pen-box.tech API Documentation v.1.0

Quick Recap: What & Why an API

A restful API is arguably the quickest & easiest way to send data directly into a database (for storage and sorting) from a command line tool such as PowerShell or Python.

How Does it Work?

First off, trying to describe the file hierarchy and then moving up into how it applies to you. For 2/5 of the Group, you have nine classes to work with, Anson has five in a 'further in' directory. Each class has two pages at the moment, a create.php (where you send json entries with all info into) and a read.php (which pulls all of the entries tied to either that business or that computer in that business, based on whether that class has a one to one, or one to many attribute with a computer).

pen-box.tech/computer_information/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(50)	Machine_ID
os	varchar(160)	Operating_System
os_sn	varchar(160)	OS_Serial_Number
user	varchar(256)	User_Current
uptime	varchar(160)	System_Uptime
manufacturer	varchar(160)	System_Manufacturer
model	varchar(160)	System_Model
serial_number	varchar(160)	Serial_Number
firewall	tinyint(1) [acts as a bool]	Firewall_Active
drive_size	varchar(10)	Drive_Size
drive_free_space	varchar(10)	Drive_Free_Space
ram	varchar(10)	RAM
processor	varchar(160)	Processor
cd_drive	varchar(160)	CD_Drive
graphics	varchar(160)	Graphics_Card

[I know that the field values have different names going in vs. coming out. It just hit me this morning, I'm sorry. Given more time I'd fix it.]

computer_information/create.php
[running off the top of my head, so these might not be *exact*]
First off, you will need your Token (a 256-long randomized string of characters that I'll generate and hand off to you to put into your code). Each device we run will have it's own unique Token.

```
$url = 'https://pen-box.tech/computer_information/create.php?
Token=[your token goes here]';
$payload = @{
     business_id     = 'business_id_value_here';
     machine_id      = 'machine_id_value_here';
     os              = 'os_value_here';
     os_sn           = 'os_sn_value_here';
     user            = 'user_current_goes_here';
     uptime          = 'uptime_goes_here';
     manufacturer    = 'manufacturer_value';
     model           = 'model_here';
     serial_number   = 'serial_number_here';
     firewall        = 'firewall_value_here';
     drive_size      = 'drive_size_value_here';
     drive_free_space= 'drive_free_space_here';
     ram             = 'string of the ram here';
     processor       = 'processor_value_here';
     cd_drive        = 'cd_drive_value_here';
     graphics        = 'graphics_value_here';
};
$the_json = $payload | ConvertTo-Json;
Invoke-RestMethod -Method PUT -Uri $url -Body $the_json;
```

computer_information/read.php
To access read.php for computer information (because a computer's info is one to one per computer) you must also put the business_id in the url so that the API knows which business to search for. If you don't you will receive an error 401. You will get back something like:

```
$url = 'https://pen-box.tech/computer_information/read.php?
bus=[business id goes here]&Token=[your token goes here]';
$response = Invoke-RestMethod -Uri $url;
$response

computer_information
--------------------
{@{Business_ID=x; Machine_ID=x; Operating_System=X;
OS_Serial_Number=x;...
```

pen-box.tech/installed_programs/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
name	varchar(160)	Name
version	varchar(50)	Version
install_date	varchar(10)	Install_Date

installed_programs/create.php
[Extremely similar to computer_information/create.php's inserting method. Just edit the url, and set the payload to the fields available here. Will type it out once I have more time, in a few hours maybe?]

installed_programs/read.php
[Same thing as above, just change $url to point to 'https://pen-box.tech/installed_programs/read.php?bus=**[business id**]&mac=[**machine id**]&Token=[**token here**]'. Machine ID needed because one machine has many programs installed, and you get error 401 otherwise.]

pen-box.tech/installed_updates/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
description	varchar(50)	Description
hf_id	varchar(20)	Hot_Fix_Id
installed_on	varchar(25)	Installed_On
installed_by	varchar(50)	Installed_By

installed_updates/create.php
[See installed_programs and follow those general instructions.]

installed_updates/read.php
[Many to one on updates, so follow installed_programs general picture.]

pen-box.tech/machines/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
machine_name	varchar(160)	Machine_Name

machines/create.php
[follow general instructions as above]

machines/read.php
[one to one, so only Token and business_id (bus) are needed. Follow Computer_information.]

pen-box.tech/network_information/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
description	varchar(160)	Description
dns	varchar(160)	DNS_Hostname
ip	varchar(160)	IP_Address
mac	varchar(20)	MAC_Address

network_information/create.php
[follow computer_information's general example]

network_information/read.php
[one to one here too, so pass in bus & token]

pen-box.tech/port_scan/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
port_num	int(5)	Port_Number
port_stat	tinyint(1) [0:closed, 1:open]	Port_Status

port_scan/create.php
[follow general instructions]

port_scan/read.php
[one to many, so pass in bus, mac, & Token]

pen-box.tech/shared_devices/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
name	varchar(160)	Name
description	varchar(160)	Description
path	varchar(1024)	Path

shared_devices/create.php
[follow general insctructions]

shared_devices/read.php
[one to many, so pass in bus, mac & Token]

pen-box.tech/usb_devices/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
drive_name	varchar(160)	Drive_Name
free_space	varchar(50)	Free_Space

usb_devices/create.php
[follow general instructions]

usb_devices/read.php
[one to many, so pass in bus, mac & Token]

pen-box.tech/user_documents/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
machine_id	varchar(160)	Machine_ID
name	varchar(256)	Name
type	varchar(160)	Extension_Type
path	varchar(1024)	Path
last_access	varchar(25)	Last_Access_Time

user_documents/create.php
[follow general instructions]

user_documents/read.php
[one to many, so pass in bus, mac, & Token]

pen-box.tech/user_information/

Insert into create.php as	Type in the database		Returned from read.php as
business_id	varchar(160)		Business_ID
machine_id	varchar(160)		Machine_ID
name	varchar(50)		Current_User_Name
type	int(4)		Account_Type
sid	varchar(50)		User_SID
domain	varchar(160)		Account_Domain
password	tinyint(1)	[0:no 1:yes]	Password_Required
admin	tinyint(1)	[0:no 1:yes]	Is_Admin

user_information/create.php
[follow the general instructions]

user_information/read.php
[Hypothetically more than one user could exist on a computer, and I didn't want something that trivial to trip up this system, so it is considered one to many. Pass in bus, mac & Token.]

Anson Gets his own subdirectory

because he is special. Also it seemed right at 2:00 AM somewhere between a Saturday and a Sunday. Also, these laziest of tables all only have two rows. ALSO, also, I built these under the assumption that a business would only get one test run on it, we could easily toss in a third row Test_Number to let us keep results of more tests in the future.

pen-box.tech/web_test/main/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
value	text	Value

web_test/main/create.php
and hand off to you to put into your code). Each device we run will have it's own unique Token.

```
$url = 'https://pen-box.tech/web_test/main/create.php?Token=[your
token goes here]';
$payload = @{
     business_id     = 'business_id_value_here';
     value           = 'value's_value_here';
};
$the_json = $payload | ConvertTo-Json;
Invoke-RestMethod -Method PUT -Uri $url -Body $the_json;
```

web_test/main/read.php
You're going to get **a whole lot** back for your chunks of data so watch out for that. ALSO, because of the sanitization process, useful things such as < and > have been torn apart into htmlspecialchar and strip_tags representations. I can look into getting PHP to revert these changes when returning them, but I've been a little pressed for time thus far.
It might be handy to say that you just need the general "read" call, but just pass bus and Token in the address.

pen-box.tech/web_test/ssl_data

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
value	mediumtext (apparently could store a book, seemed appropriate)	Value

web_test/ssl_data/create.php
[see example right above this]

web_test/ssl_data/read.php

[pass bus and Token in through the address

pen-box.tech/web_test/ssl_log

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
value	mediumtext	Value

web_test/ssl_log/create.php
[see your first example]

web_test/ssl_log/read.php
[pass bus and Token through address]

pen-box.tech/web_test/wp_scan_content/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
value	varchar(1000)	Value

web_test/wp_scan_content/create.php
[see first example]

web_test/wp_scan_content/read.php
[pass bus and Token in through address]

pen-box.tech/web_test/wp_scan_wpcontent/

Insert into create.php as	Type in the database	Returned from read.php as
business_id	varchar(160)	Business_ID
value	varchar(1000)	Value

web_test/wp_scan_wpcontent/create.php
[see your first example]

web_test/wp_scan_wpcontent/read.php
[pass bus and Token in through address]

Sorry for any/all typos, and noncoherent ramblings in this doc, I definitely rushed these pages out to have some rough instructions that hopefully can be followed without *too* much trouble.

Figured it would be handy to have one example, and show what datatype each row was made out of.