AGILE PROJECTS 2018

Elemer Lelik, Ericsson Hungary

elemer.lelik@ericsson.com

2018.02.07

1. OER

- Create an OER (Octet Encoding Rules) encoder/decoder for ASN.1 at minimum for the simple types and simple structures. Implementation can based on the language of your choice. Any available open source libraries can be used.

- Reference: http://www.oss.com/asn1/resources/books-whitepapers-pubs/Overview%20of%20OER.pdf

2. REST API I

- Assume you have a web service based on a REST API with no formal specification.

- Create a client for the web service that can be used in its testing. Implementation can based on the language of your choice. Any available open source libraries can be used.
- Reference:
- Representational state transfer
- https://en.wikipedia.org/wiki/Representational_state_transfer

3. REST API II

- Assume you have a web service based on a REST API formally specified with openAPI 3.0
- Use available Swagger libraries to create a client for the web service that can be used in its testing. Implementation can based on the language of your choice. Any available open source libraries can be used.
- References:
- Representational state transfer
- https://en.wikipedia.org/wiki/Representational_state_transfer
- https://github.com/OAI/OpenAPI-Specification
- https://swagger.io/

4. IOT MQTT

- Assume you have an IoT device with a small number of sensors you want to connect to an MQTT broker.
- Create an MQTT client that simulates the behaviour of the device. Implementation can based on the language of your choice. Any available open source libraries can be used.

- Reference:
- http://mqtt.org/
- https://iot.eclipse.org/getting-started

5 .NOSQL

- Create a NoSQL client that connects to a NoSQL database (MongoDB, Redis, CouchDB etc.) of your choice. The client should be capable of performing basic transactions.
- Implementation can based on the language of your choice.
- Any available open source libraries can be used.

6. FUZZING JSON

- Create a function that is capable of fuzzing a JSON document supplied as input.
- Explore both mutational and generation-based fuzzing.
- The output of the function is the fuzzed JSON document.
- Implementation can based on the language of your choice.
- Any available open source libraries can be used.
- Reference:
- https://en.wikipedia.org/wiki/Fuzzing

ERICSSON