

Service Virtualization as Code

Sachin Srivastava

August 2017

Why SV as Code?

Current Scenario for Developers

- Learn DevTest
- Configure endpoints(Change Code to Configure Virtual Service Endpoint)
- Dependencies

Solution

- Should integrate with their environment and framework
- Need easy configuration
- Reduced dependencies

What is SV as Code?

New, lightweight way to define virtual services

- Helps save valuable time

Delivers a simple yet powerful Java library

- Provides an easy to use API

Create and run VS as part of Junit testing

```
import static com.ca.svcode.protocols.http.fluent.HttpFluentInterface.*;
import static org.junit.Assert.*;

import com.ca.svcode.engine.junit4.VirtualServerRule;
import org.junit.*;

public class ExampleTest {

    @Rule
    public VirtualServerRule vs = new VirtualServerRule();

    @Test
    public void exampleTest() {
        // virtual service definition
        forGet("http://www.example.com/time").doReturn(
            okMessage()
                .withJsonBody("{\"timestamp\":\"1498838896\"}")
        );

        // application connects to http://www.example.com/test and retrieves JSON response
        int currentTimeStamp = Application.retrieveCurrentTimestamp();

        // received timestamp check
        assertEquals(1498838896, currentTimeStamp);
    }
}
```

Key Features

Create and run VS directly in unit testing code

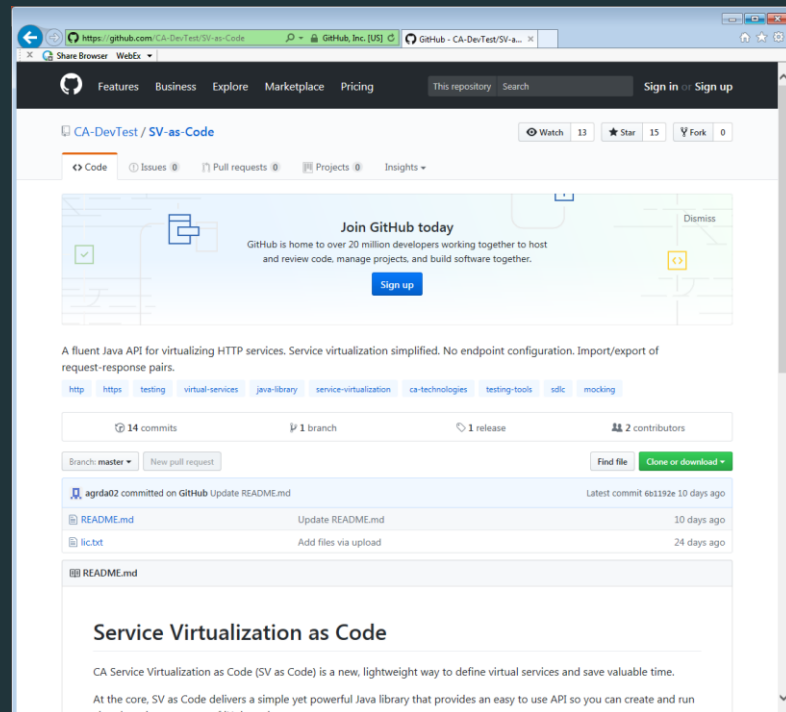
- Convenient to developers

Reference libraries

- Easy to download ready to use
- Available in github

No need for configuration tweaks

Runs VS within app memory



<https://github.com/CA-DevTest/SV-as-Code>

Features

Easy to use interface

- Integrate with existing IDE (Eclipse, IntelliJ, Netbeans)

Auto generated end points

- In process virtualization

Supports Asset Sharing

- Ability to import and export request / response pairs

Virtualization support

- Supports web service virtualization needs (REST and SOAP services)

SV as Code vs DevTest

SV as Code

- Targeted at Developers
- Support for unit testing
- Configurable only at code level

DevTest

- Targeted at power users, testing teams etc.
- For all types of testing
- Expansive support for use cases, protocols etc.



Sachin Srivastava

Engineering Services Architect

Sachin.Srivastava@ca.com

