Curiosity browser build

# Abbreviations

The following abbreviations will be used below:

|  |  |
| --- | --- |
| Name | Abbreviation |
| curiosity\_browser | *APP* |
| dev.mars-browser.co.uk | *hostname* |
| document root for the PHP code | *DOCROOT* |
| jsinc folder | *jsinc* |
| web page https://www.mars-browser.co.uk/curiosity | *website* |

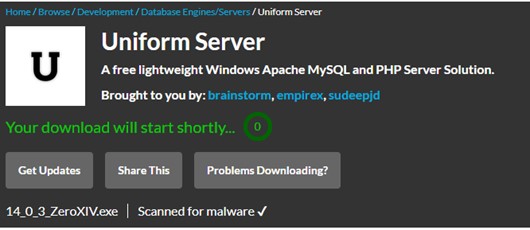
# Repos

* <https://github.com/open768/curiosity_browser>
* <https://github.com/open768/phpinc>
* <https://github.com/open768/spaceinc>
* <https://github.com/open768/jsinc>

# PHP for windows

## Downloading Uniform Server

I use the lightweight [Uniform Zero XV](https://www.uniformserver.com/) LAMP distribution - self extracting zip files can be downloaded at [sourceforge](https://sourceforge.net/projects/miniserver/files/)



Always use a virus checker to confirm that downloaded files are safe.

## Installing Uniform Server

The downloaded file is a self-extracting zip file that runs as a [portable application](https://portableapps.com/) (i.e. it doesn’t require any installation, just unpack and go). You can extract it yourself into the directory that you want uniserver to run from.

Please also download Visual C++ Redistributable for Visual Studio 2019 as noted in the file MUST\_READ.txt from

**It can be downloaded from:**

**https://docs.microsoft.com/en-US/cpp/windows/latest-supported-vc-redist?view=msvc-170#visual-studio-2015-2017-2019-and-2022**

**Select: vc\_redist.x86.exe**

### shortcuts

From now on we will use the following shortcuts in this document

$UNISERVER as the location of where uniserver is installed

$APACHE as a shortcut for $UNISERVER/core/apache2

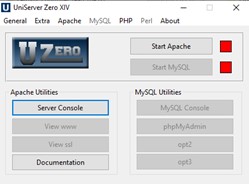
## Configuring uniform server

Uniform zero is designed to run with minimal configuration. There are some things we need to do from the uniserver console.

Optional: In my version of uniserver I disable MySQL support by deleting the folder $UNISERVER/core/mysql

### Start uniserver

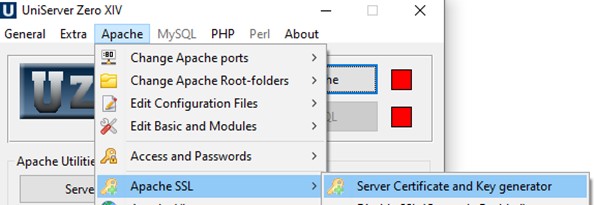
Start the console: by running $UNISERVER/UniController.exe (or start it from portable Apps)



### Create an SSL certificate

Rapport runs on the cloud as https, even the on the development environment you must run https.

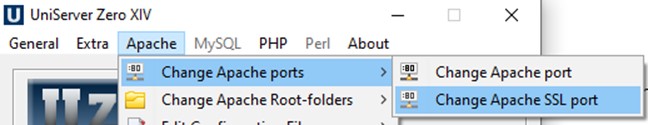
Luckily this is super easy on uniserver using the server cert and key generator



If the Apache SSL option is disabled stop Apache first to enable the option.

### Change the SSL port

Port 443 is usually privileged – best to change the port to 8443



### Select the PHP version

Without this when you run Apache, PHP won’t run. Pick the latest version of PHP available.

## php ini

ensure that the following are enabled in the PHP configuration

* extension=curl
* extension=openssl

# Apache

Its best to separate your development into a virtual host to keep the project self-contained and log files separate

It is preferable to create an environment variables where needed

If using environment variables remember to restart portable apps and uniserver to pick up changes

## Create a vhost

### Create hosts entry for your virtual host

E.g. for https:// *hostname*:8443/

In this case edit the windows hosts file (or use [powertoys](https://aka.ms/installpowertoys))

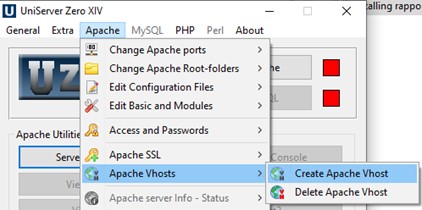
%windir%\System32\drivers\etc\hosts

And add the line

127.0.0.1 *hostname*

### Create a non SSL vhost (optional)

You can use the uniserver zero ui to create a non SSL virtual host



### Create a SSL virtual host

1. Edit the file $APACHE/conf/extra/httpd-ssl.conf
2. Copy the vhost section

<VirtualHost \_default\_:${AP\_SSL\_PORT}>

…

</VirtualHost>

1. Paste the vhost section changing the label to *hostname*:${AP\_SSL\_PORT}
2. In the new *hostname* vhost section change the following

<VirtualHost *hostname*:${AP\_SSL\_PORT}>

ErrorLog "${US\_ROOTF}/core/apache2/logs/ *APP*\_error\_ssl.log"

TransferLog "${US\_ROOTF}/core/apache2/logs/ *APP*\_access\_ssl.log"  
DocumentRoot "*DOCROOT*"

ServerName " *hostname* "

Options Indexes

<Directory "*DOCROOT* ">

## Add an alias

If the js folder isn’t at the top level of the *DOCROOT*, it needs to be added as an alias in the vhost definition

Alias "/js" "*js*"

<Directory "*js*">

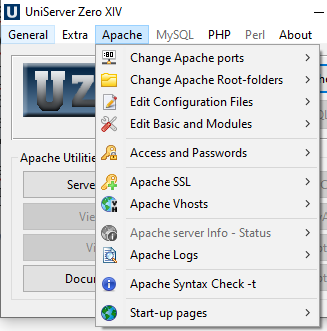
Require all granted

SSLRequireSSL

</Directory>

## Do an Apache syntax check

Just in case an error was introduced



## Example vhost configuration

VirtualHost \*:${AP\_SSL\_PORT}>

# General setup for the virtual host

DocumentRoot F:\projects\php

ServerName dev.mars-browser.co.uk

ServerAlias www.dev.mars-browser.co.uk \*.dev.mars-browser.co.uk

ServerAdmin you@example.com

ErrorLog "${US\_ROOTF}/core/apache2/logs/error\_ssl.log"

TransferLog "${US\_ROOTF}/core/apache2/logs/access\_ssl.log"

# SSL Engine Switch:

SSLEngine on

SSLCertificateFile "${US\_ROOTF}/core/apache2/server\_certs/server.crt"

SSLCertificateKeyFile "${US\_ROOTF}/core/apache2/server\_certs/server.key"

SSLVerifyClient none

SSLProxyEngine off

#aliases

Alias /js "F:\ projects\js"

#== permissions for Server Root folder:

<Directory "F:\projects\php">

Options Indexes Includes FollowSymLinks

AllowOverride All

Require all granted

SSLRequireSSL

</Directory>

<Directory "F:\projects\js">

Options Indexes Includes FollowSymLinks

AllowOverride All

Require all granted

SSLRequireSSL

</Directory>

<FilesMatch "\.php$">

SSLOptions +StdEnvVars

</FilesMatch>

</VirtualHost>

# Application configuration

## Install composer

Composer is already installed in the folder *APP*/bin

Composer is not currently used. This is a placeholder

## Check configuration

From the command line

* change directory to *APP*/bin
* run the command check.bat

this will identify whether any configuration changes need to be made

## Edit application Paths

Verify the locations configured in *APP*/php/app-config/app-config.php

# using the app

## Code structure

## Page Structure

Each php page

* defines the global $root which links back to the top curiosity browser folder
* pulls in $root/php/app-common.php which sets up other globals and checks
* has an associated <page>.js which loads the jQueryUI definition
* outputs HTML
  + pulls in $home/php/fragments/header.php
  + contains the html page structure
  + attaches an event handler to the body onload event
    - which calls the appropriate function from the associated <page>.js
  + pulls in $home/php/fragments/footer.php

## becoming an admin

* log into the application
* user enters admin page (*website* /php/pages/admin/admin.php) in debug mode – this
  + https://www.mars-browser.co.uk/curiosity/php/pages/admin/admin.php?debug2
* shows the user copy the shown ID

17-08-2024 10:37:37: ❖ Enter> cAuth.get\_user\_id

      17-08-2024 10:37:37: user ID is 101597#########

17-08-2024 10:37:37: ❖ Leave > cAuth.get\_user\_id

* + take a copy of the shown USERID
  + create a file ckadmin.json in *DOCROOT*/php/app-config folder with the following contents

{ "admin\_ids": ["adminid1","adminid2"] }

* reload the admin page in debug to confirm that the userid has been ingested and user is admin

17-08-2024 10:45:36: ❖ Enter> cAuth.check\_for\_admin\_id\_file

17-08-2024 10:45:36: checking for ID file '/homepages/14/d542224021/htdocs/mars/curiosity/php/app-config/ckadmin.json'

17-08-2024 10:45:36: ❖ ID file found

17-08-2024 10:45:36: file contains json

17-08-2024 10:45:36: Adding 101597######### to role ckadmin

* reload the *website* to confirm that admin button is visible for user
* reload the admin page normally