Market logs:

Charlie horn<[chorn@inttechnologies.com](mailto:chorn@inttechnologies.com)>

**Sujit Kumar** <Sujit@inasolution.com>

agenda:

DevTest Portal-to increase ease of use for new user

serverside recorder-for HTTP recordings enabling shared install and ease of use.

service image editor-in web based UI with advanced searching,request/response matching and annotations

learning mode-to automatically learn changes in the live system and apply them to the virtual service

opaque data processing-adds atrtificial intelligence to the creation of virtual service

additional protocol support-for WADL,RAML and enhanced REST and updated SWIFT support

CA Service Virtualization 8.0 is the most feature rich covering most dynamic change for customers and people who need service virtualization to remove constraints in the environment. it is found in 2007 since then it is enhancing its importance.

DevTest Portal:

DevTest Portal is not comprehensive. in this release we talk about most used work flows that work in work station and probably its difficult for new user

left side- home,create,manage,monitor,application insight,settings

middle-getting started,quick links,help

under getting started:

how to record and save a virtual service

how to edit a virtual service

how to create a test

how to monitor a test

how to manage test artifacts

how to manage a ticket

how to work with the shelf

how to document a transaction for a manual test

we have preferences where we can enable help for first time user

under virtual service current status we can also see

total transactions

deployed

recorded

in left side under CREATE after expanding we can see-

API Test

virtualize website(http/s)

virtualize JDBC

virtualize using RR(request/response) pairs.---create virtual service--screenshot in phone ex:hello2-- upload rr pairs

vapi.ca.com--virtual api cloud

here we can create your own virtual API

to make sure we virtualized using RR pairs:

localhost:1505---to open devtest console localhost:1507 to open devtest portal

enter username and password login

-server console

in the left navigation select VSE---thats the way how we deploy the virtual service--in the right side we can see hello2 running which we deployed.from there we can download the backing archive for the hello2 virtual service.

virtualize website(http/s)--INTRODUCED IN 8.0

record configure sace

record:

Http transfer protocol

client---vs recorder---server

java localhost:8080

we want the client to point to port 2343 first and the recorder automatically goes to backend server

click start recording and perform transactions on front end kiosk--we can see the change of recording in right side yellow color

here we performed deposit check balance

configure: screenshot

select necessary webservices and click on update transaction view.

and save

now open virtual service recorder--kiosk

since we have recorded only deposit,check balance if we try to perform withdraw then the front end ui ie kiosk will not allow you to do that.

screenshot

adding annotations---on signature right click and add note like leena did this transaction on apr13... this is called adding annotations.

we have 9 different ways to run virtual service.

ability to pass through to live service

ability to record the changes that happened on live service

in 8.0, with the help of the client trigger the virtual service, if the virtual service doesn't have response it will pass through to the live service. so we can work with the live service and virtual service at the same time

in addition to that we have an additional functionality not just pass through the live service, we have the ability model client directly go the live service and then back to virtyal service if the live service does not have the response.

LEARNING MODE:

CHALLENGE:

Ensuring a virtual service remains in sync with live system is inherently time consuming.

SOLUTION:

1. compares responses betwenn the virtual service and live system to automate the creation and updating of virtual services.

2. assures that virtual service is a faithful simulation of the current live system.

ODP:

with opaque data processing we can figure it out how to match,request and response without needing to know about actual protocol.

screenshot

protocol support: screenshot

assertion,filters,data sets,vsi,vsm,vse.regristry,work station,local properties,moking--replicating actal backend data and virtualization--to remove dependency on backend,not data dependent,

stubing--performance testers will do this for example checking response time, mq-proof of concept quick and dirty

virtualization using webservices

virtualization using mq