

Serverless service file Typescript definitions will never be outdated again!



Frédéric Barthelet · Follow
Published in Serverless Transformation · 2 min read · Nov 24, 2020

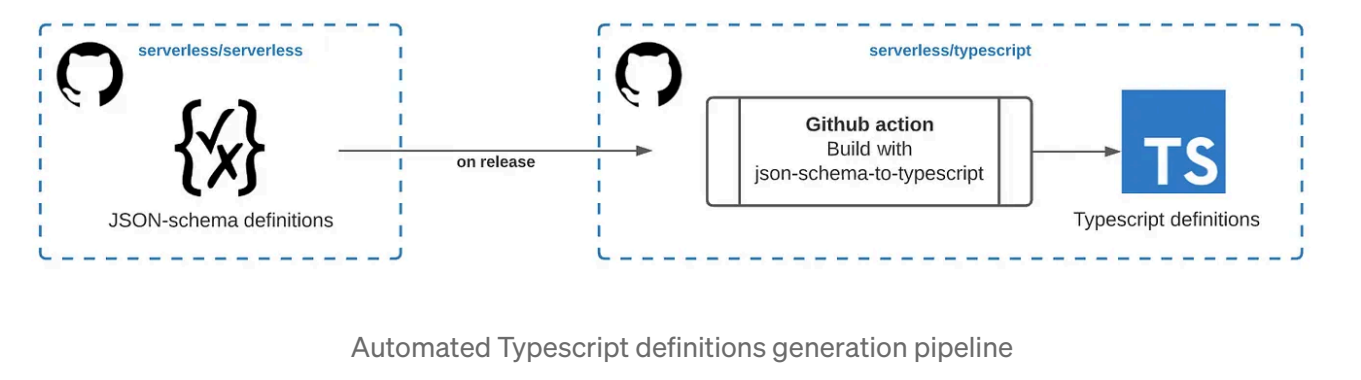
117 1



Since [v1.72.0](#), the Serverless framework accepts `serverless.ts` as a valid *service file* in addition to the usual `serverless.yml`, `serverless.json` and `serverless.js` file formats.

TL;DR

A [new repository](#) is born this month in the Serverless organization. This repository, named `serverless/typescript`, hosts an automated pipeline generating Typescript definitions based on JSON schemas in core Serverless framework.



A few simple steps to use those new definitions

- installing the definitions `npm i @serverless/typescript --save-dev`
- importing the definition in your `serverless.ts` file with `import type { AWS } from '@serverless/typescript';`
- use this `AWS` type at the root of your service file export

The reason behind an automated pipeline

I wrote a [detailed article explaining the benefits of using Typescript for IaC definition files](#). I strongly encouraged you to give it a go if you'd like to benefit from types, imports and references!

Until today, the only way to type Serverless service file with Typescript was to install the [Serverless DefinitelyTyped package](#) also known as `@types/serverless`. Once installed, this community maintained package exports a `Serverless` definition that can be used to type your `serverless.ts` service file configuration object. However, like all community maintained definitions in *DefinitelyTyped*, the quality of the definitions depends on the reactivity and accuracy of the maintainers. Therefore, from times to times, these definitions might diverge from the actual accepted service file configurations, either being outdated or simply wrong. This defeats the purpose of having definitions to rely on to ensure service definition is formatted as expected.

Fortunately, Serverless internally uses JSON-schema and `ajv` to perform service file validation at the beginning of any Serverless CLI command. This validation helps massively by failing fast before a deployment, but Typescript types have the advantage of failing even faster, with direct feedback to the developer, in his IDE.

In order to tackle the issue of outdated DefinitelyTyped definitions, I implemented an automated pipeline to generate those Typescript definitions directly from the internally used JSON-schema of Serverless. This automated pipeline is triggered every time a new release of Serverless framework is available. The pipeline ends with the publishing of the newly generated definitions to NPM, ensuring they are always up to date and consistent with the framework internal validation logic.

The [serverless/typescript repository](#) is open-source and all contributions to improve the automatically generated definitions are welcome!

Serverless Serverless Framework Typescript Json Schema Infrastructure As Code



Written by Frédéric Barthelet
104 Followers · Editor for Serverless Transformation
AWS Community Builder, @serverless/typescript maintainer and Serverless framework contributor.

More from Frédéric Barthelet and Serverless Transformation

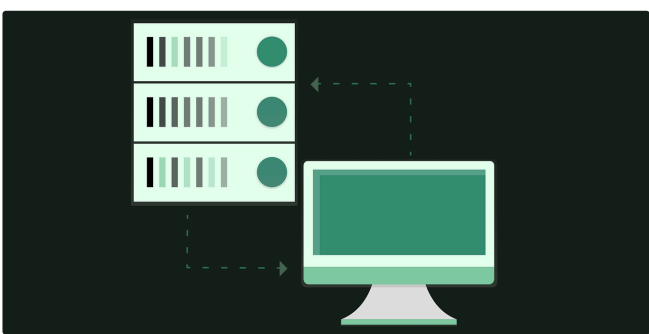


Frédéric Barthelet in Serverless Transformation

The Awesome Libraries of AWS Typescript Serverless Developers

A list of awesome open source tools for Typescript AWS Serverless development...
5 min read · Dec 3, 2020

441 1

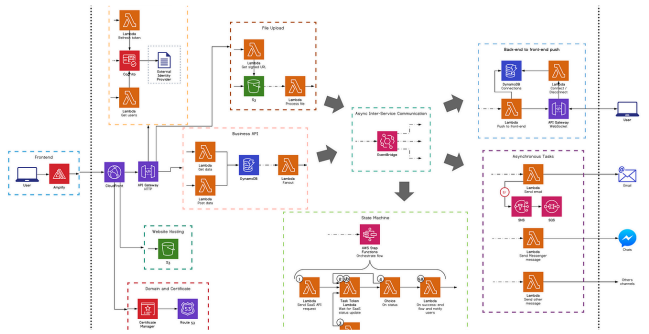


Xavier Lefèvre in Serverless Transformation

Asynchronous client interaction in AWS Serverless: Polling,...

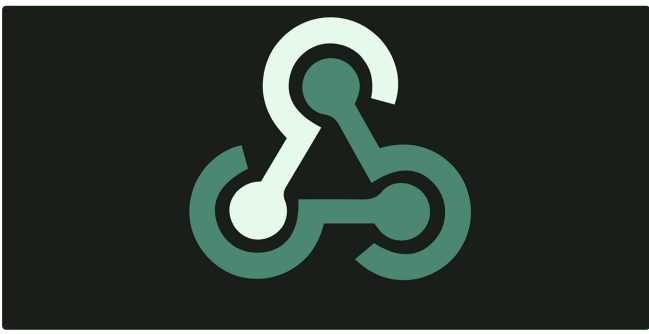
Event-driven Serverless often requires async update to the client. There are several ways ...
6 min read · Apr 18, 2020

730 7



Xavier Lefèvre in Serverless Transformation

What a typical 100% Serverless Architecture looks like in AWS!



Frédéric Barthelet in Serverless Transformation

Serverless webhooks—Designing Lift