Hands-On SIG JavaScript 🡪 Typescript

**Attachment**: Simple JavaScript Support Library for publishing Events to Azure EventGrid topic.

**Description**: Starting point for this hands-on is a simplified version of a Javascript library used at a customer. It is used for all Azure Functions that need to publish an event to an EventGrid topic.

When we consider converting our code base to TypeScript, converting our support libraries are a necessary step in this process.

**Assignment**: Convert this library to TypeScript.

**Possible starting points / technical challenges to isolate**

* An official Azure NodeJS library is used: @azure/eventgrid. Try to find out if it already has a TypeScript version or if there is an @types available. Focus on the code that calls this external library. How would those calls look in TypeScript?
* Start at the entry function for this library: publishEventToEventGridTopic. How would the interface look in TypeScript? And the interfaces for the internal functions that are called in the follow-up?
* eventtopic.js contains an Enum and a Map that works as a connection cache. How would those two be implemented in TypeScript?
* The unit tests are implemented with Jest. What libraries would be necessary to use TypeScript in Jest? Try to convert both tests to TypeScript.
* The two existing unit tests check for an error flow. Try to mock the @azure/eventgrid library and test a ‘happy’ flow.
* An advantage of TypeScript could be the support of a fixed schema/Interface that needs to be adhered when publishing to a specific EventGrid topic. Try to implement a (made-up) schema for one of the topics and experiment with different validation that could be set in the schema.

Hands-On 2 SIG JavaScript 🡪 Typescript

**Attachment**: Simple JavaScript Function with EventGrid trigger.

**Description**: Starting point for this hands-on is a simplified version of a Azure Function that triggers on an incoming EventGrid Event. It does some validation and calls an API with a POST request using the node library node-fetch.

When we consider converting our code base to TypeScript, converting our Azure Functions is the biggest bulk of this base. Keeping the same high standards for things like readability, logging and unittesting is very important.

**Assignment**: Convert this Function to TypeScript.

**Possible starting points / technical challenges to isolate**

* How could we do API calls from TypeScript. Is there a TypeScript library available for node-fetch or are there better alternatives?
* The unit test is implemented with Jest. How would we mock an incoming EventGrid event? How would our (mocked) Azure Context object look like in TypeScript?
* There are several fixed schemas in the code. (Incoming event structure, outgoing POST body, returning response body). How could TypeScript help in supporting and enforcing this schemas in the code?