

Create an App for Azure (part 2)

Cloud, CSS, Docker, HTML, PHP

Language Skills : 5

The digital transition has pushed software companies and enterprises to migrate their applications to the web. Azure cloud services have provided the ideal platform, the right services, and the tools to go digital while keeping full control over the data. This publication proposes the creation of a simple application by discovering some of these services and tools.

Published Wednesday June 22nd 2022, 17:33
Modified Monday August 26th 2024, 10:04



By Olivier Paudex

Setting up the work environment

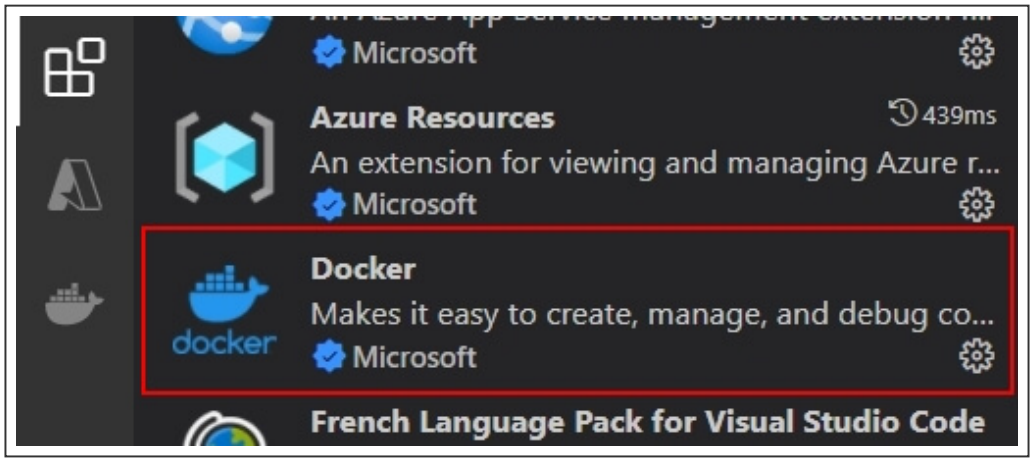
To work with the Azure cloud and containers, you need to install certain tools, a few of which are highly recommended.

> This publication is composed of several parts. This being the 2nd part.

The working tool here is a Window 11 PC; it is quite possible to do this from a Mac.

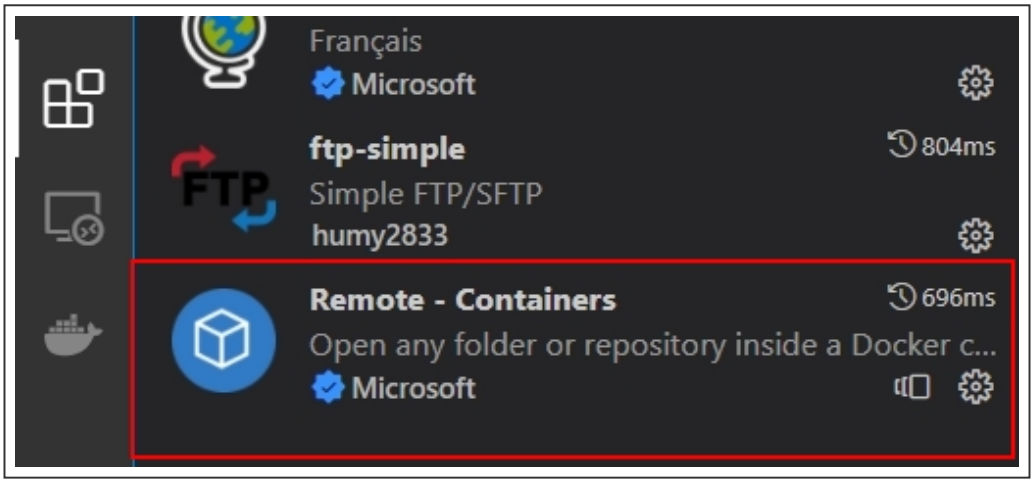
If you don't have an Azure account, now is the time to do so. Opening an account and using the various services is free for the first 30 days.

- Installation of Visual Studio Code : https://code.visualstudio.com/download.
- Docker installation on Windows : https://docs.docker.com/desktop/windows/install/.
- Installation of the Docker plugin for VSCode.



The Docker plugin for VSCode

- Installation of the plugin **"Remote – Containers"** for VSCode to edit the container code directly.



The Remote – Containers plugin for VSCode

- Install PowerShell : https://docs.microsoft.com/en-us/powershell/.
- Install Azure CLI : https://docs.microsoft.com/en-us/cli/azure/.

- From the VSCode interface, open a Powershell terminal and update Azure CLI if necessary

```
az --version
```

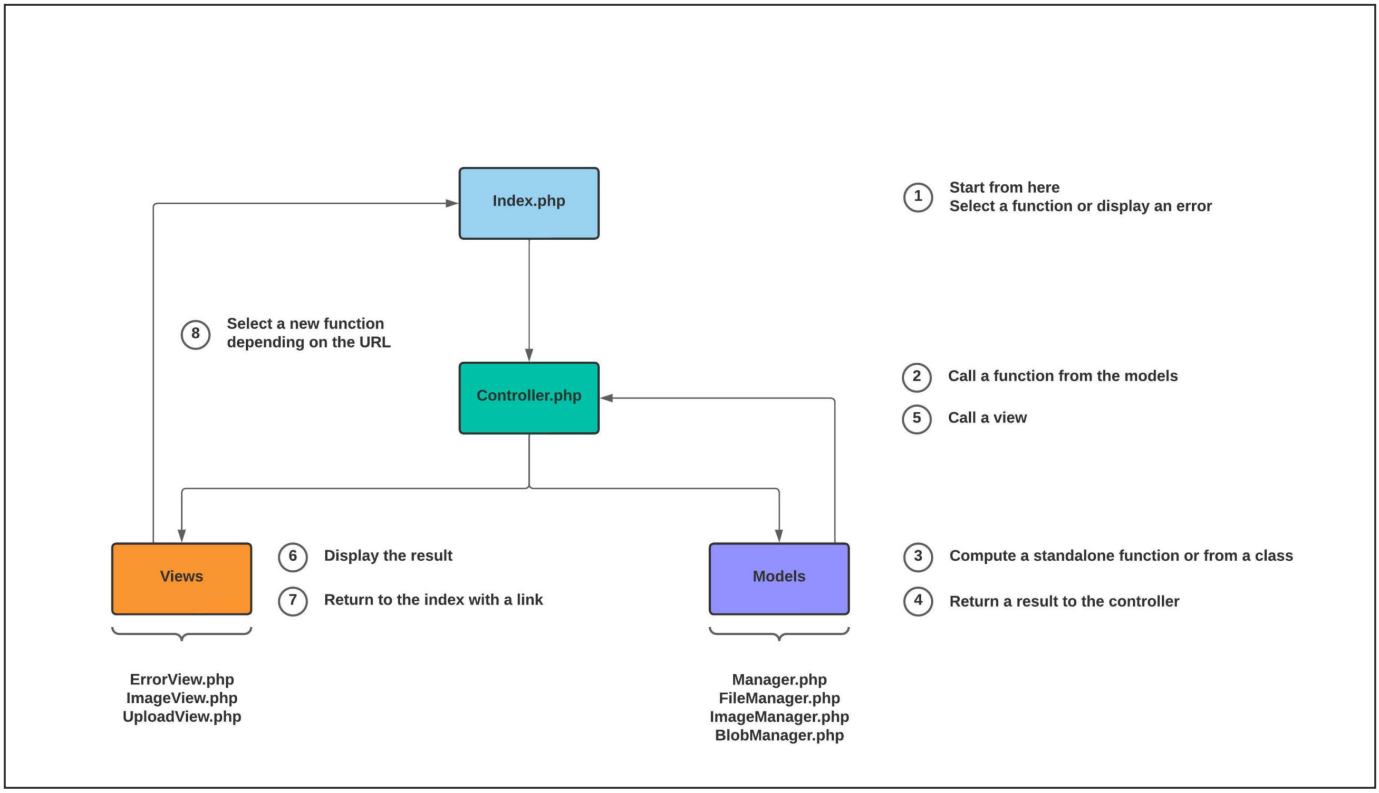
then

```
az upgrade
```

Preliminary Project

The MVC method

Before starting a project, it is always interesting to put your ideas on the table, to draw the screens, to choose a programming language. In this project, the **PHP**, the **HTML** and the **CSS** have been chosen. For the code, the **MVC** methodology, focused on models and views, will be used to keep the vision as simplistic as possible. Below is an outline of the project in MVC format.

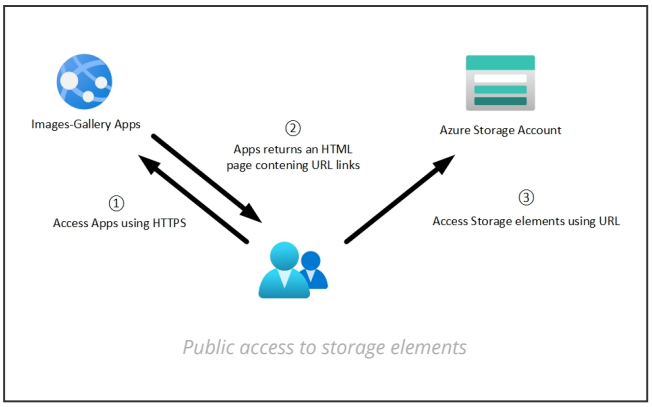


The MVC model of the application

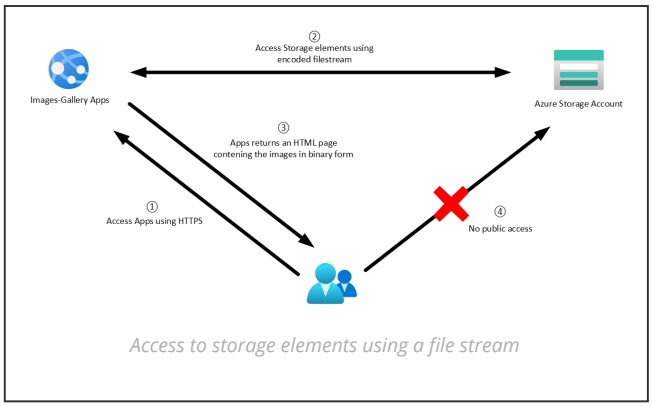
Access to data storage

Here's an explanation of how to access data in storage in a secure and non-secure manner. The application will of course use the second way.

On the Azure cloud, to access data storage, one uses the URL of an item to access it. Except that in the case of a cloud application, public access to the storage service must be closed. This does not allow using the URL of the elements to access them. The **Images Gallery** application will use a binary read of the stored images to access them, not its URL.



Public access to storage elements



Access to storage elements using a file stream

- In the image on the left, storage items are accessed using the URL.
- In the image on the right, storage items are accessed using an internal data stream from the application. This closes public access to the cloud storage account.

Below is the function that retrieves an image from the storage account. It is located in the **BlobManager.php** file.

> Download the code for this application by clicking on the "Attachment" link at the very top of the post.

```
// Get blob content from Azure storage
public function getBlobContent($containerName) {
  // Storage connexion
  $blobClient = $this->storageConnect();

  // Get blob file and mime property
  $blob = $blobClient->getBlob($containerName, $this->Name);
  $properties = $blobClient->getBlobProperties($containerName, $this->Name);
  $mimeType = $properties->getProperties()->getContentType();
  // Get base64 encoded blob file stream
  $stream = stream_get_contents($blob->getContentStream());
  $fileEncode = base64_encode($stream);
  // Return the source stream
  $src = 'data: '. $mimeType . '; base64,' . $fileEncode;
  return $src;}
```

Conclusion

Here are some good starting ideas about what software to use. A reflection on the methodology was also mentioned, a method that it is very advisable to adopt when you want to create a web application.

Finally, an explanation of the concept used when accessing a storage account was added. It was very important to outline this, as there is little to no documentation regarding secure non-public access to Azure storage.

The next chapter will talk about the project, the application code, the containerized infrastructure with Docker and the libraries used.