Service Oriented Architecture

Application Design – Recap

Lecture 10

Dr. Alessio Bonti


Recap …….

What is a web service?

CONCEPTS

What makes a piece of code, a web service?

From the perspective of security, multi-tenant applications pose additional challenges with respect to single-tenant applications.

These challenges are primarily due to keep isolated the different tenants whithin a common computing environment.

Some of the key problems are:

01 IT IS ACCESSIBLE THROUGH AN HTTP INTERFACE

02 IT IS INDIPENDENT FROM THE FRONT END IMPLEMENTATION

03 THEY SERVE A PURPOSE ..... OR SIMPLY SERVE

Security

CONCEPTS

RESTFUL SERVICES

It identifies a service which is based on rest principles :

Use http methods explicitly , be stateless, expose urls in a directory structure.

01 GET – Retrieve a Resource

02 POST – create a resource

03 PUT – change the state of a resource

04 DELETE – remove a resource

SOA 101 again

CONCEPTS | Is this a web service?

```
console.log('I am a web service')
```

SOA 101 recap

CONCEPTS | What about this one?

```
const express = require('express')
const app = express()

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

SOA 101 recap

CONCEPTS | What about this one?

```
const express = require('express')
const app = express()

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

SOA - getting complex

CONCEPTS | What can you tell me about this now?

```
const express = require('express')
const app = express()

function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
}
function getFunnyQuote(){

    var quotes=['This is funny','This is super funny', 'This is hilarious'];
    var quote=quotes[getRandomInt(0,2)]
    return quote

}

app.get('/funnyquote', (req, res) => res.send('Hello World!'))
app.get('/', (req, res) => res.send('Hello World!'))
app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

Monolithic vs SOA

CONCEPTS | This is more like a service

!!!!

MAKE SURE YOU ARE NOT MONOLITHIC

When creating your architecture, you must take into consideration many aspects, and the first one is communication , when do you keep things in one place, when not?

PROS

- *The application and the code, reside all in the same place, meaning that there is no comunication you need to wait for, also, it is simpler to monitor, and you have less strain on the resources. For example, if you had micro architectures, you may need to replicate the same server many times.*

CONS

- *Things are coupled together, meaning that integration is more difficult, and resources management will not be easy when it will be time to scale. Also, many points of failure unless every point has a solid redundancy plan in place.*


Questions?