

# CORONA QUICK START FOR AZURE

THIS GUIDE EXPLAINS HOW TO SET UP CORONA ON A VIRTUAL MACHINE IN AZURE

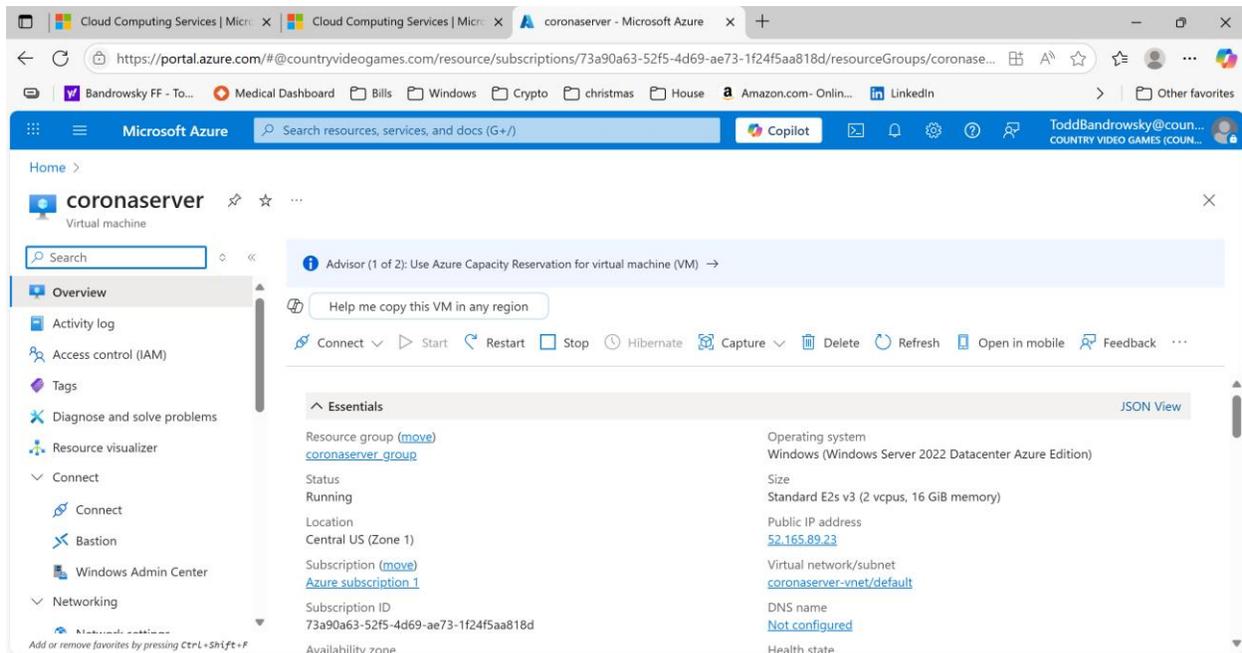
## CREATE AN AZURE VIRTUAL MACHINE

USE WINDOWS SERVER 2022

WE USED SIZING OF 2 CORE 16GB.

*ALSO TESTED ON XEON 9480 MAX 56 CORES 112 THREADS*

EXPLORING THOSE FREE MACHINES OUT THERE.



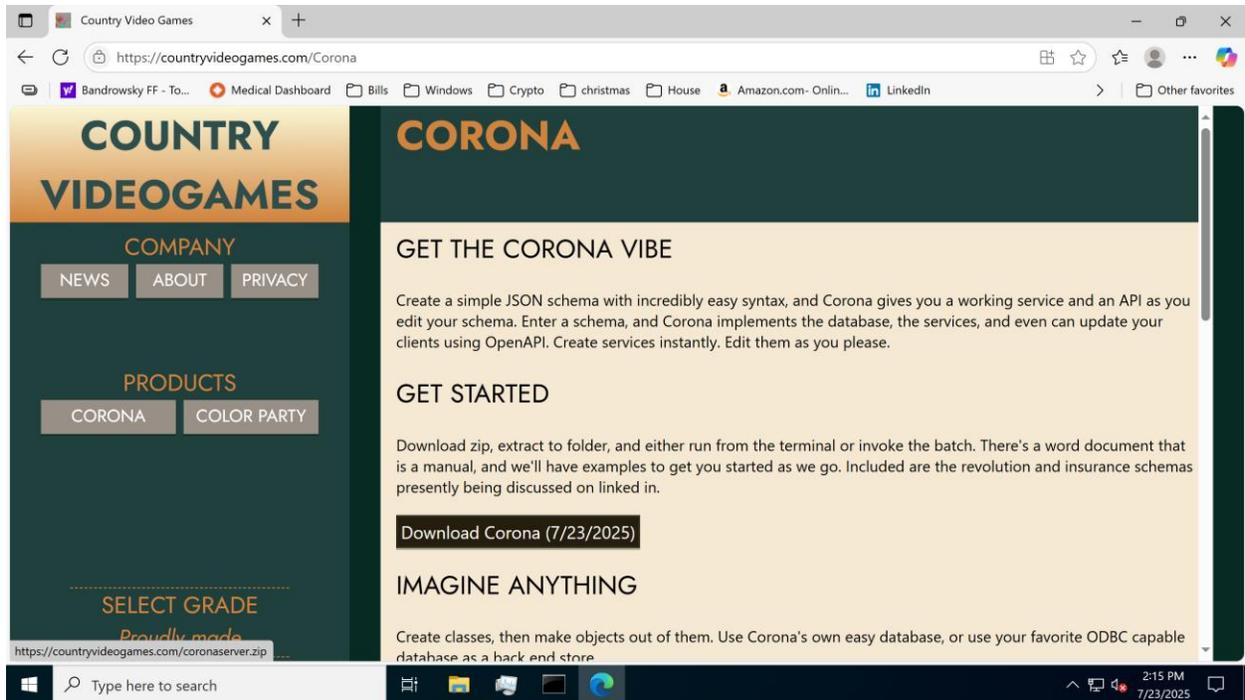
The screenshot displays the Microsoft Azure portal interface for a virtual machine named 'coronaserver'. The page shows the 'Overview' tab with various navigation options on the left sidebar. The main content area displays the 'Essentials' section, which includes the following details:

Property	Value
Resource group	<a href="#">coronaserver_group</a>
Operating system	Windows (Windows Server 2022 Datacenter Azure Edition)
Status	Running
Size	Standard E2s v3 (2 vcpus, 16 GiB memory)
Location	Central US (Zone 1)
Public IP address	<a href="#">52.165.89.23</a>
Subscription	<a href="#">Azure subscription 1</a>
Subscription ID	73a90a63-52f5-4d69-ae73-1f24f5aa818d
Virtual network/subnet	<a href="#">coronaserver-vnet/default</a>
DNS name	<a href="#">Not configured</a>
Availability zone	
Health state	

# INSTALL CORONA ON THE VIRTUAL MACHINE

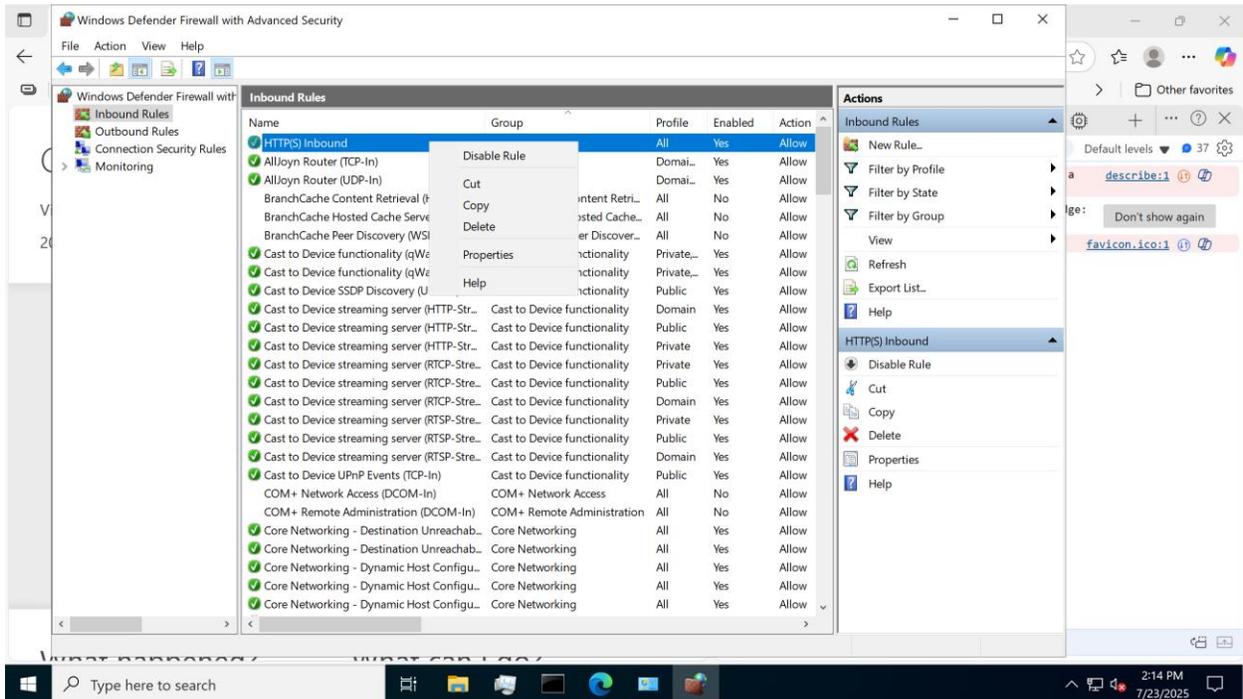
On the virtual machine you just created,

1. go to the <https://countryvideogames.com/corona>.
2. Download the zip file containing coronaserver.msi and setup.exe



# OPEN INBOUND 443 AND 80 WITH FIREWALL

1. OPEN THE PORT ON AZURE NETWORKING
2. OPEN THE PORT ON THE VM ITSELF

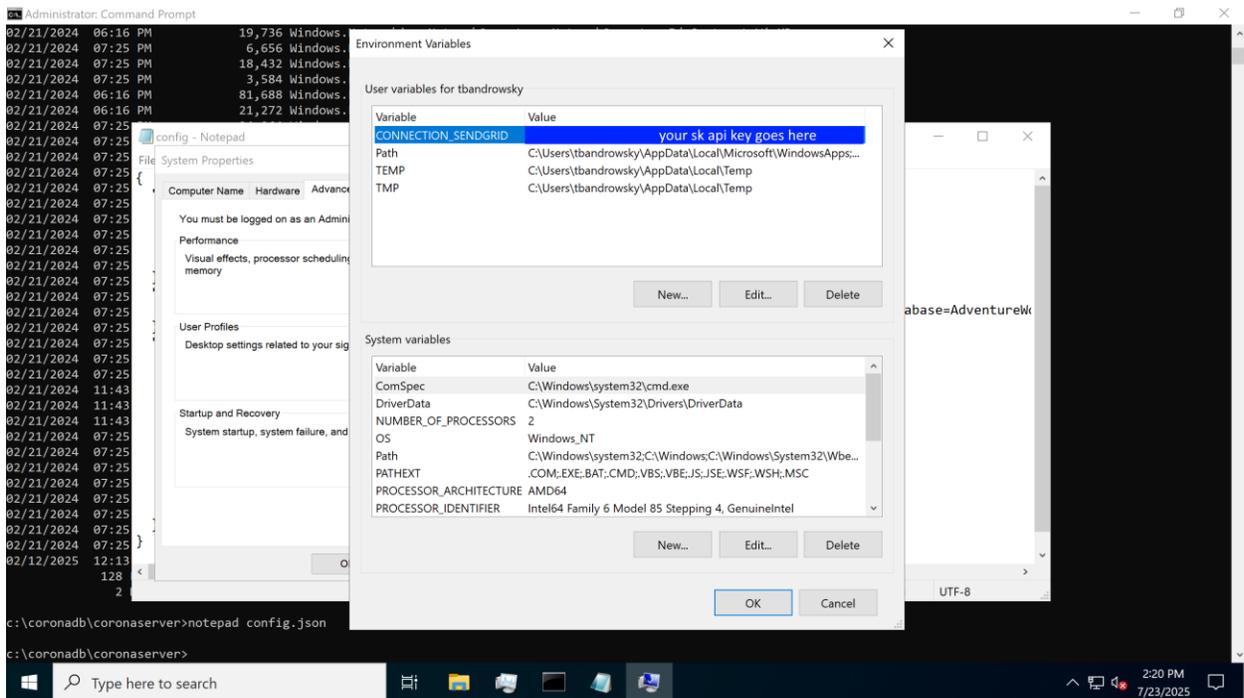


# CONFIGURE SENDGRID ENVIRONMENT VARIABLE

CORONA has the MFA ceremony built in for login, account creation and secret code send. Email rather than mobile is used, for the moment, although plans are there to support both.

CORONA will check the environment variable first for the SendGrid secret.

SendGrid is used to send emails.



# EXAMINE THE SERVER DIRECTORY

NOTICE THERE ARE TWO SETS OF .JSON FILES.

CONFIG.JSON – DATABASE SERVER CONFIGURATION

\*\_SCHEMA.JSON – A SCHEMA REFERENCED BY CONFIG.JSON. ALSO NOTE THAT DATABASE FILES CAN BE PLACED ANYWHERE.

CORONA SUPPORTS ONE DATABASE AT A TIME.

```
Administrator: Command Prompt
02/21/2024 07:25 PM 4,096 Windows.UI.Shell.SecurityAppManagerContract.winmd
02/21/2024 07:25 PM 10,752 Windows.UI.Shell.WindowTabManagerContract.winmd
02/21/2024 07:25 PM 12,288 Windows.UI.UIAutomation.UIAutomationContract.winmd
02/21/2024 07:25 PM 3,584 Windows.UI.ViewManagement.ViewManagementViewScalingContract.winmd
02/21/2024 07:25 PM 10,752 Windows.UI.WebUI.Core.WebUICommandBarContract.winmd
02/21/2024 07:25 PM 102,400 Windows.UI.Xaml.Core.Direct.XamlDirectContract.winmd
02/21/2024 07:25 PM 13,312 Windows.UI.Xaml.Hosting.HostingContract.winmd
02/21/2024 07:25 PM 9,728 Windows.Web.Http.Diagnostics.HttpDiagnosticsContract.winmd
02/12/2025 12:13 PM 1,245,240 WINHTTP.dll
128 File(s) 26,865,539 bytes
2 Dir(s) 84,255,916,032 bytes free

c:\coronadb\coronaserver>notepad config.json

c:\coronadb\coronaserver>dir *.json
Volume in drive C is Windows
Volume Serial Number is A004-E588

Directory of c:\coronadb\coronaserver

07/23/2025 02:19 PM 868 config.json
02/02/2025 12:01 PM 34,001 corona.postman_collection.json
02/02/2025 12:01 PM 16,503 country_village.json
07/19/2025 11:04 AM 4,340 insurance_schema.json
4 File(s) 55,712 bytes
0 Dir(s) 84,256,078,592 bytes free

c:\coronadb\coronaserver>dir *.exe
Volume in drive C is Windows
Volume Serial Number is A004-E588

Directory of c:\coronadb\coronaserver

07/23/2025 01:30 PM 1,518,080 coronaserver.exe
07/23/2025 02:00 PM 549,888 setup.exe
2 File(s) 2,067,968 bytes
0 Dir(s) 84,252,684,288 bytes free

c:\coronadb\coronaserver>notepad insurance_schema.json

c:\coronadb\coronaserver>
```

# CONFIGURE CONFIG.JSON

SenderName, SenderEmail, and UserConfirmationTitle are used to create the body of the email(s) being sent. Change those to you.

Server Listen\_Point is the bind to. The port in this URL *MUST* be specified, even if 80/443.

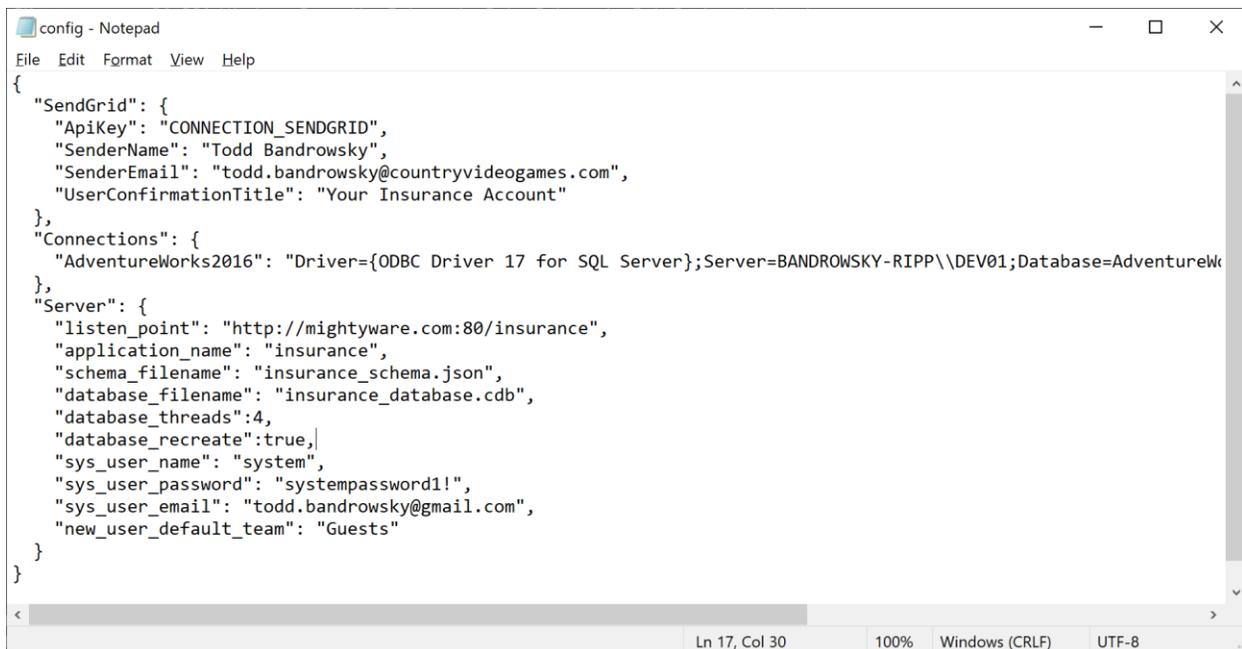
Application\_Name is your friendliest application name.

Schema\_filename says where your schema file lives.

Database\_Filename where you want the database file itself to live .

Database\_threads how many server threads to devote to corona.

Sys\_user\_name and sys\_user\_password, etc, are all the SA stuff, so like, you'll want to change all this

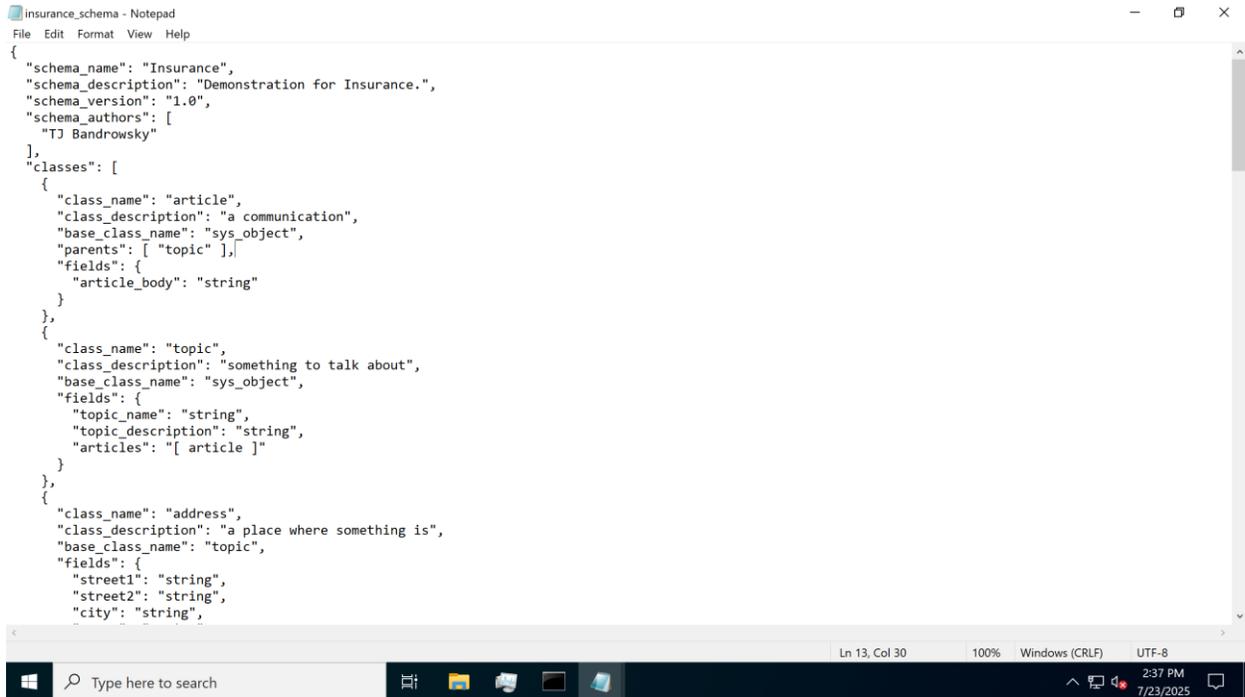


```
{
  "SendGrid": {
    "ApiKey": "CONNECTION_SENDGRID",
    "SenderName": "Todd Bandrowsky",
    "SenderEmail": "todd.bandrowsky@countryvideogames.com",
    "UserConfirmationTitle": "Your Insurance Account"
  },
  "Connections": {
    "AdventureWorks2016": "Driver={ODBC Driver 17 for SQL Server};Server=BANDROWSKY-RIPP\\DEV01;Database=AdventureWk"
  },
  "Server": {
    "listen_point": "http://mightyware.com:80/insurance",
    "application_name": "insurance",
    "schema_filename": "insurance_schema.json",
    "database_filename": "insurance_database.cdb",
    "database_threads": 4,
    "database_recreate": true,
    "sys_user_name": "system",
    "sys_user_password": "systempassword1!",
    "sys_user_email": "todd.bandrowsky@gmail.com",
    "new_user_default_team": "Guests"
  }
}
```

# CONFIGURE SCHEMA

A Schema in Corona consists of class definitions, and data definitions. Schemas are named and stored in the database.

Changes to the Schema are applied immediately, even when Corona is running.



```
insurance_schema - Notepad
File Edit Format View Help
{
  "schema_name": "Insurance",
  "schema_description": "Demonstration for Insurance.",
  "schema_version": "1.0",
  "schema_authors": [
    "TJ Bandrowsky"
  ],
  "classes": [
    {
      "class_name": "article",
      "class_description": "a communication",
      "base_class_name": "sys_object",
      "parents": [ "topic" ],
      "fields": {
        "article_body": "string"
      }
    },
    {
      "class_name": "topic",
      "class_description": "something to talk about",
      "base_class_name": "sys_object",
      "fields": {
        "topic_name": "string",
        "topic_description": "string",
        "articles": "[ article ]"
      }
    },
    {
      "class_name": "address",
      "class_description": "a place where something is",
      "base_class_name": "topic",
      "fields": {
        "street1": "string",
        "street2": "string",
        "city": "string",

```

# CONSIDER CLOUDFLARE

The screenshot shows the Cloudflare dashboard for the account 'Todd.bandrowsky@gmail.com'. The 'Domains' tab is selected, displaying a list of domains. The table includes columns for Domain, Status, Security insights, Unique visitors, and Plan. Two domains are listed: 'countrybit.com' and 'mightyware.com', both with 'Active' status and 'Free' plans. The 'Unique visitors' column shows line graphs and values of 703 and 1.41k respectively. A 'Filter by: Starred' option is visible above the table. The sidebar on the left contains navigation options like 'Account Home', 'Discover', 'Domain Registration', 'Log Explorer', 'Analytics & Logs', 'Security Center', 'Trace', 'WAF', and 'Turnstile'. The top navigation bar includes a search bar, 'Support', '+ Add', and language settings.

Domain	Status	Security insights	Unique visitors	Plan
countrybit.com	Active	Enable	703	Free
mightyware.com	Active	Enable	1.41k	Free

# LAUNCH CORONA

CD to your Corona directory.

Open Command Prompt

**./coronaserver console**

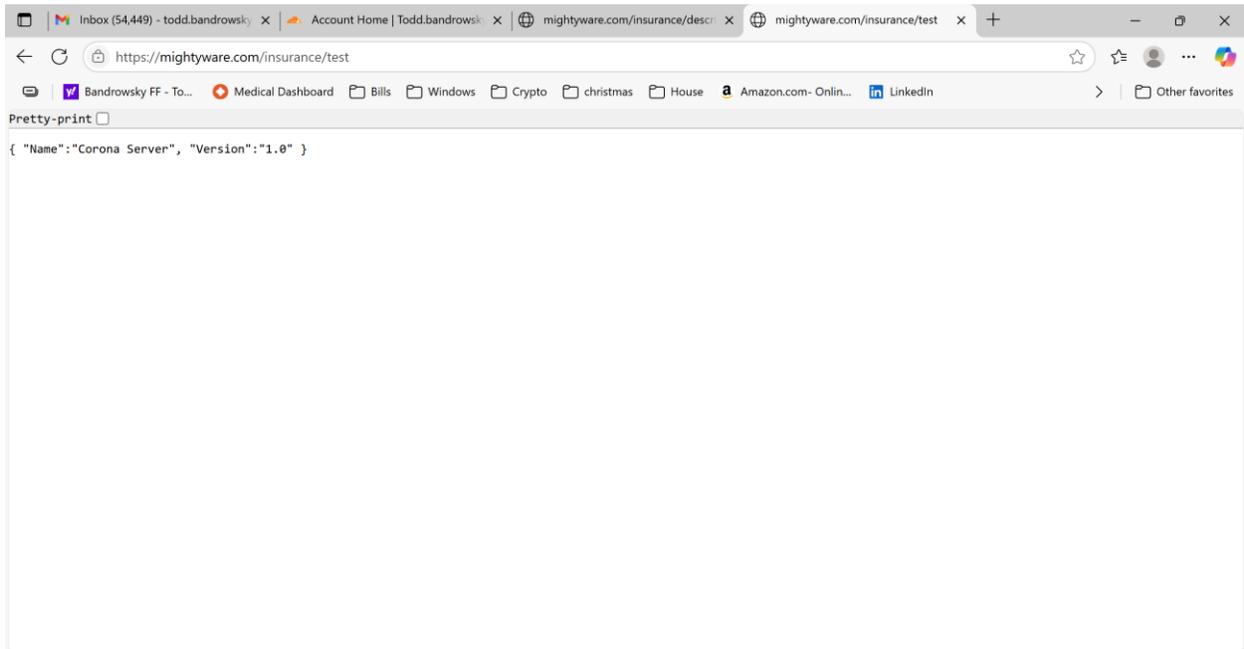
End points appear in list.

```
Administrator: Command Prompt - coronaserver console
Users          Users          1824    07/23/25 14:41 start    corona-database-engine.hpp 5660
Users          Users          1824    0.0021165 secs corona-database-engine.hpp 5676
DataSets       DataSets       1824    07/23/25 14:41 start    corona-database-engine.hpp 5683
DataSets       DataSets       1824    0.0016667 secs corona-database-engine.hpp 5893
DataSets       create_object  start     1824    07/23/25 14:41 start    corona-database-engine.hpp 7093
DataSets       create_object  complete  1824    0.0017106 secs corona-database-engine.hpp 7183
DataSets       put_object     start     1824    07/23/25 14:41 start    corona-database-engine.hpp 7209
DataSets       put_object     complete  1824    0.3354459 secs corona-database-engine.hpp 7265
DataSets       Object(s) created
DataSets       Object(s) created 1824
DataSets       schema applied 1824    4.214448 secs corona-database-engine.hpp 5920
poll_db        schema applied 1824    4.2188787 secs corona-comm-service-bus.hpp 442
poll_db        apply config   1824    07/23/25 14:41 start    corona-comm-service-bus.hpp 447
poll_db        apply_config  start     1824    07/23/25 14:41 start    corona-database-engine.hpp 5405
poll_db        apply_config  complete  1824    0.0012264 secs corona-database-engine.hpp 5447
poll_db        config applied 1824    0.0054486 secs corona-comm-service-bus.hpp 449
poll_db        updated server, and listening on :http://mightyware.com:80/insurance
poll_db        test/         1824
poll_db        login/createuser/ 1824
poll_db        login/loginuser/ 1824
poll_db        login/confirmuser/ 1824
poll_db        login/senduser/ 1824
poll_db        login/passworduser/ 1824
poll_db        classes/get/ 1824
poll_db        classes/get/details/ 1824
poll_db        classes/put/ 1824
poll_db        objects/get/ 1824
poll_db        objects/query/ 1824
poll_db        objects/create/ 1824
poll_db        objects/put/ 1824
poll_db        objects/delete/ 1824
poll_db        objects/edit/ 1824
poll_db        objects/run/ 1824
poll_db        objects/copy/ 1824
poll_db        describe/ 1824
login_user    start         1824    07/23/25 14:41 start    corona-database-engine.hpp 6417
login_user    complete     1824    0.0017535 secs corona-database-engine.hpp 6492
update        start         1824    07/23/25 14:41 start    corona-database-engine.hpp 6824
update        complete     1824    0.0014277 secs corona-database-engine.hpp 6877
query         complete     1824    3.57e-05 secs corona-database-engine.hpp 7073
```

# QUICK TEST.

First off, you can see the URLs for any of the end points in the console window.

<http://mightyware.com/insurance/test>



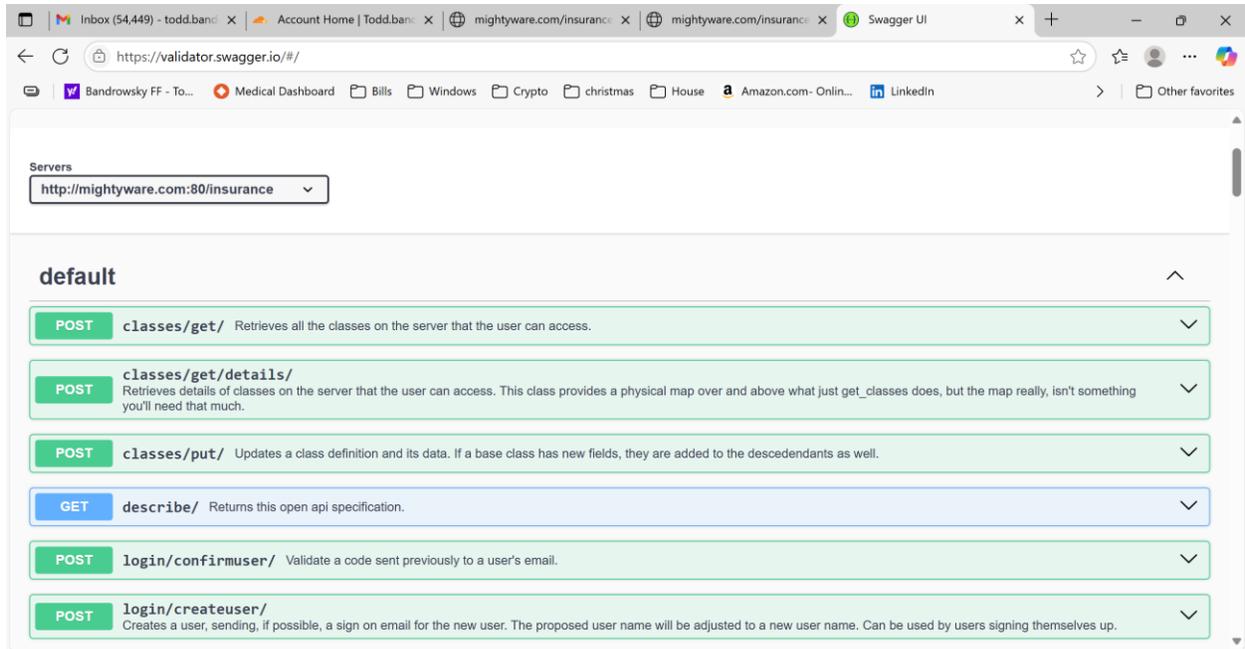
# OPEN API CALL

<http://mightyware.com/insurance/describe>

```
{ "components": { "schemas": { "address": { "description": "a place where something is", "properties": { "articles": { "type": "array" }, "city": { "type": "string" }, "class_name": { "type": "string" }, "country": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "object_id": { "type": "number" }, "state": { "type": "string" }, "street1": { "type": "string" }, "street2": { "type": "string" }, "topic_description": { "type": "string" }, "topic_name": { "type": "string" }, "updated": { "type": "string" }, "updated_by": { "type": "string" }, "zip": { "type": "string" } }, "type": "object" }, "article": { "description": "a communication", "properties": { "article_body": { "type": "string" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "object_id": { "type": "number" }, "topic": { "type": "number" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "asset_account": { "description": "a comment on an article", "properties": { "account_name": { "type": "string" }, "account_number": { "type": "string" }, "accounts": { "type": "array" }, "balance": { "type": "number" }, "base_account": { "type": "number" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "credits": { "type": "number" }, "debits": { "type": "number" }, "object_id": { "type": "number" }, "transactions": { "type": "array" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "base_account": { "description": "an entity which can be posted to", "properties": { "account_name": { "type": "string" }, "account_number": { "type": "string" }, "accounts": { "type": "array" }, "balance": { "type": "number" }, "base_account": { "type": "number" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "credits": { "type": "number" }, "debits": { "type": "number" }, "object_id": { "type": "number" }, "transactions": { "type": "array" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "contact": { "description": "a someone", "properties": { "articles": { "type": "array" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "current_email": { "type": "string" }, "first_name": { "type": "string" }, "last_name": { "type": "string" }, "location": { "type": "number" }, "mobile_phone": { "type": "string" }, "object_id": { "type": "number" }, "other_email": { "type": "string" }, "other_phone": { "type": "string" }, "topic_description": { "type": "string" }, "topic_name": { "type": "string" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "customer": { "description": "a comment on an article", "properties": { "articles": { "type": "array" }, "class_name": { "type": "string" }, "contacts": { "type": "array" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "locations": { "type": "array" }, "object_id": { "type": "number" }, "topic_description": { "type": "string" }, "topic_name": { "type": "string" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "employees": { "description": "a comment on an article", "properties": { "articles": { "type": "array" }, "class_name": { "type": "string" }, "contacts": { "type": "array" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "locations": { "type": "array" }, "object_id": { "type": "number" }, "topic_description": { "type": "string" }, "topic_name": { "type": "string" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "equity_account": { "description": "a comment on an article", "properties": { "account_name": { "type": "string" }, "account_number": { "type": "string" }, "accounts": { "type": "array" }, "balance": { "type": "number" }, "base_account": { "type": "number" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "credits": { "type": "number" }, "debits": { "type": "number" }, "object_id": { "type": "number" }, "transactions": { "type": "array" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "general_ledger_account": { "description": "a comment on an article", "properties": { "account_name": { "type": "string" }, "account_number": { "type": "string" }, "accounts": { "type": "array" }, "balance": { "type": "number" }, "base_account": { "type": "number" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "credits": { "type": "number" }, "debits": { "type": "number" }, "object_id": { "type": "number" }, "transactions": { "type": "array" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "liability_account": { "description": "a comment on an article", "properties": { "account_name": { "type": "string" }, "account_number": { "type": "string" }, "accounts": { "type": "array" }, "balance": { "type": "number" }, "base_account": { "type": "number" }, "class_name": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "credits": { "type": "number" }, "debits": { "type": "number" }, "object_id": { "type": "number" }, "transactions": { "type": "array" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" }, "location": { "description": "a place where people are", "properties": { "articles": { "type": "array" }, "city": { "type": "string" }, "class_name": { "type": "string" }, "contacts": { "type": "array" }, "country": { "type": "string" }, "created": { "type": "string" }, "created_by": { "type": "string" }, "object_id": { "type": "number" }, "partner": { "type": "number" }, "state": { "type": "string" }, "street1": { "type": "string" }, "street2": { "type": "string" }, "topic_description": { "type": "string" }, "topic_name": { "type": "string" }, "updated": { "type": "string" }, "updated_by": { "type": "string" } }, "type": "object" } } }
```

# FIRE UP OPENAPI

You'll give your url to open API to try this out.



The screenshot shows a web browser window with the Swagger UI interface. The browser's address bar displays `https://validator.swagger.io/#/`. The page title is "Swagger UI". The "Servers" dropdown menu is set to `http://mightyware.com:80/insurance`. The "default" server is expanded, showing a list of API endpoints:

- POST** `classes/get/` Retrieves all the classes on the server that the user can access.
- POST** `classes/get/details/` Retrieves details of classes on the server that the user can access. This class provides a physical map over and above what just `get_classes` does, but the map really, isn't something you'll need that much.
- POST** `classes/put/` Updates a class definition and its data. If a base class has new fields, they are added to the descendants as well.
- GET** `describe/` Returns this open api specification.
- POST** `login/confirmuser/` Validate a code sent previously to a user's email.
- POST** `login/createuser/` Creates a user, sending, if possible, a sign on email for the new user. The proposed user name will be adjusted to a new user name. Can be used by users signing themselves up.

