Hand off to Celtic Testing Experts POC

- QA Team

Table of Contents

1. Web Service 3

2. Service API details 3

2.1 App Session –API details 3

2.2 User Session –to create profile - API details 3

2.3 User Session –SSO login - API details 4

2.4 Location –Add - API details 4

2.5 Location –Retrieve - API details 4

2.6 Channel or end point –Add - API details 4

2.7 Channel or end point –Retrieve - API details 4

2.8 Demographics -Add - API details 5

2.9 Demographics -Retrieve - API details 5

3. Monitoring Linux server 5

4. Analyzing performance metrics 5

5. User journey recommendations 5

6. Appendices 5

6.1 Snapshots from cocoa rest client for mac 5

# 1. Web Service

The URL to access the webservice exposed for proof of concept demo is below.

* <http://qcups0b00.dev.weather.com:8080>

(Current QA env url for now, To be changed before releasing this document to Celtic testing experts guys)

# 2. Service API details

The following API’s that can be used as part of POC scenarios(User journeys)

* http://qcups0b00.dev.weather.com:8080/dsx/session
* http://qcups0b00.dev.weather.com:8080/dsx/p
* http://qcups0b00.dev.weather.com:8080/dsx/p/locations
* http://qcups0b00.dev.weather.com:8080/dsx/p/endpoints
* http://qcups0b00.dev.weather.com:8080/dsx/p/demographics
* http://qcups0b00.dev.weather.com:8080/dsx/p/settings/application

## 2.1 App Session –API details

Coco Rest client for mac or Advanced Restful client for Chrome browser are the ones widely used in our team. Please note session validity is set to 30 minutes by default.

API: <http://qcups0b00.dev.weather.com:8080/dsx/session>

Method type: POST

Request body: testApp-0000

Context-type: Json

Response: Cookie1(app session id specific to testApp-0000 profile)

Response code: 200 OK

## 2.2 User Session –to create profile - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p

Method type: POST

Request header: Cookie: Cookie1(this cookie1 is grabbed from the response body of previous request)

Request body: {"provider":"wx","id":"uma5","token":"test5"}

Context-type: Json

Response code: 200 OK

## 2.3 User Session –SSO login - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/sso

Method type: POST

Request header: Cookie: Cookie1(this cookie1 is grabbed from the response body of previous request)

Request body: {"provider":"wx","id":"uma5","token":"test5"}

Context-type: Json

Response: Cookie2 (app cum user cookie specific to uma5 and testApp-0000 combination)

Response code: 200 OK

## 2.4 Location –Add - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/locations/umaloc

Method type: PUT

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: {"tag": "home", "locId": "15217", "locType": 4, "countryCode": "US"}

Context-type: Json

Response code: 200 OK

## 2.5 Location –Retrieve - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/locations/umaloc

Method type: GET

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: None

Context-type: Json

Response: {"tag": "home", "locId": "15217", "locType": 4, "countryCode": "US"}

Response code: 200 OK

## 2.6 Channel or end point –Add - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/endpoints/umaep

Method type: PUT

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: { "chan": "smtp", "addr": "uma.kandaswamy@weather.com" }

Context-type: Json

Response code: 200 OK

## 2.7 Channel or end point –Retrieve - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/endpoints/umaep

Method type: GET

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: None

Context-type: Json

Response: { "chan": "smtp", "addr": "uma.kandaswamy@weather.com" }

Response code: 200 OK

## 2.8 Demographics -Add - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/demographics/gender

Method type: PUT

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: F

Context-type: Json

Response code: 200 OK

## 2.9 Demographics -Retrieve - API details

API: http://qcups0b00.dev.weather.com:8080/dsx/p/demographics/gender

Method type: GET

Request header: Cookie: Cookie2 (this cookie should be grabed from the response of sso login 2.3)

Request body: None

Context-type: Json

Response: F

Response code: 200 OK

# 3. Monitoring Linux server

Below are most important monitoring graphs we would want to configure in the tool to monitor all our linux servers.

Vmstat

Iostat

Sar

Mpstat for multiprocessor monitoring

Pmap

Netstat

Tcpdump

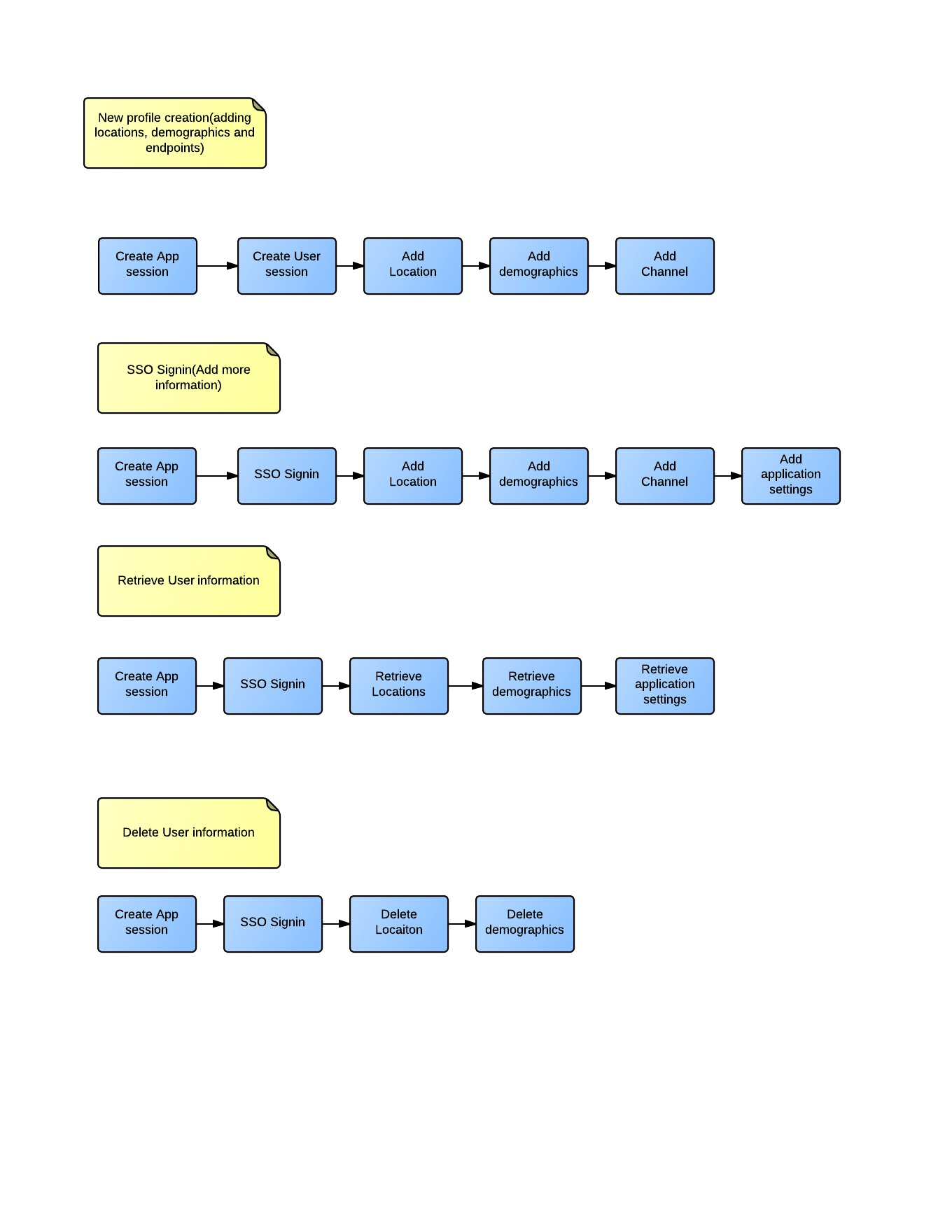
Jstat for JVM monitoring

# 4. Analyzing performance metrics

Below are some of the major graphs that we would want the tool to generate for analyzing our performance metrics other than resources monitoring.

* Active sessions graphs
* Average response time Vs Running simultaneous users
* Transactions per sec Vs Running simultaneouse users
* Erros per second Vs Running simultaneous users
* Transaction summary graphs
* Throughput graphs
* JVM monitoring graphs

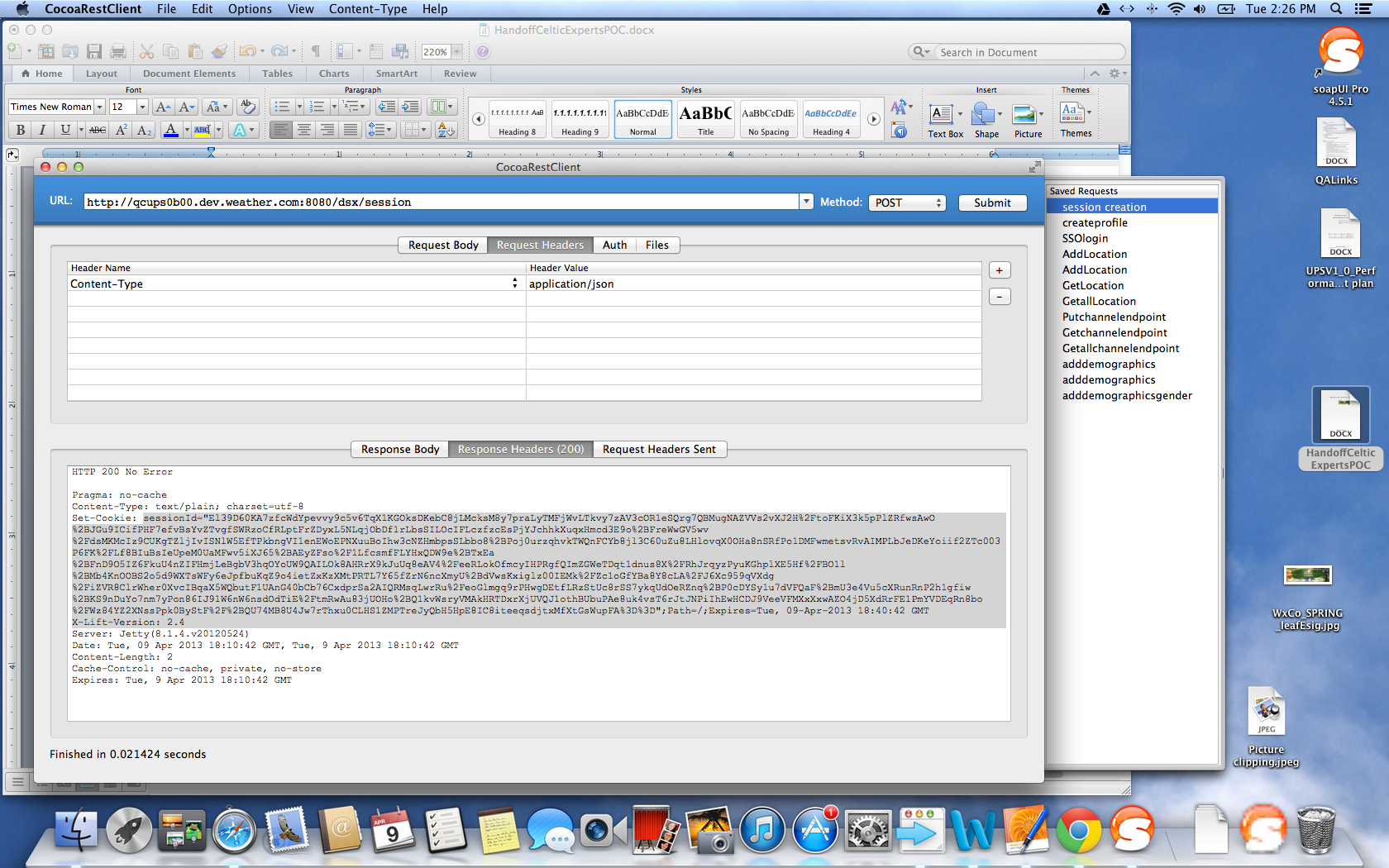
# 5. User journey recommendations



# 6. Appendices

## 6.1 Snapshots from cocoa rest client for mac

Below snapshot (appidsession 1) shows request 2.1 and response cookie in the response header to grab the dynamic data and substitute in the header for next request



appidsession 1