08:23",
           "user":"Joe Bloggs"
           },
           {
           "action":"Modified",
           "date":"2/4/2013 @ 16:52",
           "user":"Joe Bloggs"
           }
           ]
      },
      {
      "id":"223",
      "start":"5/4/2013 @ 14:15",
      "end":"5/4/2013 @ 16:30",
      "venue_id":"4",
      "venue":"Room 4",
      "client":"Some User (Some Org)",
      "type":"Community",
      "attendees":"40",
      "notes":"",
      "resources":[],
      "invoice":"MIDAS00001",
      "history":[
           {
           "action":"Added",
           "date":"2/4/2013 @ 08:25",
           "user":"Joe Bloggs"
           }
           ]
      }
]
```

Variables Returned	
id	Internal Booking ID
start	Start date/time of booking
end	End date/time of booking
venue_id	Internal ID of the venue
venue	Name of the venue
client	Client the booking is for

type	Booking Type
attendees	Estimated number of people attending booking
notes	Booking Notes
resources	Resources assigned to booking
resource	The name of the resource
qty	The quantity of the resource
invoice	Invoice in which the booking appears
history	Booking history
action	The action performed. This will be one of the following: "Booking Request Received", "Added", "Modified", "Deleted", "Restored", or "Invoice Created"
date	The date/time the action occurred
user	The user who performed the action
custom fields	Any custom booking fields setup in your MIDAS will also be returned
status	Denotes whether the booking is a "request", a "confirmed" booking, or whether it has been "deleted"

get_client

Returns client records

Required Parameters *(at least 1 required)*	**Possible Values**	**Description**
client	*<Client name>*	Client name
org	*<Organization name>*	Organization name
email	*<email address>*	Email address

Optional Parameters	**Possible Values**	**Description**
match	*exact* | *loose*	Allows specifying the closeness of the match. If match is set "`exact`" and "`client`" is set to "`Joe Bloggs`", only clients with the name "Joe Bloggs" will be returned. If match is set to "`loose`" and "`email`" is set to "`@mid.as`", any client with an "@mid.as" email address will be returned. If the match parameter is omitted only exact matches are returned.

Example Response

```
[
     {
     "id":"23",
     "name":"Joe Bloggs",
     "organization":"MIDAS",
     "email":"joe@mid.as",
     "address":"PO Box 224,Cheadle,Cheshire.SK8 4WB",
     "phone":"01234 567 890",
     "fax":"01234 567 891",
     "mobile":"07123456789",
     "notes":"",
     "added":"7/8/2012 @ 16:49"
     }
]
```

Variables Returned	
id	Unique client identification number
name	Client’s name
organization	Client’s organization
email	Client’s email address

address	Client’s postal address
phone	Client’s telephone number
fax	Client’s fax number
mobile	Client’s mobile (cell) number
notes	Notes about the client
added	Date/time when client was added to MIDAS

get_consumable_levels

Returns the current stock level of all consumable items

Required Parameters

This API command has no additional required parameters

Example Response

```
[
      {
      "consumable":"Flipchart Paper",
      "qty_remaining":"23",
      },
      {
      "consumable":"Permanent Marker Pens",
      "qty_remaining":"156",
      }
]
```

Variables Returned

consumable	The name of the consumable
qty_remaining	The current stock level. If no stock quantity has been defined, `qty_remaining` will return "`Unlimited`"

get_invoice

Retrieves a specific invoice

Required Parameters	Possible Values	Description
invoice	*<Invoice Reference>*	Invoice Reference

Example Response

```
{
      "client":"MIDAS (MIDAS2)",
      "date":"5/11/2012 @ 10:00",
      "items":[
            {
            "description":"Room 1 (6/11/2012 @ 19:30 - 21:30)",
            "qty":"2.00",
            "rate":"23.50",
            "tax_rate":"20.00",
            "tax_amount":"4.70",
            {
            "description":"Room 2 (13/11/2012 @ 19:30 - 21:30)",
            "qty":"2.00",
            "rate":"23.50",
            "tax_rate":"20.00",
            "tax_amount":"4.70",
            }
            ],
      "total":"188.00",
      "paid":"0.00",
      "history":[
            {
            "action":"Created",
            "date":"5/11/2012 @ 08:45",
            "user":"MIDAS Administrator"
            },
            {
            "action":"Modified",
            "date":"5/11/2012 @ 09:42",
            "user":"MIDAS Administrator"
            },
            {
            "action":"Emailed",
            "date":"5/11/2012 @ 10:00",
            "user":"MIDAS Administrator"
            }
            ]
}
```

Variables Returned	
client	Client
date	Invoice Date (or "Invoice Not Sent if invoice hasn’t been emailed/printed)
items	Invoice items
description	The item’s description
qty	The item’s quantity
rate	The item’s rate (charge)
tax_rate	The percentage tax rate applied to the item
tax_amount	The amount of tax to be applied
total	Invoice total (ex tax)
paid	Amount paid
history	Invoice history
action	The action performed. This will be one of the following: "Created", "Modified", "Printed", "Emailed", "Payment Received", "Payment Overdue", "Paid In Full"
date	The date/time the action occurred
user	The user who performed the action

get_invoices

Retrieves a list of invoices for a specific client

Required Parameters *(at least 1 required)*	**Possible Values**	**Description**
client	*<Client name>*	Client name
org	*<Organization name>*	Organization name
email	*<email address>*	Email address

Optional Parameters	**Possible Values**	**Description**
type	deposit | cancel | invoice | credit	The type of invoices to be returned. A value of "`deposit`" returns Deposit Invoices, a value of "`cancel`" returns Cancellation Invoices, and a value of "`credit`" returns Credit Notes. If omitted, (or set to "invoice") regular invoices will be returned
unsent	0 | 1	If "`1`", only unsent invoices will be returned

Example Response

```
{
      "client":"Joe Bloggs (Bloggs Inc)",
      "email":"joe@bloggsinc.com",
      "invoices":"MIDAS0001,MIDAS0003,MIDAS0004"
}
```

Variables Returned	
client	Client/Organization
email	Client’s email address
invoices	A comma separated list of all invoices associated with the client

get_messages

Retrieves all current internal messages for a specific user

Required Parameters *(only 1 required)*	**Possible Values**	**Description**
email	*<user's email>*	User's email address
user	*<user's name>*	User's name

Example Response

```
[
      {
      "message":"Hello World!",
      "author":"Joe Bloggs",
      "created","24/4/2015 @ 00:03",
      "expires","26/4/2015 @ 00:00"
      },
      {
      "message":"This is a test message",
      "author":"Jane Doe",
      "created","25/4/2015 @ 03:42",
      "expires","29/4/2015 @ 12:30"
      }
]
```

Variables Returned	
message	Details of the watch
author	The user who created the message
created	Date/Time the message was created
expires	Date/Time at which the message expires

get_reminders

Retrieves all current reminders for a specific user

Required Parameters *(only 1 required)*	**Possible Values**	**Description**
email	*<user's email>*	User's email address
user	*<user's name>*	User's name

Example Response

```
[
      {
      "reminder":"Reminder: Room 3: 25/4/2015 @ 13:00 - 15:00 Booking for Joe
Bloggs",
      "expires":"25/4/2015 @ 15:00"
      },
      {
      "reminder":"Reminder: Room 3: 26/4/2015 @ 14:00 - 16:30 Booking for Jane
Doe",
      "expires":"26/4/2015 @ 16:30"
      }
]
```

Variables Returned	
reminder	Details of the reminder
expires	Date/Time at which the reminder expires

get_resource

Returns resource information

Required Parameters	Possible Values	Description
resource	*<Resource name>*	Resource name

Example Response

```
[
      {
      "resource":"Laptop",
      "category":"E",
      "qty":"3",
      "charge":"120.00",
      "tax":"20.00",
      "public":"1",
      "limited_to":"Room 1,Room 2"
      }
]
```

Variables Returned	
resource	Resource name
category	The category the resource item falls under. Possible values are: "E" = Equipment "C" = Consumable "S" = Staffing
qty	The total quantity of the resource added to MIDAS
charge	The charge for a single quantity of the resource item
tax	The percentage tax rate for the resource item
public	Whether the resource is available for public requesting/booking
limited_to	If the resource has been restricted to only be available to certain venues, the "limited_to" value contains a comma separated list of these venues

get_resource_availability

Returns the quantity available of a specified resource between two dates/times

Required Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Start time
end	Valid date and time (Format: YYYYMMDDHHMM)	End time
resource	*<Resource name>*	Resource name
qty	*Number*	Quantity to check

Example Response

```
{
     "available":"2"
}
```

Variables Returned	
available	Quantity of specified resources available during times specified

get_setting

Returns a current MIDAS setting

Required Parameters	Possible Values	Description
setting	*<setting name>*	Setting name. See Appendix B for available setting names

Example Response

```
{
      "email_sendfrom":"midas@example.com"
}
```

Variables Returned	
<setting>	The current value of the specified <setting>

get_template

Returns the contents of a template

Required Parameters	Possible Values	Description
template name	*<template name>*	Template name. See below for available template names

Template Name	Description	Additional
email	Email (General)	
email_book_cancel	Email (Booking Cancelled)	Email subject
email_book_confirm	Email (Booking Confirmation)	Email subject
email_book_remind	Email (Booking Reminder)	Email subject
email_req_approved	Email (Booking Request Approved)	Email subject
email_req_approved_m	Email (Booking Request Approved – with changes)	Email subject
email_req_rejected	Email (Booking Request Rejected/Declined)	Email subject
email_req_submitted	Email (Booking Request Submitted)	Email subject
email_inv_overdue	Email (Invoice Overdue)	Email subject
email_inv_remind	Email (Invoice Reminder)	Email subject
invoice	Invoice (Regular)	
invoice_cancellation	Invoice (Cancellation)	
payonline	Online Payments	URL of external .css file
print	Print	
receipt	Invoice Receipt	
webrequest	Public Booking Requests	URL of external .css file
intro	Welcome Note (on login screen)	

Example Response

```
{
        "email":"<table style=\"width:100%\"><tr>
        <td>%LOGO%</td><td><b>%DATABASE%</b></td><td>%MIDASLOGO%</td></tr>
        </table>%CONTENT%"
}
```

Variables Returned	
<template>	The current contents of the specified <template>
additional	For templates which contain an additional value in addition to the main template content (such as email templates with an additional subject line), this value is returned in the "additional" parameter

get_user

Returns user information

Required Parameters *(at least 1 required)*	**Possible Values**	**Description**
user	*<name>*	User’s full name
email	*<email address>*	User’s email address

Optional Parameters	**Possible Values**	**Description**
match	*exact* | *loose*	Allows specifying the closeness of the match. If match is set "`exact`" and "`user`" is set to "`Joe Bloggs`", only users with the name "Joe Bloggs" will be returned. If match is set to "`loose`" and "`user`" is set to "`Joe`", any user named "Joe" will be returned. If the match parameter is omitted only exact matches are returned

Example Response

```
{
      {
      "name":"Joe Bloggs",
      "email":"joe@bloggs.com",
      "added":"25/2/2013 @ 16:49",
      "last_login":"23/4/2013 @ 20:51",
      "last_password_change":"29/3/2013 @ 18:34",
      "last_modified":"25/2/2013 @ 23:42",
      "account_locked":"0",
      "failed_login_attempts":"0",
      "day_starts_at":"00",
      "day_runs_for":"24",
      "do_not_log_activity":"0"
      "email_calendar_events":"1",
      "email_messages":"0",
      "email_pending_notifications":"1",
      "email_watch_notifications":"0",
      "force_pw_change_at_login":"0",
      "suppress_messages_popup":"0",
      "venue_group_access":"Group A,Group B",
      "permissions":[
            {
            "can_add_bookings":"1",
            "can_add_clients":"1",
            "can_add_day_notes":"1",
            "can_add_historical_bookings":"0",
            "can_add_out_of_hours_bookings":"0",
            "can_change_password":"1",
            "can_delete_bookings":"2",
            "can_delete_clients":"1",
            "can_email_clients":"1",
            "can_evac":"1",
            "can_invoice":"1",
            "can_manage_addons":"1",
            "can_manage_booking_types":"1",
            "can_manage_midas":"1",
            "can_manage_resources":"1",
            "can_manage_users":"1",
            "can_manage_venues":"1",
            "can_modify_bookings":"2",
            "can_modify_clients":"0",
            "can_print":"1",
            "can_process_requests":"1",
            "can_restore_bookings":"1",
            "can_use_mymessages":"1",
            "can_view_activity_log":"1",
            "can_view_clients":"2",
            "can_view_statistics":"1",
            "max_booking_length":"120",
            "max_bookings":"4",
            "max_bookings_per":"D"
            }
            ]
      }
}
```

Variables Returned	
name	User's name
organization	User's email address
added	Date/Time user was added to MIDAS
last_login	Date/Time user last successfully logged in
last_password_change	Date/Time user last changed their password
last_modified	Date/Time user information was last modified
account_locked	Indicates if the user has been suspended / locked out of MIDAS
failed_login_attempts	The number of unsuccessful login attempt on this account since last successful login
day_starts_at	The hour (in 24-hour clock mode) the user's booking grid is displayed from
day_runs_for	The number of hours the user's booking grid displays
do_not_log_activity	If "0" all user activity within MIDAS will be recorded in the Recent Activity Log If "1" user activity within MIDAS will appear in the Recent Activity Log, except for logins/logouts

	If "2" no user activity will be recorded in the Recent Activity Log
email_calendar_events	Indicates whether reminders should also be sent to user's email as calendar events
email_messages	Indicates whether messages are forwarded to user's email
email_pending_notifications	Indicates whether new booking request notifications are forwarded to user's email
email_watch_notifications	Indicates whether watch notifications are forwarded to user's email
force_pw_change_at_login	Indicates whether user is required to change their password upon next login
suppress_messages_popup	Indicates whether the "My Messages" pop-up is suppressed (not shown) after login
venue_group_access	Indicates which Venue Groups the user has access to in the Booking Grid. A value of "*" indicates user can view all Venue Groups
permissions	The various permissions associated with the user account. Most take a value of either "1" meaning user has been granted a permission, "0" meaning the user does not have a permission. Permissions with additional possible values are indicated.
can_add_bookings	User can add bookings. "0" indicates user may not make bookings "1" indicates user may make bookings "2" indicates user may only make booking requests
can_add_clients	User can add clients
can_add_day_notes	Use can add notes to calendar dates
can_add_historical_bookings	User can add bookings for dates occurring in the past
can_add_out_of_hours_bookings	User can add bookings outside of a venue's operating hours
can_change_password	User can change their password
can_delete_bookings	User can delete bookings "0" indicates user cannot delete any bookings "1" indicates user may delete only those bookings originally added by them "2" indicates user may delete any booking
can_delete_clients	User can delete clients
can_email_clients	User can email clients directly from MIDAS
can_evac	User can print Emergency Evacuation data
can_invoice	User can use invoicing

can_manage_addons	User can manage MIDAS addon settings (For more information on available addons for MIDAS, please see https://mid.as/addons)
can_manage_booking_types	User can manage Booking Types
can_manage_midas	User can manage MIDAS
can_manage_resources	User can manage Resources
can_manage_users	User can manage Users & Permissions
can_manage_venues	User can manage Venues
can_modify_bookings	User can modify bookings "0" indicates user cannot modify any bookings "1" indicates user may modify only those bookings originally added by them "2" indicates user may modify any booking
can_modify_clients	User can modify clients
can_print	User can use print functions
can_process_requests	User can approve/reject pending booking requests
can_restore_bookings	User can restore previously deleted bookings
can_use_mymessages	Use can use My Messages (Messages, Reminders, and Watches)
can_view_activity_log	Use can access the Recent Activity log
can_view_clients	User can view client information "0" indicates user cannot view any client data "1" indicates user can view client and organization names only "2" indicates user can view full client info for any client "3" indicates user can only view full client into for clients they've added bookings for
can_view_statistics	Use can access Statistics
max_booking_length	The maximum duration (in minutes) that the user is permitted to make any single booking for
max_bookings	The maximum number of bookings the user may add to any given period (determined by "max_bookings_per" value)
max_bookings_per	The period over which the "max_bookings" value applies. Possible values are: "D" – Single date "W" – Week "M" – Month "Y" – Year

get_users_logged_in

Returns a list of currently logged in users

Required Parameters
This API command has no additional required parameters

Example Response

```
[
	{
	"name":"Joe Bloggs",
	"sessions":"2",
	"browser":"Edge 89.0.774.68 (Windows 10 x64),Chrome 89.0.4389.90 
(Windows 10 x64)"
	},
	"name":"MIDAS Administrator",
	"sessions":"1",
	"browser":"Chrome 89.0.4389.114 (Windows 10 x64)"
	}
]
```

Variables Returned	
name	The user’s name
sessions	The number of currently active sessions. A number greater than 1 indicates the user account is currently logged in on multiple browsers or devices.
browser	The user's browser. If the user is logged in from multiple browsers, each browser will be shown in a comma separated list.

get_venue

Returns venue information

Required Parameters	Possible Values	Description
venue	*<Venue name>* | *<Venue ID>*	Venue name or internal venue ID

Optional Parameters	Possible Values	Description
match	*exact* | *loose*	Allows specifying the closeness of the match. If match is set "`exact`" and "`venue`" is set to "`Meeting Room`", only the venue with the name "Meeting Room" will be returned. If match is set to "`loose`" and "`venue`" is set to "`Meeting Room`", any venue containing the name "Meeting Room", such as "Meeting Room A", "Meeting Room B", etc will be returned. If the match parameter is omitted only exact matches are returned

Example Response

```
[
        {
        "id":"1",
        "venue":"Room 1",
        "capacity":"20",
        "description":"A small meeting room",
        "public":"1",
        "rates":[
                {
                "Mon":[
                        {
                        "rate":"100.00",
                        "rate_length":"1",
                        "rate_type":"Hourly"
                        },
                        {
                        "rate":"80.00",
                        "rate_length":"4",
                        "rate_type":"Hourly"
                        },
                        {
                        "rate":"450.00",
                        "rate_length":"6",
                        "rate_type":"Booking"
                        }
```

```
                  ],
            "Tue":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ],
            "Wed":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ],
            "Thu":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ],
            "Fri":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ],
            "Sat":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ],
            "Sun":[
                  {
                  "rate":"20.00",
                  "rate_type":"Hourly"
                  }
                  ]
            }
            ],
      "tax":"20.00",
      "operating_hours":[
            {
            "day":"Mon",
            "closed":"0000-0800,2100-2400",
            },
            {
            "day":"Tue",
            "closed":"0000-0800,2100-2400",
            },
            {
            "day":"Wed",
            "closed":"0000-0800,2100-2400",
            },
            {
```

```
            "day":"Thu",
            "closed":"0000-0800,2100-2400",
            },
            {
            "day":"Fri",
            "closed":"0000-0800,2100-2400",
            },
            {
            "day":"Sat",
            "closed":"0000-0930,2000-2400",
            },
            {
            "day":"Sun",
            "closed":"0000-1000,1600-2400"
            }
            ],
      "closed_from":"",
      "closed_until":"1/4/2015",
      "managers":"Jane Doe,Joe Bloggs",
      "member_of":"Group A,Group B",
      "blocks":"Room 4",
      "alternatives":"Room 2,Room 3"
      }
]
```

Variables Returned	
id	Internal venue ID
venue	Venue name
capacity	The maximum occupancy of the venue
description	The venue’s description
public	Whether the venue is available for public requesting/booking
rates	The room rate/hire charge for each day of the week
<day>	Can be one of the following: "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun" to denote subsequent rate information for each day of the week
rate	The room/rate hire charge amount
rate_length	The minimum length of the booking the rate applies to (in hours)
rate_type	If "Hourly", the rate is per-hour, if "Daily", the rate is per-day, if "Booking", the rate is per-booking
tax	The percentage tax rate to charge on venue rates
operating_hours	The times between which the venue is closed during each day of the week
day	The day of the week
closed	A comma separated list of the times (24 hour format) the venue is closed
closed_from	The date after which the venue is considered closed
closed_until	The date until which the venue is considered closed

managers	A comma separated list of user who are "Managers" of the venue i.e. can approve Booking Requests for the venue
member_of	A comma separated list of Venue Groups to which the venue belongs to
blocks	A comma separated list of venues which are automatically blocked (Unavailable) when this venue has an ongoing booking
alternatives	A comma separated list of venues which should be considered as alternatives to this venue

get_venue_groups

Returns a complete set of venue group names together with a list of IDs of venues assigned to each group

Example Response

```
[
	{
	"group":"First Floor",
	"venues":"1,2,3,4,5"
	},
	{
	"group":"Second Floor",
	"venues":"6,7,8,9,10"
	},
	{
	"group":"Basement",
	"venues":"11,12"
	}
]
```

Variables Returned	
group	Name of the venue group
venues	Comma separated list of internal IDs of venues in the venue group

get_venues

Returns a complete set of venue names together with their corresponding internal venue IDs

Required Parameters

This API command has no additional required parameters

Example Response

```
[
      {
      "id":"21",
      "venue":"Room 1"
      },
      {
      "id":"2",
      "venue":"Room 2"
      },
      {
      "id":"3",
      "venue":"Room 3"
      }
]
```

Variables Returned	
Id	Internal venue ID (this ID remains fixed, even if the venue is subsequently renamed)
venue	Venue name

get_venues_in_group

Returns a list of venues in a venue group

Required Parameters	Possible Values	Description
group	*<venue group>*	Name of a venue group

Example Response

```
{
      "Group A":"Room 1,Room 2,Room3"
}
```

Variables Returned	
<group name>	Comma separated list of current venues in group

get_venues_in_use

Returns a list of all venues in use at the point in time when the API call is made. If optional start and end parameters are supplied, returns a list of all venues in use between start and end parameters.

Required Parameters
This API command has no additional required parameters

Optional Parameters	Possible Values	Description
start	Valid date and time (Format: YYYYMMDDHHMM)	Start time
end	Valid date and time (Format: YYYYMMDDHHMM)	End time

Example Response

```
{
	"venues":"Room 1,Room 2,Room3"
}
```

Variables Returned	
venues	Comma separated list of venues in use

get_watches

Retrieves all current watches for a specific user

Required Parameters *(only 1 required)*	Possible Values	Description
email	*<user’s email>*	User’s email address
user	*<user’s name>*	User’s name

Example Response

```
[
      {
      "watch":"Booking Matching [Venue: Room 1] on 25/4/2015 @ 13:00 - 15:00
For Joe Bloggs Added",
      "expires":"25/4/2015 @ 15:00"
      },
      {
      "watch":"Booking Matching [Venue: Room 4] on 25/4/2015 @ 14:00 - 16:30
For Jane Doe Deleted",
      "expires":"26/4/2015 @ 16:30"
      }
]
```

Variables Returned	
watch	Details of the watch
expires	Date/Time at which the watch expires

mod_booking

Modifies an existing booking.

Required Parameters	Possible Values	Description
id	Booking ID	The unique numeric ID of the booking to be modified

In addition, one or more of the parameters from the following table must also be included with the call. Only values for the parameters you specify will be changed. For example, to only change a booking type, whilst retaining the booking's original start, end, and venue details, only pass the "`new_type`" parameter. To change the venue and the type and the venue, pass both "`new_type`" and "`new_venue`".

Optional Parameters (at least 1 required)	Possible Values	Description
new_start	Valid date and time (Format: YYYYMMDDHHMM)	The new start date/time the booking should be changed to
new_end	Valid date and time (Format: YYYYMMDDHHMM)	The new end date/time the booking should be changed to
new_venue	Venue ID | Venue Name	The numeric ID or name of the venue the booking should be changed to
new_client	Client ID	The numeric ID of the client the booking should be changed to
new_type	Booking Type ID | Booking Type Name	The numeric ID of the Booking Type, or the name of the Booking Type the booking should be changed to
new_attendees	Number	The new number of booking attendees
new_notes	String	New notes about the booking
new_resources	Quantity & Resource ID |Resource Name	New resources to be added to the booking. Values should be passed as "`qty|resource` (ID or name)". For example, to add 2 Tables to a booking, pass "`2|Tables`". This parameter can be added multiple times to add further resources Note: Passing the "`new_resources`" parameter will remove any previous resources that had been assigned to the booking

<new custom fields>	Custom Value	If you've defined custom booking fields in your MIDAS, you can optionally specify new values for these fields for the booking you're modifying. For example, if you have a custom field named "Insurance", you would pass the parameter "`new_insurance`" together with the new value you wish to set

Optional Parameters	Possible Values	Description
accept_limited_resources	0 | 1	If you're specifying new resources for a booking, this parameter determines whether the bookings should still be modified if the quantity of resources requested isn't available. For example, if you add 10 Tables to a booking, yet only 5 Tables are available, an error will be returned and the booking won't be modified. However, passing the "`accept_limited_resources`" parameter with a value of "`1`" will continue the modify booking, but with reduced resource quantities (in the previous example, this would mean that only 5 Tables would be added to the booking, instead of the desired 10)
no_watches	0 | 1	Setting the "`no_watches`" parameter to "`1`" will suppress generation of Watch notifications for users who are monitoring modified bookings, otherwise omitting this parameter (or setting it to "`0`") will allow Watch notifications to be generated accordingly

Example Response

```
{
      "booking_modified":"1"
}
```

Variables Returned (on success)	
booking_modified	Returns "1" if the booking was successfully modified

Variables Returned (on error)	

error	The reason why the booking couldn't be modified. Typical examples include:
	`Unavailable` *(clashes with an existing booking)*
	`Unavailable - Blocked By Meeting Room 1`
	`45 Attendees will exceed the maximum capacity of Meeting Room 1 (15)`
	`Outside Operating Hours`
	`Venue unavailable from 1/1/2015`
	`Venue unavailable between 1/1/2015-1/2/2015`
	`Venue not available until 1/1/2015`
	`Only x5 Tables Available`

mod_client

Modifies an existing client record.

Two sets of parameters are required (outlined in the tables below) - the first set controls which client(s) to modify, the second set specifies which parameters of the selected client(s) to modify.

Controlling which client(s) to modify:

Required Parameters (at least 1 required)	**Possible Values**	**Description**
id	Client ID	The unique numerical id of the client (which can be determined from the "get_client" call) Passing this parameter will override all other required parameters in this table
client	String	The name of the existing client to be modified
org	String	Organization name
email	Email address	Email address
address	String	Postal/mailing address
phone	Number	Telephone number
fax	Number	Fax number
mobile	Number	Mobile/cell number
notes	String	Notes about the client

To ensure that only a single client is modified, pass only the "`id`" parameter, or alternatively as many of the other above parameters as possible to ensure a specific individual client is modified, rather than a group of clients. For example, if you only pass the "`org`" parameter alone ALL clients in the database that match the organization name you supply will be modified. Whereas passing both the client name and the organization name will ensure that only client records that match BOTH will be modified. Passing "`client_id`" alone will ensure that only one client will be modified.

Setting new values:

Required Parameters (at least 1 required)	**Possible Values**	**Description**
new_client	String	The new Client name
new_org	String	The new Organization name
new_email	Email address	The new Email address
new_address	String	The new postal/mailing address

new_phone	Number	The new telephone number
new_fax	Number	The new fax number
new_mobile	Number	The new mobile/cell number
new_notes	String	New notes about the client
new_tax_exempt	0 | 1	Specifies whether the client is exempt from being charged tax on their invoices

Omitting any of the above parameters will leave its current value unchanged. Passing any of the above parameters with a "blank" value will clear/delete the current value of the corresponding field.

Example Response

```
{
	"clients_modified":"1"
}
```

Variables Returned	
clients_modifed	The number of clients modified

mod_invoice

Modifies/Updates the status of an existing invoice.

Required Parameters	Possible Values	Description
invoice	Invoice ID	The number/reference of the invoice to update

Required Parameters (at least 1 required)	**Possible Values**	**Description**
client	Client ID	The numeric ID of the existing client to change the invoice for. Note: Only unsent invoices can have their client changed
notes	String	Additional notes to appear on the invoice
internal_notes	String	Additional internal notes associated with the invoice
paid	Decimal	The amount paid against the invoice (without currency symbol - i.e. 10.00)
credit	Decimal	The amount of credit applied to the invoice (without currency symbol - i.e. 10.00)
refunded	Decimal	The amount refunded against the invoice (without currency symbol - i.e. 10.00)

Example Response

```
{
      "invoice_updated":"MIDAS00075"
}
```

Variables Returned	
invoice_updated	The Invoice number/reference of the updated invoice

mod_resource

Modifies an existing resource.

Required Parameters	Possible Values	Description
resource	Resource ID	The numerical ID of the existing resource to modify

Optional Parameters (at least 1 required)	Possible Values	Description
new_name	String	The new name for the existing resource
new_type	equipment | consumable | staffing | <custom category>	The new type (category) of the resource
new_qty	Number	The new available quantity of the resource. Passing this parameter with a blank value will reset the available quantity to "Unlimited"
new_charge	Decimal	The new charge (without currency symbol - i.e. 10.00) for the resource. This will be a per-hour charge where the resource is of "staffing" type/category, otherwise this is a per-booking charge
new_public	0 | 1	Sets whether the resource is publicly requestable (can appear on the Public Booking Request screen)
new_venues	Venue IDs	A comma-separated list of Venue ID's the resource is available to. Passing this parameter with a blank value will reset the resource to be available to all venues

Omitting any of the above parameters will leave its current value unchanged. Passing any of the above parameters with a "blank" value will clear/delete the current value of the corresponding field.

Example Response

```
{
	"resource_modified":"67"
}
```

Variables Returned	
resource_modified	The unique resource ID corresponding the modified resource

mod_venue

Modifies an existing venue.

Required Parameters	Possible Values	Description
venue	Venue ID / Name	The name or numerical ID of the existing venue to modify

Optional Parameters (at least 1 required)	Possible Values	Description
new_name	String	The new name for the existing venue
new_description	String	The new description for the existing venue (may include HTML)
new_capacity	Integer	The new capacity (maximum occupancy) of the venue
new_closed_all	String	Sets the periods during all days of the week during which the venue is "closed". Values are all in 24 hour clock mode, in the format HHMM. To "close" the venue between 11am and 2.30pm, set "`new_closed_all`" to "`11001430`". To set multiple closed periods, add these on to the string. For example, to close a venue between 11am and 2.30pm and 5pm - 6pm, set "`1100143017001800`".
new_closed_mon	String	As "`new_closed_all`" but applies to Mondays only
new_closed_tue	String	As "`new_closed_all`" but applies to Tuesdays only
new_closed_wed	String	As "`new_closed_all`" but applies to Wednesdays only
new_closed_thr	String	As "`new_closed_all`" but applies to Thursdays only
new_closed_fri	String	As "`new_closed_all`" but applies to Fridays only
new_closed_sat	String	As "`new_closed_all`" but applies to Saturdays only
new_closed_sun	String	As "`new_closed_all`" but applies to Sundays only
new_rate_all	Decimal	Sets a new venue rate for all days (use in conjunction with the "`new_rate_all_type`" parameter)
new_rate_mon	Decimal	Sets a new venue rate for Mondays (use in conjunction with the "`new_rate_mon_type`" parameter)
new_rate_tue	Decimal	Sets a new venue rate for Tuesdays (use in conjunction with the "`new_rate_tue_type`" parameter)
new_rate_wed	Decimal	Sets a new venue rate for Wednesdays (use in conjunction with the "`new_rate_wed_type`" parameter)
new_rate_thu	Decimal	Sets a new venue rate for Thursdays (use in conjunction with the "`new_rate_thu_type`" parameter)
new_rate_fri	Decimal	Sets a new venue rate for Fridays (use in conjunction with the "`new_rate_fri_type`" parameter)

new_rate_sat	Decimal	Sets a new venue rate for Saturdays (use in conjunction with the "`new_rate_sat_type`" parameter)
new_rate_sun	Decimal	Sets a new venue rate for Sundays (use in conjunction with the "`new_rate_sun_type`" parameter)
new_rate_all_type	D | H | F | P	Sets the type of venue rate being defined (use in conjunction with the "`new_rate_all`" parameter). Possible options are "`D`" (Per day), "`H`" (Per hour), "`F`" (Fixed), or "`P`" (Per person)
new_rate_mon_type	D | H | F | P	As "`new_rate_all_type`" but applies to Mondays only (use in conjunction with the "`new_rate_mon`" parameter)
new_rate_tue_type	D | H | F | P	As "`new_rate_all_type`" but applies to Tuesdays only (use in conjunction with the "`new_rate_tue`" parameter)
new_rate_wed_type	D | H | F | P	As "`new_rate_all_type`" but applies to Wednesdays only (use in conjunction with the "`new_rate_wed`" parameter)
new_rate_thu_type	D | H | F | P	As "`new_rate_all_type`" but applies to Thursdays only (use in conjunction with the "`new_rate_thu`" parameter)
new_rate_fri_type	D | H | F | P	As "`new_rate_all_type`" but applies to Fridays only (use in conjunction with the "`new_rate_fri`" parameter)
new_rate_sat_type	D | H | F | P	As "`new_rate_all_type`" but applies to Saturdays only (use in conjunction with the "`new_rate_sat`" parameter)
new_rate_sun_type	D | H | F | P	As "`new_rate_all_type`" but applies to Sundays only (use in conjunction with the "`new_rate_sun`" parameter)
new_tax	Decimal	Sets a new (percentage) tax rate for charges for this venue
new_public	0 | 1	Sets whether the venue should be available for public booking/requesting
new_color	RRGGBB	A hexadecimal color that the venue should appear in in the booking grid
new_closed	Comma separated list	A comma separated list of date ranges during which the venue is "closed". Values are either in the format YYYYMMDD-YYYYMMDD or epoch-epoch times. Examples: "`20220601-20220614`" (Venue closed 1-14 June 2022 (inclusive)) "`20220601-`" (Venue closed from 1 June 2022 onwards) "`-20220601`" (Venue closed until 1 June 2022) "`20220601-20220607,20220614-20220620`" (Venue closed 1-7 and 14-20 June 2022 (inclusive))
new_managers	Comma separated list	A comma separated list of numeric user IDs to assign as the "managers" of the venue
new_blocks	Comma separated list	A comma separated list of numeric venue IDs to prevent (block) from being able to be booked at the same time as the venue being modified

new_alternatives	Comma separated list	A comma separated list of numeric venue IDs to offer as suitable alternative venues in the event of unavailability for the venue being modified

Omitting any of the above parameters will leave its current value unchanged. Passing any of the above parameters with a "blank" value will clear/delete the current value of the corresponding field.

Example Response

```
{
      "venue_modified":"6"
}
```

Variables Returned	
venue_modified	The unique internal ID corresponding the modified venue

mod_venue_ group

Modifies a existing venue group

Required Parameters *(Only 1 required)*	**Possible Values**	**Description**
id	*<Venue Group ID>*	Venue Group ID
group	*String*	Venue Group name

Optional Parameters *(at least 1 required)*	**Possible Values**	**Description**
new_name	*String*	New name for venue group
venues	*<Venue IDs>*	Comma separated list of venue IDs to include in venue group

Example Response

```
{
	"venue_group_modified":"4"
}
```

Variables Returned	
venue_group_modified	The unique venue group ID corresponding the modified venue group

notify_user

Sends an internal notification to a MIDAS user.

The notification will appear in the user's "Notification Center" and may also be forwarded to their email address if they have configured message forwarding.

Required Parameters	Possible Values	Description
user	User ID | User email	The numeric ID of the existing user to message, or their email address (Note: If an email address is supplied which doesn't match an existing user, the message will fail)
message	Text	The message to send
expires	Valid date and time (Format: YYYYMMDDHHMM)	The date/time after which the message should expire and be automatically removed from the recipient's My Messages screen (The user may choose to manually remove the message sooner)

Optional Parameters	Possible Values	Description
type	message | reminder	Sets the type of notification to send. If omitted, the type will be set to "message"

Example Response

```
{
      "completed":"1"
}
```

Variables Returned	
completed	Returns "1" once the API call has completed

reject_request

Rejects/declines a booking request.

Required Parameters	Possible Values	Description
id	Booking ID	The numeric ID of the booking request to be rejected

Optional Parameters	Possible Values	Description
reason	String	A reason why the request is being rejected (will be included in the rejection email notification sent to the original requestor)
silent	0 | 1	By default when a booking request is rejected, the original requestor is sent an email notification informing them as such. Setting "`silent`" to "`1`" will suppress the sending of this notification

Example Response

```
{
      "request_rejected":"1"
}
```

Variables Returned	
request_rejected	Returns "1" after a successful rejected

restore_booking

Re-instates a recently deleted booking, or bookings.

Required Parameters	Possible Values	Description
id	Booking ID	The unique numerical reference (or comma separated list of references) of the deleted booking(s) to be re-instated (which can be determined from the "get_bookings" call)

Example Response

```
{
      "restored":"925",
      "not_restored":"927,929,1024"
}
```

Variables Returned	
restored	Comma-separated list of all deleted booking references successfully
not_restored	Comma-separated list of all booking references that couldn't be restored (i.e. those which would now clash with other bookings if restored)

set_setting

Changes the value of a current MIDAS setting

Required Parameters	Possible Values	Description
setting	*<setting name>*	Setting name. See Appendix B for available setting names
value	*<new value>*	The new value for the setting

Example Response

```
{
      "response":"ok"
}
```

Variables Returned	
response	Returns "ok" after a successful setting change

set_template

Set/Modify a template

Required Parameters	Possible Values	Description
template name	*<template name>*	Template name. See below for available template names
content	Text	New HTML template content

Optional Parameters	Possible Values	Description
additional	String	For templates where an additional value can be set (such as email templates where a subject line may be specified), this new value can be passed via the "additional" parameter. See below for which templates support this parameter

Template Name	Description	Additional
email	Email (General)	
email_book_cancel	Email (Booking Cancelled)	Email Subject
email_book_confirm	Email (Booking Confirmation)	Email subject
email_book_remind	Email (Booking Reminder)	Email subject
email_req_approved	Email (Booking Request Approved)	Email subject
email_req_approved_m	Email (Booking Request Approved – with changes)	Email subject
email_req_rejected	Email (Booking Request Rejected/Declined)	Email subject
email_req_submitted	Email (Booking Request Submitted)	Email subject
email_inv_overdue	Email (Invoice Overdue)	Email subject
email_inv_remind	Email (Invoice Reminder)	Email subject
invoice	Invoice (Regular)	
invoice_cancellation	Invoice (Cancellation)	
payonline	Online Payments	URL of external .css file
print	Print	
receipt	Invoice Receipt	
webrequest	Public Booking Requests	URL of external .css file
intro	Welcome Note (on login screen)	

Example Response

```
{
      "response":"ok"
}
```

Variables Returned

response	Returns "ok" after a successful template change

util_api_usage

Returns details of recent API usage

Required Parameters
This API command has no required parameters

Optional Parameters	Possible Values	Description
period	integer | comma separated list of integers	The period of time (in hours) to calculate API usage over. A comma separated list of values may be passed. For example, passing "1,24,168" will return API usage over the past hour, past day (24 hours), and past week (168 hours)
end	Valid date and time (Format: YYYYMMDDHHMM)	By default, API usage statistics are included up to the current time. To view API usage for a period up until a specific date/time, pass the "end" parameter

Example Response

```
[
        {
        "previous_hours":"1",
        "calls":"12",
        "data_nice":"120.23 KB",
        "data_raw":"120231"
        },
        {
        "previous_hours":"24",
        "calls":"288",
        "data_nice":"2.89 MB",
        "data_raw":"2891230"
        },
        {
        "previous_hours":"168",
        "calls":"2016",
        "data_nice":"20.20 MB",
        "data_raw":"20202350"
        }
]
```

Variables Returned	
previous_hours	The period (in hours) up until the current time (or up until the time supplied in the "end" parameter) which the values of the "calls" and "data" variables relate to
calls	The number of API calls made during the "`previous_hours`" period
data_nice	The amount of data returned by the API during the "`previous_hours`" period, formatted in KB, MB, or GB and rounded to two decimal places

data_raw	The amount of data returned by the API during the "`previous_hours`" period in bytes

util_from_epoch

Converts epoch seconds to a standard date/time format

Required Parameters	Possible Values	Description
data	Epoch seconds	Epoch seconds to convert

Optional Parameters	Possible Values	Description
format	1 | 0	If omitted (or "`0`"), the returned date/time string will be in the format "`YYYYMMDDHHMM`". If "`1`", the returned date/time string will reflect the current time/date format settings in MIDAS, for instance, "`DD/MM/YYYY @ HH:MM`"

Example Response

```
{
      "response":"201305121008"
}
```

Variables Returned	
response	The converted epoch time, returned in a date/time format specified by the "`format`" parameter. If no "`format`" parameter specified, response is returned as "`YYYYMMDDHHMM`"

util_to_epoch

Converts a date/time to epoch seconds

Required Parameters	Possible Values	Description
data	Valid date and time (Format: YYYYMMDDHHMM)	Date/Time string to convert to epoch seconds

Example Response

```
{
      "response":" 1357066800"
}
```

Variables Returned	
response	The converted date/time, returned as epoch seconds

Appendix A – Activity Codes

The following codes may be used in conjunction with the "`get_activity`" call

CODE	Description	Data1	Data2	Data3	Data4
ADD	User added booking	Date/Time			
ADD2VG	User added venue to group	Venue	Venue Group		
ADDBT	User added booking type	Type	Color		
ADDC	User added client	Client	Organization		
ADDDN	User added day notes	Date	Date		
ADDI	User created invoice	Invoice			
ADDMAP	User added floorplan/map	Venue Group			
ADDRES	User added resource	Resource			
ADDU	User added User	Name	Email		
ADDV	User added venue	Venue			
ADDVG	User added venue group	Venue Group			
BKUP	MIDAS backup (Automatic)				
CANR	Booking request cancelled	Client	Organization		
DELB	User deleted booking	Date/Time			
DELBT	User deleted booking type	Type			
DELC	User deleted client	Client	Organization		
DELDN	User deleted day notes	Date	Date		
DELI	User deleted invoice	Invoice			
DELRES	User Deleted Resource	Resource			
DELV	User deleted venue	Venue			
DELVG	User deleted venue group	Venue Group			
EMS	User sent email	Client	Subject		
EMSI	User sent email invoice	Client	Invoice		
EXPB	User exported bookings				
EXPC	User exported clients				
EXPIS	User exported invoice summaries				
EXPR	User exported resources				
GENI	User generated invoice (Regular)	Invoice			
GENIC	User generated invoice (Cancellation)	Invoice			
IN	User logged in				
LFAIL	Failed login attempt	Email address			
LOCKD	User account locked for excessive failed login attempts	User account	Number of failed login attempts		
LSUSP	User account suspended	User account			
MBKUP	MIDAS backup (Manual)				
MOD	User modified booking	Date/Time			
MODBT	User modified booking type	Previous Type	Previous Color	New Type	New Color
MODC	User modified client	Client	Organization		
MODDN	User modified day notes	Date	Date		
MODI	User modified invoice	Invoice			
MODMAP	User modified floorplan/map	Venue Group			
MODRES	User Modified Resource	Resource			
MODU	User modified User	Name	Email		
MODV	User modified venue	Venue			
MREST	User Restored a MIDAS backup	Date/Time			
NEWR	Booking request received	Client	Organization		
OUT	User logged out				
PAYIF	Payment received in full	Invoice	Payer name	Payer email	Transaction ID
PAYIP	Partial payment received	Invoice	Payer name	Payer email	Transaction ID
PEED	User printed Emergency Evacuation Data				

PENA	User approved booking	Date/Time	Client		
PENR	User rejected booking	Date/Time	Client	Reason	
PRNT	User printed bookings	Print Title			
PRNTI	User printed invoice	Client	Invoice		
PWC	User changed password				
REMFVG	User removed venue from group	Venue	Venue Group		
REMU	User deleted User	Name	Email		
RENVG	User renamed venue group	Previous Venue Group	New Venue Group		
REST	User restored booking	Date/Time			
TENTX	Tentative booking expired	Date/Time	Client	Venue	
UPDATE	MIDAS updated	New Version	New Build Date		
UPSI	User updated invoice status	Invoice			
WEBB	Web Booking made	Client	Date/Time	Venue	Amount Paid

Appendix B – Setting Names

The following codes may be used in conjunction with the "get_setting" and "set_setting" calls. The "Settable" column denotes which settings can be changed via the "set_setting" call.

Setting	Description	Example value	Settable
api_version	Current version of the MIDAS API	2.54	
availability_alt_buffer	This setting (in Minutes) enforces a "gap" (spacing) when offering earlier/later alternative times in a venue	15	•
availability_alt_earlier	If "1" MIDAS will attempt to offer an earlier time in the event of an Unavailable venue	1	•
availability_alt_ignorervs	If "1" MIDAS will not enforce venue resource restrictions when offering alternative venues	1	•
availability_alt_later	If "1" MIDAS will attempt to offer a later time in the event of an Unavailable venue	1	•
availability_alt_venue	If "1" MIDAS will attempt to offer an alternative venue in the event that desired times are unavailable	1	•
availability_display_prepost	If "1", times shown on the Booking Availability screen will include any setup/breakdown times	1	•
availability_estimate_costs	If "1", estimated charges for bookings/resource will be indicated on the Booking Availability screen	0	•
availability_include_pending	If "1" MIDAS takes into account any pending booking requests when checking booking availability. If "0" confirmed bookings can be made over provisional booking slots	0	•
availability_post_buffer	Maintains a fixed spacing (in minutes) after each booking before the next booking may commence	15	•
availability_pre_buffer	Maintains a fixed spacing (in minutes) before each booking to that of the end of the previous booking	15	•
backup_email	The email address that automated database backups are sent to	backup@yourdomain.com	•

backup_last	The time of the last backup (epoch seconds)	1365609454	
backup_persist	The number of days to keep backups on server	7	•
bookings_overbook_post	If "1" a booking following on from another booking may start during the breakdown time of the previous	0	•
bookings_overbook_pre	If "1" a booking will be allowed to end during the setup time of a booking following on from it	0	•
bookings_prompt_add_client	If "1" MIDAS will prompt for additional client details when adding bookings for a new client	0	•
bookings_use_map	If "1" users can visually book from a floorplan or map. If "0" users select bookable spaces from a list of venues.	0	•
build_date	Current MIDAS build date	1365608000	
callto_enabled	If "1" client phone/cell numbers become clickable "callto" links within MIDAS	1	•
checkintime	Sets the check in time (in HHMM format) for when the time selector is set to "Nights"	1500	•
checkouttime	Sets the check out time (in HHMM format) for when the time selector is set to "Nights"	1100	•
cron_enable_invoice_overdue	If "1" automated invoice overdue notifications are enabled	1	•
cron_enable_invoice_remind	If "1" automated upcoming invoice reminders are enabled	1	•
cron_enable_invoice_send	If "1" automated sending of unsent invoices is enabled	1	•
cron_enable_post_feedback	If "1" automated post-booking follow up emails are enabled	1	•
cron_enable_start_remind	If "1" automated upcoming booking reminders are enabled	1	•
cron_hour	The hour (0-24) during which daily scheduled tasks should run	11	•
cron_invoice_overdue	The number of days after an unpaid invoice was due to automatically send an overdue notification to the client	7	•

cron_invice_remind	The number of days before an unpaid invoice is due to automatically send a payment reminder to the client	3	•
cron_last	The time that scheduled tasks were last run (epoch second)	1407123529	
cron_post_feedback	The number of hours after bookings have completed to send automated post-booking follow up emails	24	•
cron_start_remind	If "1" automated upcoming booking reminders are enabled	1	•
datetime_amsymbol	The symbol denoting the first 12 hours of the day (when running in 12 hour clock mode)	AM	•
datetime_datefirst	If "1" dates are shown before times, otherwise times are shown before dates	1	•
datetime_dateformat	The date format	DD/MM/YYYY	•
datetime_dtlink	The link symbol between date and time	@	•
datetime_gmtoffset	The timezone's GMT offset	0	•
datetime_minterval	The granularity of minutes	5	•
datetime_pmsymbol	The symbol denoting the first 12 hours of the day (when running in 12 hour clock mode)	PM	•
datetime_startofweek	The day that should be considered the start of the week (0 = Sunday, 1 = Monday, etc)	1	•
datetime_timeformat	The time format	HH:mm	•
datetime_timenow	The current time (in epoch seconds)	1365696187	•
datetime_timezone	The timezone	Europe/London	•
datetime_ttlink	The link symbol between two times	-	•
datetime_vcaltzfix	If "1" times in VCALS are adjusted to UTC times for compatibility with some 3rd party calendar apps don't support "local" times	0	•
default_booktype	The numeric ID of a booking type set as the default booking type	3	•
email_allsendfrom	If "1" all outgoing email from MIDAS will be sent from the "`email_sendfrom`" address. If "0", only system-generated emails (such as password reset requests, etc) will be sent from the	0	•

	"email_sendfrom" address. User-initiated email (i.e. sending an email or invoice to a client, etc) will instead send mail appearing to be sent from the email address associated with the user's account		
email_bcc	The email address that outgoing email should be bcc'd to	bcc@yourdomain.com	•
email_bcc_what	A string of 4-character codes denoting which type of emails should be bcc'd. Possible codes are: "mail" (Generic email), "bcan" (Booking cancellations), "bcon" (Booking confirmations), "brem" (Booking reminders), "rapp" (Booking request approved), "rapm" (booking request approved with changes), "rrej" (Booking request rejected), "rsub" (Booking request submitted), "invo" (Invoice), "iovr" (Invoice Overdue), "irmd" (Invoice Reminder), "rcpt" (Receipt)	mailbcanbconrsubiovr	•
email_compat	If "1" outgoing emails will be sent in compatibility mode for the benefit of older email clients	0	•
email_merge	If "1" MIDAS will attempt to reduce outgoing email volume by automatically combining similar emails, otherwise if "0" MIDAS will send emails individually	1	•
email_reply_to	Optionally specify a "reply to" email address for all outgoing mail	midas@example.com	•
invoicing_amounts_ex_tax	If "1", invoice amounts are show excluding tax	1	•
invoicing_apply_disc	If "1", booking type discounts apply to venues only. If "2", discounts apply to resources only. If "12", discounts apply to both venues + resources. If "0", no discounts will be applied	1	•
invoicing_cancel_amount	The amount (fixed or percentage) to charge in late cancellation fees	20%	•
invoicing_cancel_enabled	If "1" and a booking is cancelled (deleted) within invoicing_cancel_leadtime hours of	1	•

	when the booking was due to commence, a cancellation invoice will be created		
invoicing_cancel_leadtime	Specifies (in hours) the cut-off point before a booking commences after which a cancellation invoice can be generated if the booking is cancelled (deleted)	168	•
invoicing_cancel_tax	Specifies a percentage tax rate to apply to cancellation invoices	20	•
invoicing_counter	The next invoice number to be generated	4	•
invoicing_counter_cancel	The next cancellation invoice number to be generated	4	•
invoicing_counter_credit	The next credit note number to be generated	4	•
invoicing_counter_deposit	The next deposit invoice number to be generated	4	•
invoicing_createbydefault	If "1" the "Create Invoice" option is checked when adding bookings	0	•
invoicing_createifzero	If "1" invoices will be created even if the calculated invoice total is zero. If "0" invoices will only be created if their values are non-zero	1	•
invoicing_currencycode	The currency code	USD	•
invoicing_currencysymbol	The currency symbol	$	•
invoicing_decimalsep	The decimal separator	.	•
invoicing_deposit_amount	The amount to charge on deposit invoices. Can be an absolute value, or a percentage of total booking cost	20%	•
invoicing_deposit_deduct	If "1" the value of a deposit invoice is deducted from the total value of the corresponding regular invoice	0	•
invoicing_deposit_enabled	If "1" deposit invoices are enabled	1	•
invoicing_deposit_tax	Specifies a percentage tax rate to apply to deposit invoices	20	•
invoicing_generate_notes	The internal name of a booking field to optionally include in the "notes" section on generated invoices	notes	•
invoicing_include_prepost	If "1" booking length takes into account setup/breakdown times for the purposes of invoicing. If "0" setup/breakdown periods are not invoiced for	0	•

invoicing_includezero	If "1" items will be included on invoices even if their total value is zero. If "0" invoices will only include items that have an associated cost	1	•
invoicing_itemize_notes	If "1" notes on invoices will be itemized per booking. If "0" invoice notes will be combined	1	•
invoicing_no_booking_mod	If "1" bookings from which invoices have been generated can then no longer be modified	0	•
invoicing_no_invoice_delete	If "1" invoices cannot be deleted/removed from the system until they have been paid in full	0	•
invoicing_paid_if_zero	If "1" when generating an invoice totaling zero, the invoice will be marked as "Paid In Full". If "0" a generated invoice totaling zero will not automatically be marked as paid	0	•
invoicing_partial_payments	If "1" partial invoice payments are allowed. If "0" invoices can only be paid in full	0	•
invoicing_paypal_account	Your PayPal email address (Required to accept payments via PayPal)	paypal@mydomain.com	•
invoicing_paypal_enabled	If "1" online invoice payments are allowed via PayPal	1	•
invoicing_pay_require_email	If "1" clients are required to enter their email address in order to view their invoice online	1	•
invoicing_prefix	The prefix that is prepended to all regular invoice numbers	MIDAS	•
invoicing_prefix_cancel	The prefix that is prepended to all cancellation invoice numbers	CANCEL	•
invoicing_prefix_credit	The prefix that is prepended to all credit note numbers	CREDIT	•
invoicing_prefix_deposit	The prefix that is prepended to all deposit invoice numbers	DEPOSIT	•
invoicing_resource_additional	How resource items appear on invoices. "0" = Name "1" = Name (Venue) "2" = Name (Date/Times) "3" = Name (Venue, Date/Times)	0	•
invoicing_rounding	If "1" invoice values are rounded up to the nearest whole number 0	1	•

invoicing_send_as_pdf	If "1" invoices are sent via email to clients as PDF attachments. If "0" invoices are sent to clients enclosed within the body of the email	0	•
invoicing_silentdisc	If "1" booking type discounts are applied to invoices without any indication. If "0" invoices will denote that a discount has been applied	0	•
invoicing_stripe_enabled	If "1" online invoice payments are allowed via Stripe	1	•
invoicing_stripe_pk	Your Stripe Publishable Key (Required to accept payments via Stripe)	pk_live_xxxxxxxxxxxxxxxx	•
invoicing_stripe_sk	Your Stripe Secret Key (Required to accept payments via Stripe)	sk_live_xxxxxxxxxxxxxxxxx	•
invoicing_tax	The default tax percentage to be added to invoices	0	•
invoicing_terms	The default payment terms (days) for invoices	30	•
invoicing_thousandsep	The thousand separator	,	•
maxnights	The maximum number of nights stay to permit when the time selector is set to "Nights"	14	•
occupancy_max	The maximum number of people allowed on your site at any one time	510	•
occupancy_show	Show the occupancy level when adding/modifying bookings	1	•
occupancy_warn	Warn if the number of people on site exceeds this value	300	•
pdf_margins	A pipe separated list of PDF page margins to apply (in the order top|bottom|left|right). The final value is the unit of measurement of the margins (i.e. "mm" for millimeters)	10|10|10|10|mm	•
pdf_orientation	The page orientation (Portrait or Landscape) of generated PDFs	Portrait	•
pdf_size	The page size of generated PDFs	A4	•
print_evactolerance	When printing Emergency Evacuation Data, this number reflects how many minutes either side of the current time the calculation of the number of people on site should account for	15	•

print_sortby	The booking field to sort booking print outs by	start	•
print_sortdir	The sort order of print outs (0 = ascending, 1 = descending)	0	•
print_split_resources	If "1" resources will be split into their categories on printouts	0	•
public_update_clients	If "1" existing client records will be updated based on data entered on public booking/request screens	0	•
resources_overbook_post	If "1" a booking following on from another booking may use the same resources during the breakdown time of the first booking and setup time of the second	0	•
resources_overbook_pre	If "1" a booking following on from another booking may use the same resources during the setup time of that booking during the breakdown time of the previous	0	•
search_maxresults	The maximum number of search results to return per page	50	•
security_hibp	If "1" new user passwords are checked against the "Have I been Pwned?" service	1	•
security_new_device_alert	If "1" users are alerted upon logins from unfamiliar devices	1	•
tentative_enabled	If "1" tentative booking expiration is enabled, if "0" tentative bookings won't expire	1	•
version	Current MIDAS version	4.32	
view_autoclose_alerts	The number of seconds the "My Messages" pop-up should display for, if enabled by users	10	•
view_autoclose_notes	The number of seconds after which any "day notes" pop-ups should automatically close	15	•
view_changemonthcell	Allow users to choose what's displayed in the monthly overview	1	•
view_cutoffdays	The number of days to keep temporary logs for	30	•
view_default	The default view in the Booking Grid	All	•
view_gridrowheight	The height (in pixels) of each row in the booking grid	30	•

view_gridvenuewidth	The width (in pixels) of the first column in the booking grid	100	•
view_showonblocks	The booking field(s) to show on booking "blocks" in the booking grid	organisation	•
view_showonmonthcell	The booking field to show on dates in the monthly overview	client	•
view_showontools	The booking field(s) to show on tooltips when hovering over booking "blocks" in the booking grid	organisation	•
view_updatefrequency	The number of seconds between successive background data refreshes	30	•
view_viewsize	The number of days to show in the booking grid by default	1	•
webbook_enabled	If "1" public web booking is enabled, if "0" public web booking is disabled	1	•
webbook_expire	The length of time (in minutes) a public web booking is held before being automatically removed if not paid for by the client within this time frame	15	•
webbook_invoice	If "1" invoices are automatically created when web bookings are made	1	•
webrequest_alloweddomains	A comma separated list of email domains from which booking requests are permitted	@example.com, @hotmail.com	•
webrequest_autoapprove	If "1" booking requests are auto-approved. If "0" booking requests must be approved by an administrator	0	•
webrequest_autoapprove_from	A comma separated list of email address/domains from which booking requests are automatically approved	@example.com, @hotmail.com	•
webrequest_autoapprove_type	A comma separated list of internal booking type IDs for which booking requests are automatically approved	1,3,4	•
webrequest_blockcolor	The color of existing booking blocks on the public booking request screen	red	•
webrequest_bulkapprove_order	Controls the order in which booking requests are bulk approved (0 = Earliest requested approved first, 1 = Latest requested first, 2 = Earliest	0	•

	commencing first, 3 = Latest commencing first)		
webrequest_disablepast	The date (YYYYMMDD) past which booking requests are not permitted	20141231	•
webrequest_enabled	If "1" public booking requests are enabled, if "0" public booking requests are disabled	1	•
webrequest_leadintime	The number of days in advance public booking request must be made	7	•
webrequest_leadouttime	The number of days in advance public booking request are allowed to be made for	14	•
webrequest_showcapacities	If "1" selected venue's capacities are shown during public booking requesting. If "0" venue capacities are not revealed to requestors	1	•
webrequest_showclosed	If "1" selected venue's operating hours are shown during public booking requesting. If "0" operating hours are not revealed to requestors	0	•
webrequest_showonblocks	What information should be shown on booking blocks in the public booking request screen	times	•
webrequest_use_map	If "1" enables visually booking/requesting from a floorplan or map on the public bookings/request screen. If "0" users select spaces to book/request from a list of venues on public booking/requesting screens.	0	•
webrequest_venue_auto_select	If "1" venue auto selection is enabled for public booking/requests	0	•

Appendix C – Error Handling

When making API calls, your applications should take into account possible errors that may be returned.

Unless you've configured your API to return http status code 200 (OK) for every request, successful API calls should return either a 200 (OK) or 201 (Created) http status code. Unsuccessful API calls may return 4xx/5xx http status codes, as outlined in the Error Response table below.

API errors will be returned in JSON format as follows:

Example Error Response

```
{"error":"error response"}
```

Error Response	HTTP Status Code	Meaning	Applies To

"API running in read only mode"	403	The API is currently running in read-only mode, and you have made a call that requires full-access. Read-Only mode can be turned off via MIDAS Admin Options → Manage Addons → API Access → Read-Only access	`add_*`
			`approve_request`
			`del_*`
			`email_client`
			`gen_invoice`
			`mod_*`
			`notify_user`
			`reject_request`
			`set_*`

"booking not found"	404	The specified booking was not found	`del_booking` `mod_invoice`
"client not found"	404	The specified client was not found	`email_client`
"exceeds invoice total"	400	The sum of amounts paid, credited, and refunded exceed the specified invoice's total	`mod_invoice`
"filter field not found"	400	The specified filter field is invalid	`get_bookings`
"GET method not allowed. Use POST instead"	405	An http GET request has been made to the API, but the API has been set to accept http POST requests only. This setting may be changed via MIDAS Admin Options → Manage Addons → API Access → Allow GET (as well as POST) method	*All*
"invalid api key"	401	The API key supplied is invalid	*All*

"invoice x is already associated with this booking"	409	An attempt was made to generate an invoice for a booking which already has an invoice associated with it	`gen_invoice`
"invoice not found"	404	The specified invoice was not found	`del_invoice` `mod_booking`
"missing api key"	401	You have not supplied your API key with your call	*All*
"no email address on record"	400	The specified client record has no email address associated with it	`email_`client

"no matches found"	404	No data was returned for the API call. Try broadening the search parameters	`del_bookings`
			`gen_invoice`
			`get_activity`
			`get_bookings`
			`get_client`
			`get_consuable_levels`
			`get_invoice`
			`get_invoices`
			`get_messages`
			`get_reminders`
			`get_user`
			`get_venues_in_group`
			`get_watches`

"not available with LDAP enable"	400	The API call is unavailable when Active Directory sign-on in enabled	`authenticate_user`
"not enabled"	503	API access is not enabled at this time. API access can be enabled from within MIDAS via MIDAS Admin Options → Manage Addons → API Access → Enabled	*All*
"notification expired"	400	The notification's expiry date/time has passed	`notify_user`
"protected setting"	401	The setting you're trying to modify is read-only and cannot be modified	`set_setting`
"referrer not allowed"	403	The API call originates from a blocked domain/IP address. Allowed domains/IP's may be specified from within MIDAS via MIDAS Admin Options → Manage Addons → API Access → Allowed Referrers	*All*
"required parameter missing"	400	One or more parameters are required for the particular call you're making is missing	*All*

"required permission missing"	401	The API call was made under a user account which does not have sufficient privileges to make the particular API call	*Various*
"request not found"	404	The specified booking request was not found	`approve_request` `reject_request`
"resource not found"	404	The specified resource was not found	`del_resource` `mod_resource`
"unable to …"	502	A generic error indicated that the API has behaved in an unexpected way. Please contact our support team for assistance	*Various*
"unable to connect to database"	503	The API was unable to connect to the database	*All*
"unable to delete – invoice not yet paid"	401	Software settings prevent the specified invoice from being deleted until it has been paid in full	`del_invoice`
"unable to modify – invoice has been sent"	401	Software settings prevent the specified invoice from being modified once it has been finalized	`mod_invoice`

"unknown booking type"	404	Indicates that the specified booking type doesn't exist	`del_bookings`
			`get_bookings`
			(when the type parameter is specified)

"unknown command"	400	The API command you've specified was unrecognized. Please check you're using a valid command	*All*
"unknown resource"	404	Indicates that the specified resource doesn't exist	`get_resource_availability`
"unknown setting"	404	Indicates that the specified setting isn't recognized	`get_setting` `set_setting`
"unknown template"	404	Indicates that the specified template isn't recognized	`get_template` `set_template`

"unknown user"	404	Indicates that the specified user doesn't exist	`get_activity`
			`get_messeges`
			`get_reminders`
			`get_watches`

"unknown venue"	404	Indicates that the specified venue doesn't exist	`del_bookings`
			`get_availability`
			`get_bookings`
			`get_venue`
			(when the venue parameter is specified)

"unknown venue group"	404	Indicates that the specified venue group doesn't exist	`get_venues_in_group`

Code Samples

Below are some typical examples of code for common programming languages of how you could access the MIDAS API.

In all of the following examples the "`get_setting`" API call is made in order to retrieve the current version of MIDAS

C#

```
using(WebClient client = new WebClient())
{
    System.Collections.Specialized.NameValueCollection reqparm = new
System.Collections.Specialized.NameValueCollection();
    parameters.Add("key", "your_api_key");
    parameters.Add("action", "get_setting");
    parameters.Add("setting", "version");
    byte[] responsebytes =
client.UploadValues("https://your_midas_url/api.pl", "POST", parameters);
    string response = Encoding.UTF8.GetString(responsebytes);
}
```

Go

```
package main

import (
    "bytes"
    "encoding/json"
    "fmt"
    "log"
    "net/http"
)

func main() {
    values :=
map[string]string{"key":"your_api_key","action":"get_setting","setting":"vers
ion"}
    json_data, err := json.Marshal(values)
    if err != nil {
        log.Fatal(err)
    }
    resp, err := http.Post("https://your_midas_url/api.pl",
"application/json",
        bytes.NewBuffer(json_data))
    if err != nil {
        log.Fatal(err)
    }
    var res map[string]interface{}
    json.NewDecoder(resp.Body).Decode(&res)
    fmt.Println(res["json"])
}
```

Java

```
HttpClient httpclient = HttpClients.createDefault();
HttpPost httppost = new HttpPost("https://your_midas_url/api.pl");

List params = new ArrayList(3);
params.add(new BasicNameValuePair("key", "your_api_key"));
params.add(new BasicNameValuePair("action", "get_setting"));
params.add(new BasicNameValuePair("setting", "version"));
httppost.setEntity(new UrlEncodedFormEntity(params, "UTF-8"));

HttpResponse response = httpclient.execute(httppost);
HttpEntity entity = response.getEntity();

if (entity != null) {
    InputStream instream = entity.getContent();
    try {
        // process response
    } finally {
        instream.close();
    }
}
```

jQuery

```
$.post("https://your_midas_url/api.pl", { key: "your_api_key", action:
"get_setting", setting: "version" })
.done(function(response) {
 alert("Response: " + response);
}, "JSON");
```

.net

```
using System;
using System.Collections.Specialized;
using System.Net;

public static class Http
{
    public static byte[] Post(string uri, NameValueCollection pairs)
    {
        byte[] response = null;
        using (WebClient client = new WebClient())
        {
            response = client.UploadValues(uri, pairs);
        }
        return response;
    }
}
var response = Http.Post("https://your_midas_url/api.pl", new
NameValueCollection() {
    { "key", "your_api_key" },
    { "action", "get_setting" },
    { "setting", "version" }
});
```

Perl

```
use LWP::UserAgent;
my$ua = LWP::UserAgent->new(env_proxy => 0,keep_alive => 0,timeout =>
30,agent =>'Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 5.0)');
my$r = $ua-
>post("https://your_midas_url/api.pl",[key=>"your_api_key",action=>"get_setti
ng",setting=>"version"]);
if ($r->is_success) {
      $response=$r->content;
}
```

PHP

```
$myvars = "key=your_api_key&action=get_setting&setting=version";

$ch = curl_init("https://your_midas_url/api.pl");
curl_setopt( $ch, CURLOPT_POST, 1);
curl_setopt( $ch, CURLOPT_POSTFIELDS, $myvars);
curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, 1);
curl_setopt( $ch, CURLOPT_HEADER, 0);
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1);

$response = curl_exec( $ch );
```

Python

```
import urllib
import urllib2
parameters = {'key' : 'your_api_key',
          'action' : 'get_setting',
          'setting' : 'version' }
data = urllib.urlencode(parameters)
req = urllib2.Request("https://your_midas_url/api.pl", data)
getresponse = urllib2.urlopen(req)
response = getresponse.read()
```

Ruby

```
require "net/http"
require "uri"

uri = URI.parse("https://your_midas_url/api.pl")

response = Net::HTTP.post_form(uri, {"key" => "your_api_key", "action" =>
"get_setting", "setting" => "version"})
```

Troubleshooting

If you run into difficulties making API calls, the following solutions may be of use…

When making API calls with a valid key, the only response returned is 401 Unauthorized status with an "invalid api key" message	
Applies To:	Making API calls via POST if your MIDAS system was purchased before 2016 and is cloud-hosted by us
Cause:	Making API calls via POST if your MIDAS system was purchased before 2016 and is cloud-hosted by us
Solution 1:	If you're making API calls to the domain "midas.hosting/*your_midas_url*/api.pl", update your code to make calls to "*your_midas_url*.mid.as/api.pl" instead, and ensure you're making calls via https instead of http
Solution 2:	Make API calls via GET instead

When making a valid API calls, a 404 Not Found HTTP status is returned	
Applies To:	API calls made in API v2.45 (or later)
Cause:	Successful API calls should return either a 200 (OK) or 201 (Created) http status code. Unsuccessful API calls may return 4xx/5xx http status codes. An API call can be both valid and unsuccessful at the same time. For example, if you attempt to delete a non-existent booking, a 404 (Not Found) status may be returned, indicating that the booking you were attempting to delete via the API was not found. A complete list of possible HTTP status codes returned by the API is outlined in the Error Response table
Solution:	Configure your API to return http status code 200 (OK) for every request by enabling the "Force HTTP Status 200 for all responses?" API setting

When making API calls, the only response returned is "NULL"	
Applies To:	Using cURL in PHP to make API calls via HTTPS
Cause:	Receiving a "NULL" response when making API calls in this way will likely be due to the problem that cURL on the server where your PHP script is running has not been configured to trust the HTTPS Certificate Authority (CA) of the server where your MIDAS system resides. There are two solutions to this problem:
Solution 1:	Add the following line to your PHP code (before you call curl_exec): `curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);` …this simply configures cURL to accept any server(peer) certificate. Whilst this is a quick solution, it isn't as secure as Solution 2:
Solution 2:	The more secure option is to use the "CURLOPT_CAINFO" parameter. This is used to point towards a CA certificate that cURL should trust (and therefore any server/peer certificates issued by that CA will also be trusted). In order to

do this, you'll need the CA certificate for the server where your MIDAS resides (if your MIDAS is cloud-hosted by us, you can obtain this certificate in a zip file by clicking here).

Upload the CA certificate to the server where your PHP script is running from, and then add the following lines to your PHP code:

```
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
curl_setopt($ch, CURLOPT_CAINFO, getcwd() .
"/midas.hosting.crt");
```

(This third line assumes the .crt certification can be found in the same directory as your PHP script, otherwise you'll need to adjust the above path according)

Blank responses are received when making API calls

Applies To:	Making API calls if your MIDAS system is cloud-hosted by us
Cause:	Your IP address has been automatically blocked by our systems due to an excessive API call rate (i.e. flooding our servers with a high number of API calls in a short period of time)
Solution:	Check the frequency with which calls are being made to the MIDAS API from your code, and if necessary reduce the rate at which these calls are made (to no more than 60 over a 30 second period). Once you've identified the cause of the excessive API calls in your code, and rectified, please contact us to have your IP address unblocked so that you can resume making API calls

Unable to connect to the API since 1st July 2017

Applies To:	Making API calls if your MIDAS system is cloud-hosted by us
Cause:	As of 1st July 2017 we no longer accept legacy TLS 1.0 connections across our network [Read More]
Solution:	Please ensure that your applications and the underlying programming language you develop in can support (and are correctly configured for) TLS 1.2/1.3 connections. For instance Java 6 (1.6) (and lower) and .NET 3.5 (and lower) languages don't support TLS 1.2

Unable to connect to the API since 4th July 2020

Applies To:	Making API calls if your MIDAS system is cloud-hosted by us
Cause:	As of 4th July 2020 we no longer accept legacy TLS 1.1 connections across our network [Read More]
Solution:	Please ensure that your applications and the underlying programming language you develop in can support (and are correctly configured for) TLS 1.2/1.3 connections. For instance Java 6 (1.6) (and lower) and .NET 3.5 (and lower) languages don't support TLS 1.2

API Release Notes

See https://mid.as/api/release-notes

For the most up-to-date API documentation,
please visit https://mid.as/api