Creating an open source learning project

Azure Functions University is an educational project for learning about Azure Functions - the Functions as a Service offering in Azure. The content is aimed at people who do not have previous experience with serverless technology and want to learn by following exercises and writing code.

Author Marc Duiker

The idea behind the project

I started this project because I want to enable newcomers to serverless technology to get up and running with Azure Functions in a very low friction way. Learning new things can be challenging, and frequently, the official documentation alone is not enough to understand a new topic and put it into practice.

The dual-channel delivery, lessons on GitHub and videos on YouTube, is intentional because some people prefer watching (or listening) to videos, while others prefer reading.

How it started

The Azure Functions University project started in October 2020. I have had quite some content on both GitHub and YouTube for some years now, but most of that was intended for intermediate or experienced users of Azure Functions. Since there is a huge increase in people new to programming, I want to help out that group and make it easy for them to start with serverless technology.

I consider myself reasonably experienced with Azure Functions. On the one hand, that's good for the project, so I can share a lot of what I know. But on the other hand, this can be a pitfall because I'm likely to have assumptions on topics that people new to serverless don't have. To prevent too much bias from my side, I wanted someone relatively new to the technology to co-create the content and co-host the live streams. I was following Gwyneth Pena (US) on Twitter, and since I really like her personality and the style of her videos, I asked her to join. I was thrilled she said yes immediately.

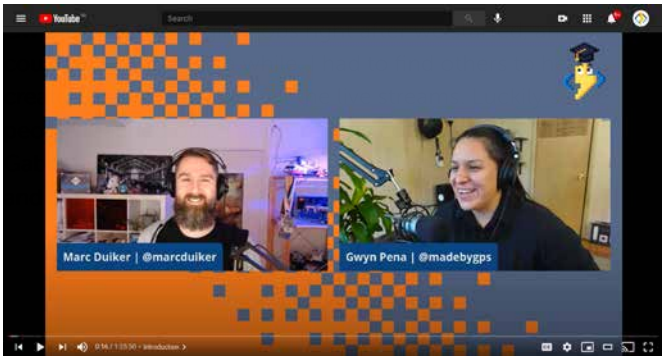


Figure 1. First Azure Functions University video about HTTP triggers

README.md

Azure Functions University

Welcome to Azure Functions University! 🎓

This repository contains everything you need to learn about Azure Functions <⚡> and complements the learning videos made by Marc and an amazing group of contributors!

You can watch the videos on the *Serverless on Azure* YouTube channel.

Figure 2. Azure Functions University GitHub repo

Curriculum

At this moment, the curriculum contains the following lessons:

› **HTTP;** How to do GET requests and use query string parameters and do POST requests where the data is read from the request body.
› **Blob;** How to use output and input bindings to read/write data from/to Blob storage using different binding types, using the BlobTrigger to start a function when a blob is written to storage.
› **Queue;** How to use output bindings with various binding types, using the QueueTrigger to start a function when a message is put in a queue.
› **Table;** How to use output and input bindings to read/write data from/to Table storage with various binding types.
› **Deployment;** How to deploy your Function App to Azure using VSCode, Azure CLI, and GitHUb Actions.
› **Configuration;** Why and how to use app settings in your Function App, using App Configuration service for easier management for app settings across multiple resources.
› **CosmosDB;** How to use the output and input bindings to read/write data from/to CosmosDB, using the CosmosDBTrigger to start a function when a new document is added to a collection, and using KeyVault to store the CosmosDB connection string.
› **Durable Functions I;** Why using Durable Functions is beneficial when dealing with multiple functions. This demonstrated by using the function chaining pattern to illustrate how orchestrations work.

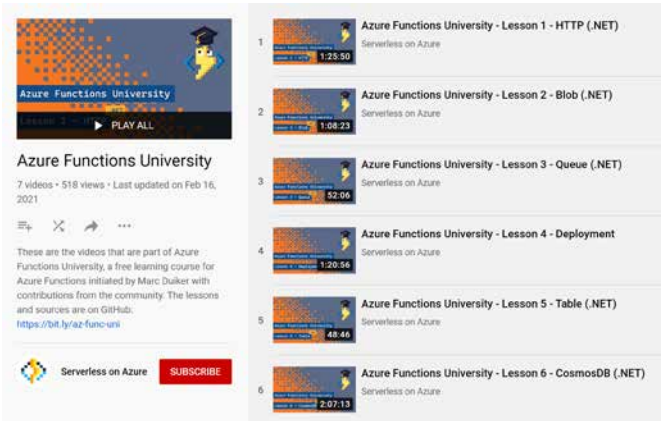


Figure 3. The Azure Functions University playlist on YouTube

I believe that consistency is key when creating educational content. Therefore each lesson follows the same structure:

› there are several exercises written in markdown, including code snippets;
› three types of call-outs are used: tips , observations and questions ;
› a complete Function App project is available as reference;
› at the end of each lesson, there's a homework assignment.

All coding exercises use VSCode as the code editor because this is a more beginner-friendly environment than Visual Studio 2019.

Although we started creating content for .NET functions, we're now also accepting contributions for other languages. TypeScript is the second language we have some lessons for now.

Challenges

Creating quality content is hard, and it is very time-consuming. For the first couple of lessons, I created most of the content myself, which was hard to combine with a full-time job. Since more people are helping now, it gets easier, although reviewing the pull requests is a considerable effort. I want to ensure the tone of the lessons remains constant and that inclusive language is used. I now realize what it feels like to be a maintainer of a small open source project.

The frequency between the lessons varies between two to four weeks. Ideally, I'd like to have a livestream every other week. However, planning is tricky since schedules and priorities shift, not only mine but also the contributors. This is voluntary work we all do in our free time, and sometimes other things are more important, and that's OK. Working on this project should be enjoyable, not stressful.

Keeping the lessons up-to-date is becoming a challenge right now. The current .NET content is targeted for .NET Core 3.1. Since Functions can now also be written in .NET 5, additional content needs to be created soon to reflect this. The .NET Core content will remain since .NET Core 3.1 has long-term support, and I expect the content will remain relevant for a while.

This brings us to another challenge, and that is the Azure Functions University GitHub repository. At the moment, there are eight lessons across two programming languages, .NET Core and TypeScript. Sub-folders are used for each language in order to keep everything tidy, but eventually, the source code needs to be split into separate repositories for each language/runtime. This will make the source code easier to manage, and VSCode will be less confused about which projects to run.

Marc Duiker
Consultant

xpirit.com/marc

What's next?

There's a lot of progress to be made. First, there is still a lot of new content to be written. Many topics have not been touched yet, e.g., security, SignalR, EventGrid. There are also content translations to the other languages that Azure Functions supports. Some people did show interest in helping out with Python and TypeScript, but it's still a long way to go until that's on the same level as the .NET lessons.

Secondly, I want to have better insight into how many people are using the GitHub repo and how they experience it. I'll be looking into GitHub classroom to see if I can get a better grip on the usage of the lessons. I prefer to have as little friction as possible, because additional sign-up boundaries might prevent people from using the material.

Will this project ever be finished? Not any time soon, I think. The Azure Functions team recently presented their roadmap for the next major releases. I expect plenty of opportunities to create new lessons and help more people to use serverless technology.

Help us!

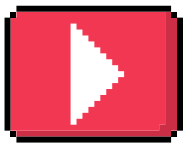
We're always looking for contributors who can help create content and co-host a live stream! Contributions can be new lessons, additions to existing lessons, or 'translations' to other programming languages (TypeScript, Python, PowerShell, Java).

Please have a look at the existing issues to see if you can contribute to those. If there is nothing to your liking, you can submit a new issue. You don't need to be an expert on the topic. We can work on the content together. </>

Links

YouTube playlist: https://bit.ly/az-func-uni-playlist
Azure Functions University GitHub repo:
http://bit.ly/az-func-uni
GitHub issue list: http://bit.ly/az-func-uni-issues

96
Stars on the GitHub repo



650
YouTube subscribers

>1850
Views of the first lesson