Serverless Computing - just a hype?

An introduction to Azure Functions

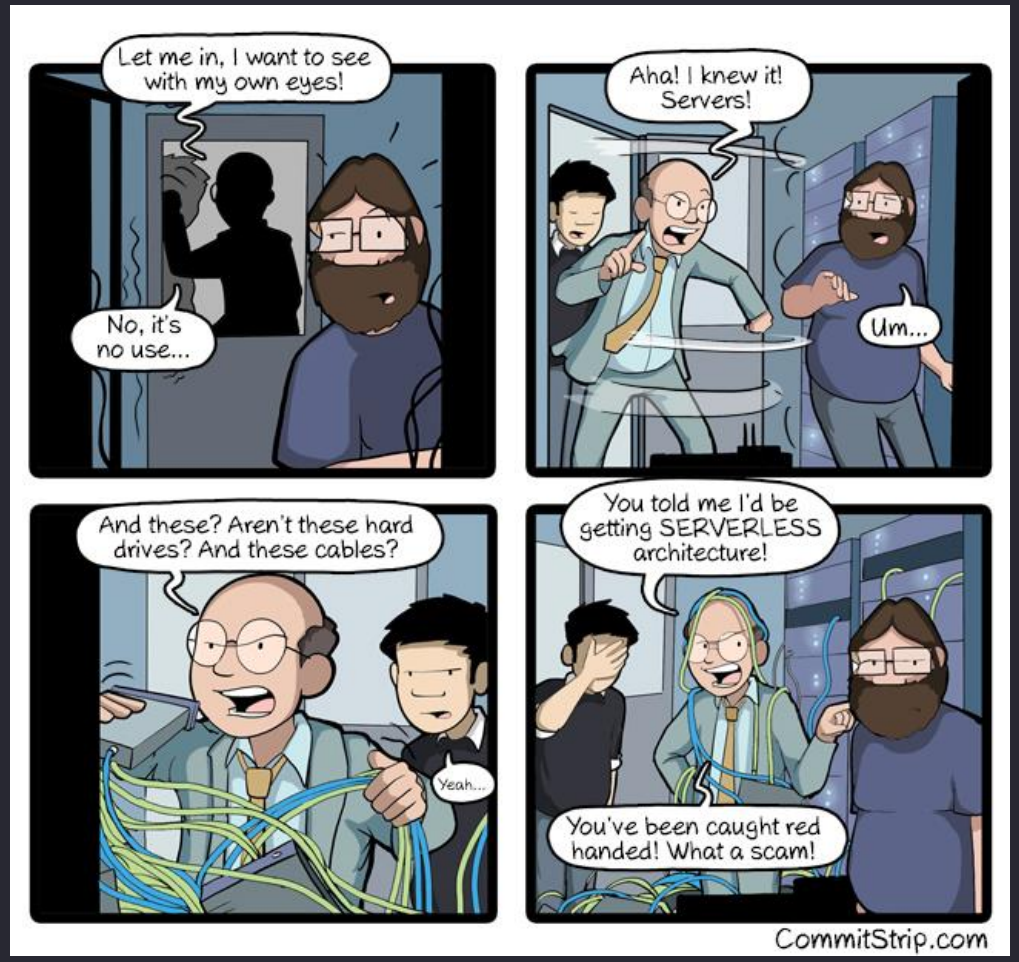
Robert Schlaeger
Developer
schlaeger@medialesson.de

@RobAnybody_

Sebastian Jensen
Developer
jensen@medialesson.de

@tsjdevapps

Serverless?!



http://www.commitstrip.com/en/2017/04/26/servers-there-are-no-servers-here/

What are Azure Functions?

Code + Events + Data = Azure Functions

Some Benefits

Abstraction of Servers

Sub-second billing

Speed & Availability

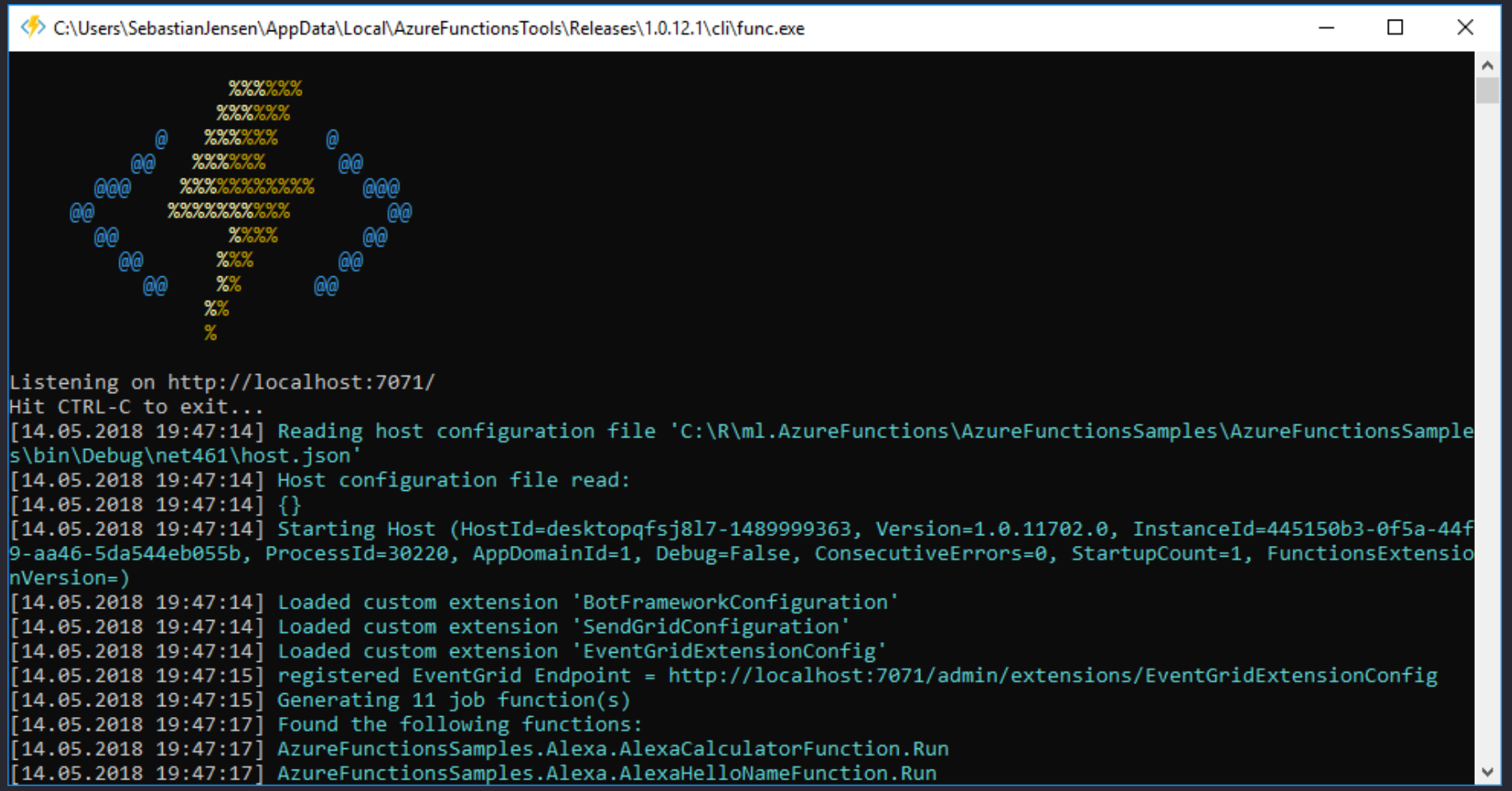
Instant Scale

Focus on Logic

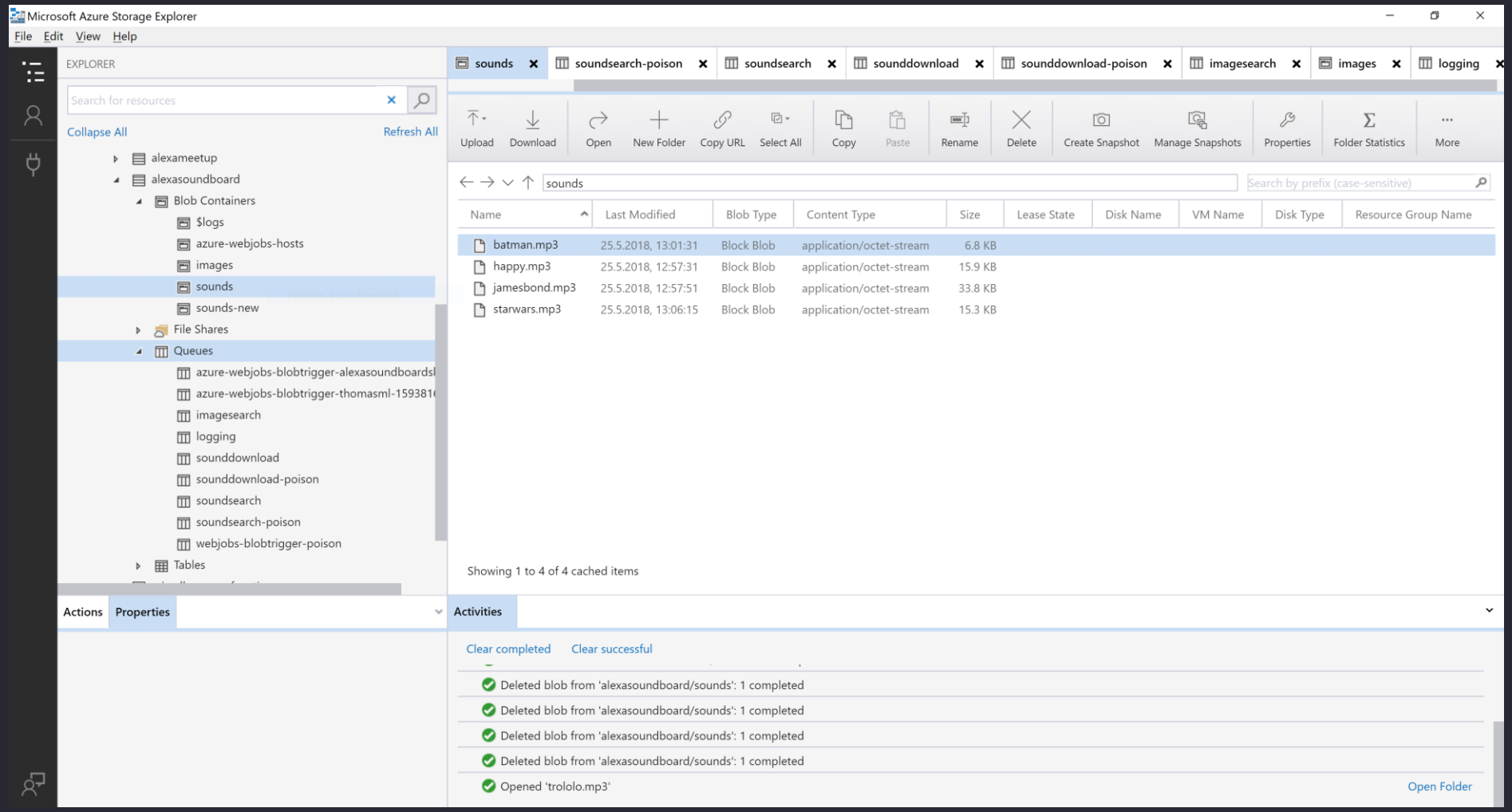
Different Languages

Language Support

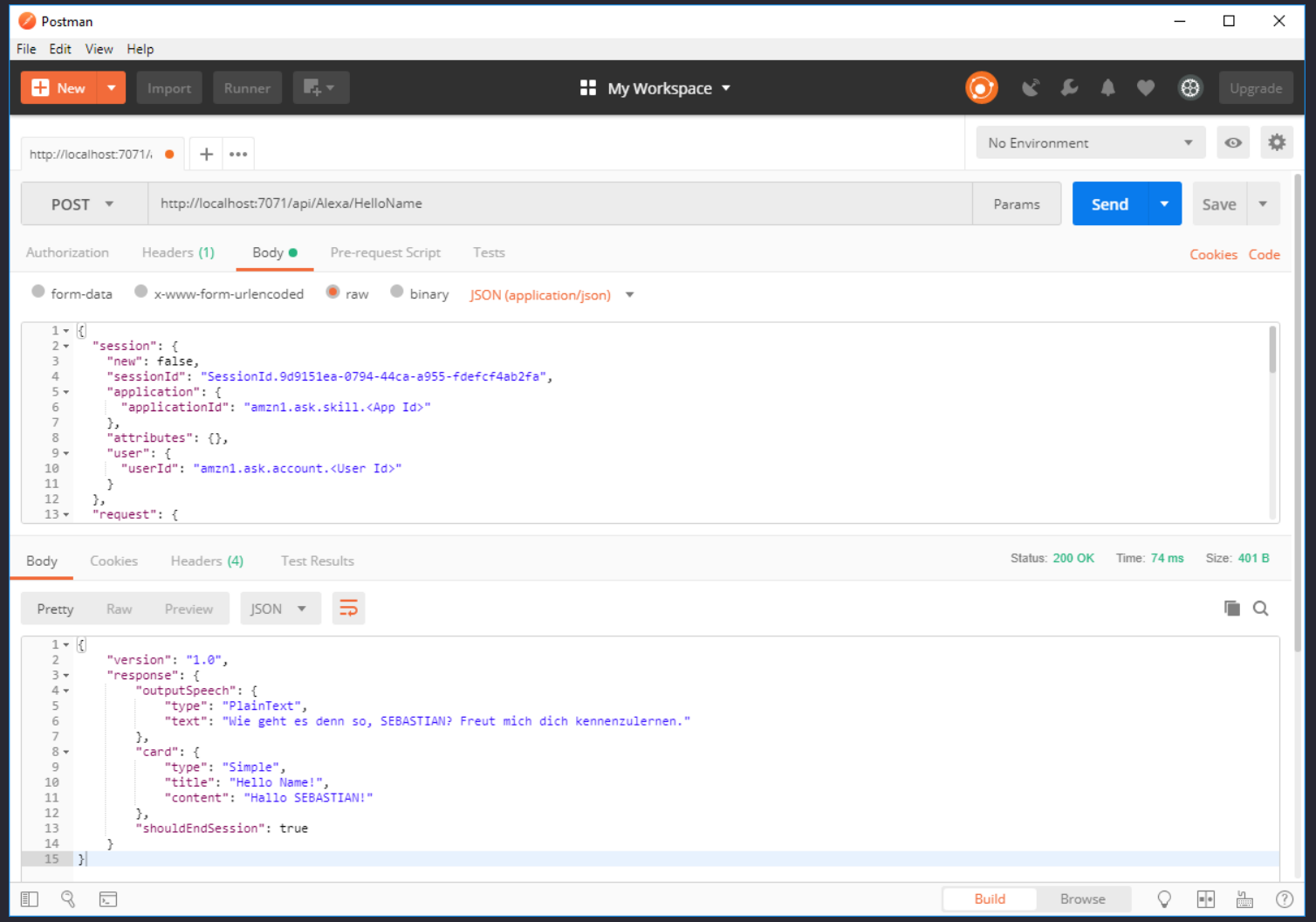
Local Testing



Local Testing



Local Testing



Hello World/Name – C#

```
using System.Net;

public static async Task<HttpResponseMessage> Run(HttpRequestMessage req, TraceWriter log)
{
    log.Info("C# HTTP trigger function processed a request.");

    // parse query parameter
    string name = req.GetQueryNameValuePairs()
      .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0).Value;

    if (name == null)
    {
        // Get request body
        dynamic data = await req.Content.ReadAsAsync<object>();
        name = data?.name;
    }

    string response = "Hello " + name + ", welcome to the functions real world!";

    return name == null
      ? req.CreateResponse(HttpStatusCode.BadRequest,
                           "Please pass a name on the query string or in the request body")
      : req.CreateResponse(HttpStatusCode.OK, response);
}
```

Hello World/Name – JS

```
module.exports = function (context, req) {
    context.log('JavaScript HTTP trigger function processed a request.');

    if (req.query.name || (req.body && req.body.name)) {
        context.res = {
            // status: 200, /* Defaults to 200 */
            body: "Hello " + (req.query.name || req.body.name) +
              ", welcome to the functions real world!"
        };
    }
    else {
        context.res = {
            status: 400,
            body: "Please pass a name on the query " +
              "string or in the request body"
        };
    }
    context.done();
};
```

Hello World/Name – F#

```
#r "System.Net.Http"
#r "Newtonsoft.Json"

open System.Net
open System.Net.Http
open Newtonsoft.Json

type Named = {
    name: string
}

let Run(req: HttpRequestMessage, log: TraceWriter) =
    async {
        log.Info(sprintf
            "F# HTTP trigger function processed a request.")

        // Set name to query string
        let name =
            req.GetQueryNameValuePairs()
            |> Seq.tryFind (fun q -> q.Key = "name")

        match name with
        | Some x ->
            return req.CreateResponse(HttpStatusCode.OK, "Hello " + x.Value
                                      + ", welcome to the functions real world!");
        | None ->
            let! data = req.Content.ReadAsStringAsync() |> Async.AwaitTask

            if not (String.IsNullOrEmpty(data)) then
                let named = JsonConvert.DeserializeObject<Named>(data)
                return req.CreateResponse(HttpStatusCode.OK, "Hello " + named.name
                                          + ", welcome to the functions real world!");
            else
                return req.CreateResponse(HttpStatusCode.BadRequest, "Specify a Name value");
    } |> Async.RunSynchronously
```

Demo

Triggers

- Defines the invocation of the function
- Must have exactly one trigger
- A trigger has some associated data with it
- Contains the payload that triggered the functions

Bindings

- Means of connecting to data from the code
- 2 types of bindings: Input Bindings and Output Bindings
- Bindings are optional
- Can have multiple input and output bindings

Voice Assistant: Alexa

Alexa

- Amazon
- April 2014
- Echo Family
- Windows-PCs later this year…

Wording: Skill

- Voice Experiences, which can be developed by third parties
- Extends the available functions
- "Voice Apps" with focused functionality
- Are currently available for free
- Be activated or deactivated via voice commands or companion apps

Demo

Pros of Azure Functions

- Azure Functions are cheap.
- Azure Functions are simple for simple scenarios.
- The amount of code you write in a function will probably be less than writing the same behavior outside of Azure Functions.

Cons of Azure Functions

- The languages and the runtimes for Azure Functions are not specialized.
- Deploying, authoring, testing, and executing a function is difficult outside of Azure.
- Startup time of the function is sometimes a little bit slow.

Resources

- Azure Functions
  - https://azure.microsoft.com/en-us/services/functions/
- Cognitive Services
  - https://azure.microsoft.com/en-us/services/cognitive-services/
- Botframework
  - https://dev.botframework.com/
- Slides & Code
  - https://github.com/tsjdev-apps/meetup-azurefunctions-demos

Any questions?

medialesson

Thank you for your interest!

www.medialesson.de

medialesson