



# SALT Maintenance Guide

# Table of Contents (Part 1)

---

- ❖ How to Test the Application's Functionality in XAMPP
  - ❖ How to Download XAMPP, PHP 7.3 and MongoDB
  - ❖ How to Download Composer
- ❖ How to Upload the Application to Azure
  - ❖ How to Download Git and Upload the Website to Azure
  - ❖ How to Update the Repository in Azure
  - ❖ How to Upload the Database to Azure
  - ❖ Installing Mongodb.dll on Azure's Server and Connecting the Database to the Application
  - ❖ Installing the Mail Server on Azure

# Table of Contents (Part 2)

---

- ❖ How to Install the Data Scraper on your machine
  - ❖ How to Run a Python Server using your local host
- ❖ How to Create Inbound Rules
- ❖ Database Table Names and Contents
- ❖ Required PHP Statements
- ❖ Process Maps/Data Flow

# How to Test the Application's Functionality in XAMPP, Pt. 1

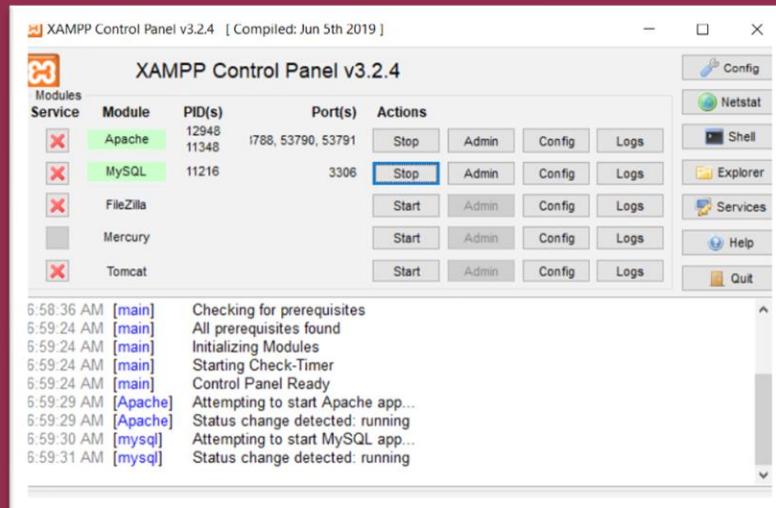


## How to Download XAMPP

- It's a good idea to run your code on a test server before you push it to the live site. To create a test environment, download XAMPP from here: <https://www.apachefriends.org/download.html>
- Once the executable has downloaded, run the installation wizard and select all the default options. Afterword, start the XAMPP server by opening the XAMPP Control Panel in C:\xampp.
- If you are not able to access the XAMPP Control Panel, right click on the Control Panel, click the Properties button, go to the Compatibility tab at the tab at the top check the "Run this Program as an Administrator" box and press apply, like so: <https://www.youtube.com/watch?v=MmuXI3P5aBM>

# How to Test the Application's Functionality in XAMPP, Pt. 2

- Once you are able to open the XAMPP Control Panel, click star the server by pressing “start” next to the Apache and MySQL modules like so:



# How to Test the Application's Functionality in XAMPP, Pt. 3



## How to Download MongoDB

- Download the Mongo server to your local machine if you haven't already. make sure you can log in and everything.
- Do what this video says, but stop after 2:30 because the rest of the things he does are outdated and won't work: <https://www.youtube.com/watch?v=p4QMTS9IFFg>

# How to Test the Application's Functionality in XAMPP, Pt. 4



## How to Download PHP 7.3 (thread safe x86)

- Download this exact runtime stack from PECL as a zip file:

<https://pecl.php.net/package/mongodb/1.7.4/windows>

- Unzip only the .dll file from this file, then put it in C:xampp\php\ext
- Press back. Now that you're in C:xampp>php, look for the "php.ini-developer" file and rename it php.ini [if it hasn't been renamed already]
- Add the php string where he tells you to add it, but add "db" to the end of mongo so it matches the exact name of the file you dragged into the "ext" folder [php\_mongodb.dll]. There's been an update to the php extension since this video was released.

# How to Test the Application's Functionality in XAMPP, Pt. 5



- Add xampp php to the PATH environment variable [Control Panel\System and Security\System > Env Variables], then double click the Path under "system variables". Click new, then copy and paste the path to the xampp php file. Step 5 in this document shows how to do this better:  
[\[https://www.thegeekstuff.com/2015/10/php-mongodb-for-xampp/\]](https://www.thegeekstuff.com/2015/10/php-mongodb-for-xampp/)
- To verify, type localhost into your browser, click on phpinfo in the right corner, ctrl+f mongo and make sure there's approx. 20 different instances of mongo services running. You can see what I'm talking about if you go to step 6 in the geekstuff doc.

# How to Test the Application's Functionality in XAMPP, Pt. 6



## How to Install Composer

- Download composer from here: <https://getcomposer.org/download/>
- Next, go to the xampp folder in your c drive, then click on the htdocs folder. Create a new folder inside of it.

This is where you will place the html and PHP files you want to connect to the db.

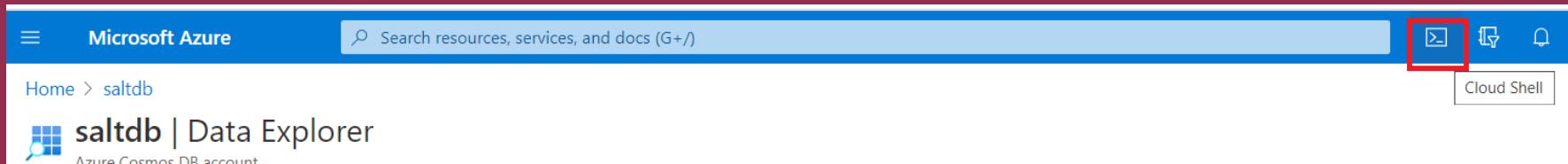
- open Powershell and navigate to the folder you just made [the commands are the same as any CLI by the way, so type: cd C:\xampp\htdocs\NewFolder]
- Type: composer require mongodb/mongodb
- Press enter and wait for PowerShell to load. It may take a few minutes. Once it finishes, your test environment will be completely ready to use.

# How to Upload the Application to Azure, Pt. 1

---

## How to Set Up Your Environment on Azure

- Go to Azure's website and create an account.
- Once your account has been created, click on the cloud shell next to the search bar



# How to Upload the Application to Azure, Pt. 2

---

Enter the following commands into the Cloud Shell

- az webapp deployment user set --user-name <username> --password <password>
- az group create --name myResourceGroup --location "West US"
- az appservice plan create --name myAppServicePlan --resource-group myResourceGroup --sku FREE
- az webapp create --resource-group myResourceGroup --plan myAppServicePlan --name phpsalt --runtime "PHP|7.3" --deployment-local-git
- grab the link they give you after the deployment finishes. mine was:

<https://ttufford@phpsalt.scm.azurewebsites.net/phpsalt.git>

# How to Upload the Application to Azure, Pt. 3

---

## How to Install & Use Git to Push the Application to Azure

- Minimize Azure and Download Git from here: <https://git-scm.com/download/win>
- Open up the command prompt on your local machine, then navigate to your htdocs folder from within the CLI: cd C:\xampp\htdocs\composer\saltphp
- type the following
- git init
- git remote add azure https://ttufford@phpsalt.scm.azurewebsites.net/phpsalt.git
- git commit -m 'message'
- git push azure master
- [takes a long time to copy everything over]

# How to Upload the Application to Azure, Pt. 4

---

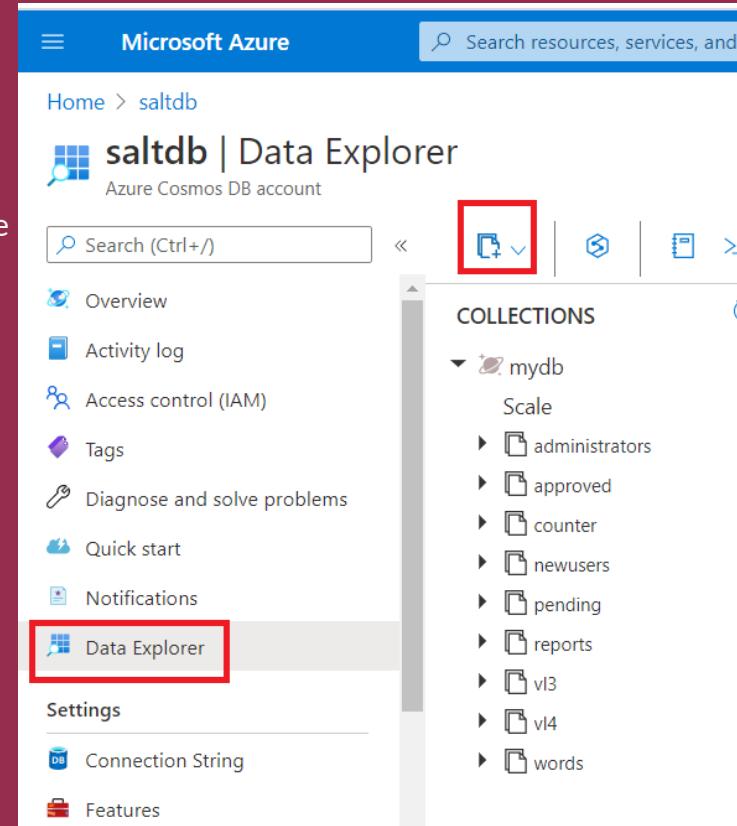
## How to Update the Repository through Git

- to update, open the CLI and type:
- cd C:\xampp\htdocs\composer\saltphp
- git commit -am "updated output"
- git push azure master
- ..sometimes you have to type: git add --all
- ..you also may have to force if it gives you that yellow msg: git push -f azure master

# How to Upload the Application to Azure, Pt. 5

## How to Upload the Database

- First, deploy Azure's Cosmos DB inside of Azure
- Next, open it up and go to the Data Explorer on the left side and create a new database called mydb
- Then make the following collections (tables) with “Fixed” capacity:
  - approved
  - Counter
  - Newusers
  - Vi3
  - Vi4
  - words



# How to Upload the Application to Azure, Pt. 6

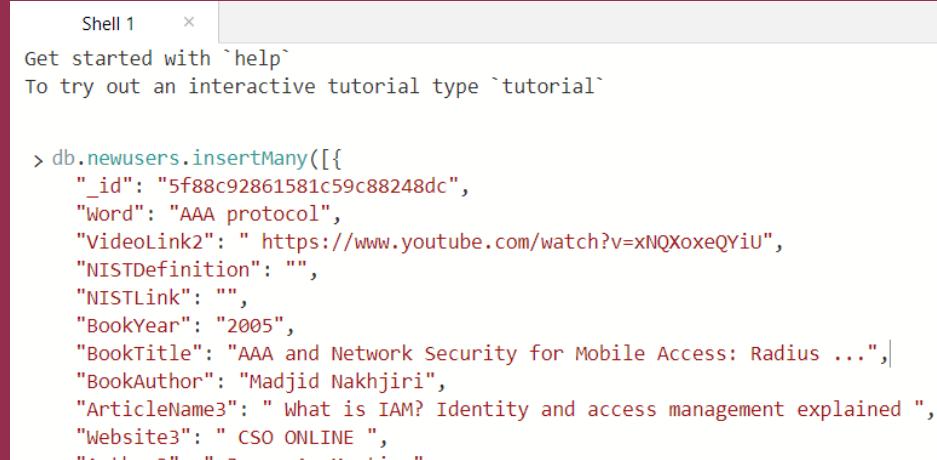
## How to get the Data

- Open all the json docs in the “database” folder inside

C:\xampp\htdocs

## How to Migrate the Data

- Click on a table, then click on the Mongo Shell button
- Type db.<tablename>.insertMany()
- Then replace <tablename> with the name of the table, but don't press enter yet
- In between the parentheses of the db.<tablename>.insertMany() statement, then press enter
- Repeat the process until all the collections are full

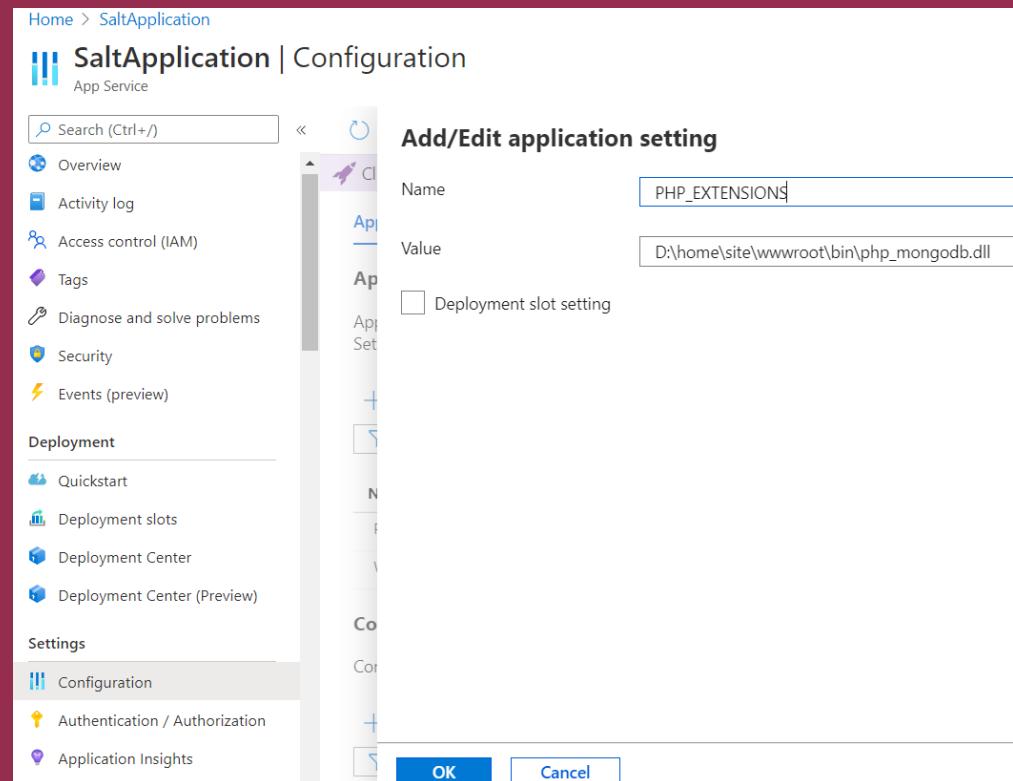


```
Shell 1 x
Get started with `help`
To try out an interactive tutorial type `tutorial`  
  
> db.newusers.insertMany([{
    "_id": "5f88c92861581c59c88248dc",
    "Word": "AAA protocol",
    "VideoLink2": " https://www.youtube.com/watch?v=xNQXoxeQyiu",
    "NISTDefinition": "",
    "NISTLink": "",
    "BookYear": "2005",
    "BookTitle": "AAA and Network Security for Mobile Access: Radius ...",
    "BookAuthor": "Madjid Nakjiri",
    "ArticleName3": " What is IAM? Identity and access management explained ",
    "Website3": " CSO ONLINE ",
    "ArticleName4": " What is IAM? Identity and access management explained "
}] )
```

# How to Upload the Application to Azure, Pt. 7

## Installing Mongodb.dll

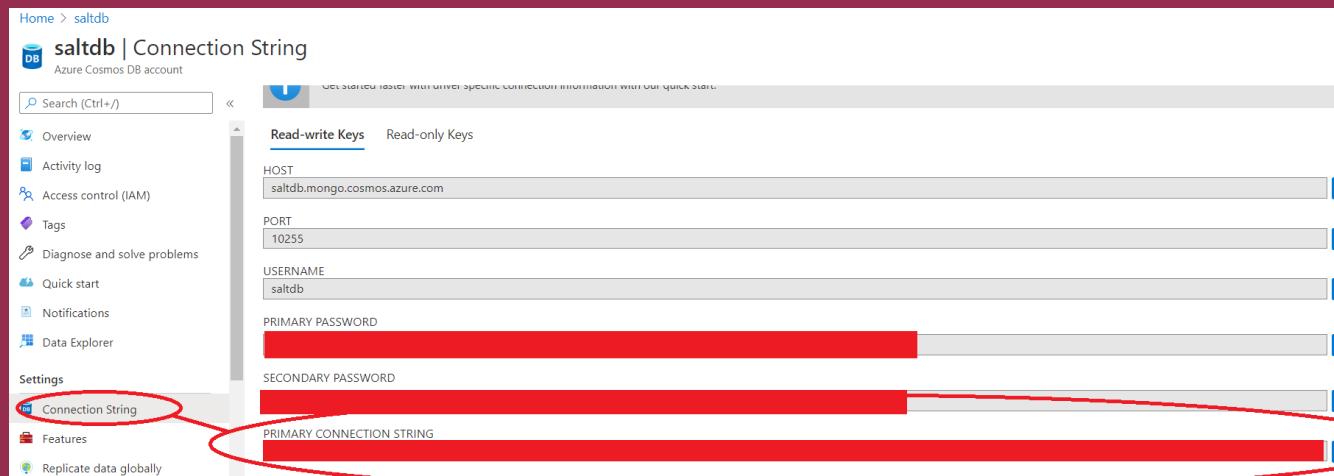
- To install the driver, open your application service manager in Azure, then "click Configuration" on the left side.
- Click new Application service and type:
- Name: PHP\_EXTENSIONS
- Value: D:\home\site\ext\php\_mongodb.dll



# How to Upload the Application to Azure, Pt. 8

## Connect the Database to the Application

- Navigate to CosmosDB in Azure and click the Connection String tab on the left side of the page.
- Copy the value beneath the Primary Connection String header



# How to Upload the Application to Azure, Pt. 8

---

## Add the Connection String to Your Application

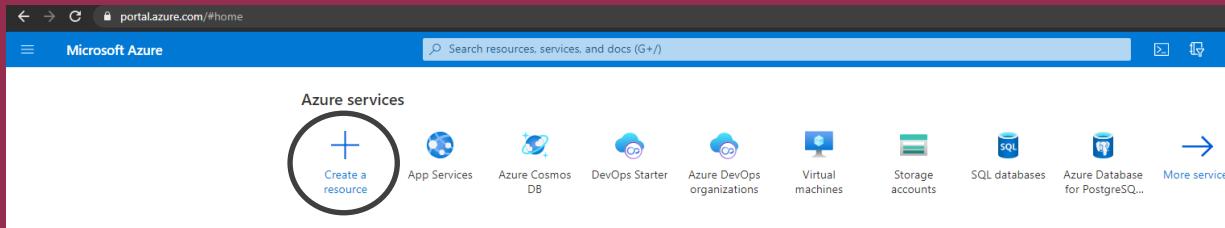
- Open up 'dbconnect.php' in cd C:\xampp\htdocs\yourfolder
- Modify line 6 by pasting the Connection String in between the single quotes then press save.
- Update the git repository and test the site now; it should work.

```
1 <?php
2 require_once __DIR__ . "/vendor/autoload.php";
3
4 //db connection
5 $manager = new MongoDB\Client(
6   ''
7 )
```

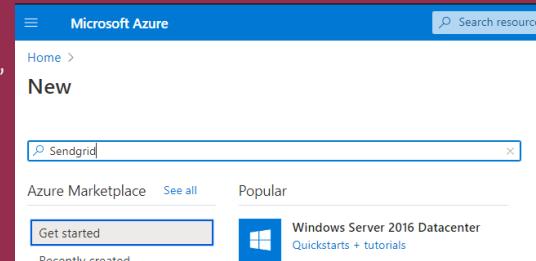
# How to Upload the Application to Azure, Pt. 9

## Install the Mail Server, Pt. 1

- Click on create a resource on the Azure dashboard



- This will take you to a page that lets you search for services you wish to add.



# How to Upload the Application to Azure, Pt. 10

## Install the Mail Server, Pt. 2

- A new page will load. Hover over the “Create” box in the “SendGrid” square.
  - A list of tiers will show up, click on the free tier.  
\*Free tier allows for 25,000 emails to be sent a month for free.
- After selecting the free tier, a form will appear.

Create SendGrid Account

SentGrid

Basics Tags Review + Create

Configure your SendGrid Account to deliver customer communication that drives engagement and growth using the cloud. [Learn more](#)

Project details

Subscription \*

Resource group \*  [Create new](#)

Location \*

Account details

Name \*

Password \*

Confirm password \*

Pricing Tier

**Free** 25,000 emails/month [Change plan](#)

Contact details

The provided information will be used as contact info for support agents / technical contacts.

First Name \*

Last Name \*

Email \*

Company \*

Website \*

[Review + Create](#) [Previous](#) [Next: Tags >](#)

Microsoft Azure

Search resources, services, and docs (G+)

Home > New > Marketplace

Private Marketplace (PREVIEW)

Pricing : All Operating System : All

Showing All Results

SendGrid Twilio SendGrid Eliminate your email headaches and save money. The email infrastructure of tomorrow, delivered today

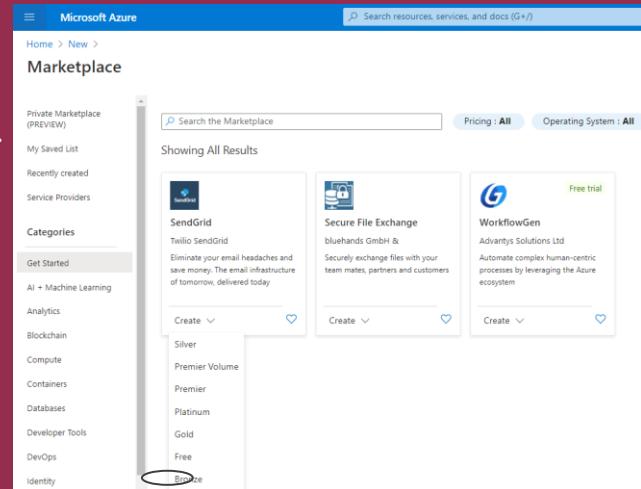
Secure File Exchange bluehands GmbH & Securely exchange files with your team mates, partners and customers

WorkflowGen Advantys Solutions Ltd Automate complex human-centric processes by leveraging the Azure ecosystem

Categories

Get Started AI + Machine Learning Analytics Blockchain Compute Containers Databases Developer Tools DevOps Identity

Create Silver Premier Volume Premier Platinum Gold Free Bronze



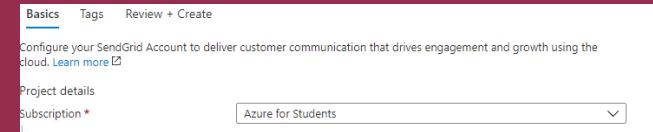
# How to Upload the Application to Azure, Pt. 11

## Install the Mail Server, Pt. 3

- Fill in the form

- The subscription field will automatically be filled with your current subscription.
- For resource group, click on the drop down and select the resource group that was created in the process earlier.

A screenshot of the Azure portal's 'Create a new resource' page. The 'Subscription' dropdown is set to 'Azure for Students'. The 'Resource group' dropdown is open, showing 'myResourceGroup' as the selected item. The 'Location' dropdown shows 'cloud-shell-storage-westeurope'. The 'Account details' section at the bottom has a circled 'myResourceGroup' link.



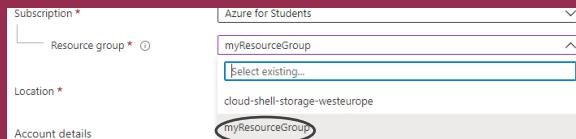
- For location, click on the input box then search for "(US) West US", then click on it.

A screenshot of the Azure portal's 'Create a new resource' page. The 'Location' dropdown is open, showing '(US) West US' as the selected item. Other options listed include '(Asia Pacific) Southeast Asia', '(Europe) North Europe', '(Europe) UK South', '(Europe) West Europe', '(US) Central US', '(US) North Central US', '(Africa) South Africa North', and '(US) West US' (which is circled).

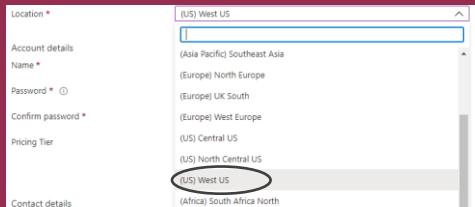
# How to Upload the Application to Azure, Pt. 12

## Install the Mail Server, Pt. 4

- Fill in the form continued
  - The subscription field will automatically be filled with your current subscription.
  - For resource group, click on the drop down and select the resource group that was created in the process earlier.



- For location, click on the input box then search for "(US) West US", then click on it.



# How to Upload the Application to Azure, Pt. 13

## Install the Mail Server, Pt. 5

- Fill in the form continued

- For the Account Details portion, fill in your name and create a password. This password will be used for the SendGrid service.
- The last portion of this form is contact details. This information will be used by SendGrid support agents if you ever run into trouble with the service.

The screenshot shows the 'Account details' section of a web form. It includes fields for Name, Password, Confirm password, and Pricing Tier. The Name field contains 'Issa Yacoub'. The Password and Confirm password fields both contain masked text. The Pricing Tier section shows 'Free' with '25,000 email/month' and a 'Change plan' link.

Account details	
Name *	Issa Yacoub
Password *	*****
Confirm password *	*****
Pricing Tier	Free 25,000 email/month <a href="#">Change plan</a>

The screenshot shows the 'Contact details' section of a web form. It includes fields for First Name, Last Name, Email, Company, and Website. The First Name and Last Name fields contain 'Issa' and 'Yacoub' respectively. The Email field contains 'issay14@gmail.com'. The Company field contains 'Arizona State University'. The Website field is empty and has a validation error message: 'The value must not be empty.' Below the form are buttons for 'Review + Create', 'Previous', and 'Next: Tags >'.  
  
Contact details  
The provided information will be used as contact info for support agents / technical contacts.  
  
First Name \* Issa  
Last Name \* Yacoub  
Email \* issay14@gmail.com  
Company \* Arizona State University  
Website \*  
The value must not be empty.  
  
Review + Create Previous Next: Tags >

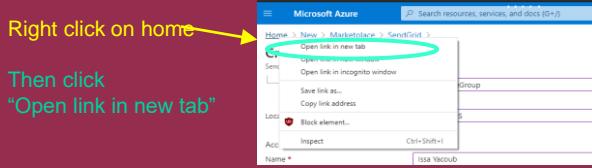
- For the website portion, you should add SALT's URL. Since we created the application in an earlier step, the URL can be accessed by opening the dashboard in a new tab and selecting app service.

# How to Upload the Application to Azure, Pt. 14

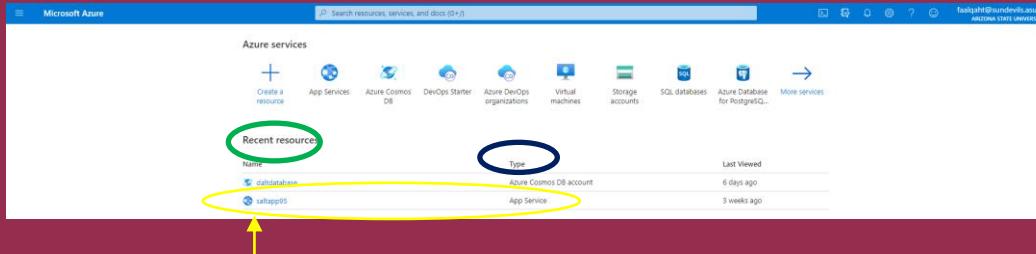
## Install the Mail Server, Pt. 6

- Fill in the form continued

- For the website portion, you should add SALT's URL. Since we created the application in an earlier step, the URL can be accessed by opening the dashboard in a new tab and selecting app service.



- The dashboard will open in a new tab. Look for “App Service” in “Recent resources” “Type” section



Click on your app  
service

# How to Upload the Application to Azure, Pt. 15

## Install the Mail Server, Pt. 7

- Fill in the form continued

- After clicking your app service, copy its URL.

The screenshot shows the Microsoft Azure portal interface. On the left, the 'saltapp95' App Service overview page is displayed. The URL field is highlighted with a green oval. On the right, a 'Contact details' form is overlaid, containing fields for First Name, Last Name, Email, Company, and Website. The 'Website' field is highlighted with a yellow box and contains the URL 'https://saltapp95.azurewebsites.net'. A red arrow points from the 'Website' field to the 'Review + Create' button at the bottom of the form.

Microsoft Azure

Search resources, services, and docs (G+)

saltapp95

App Service

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Security

Events (preview)

Resource group (change) : myResourceGroup

Status : Running

Location : West US

Subscription (change) : Azure for Students

Subscription ID : 559951da-b949-49c4-a6eb-ac1e4e6fadac

Tags (change) : Click here to add tags

URL : https://saltapp95.azurewebsites.net

App Service Plan : saltapp95

Git/Deployment username : faalqah

Git clone url : https://faalqah@saltapp95.scm.azurewebsites.net/saltapp95.git

FTP hostname : ftp://waws-prod-bay-039.ftp.azurewebsites.windows.net/site/wwwroot

Contact details

The provided information will be used as contact info for support agents / technical contacts.

First Name \* Issa

Last Name \* Yacoub

Email \* issay14@gmail.com

Company \* Arizona State University

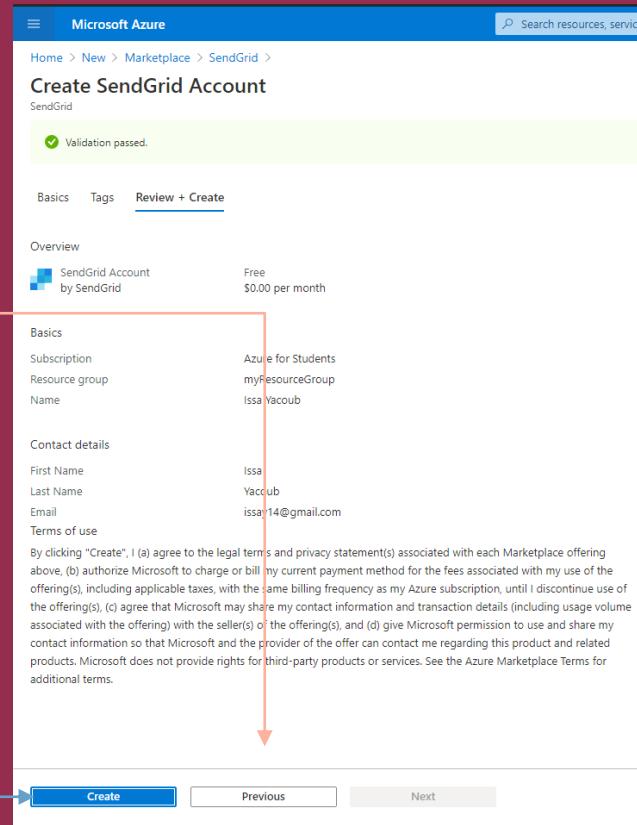
Website \* https://saltapp95.azurewebsites.net

Review + Create Previous Next: Tags >

# How to Upload the Application to Azure, Pt. 16

## Install the Mail Server, Pt. 8

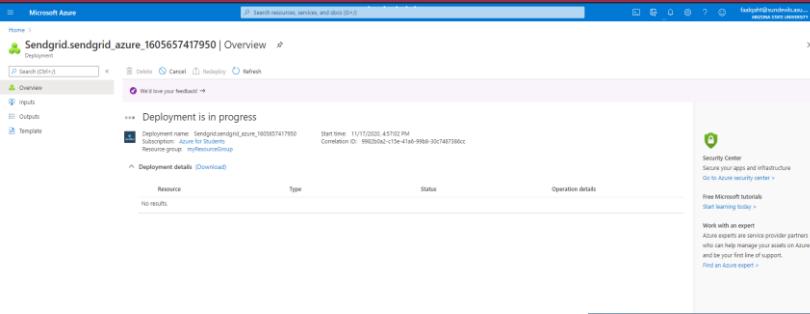
- Fill in the form continued
  - After clicking the “Review + Create” button, it will make you look over the form and make sure everything is correct.
  - If you made a mistake, click on “Previous” to edit the form
  - If everything looks good, press the blue “Create” button.



# How to Upload the Application to Azure, Pt. 17

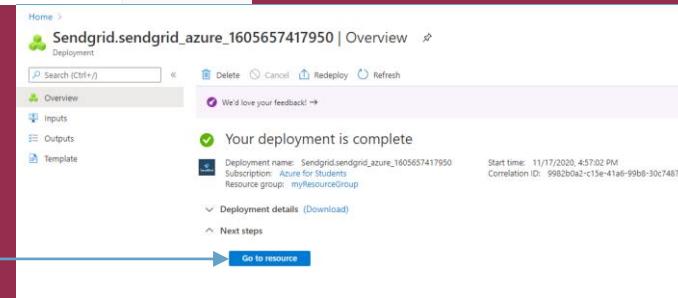
## Install the Mail Server, Pt. 9

- Fill in the form continued
  - After pressing “Create”, the SendGrid service will deploy. This process will take at least 5 minutes.



- The page will automatically update when it is finished deploying and will look like this:

- Click on the blue “Go to resource” button

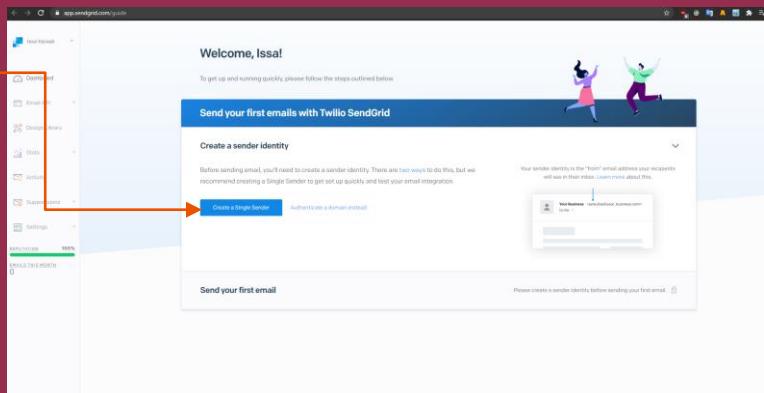
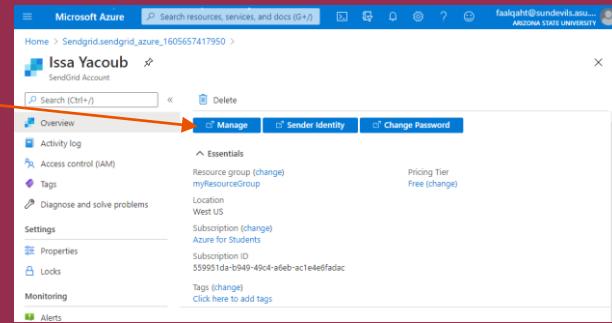


# How to Upload the Application to Azure, Pt. 18

## Install the Mail Server, Pt. 10

- Once you are on SendGrid's resource page, click on the blue “Manage button”

- This will take you to SendGrid's website. This is the first thing you will see:
- SendGrid asks you to create a sender email account that will be used to send out emails.
- To do this, click on “Create a Single Sender”



# How to Upload the Application to Azure, Pt. 19

## Install the Mail Server, Pt. 11

- This form will appear
  - From Name: The name that will appear with the email address in emails  
From Email Address: The email address that will send emails  
Reply To: The email address that will receive emails  
Company Address: Your company's address  
Company Address Line 2: A second line for your company's address if needed  
City: Your company's city  
State: Your company's state  
Zip Code: Your company's zip code  
Country: Your company's country  
Nickname: a nickname for this sender address that only you will be able to see.
- Fill out the form with your details.
  - You can create an email address with Gmail or other email sites. You can also use an existing email address you have. This email will be in charge of sending and receiving emails.
  - Make sure you have access to this email, because the next step will have you log in to that email account and verify that it is yours.
- After filling in the form, press the blue “Create” button.

**Create a Sender**

You are required to include your contact information, including a physical mailing address, inside every promotional email you send in order to comply with the anti-spam laws such as CAN-SPAM and CASL. You'll find replacement tags for this information in the footer of all the email designs SendGrid provides. [Learn more](#)

From Name •  
SALT Application

From Email Address •  
SaltEmailTesting1@gmail.com 

Attempting to send from a free email address domain like [gmail.com](#) is not recommended. [Learn more](#)

Reply To •  
SaltEmailTesting1@gmail.com

Company Address •  
7001 E Williams Field Rd

Company Address Line 2

City • Mesa State Arizona

Zip Code 85212 Country • United States

Nickname • SaltAppEmail

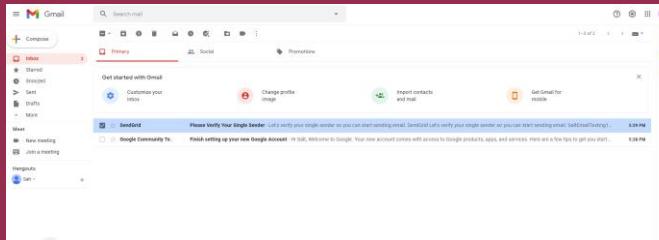
Nickname is for your reference only; this field will not be displayed to recipients.

[Cancel](#) **Create** 

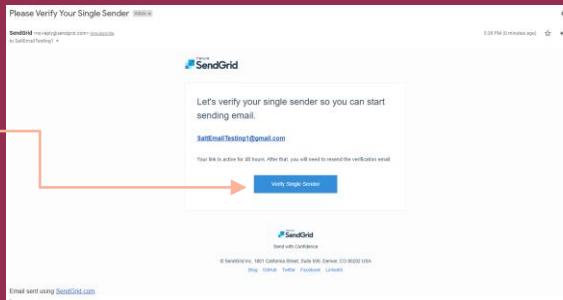
# How to Upload the Application to Azure, Pt. 20

## Install the Mail Server, Pt. 12

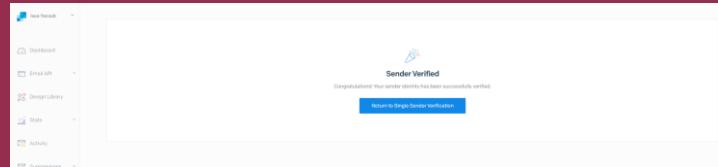
- After pressing “Create”, SendGrid will send a verification email to the email address that you inputted into the form



- Open the email then click on the “Verify Single Sender” button.



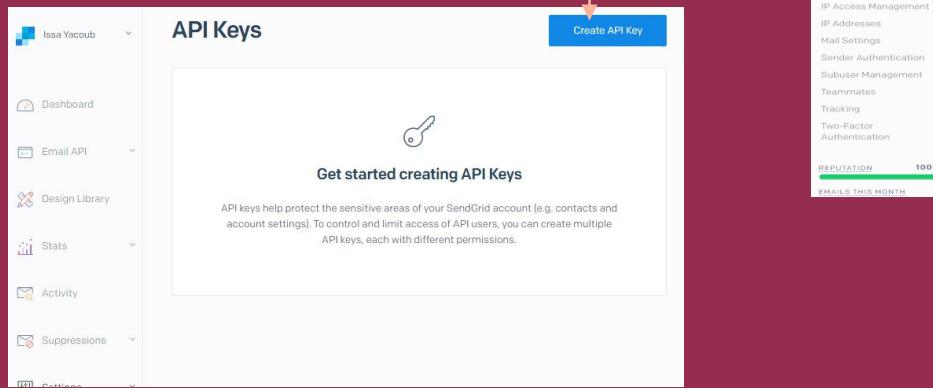
- This email address will now be able to send emails via SendGrid.



# How to Upload the Application to Azure, Pt. 21

## Install the Mail Server, Pt. 13

- The next step is to create an API key.
  - On the left-side menu bar, click on API keys
  - The API Key menu will appear
  - Click on the blue “Create API Key” button



# How to Upload the Application to Azure, Pt. 22

## Install the Mail Server, Pt. 14

- Fill out the API form

- Give a name to the API key.
- Then click on Restricted Access
- Under restricted access, Allow the API to send mail by going to the "Mail Send" field and giving it full access.
- We are giving the API key restricted access as an extra security measure. It will only be able to send mail and nothing else.
- After that, scroll to the bottom And click on the blue "Create & View" button.

The screenshot shows three overlapping windows:

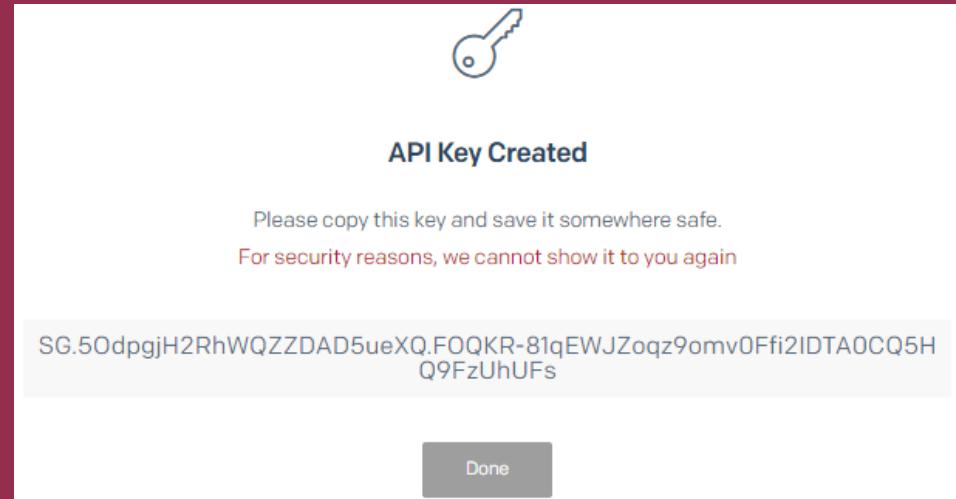
- Create API Key**: A top-level window for creating an API key. It has fields for "API Key Name" (containing "SALT\_API\_Key") and "API Key Permissions". Under "Full Access", the "Mail Send" permission is selected. There are also "Restricted Access" and "Billing Access" sections.
- Restricted Access**: A modal window showing access levels for various account components. The "Mail Send" component is highlighted with a large black oval and a blue arrow pointing from the "Create & View" button below.
- User Account**: A bottom-level window with a "User Account" section and "Create & View" buttons.

# How to Upload the Application to Azure, Pt. 23

---

## Install the Mail Server, Pt. 15

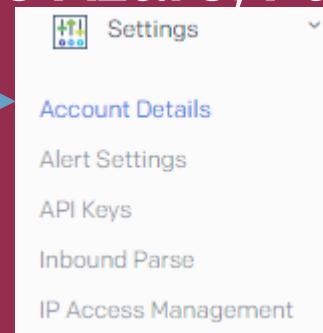
- Your API key was created and is displayed in the grey box.
- You will not be able to view the key again, so make sure to copy the key and save it in a notepad file.



# How to Upload the Application to Azure, Pt. 24

## Install the Mail Server, Pt. 16

- Next, we will find your SendGrid username.
- Click on “Account Details” on the left-side menu
- Once on the “Account Details” page,
- Find your username and copy and save it into a notepad file
- It will end with “@azure.com”



**Account Details**

Account Billing Your Products

Your Account

Personal Info	Issa Yacoub issay14@gmail.com 555-555-5555
Username	azure_8f75935deea5952b326e69ed781706a8@azure.com
Password	*****
Timezone	GMT-06:00 - Central Time, US & Canada America/Chicago

A screenshot of the SendGrid Account Details page. The "Account" tab is selected. Under "Your Account", there is a table with four rows: Personal Info (Issa Yacoub, issay14@gmail.com, 555-555-5555), Username (azure\_8f75935deea5952b326e69ed781706a8@azure.com), Password (redacted), and Timezone (GMT-06:00 - Central Time, US & Canada America/Chicago). The "Username" field is circled in black.

# How to Upload the Application to Azure, Pt. 25

## Install the Mail Server, Pt. 17

- Now that we have an API key and your SendGrid username, we will edit 2 PHP files
- Open "register\_action.php" in your editor.
- Scroll down until you see  
\$email = "YOUR SENDGRID USERNAME GOES HERE"  
It will be on line 41
- Replace the text inside the quotes with your SendGrid username.
- On line 48 replace,  
YOUR SENDGRID API KEY GOES HERE  
with your SendGrid API key.
- On line 64, replace salt.application@gmail.com  
with the sender email you created.
- Save the file

```
39 ~ if($query){  
40   register($arrays);  
41   $email = "YOUR SENDGRID USERNAME GOES HERE";  
42   $name = "SALT - Security Awareness Library Tool";  
43   $body = "<h3>Please click the link below to complete the SALT verification process</h3>  
44   <br><p>https://saltapplication.azurewebsites.net/verify.php?vkey=$vdkey</p>";  
45   $subject = "SALT | Please reset your account";  
46  
47   $headers = array(  
48     'Authorization: Bearer YOUR SENDGRID API KEY GOES HERE',  
49     'Content-Type: application/json'  
50   );
```

```
62 ~           "from" => array(  
63   "email" => "salt.application@gmail.com",  
64   "name" => $name  
65 ),
```

- The file should look like this when finished, but with your details.
- Save the file

```
38     $query = chkemail($sendemail);
39     if($query){
40         register($arrays);
41         $email = "azure_0d2d394690009b4de41f31c498f29d65@azure.com";
42         $name = "SALT - Security Awareness Library Tool";
43         $body = "<h3>Please click the link below to complete the SALT verification process</h3>
44         <br><p>https://saltapplication.azurewebsites.net/verify.php?vkey=$vdkey</p>";
45         $subject = "SALT | Please verify your account";
46
47     }
48
49     $headers = array(
50         'Authorization: Bearer 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',
51         'Content-Type: application/json'
52     );
53
54     $data = array(
55         "personalizations" => array(
56             array(
57                 "to" => array(
58                     array(
59                         "email" => $sendemail
60                     )
61                 ),
62             "from" => array(
63                 "email" => "salt.application@gmail.com",
64                 "name" => $name
65             ),
66         ),
67     );
68
69     $options = [
70         "method" => "POST",
71         "url" => "https://api.sendgrid.com/v3/mail/send",
72         "headers" => [
73             "Content-Type" => "application/json"
74         ],
75         "body" => [
76             "personalizations" => array(
77                 "to" => array(
78                     array(
79                         "email" => $sendemail
80                     )
81                 ),
82             "from" => array(
83                 "email" => "salt.application@gmail.com",
84                 "name" => $name
85             ),
86         ],
87         "auth" => [
88             "username" => "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",
89             "password" => "0d2d394690009b4de41f31c498f29d65"
90         ]
91     );
92
93     $ch = curl_init();
94     curl_setopt_array($ch, $options);
95     curl_exec($ch);
96     curl_close($ch);
97
98     echo "Email sent successfully!";
99 
```

# How to Upload the Application to Azure, Pt. 26

## Install the Mail Server, Pt. 18

- We will repeat the process for the password Reset file
- Open “recovery-action.php” in your editor.
- Scroll down until you see  
\$email = “YOUR SENDGRID-USERNAME”; It will be on line 26  
Delete the text in the quotations, then paste your SendGrid username.
- On line 34, in the quotation, replace YOUR-SENDGRID-API-KEY \_\_\_\_\_ with your API key.
- On line 49, replace salt.application@gmail.com \_\_\_\_\_ with the sender email you created.
- Save the file

```
26 $email = "YOUR_SENDGRID-USERNAME";
27 $name = "SALT - Security Awareness Library Tool";
28 $body = "<h3>Please click the link below to complete the SALT verification process</h3>
29 <br><p>https://saltapplication.azurewebsites.net/new-reg.php?vkey=\$vdkey2</p>
30 <br>Your verification key is $vdkey2";
31 $subject = "SALT | Please reset your account";
32 
33 $headers = array(
34     'Authorization: Bearer YOUR-SENDGRID-API-KEY',
35     'Content-Type: application/json'
36 );
```

```
48 "from" => array(
49     "email" => "salt.application@gmail.com",
50     "name" => $name
51 ),
```

- The file should look like this when finished, but with your details.
- Save the file

```
26 |     $email = "azure_0d2d394690009b4de41f31c498f29d65@azure.com";
27 |     $name = "SALT - Security Awareness Library Tool";
28 |     $body = "<h3>Did you request to change your password? If so, please click the link and enter your ver
29 |     <br><p>https://saltapplication.azurewebsites.net/new-reg.php?vkey=\$vdkey2</p>
30 |     <br>Your verification key is $vdkey2";
31 |     $subject = "SALT | Did you request to change your password?";
32 |
33 |     $headers = array(
34 |         'Authorization: Bearer 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',
35 |         'Content-Type: application/json'
36 |     );
37 |
38 |     $data = array(
39 |         "personalizations" => array(
40 |             array(
41 |                 "to" => array(
42 |                     array(
43 |                         "email" => $sendemail
44 |                     )
45 |                 )
46 |             ),
47 |             "from" => array(
48 |                 "email" => "salt.application@gmail.com",
49 |                 "name" => $name
50 |             ),
51 |         ),
52 |     );
53 | 
```

# How to Upload the Application to Azure, Pt. 27



## Install the Mail Server, Pt. 19

- The mail server will be up and running and the two PHP files that were edited will be able to access them and send emails.

# How to Install the Data Scraper on Your Machine, Pt. 1



- Download the Django application
- Open PyCharm and import the program in your venv
- Type the following in the terminal: `manage.py runserver`
- The terminal will return a link to your localhost, once you click on it, you will be able to scrape whatever you like.

# How to Create Inbound Rules

- Go to the Firewall & Virtual Network Tab in Azure's CosmosDB service
- Click selected networks, then add all the IPs you have permission to scrape
- This will take 5-10 minutes to update.

The screenshot shows the 'Firewall and virtual networks' settings for the 'saltdb' Azure Cosmos DB account. The left sidebar lists various account settings like Connection String, Features, and Firewall and virtual networks, with 'Firewall and virtual networks' highlighted and circled in red. The main pane shows the configuration for allowing access from selected networks. It includes sections for 'Virtual networks' (with options to add existing or new virtual networks) and 'Firewall' (where users can add IP ranges). A large red circle highlights the 'IP (Single IPv4 or CIDR range)' input field where IP addresses would be entered.

Home > saltdb

saltdb | Firewall and virtual networks

Azure Cosmos DB account

Search (Ctrl+ /)

Allow access from

All networks  Selected networks

Configure network security for your Azure Cosmos DB account. [Learn more.](#)

Virtual networks

Secure your Azure Cosmos DB account with virtual networks. [+ Add existing virtual network](#) [+ Add new virtual network](#)

Virtual Network	Subnet	Address range	Endpoint Status
No network selected.			

Firewall

Add IP ranges to allow access from the internet or your on-premises networks.

IP (Single IPv4 or CIDR range)

Connection String

Features

Replicate data globally

Default consistency

Backup & Restore

Firewall and virtual networks

Private Endpoint Connections

Data Migration

Preview Features

# Database Table Names & Contents

---

- Approved – This is where all the words go once an administrator has approved them. They come from the words table, and the \_id field does not change at any time.
- Counter – This is where all the scraped data goes. The documents in this collection are only here to overwrite the article and video fields in the ‘words’ table.
- Newusers – This is the ‘profile’ database where all users’ data is kept. This includes email, password, first name, last name and all saved flashcards. All passwords are hashed and secure.
- Words – This is the Transaction db. It contains all the words that have not yet been approved.
- Reports – This is where all the reports are sent

# Database Table Names & Contents

---

- Don't use MongoDB or the shell to create a document. If you do this, it will create a dataset with an object ID, which is a nested data type. If you have a nested data type, it will not print on the page unless you specify in PHP that you are echoing a nested variable.
- MongoDB is not a relational database, so there is no real reason to create foreign keys or to join any of the tables together. If you want to get a variable from another table, you should query the table and echo the new variable on the page in PHP instead of joining.

# Required PHP Statements

---

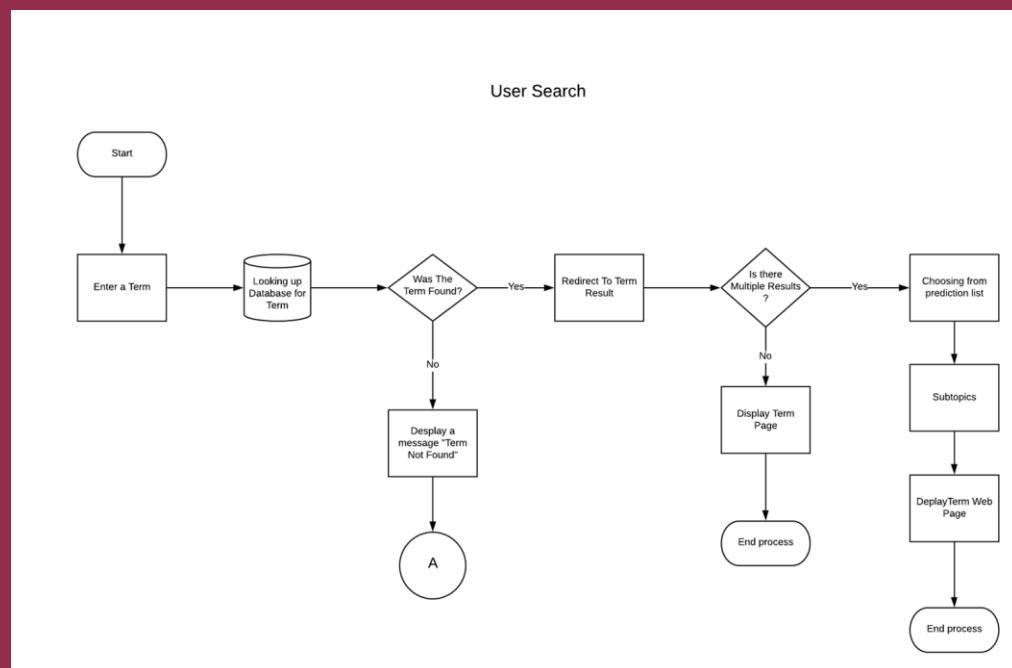
- Every new php file you create needs to have the following line at the top of the page:
  - require\_once 'dbconnect.php';
- If you want a certain page to be accessible for administrators only, include the following lines at the top of the page:

```
1 <?php
2 require_once 'library.php';
3     require_once 'dbconnect.php';
4     if(chkLogin()){
5         $name = $_SESSION["uname"];
6         $collection = $manager->mydb->newusers;
7         $query = $collection->find(['FirstName'=>$name]);
8
9     foreach($query as $doc){
10         $status = $doc->Admin;
11     }
12 }
13 if($status!="yes"){
14     header("Location:login.php");
15 }
```

# Main Requirements: 1. User Search

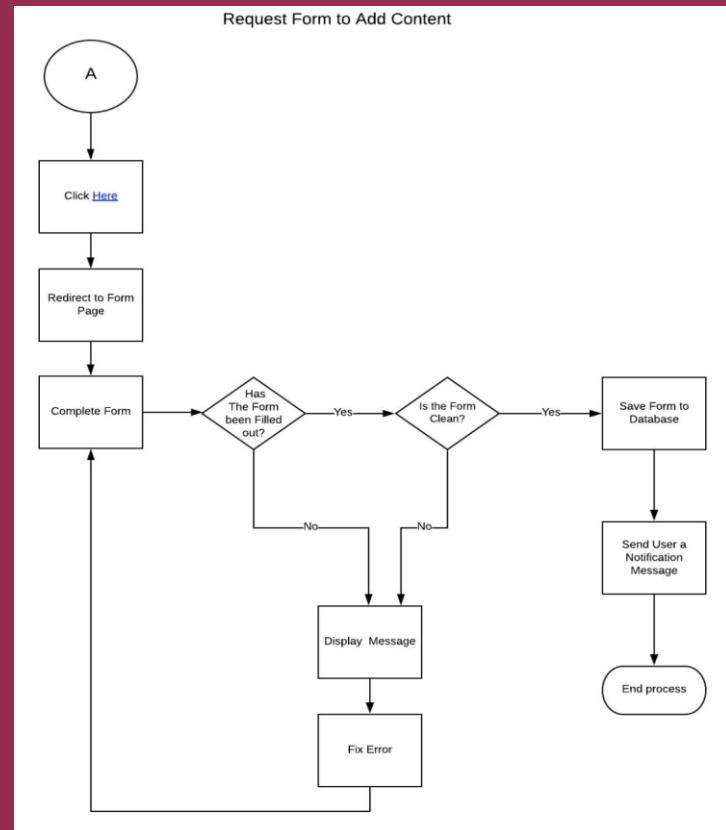
---

- ❖ User will be able to type a word in the search bar and view the full result
- ❖ The system will display potential search results as the users type
- ❖ Be told if it's not in the library
- ❖ Value: To provide the users with results that are peer-reviewed and easy to understand.



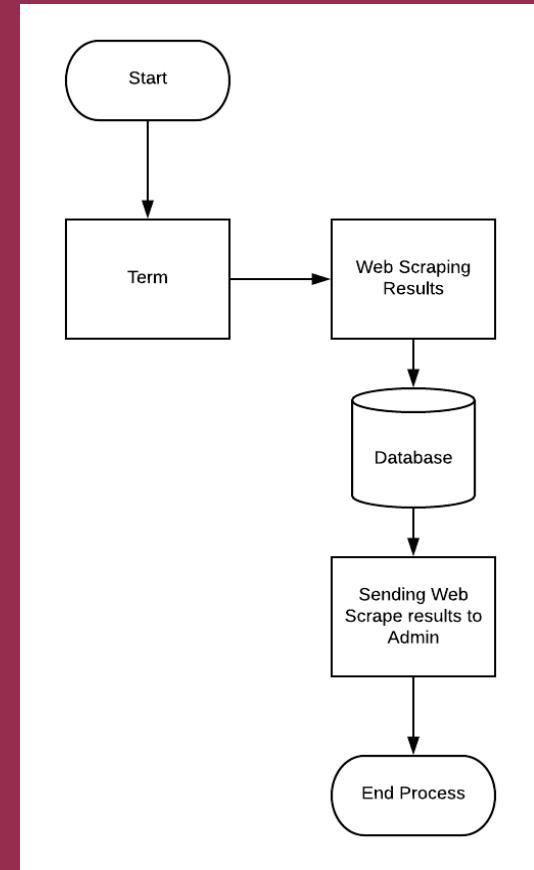
# Main Requirements: 2. Request Form to Add Content

- ❖ If a user is unable to find a term, he can request to have it added.
- ❖ Get redirected to a form
- ❖ Form has to have filters (No profanity or empty fields)
- ❖ Value: To ensure that students have their questions answered, to keep the library fresh and to keep the faculty aware of what students want to learn about.



# Main Requirements: 3. Load Library for Administrator

- ❖ Gather requirements from admin
- ❖ Scrape the requirements from a list of approved sites
- ❖ Send them to the database an administrator can access
- ❖ Give admin the ability to approve, edit or deny each entry
- ❖ Value: This will provide the administrator with the definitions they want to see on the site. It will also automate and speed up the process of loading the library.



# Main Requirements: 4. Usage Metric Dashboard

---

- ❖ Admin can view how many times each word was searched via a dashboard
- ❖ Value: To gain quantitative feedback on how the site is performing.

