

PF Keys and Acronyms



Max Bender | Torin Reilly

About Us

- Marist College Class of 2017
- Computer Science Majors with Software Dev Concentration
- Started with IBM, January of 2015 working through the year
- Joined the zPET Integration Test team at IBM Poughkeepsie



LCST /e System z Platform Evaluation Test

The Final Verification

Z/OS | CICS | IMS | DB2 | WebSphere MQ |
WebSphere Application Server | Tivoli | InfoSphere



We are a team of system programmers and testers that run a Parallel Sysplex on which we perform the final verification of a z/OS release and System z hardware and System Storage before they become generally available to clients. We gather our experiences and recommendations and document them here in our blog.

The Beginning - Torin

- Interested in computers at a young age
- DOS games and shareware
- Father fixed computers for fun
- Imitating my favorite games
- Creating websites for my “companies”
- Android hacking and development



The Beginning - Max

- Growing up with the birth of Minecraft
 - Highly moddable, lots of logic
- How do I host a server for my friends and I?
 - Introduction to Unix systems
 - Lua
 - Java
- Website Development, saved up for Microsoft Frontpage
- Basic penetration techniques
 - Bypass parent's internet timeouts
 - Play videogames.... all night.



The First Day of Work



MQ

IMS

CICS

DB2

WAS

RACF

OMEGAMON



Unix -> ISPF (z/OS)

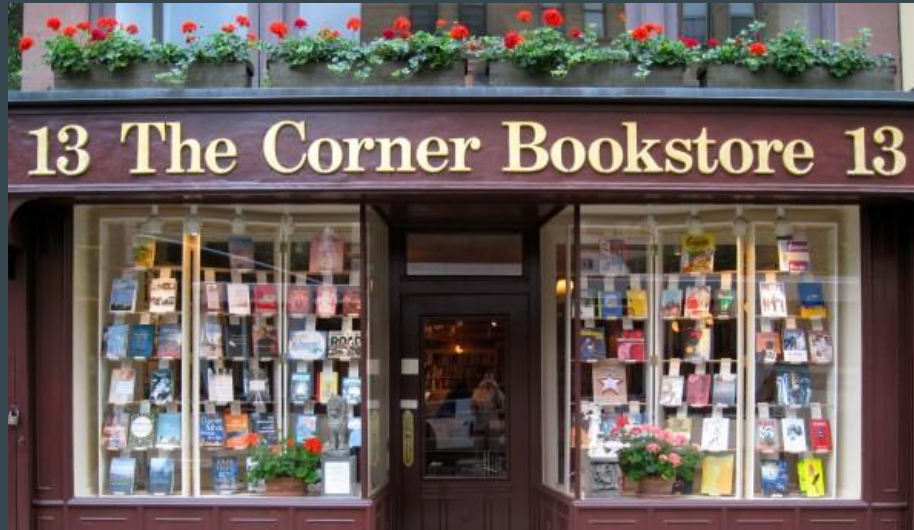
- Experience with Linux-based CLI Server Administration
 - Does not translate to ISPF panels
- PF Keys, not the same as F Keys?
- Procs/JCL
- Knowledgeable Team
 - Second Nature
- High Learning Curve

```
Menu Utilities Compilers Options Status Help
z/OS Primary Option Menu
Option ==> 13_
0 Settings      Terminal and user parameters      User ID . : MMORA
1 View          Display source data or listings   Time. . . : 12:50
2 Edit          Create or change source data      Terminal. : 3278
3 Utilities     Perform utility functions         Screen. . : 1
4 Foreground    Interactive language processing   Language. : ENGLISH
5 Batch         Submit job for language processing Appl ID . : ISP
6 Command       Enter TSO or Workstation commands TSO logon : DBAUSER
7 Dialog Test   Perform dialog testing            TSO prefix: MMORA
P IBM Products  IBM program products             System ID : TESTMVS
10 SCLM         SW Configuration Library Manager MVS acct. : 12345678
11 Workplace    ISPF Object/Action Workplace     Release . : ISPF 5.7
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications
```

The First Project

Modernizing the Bookstore

- Existing Workload → Mobile Framework
- Large codebase, many different editions of original code



What is the Bookstore?

- E-Commerce Platform
 - Modeling Amazon-like workflow

Search

Book Details

Cart

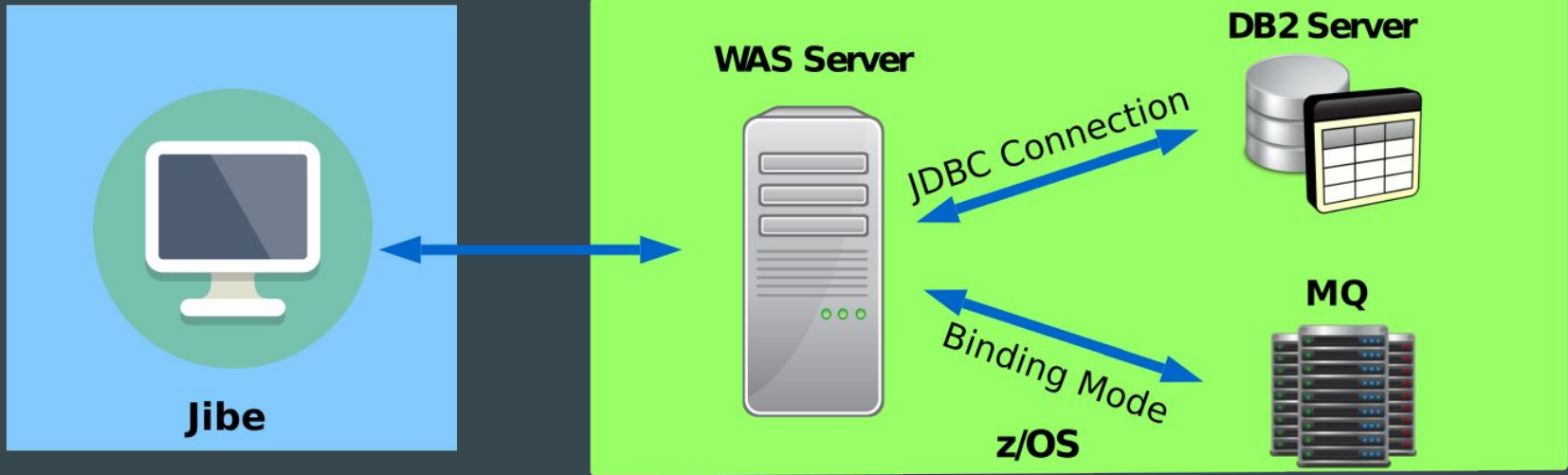
Checkout

Credit Card Processing



WebSphere

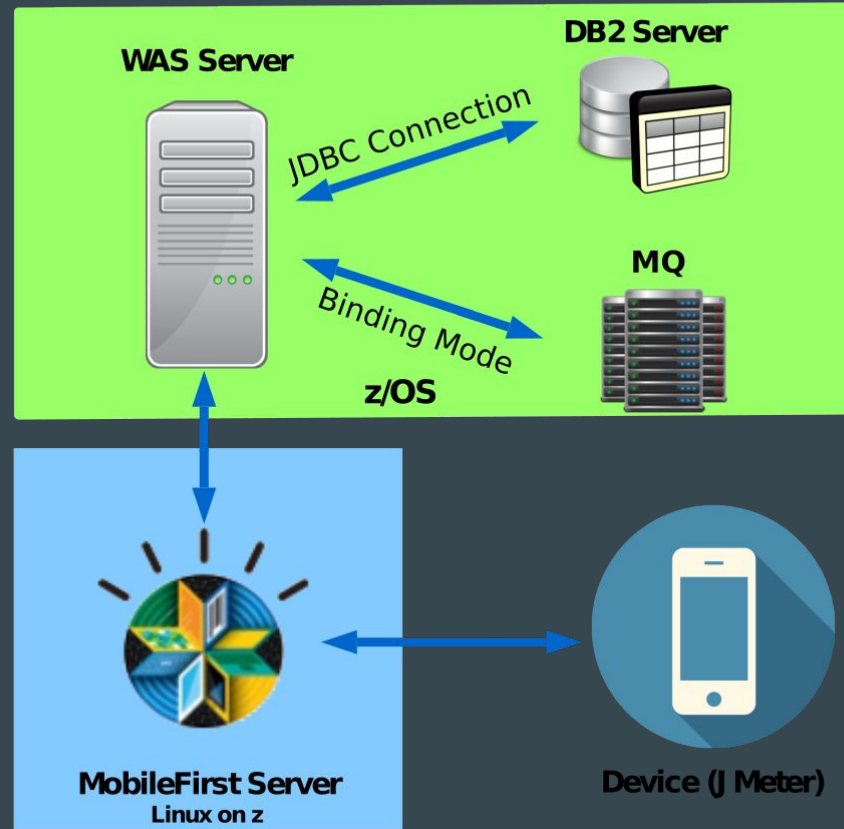
Original Bookstore Workflow



How are we going to make this work?

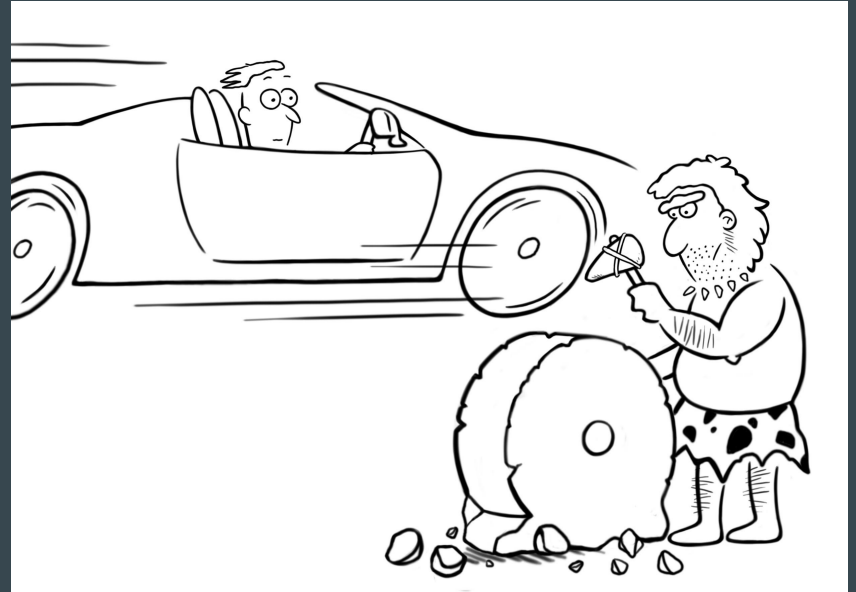
- The Old
 - DB2
 - MQ
 - Ported some Java Classes from Bookstore -> MobileBook
 - WAS Environment
- The New
 - IBM Mobile First
 - RESTful APIs Sitting on WAS server
 - Apache JMeter (Open Source)

Mobile Bookstore Workflow



What can we re-use?

- Took a look at the old code
 - Which parts are relevant? Which parts aren't..
- Don't want to reinvent the wheel
- Don't want to inherit problems



A RESTful API

What to build it in

- Java Servlets
 - Easier to port existing code
 - Infrastructure already in place
 - Familiar with Java
 - Lots of doc to help us get started
- Hosted on a WAS server on Z
 - Able to use native JDBC to interact w/ DB2
 - MQ Bindings mode, ported from Bookstore

Porting Code vs. Writing New

- Bookstore has thousands and thousand of lines of code.
- I wanted to start anew, Matt said no :(
 - School you start from scratch
 - Didn't have experience porting code/adding to existing code
- Matt talked to the importance of reusing code
 - Resilience of old code base
 - If it ain't broke, don't fix it
 - Logging standards
 - Configuration and Properties handling
- Great lesson, got experience in deciphering and reading someone else's code

Java Servlets

Java Servlets and WAS

- Project that encompasses all servlets and helper classes
- Export and Deploy as JavaEE WAR to WAS server
- WAS maps the servlets to URL endpoints
 - `x.x.x.x:1234/MobileBook/BookSearchRequest?isbn=x`
- Environments in WAS, learning to pull from native libraries, not importing.
- There were some hurdles and persistent issues we faced

The 15 Minute Wait

- Long wait times due to WAS updating policies while developing
- Make a fix/update ~> Export and upload War ~> 15 min wait to test/check updates
- With > ~150 updates to the code... a lot of waiting



General Design of Servlets

```
public class BookSearchRequest extends HttpServlet {  
  
    public AddToCart() { super(); }  
  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {  
  
        // Access GET/POST params with  
  
        var param = request.getParameter("PARAM_ID")  
  
        // Response setup  
  
        response.setContentType("application/json" | "text/xml" | "text/html");  
  
        PrintWriter out = response.getWriter();  
  
        out.print("SOME RESPONSE");  
  
    }  
  
}
```

Package Layout:

- ❖ bookstoreMobile.servlets
 - AddToCart.java
 - BookSearchRequest.java
 - Checkout.java
 - Details.java
 - Logout.java
 - RemoveFromCart.java

Request/Response

- Parameters passed to servlets as you would any HTTP request.
- Started with returning XML, ran into issues with foreign characters
- Moved to JSON later in the process, fixed issues

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
  <info>
    <session>ffd0j5mnIptVi71ziergooF</session>
    <bookResponse>
      <qty>1</qty>
      <isbn>1890627054</isbn>
      <title>The early intervention dictionary :
      <price>66.19000244140625</price>
      <author>Coleman, Jeanine G</author>
    </bookResponse>
    <sql>SELECT TITLE, AUTHOR_NAME, ISBN, RETAIL_PRICE
  </info>
```

JSON

```
{
  "session": "ffd0j5mnIptVi71ziergooF",
  "error": [],
  "bookResponse": [
    {
      "qty": 1,
      "isbn": "1890627054",
      "title": "The early intervention dictionary :
      "price": 66.19000244140625,
      "author": "Coleman, Jeanine G"
    }
  ],
  "sql": "SELECT TITLE, AUTHOR_NAME, ISBN, RETAIL_PRICE
}
```

The Cart

- Cart stored in the HTTP Session
- By doing this though, we didn't have a true RESTful API
- Concept of Stateful/Stateless APIs
 - According to the standard, each request should be independent from each other
- Let's use a database instead!
 - Cart stored in simple DB2 table, indexed by the Session Id



Adapting Bookstore Code → MobileBookstore

- Don't reinvent the wheel if you don't have to.
- Modified MQ and DB2 code to accept our version of the Cart
- Benefits
 - No need to re-write the code
 - The logging and error handling is already there
 - Same error and logging messages, consistency between the two variants
- Cons
 - If something isn't working, difficult to figure out where the program is breaking
 - Don't know the code, just accepting that it works

MobileFirst

General Overview

IBM MobileFirst



- MobileFirst Studio
 - Can deploy to a myriad of devices and Platforms (iPhone, Android, Web, Windows Phone)
 - Single Codebase
 - Create adapters to interface between MobileFirst Server and backend services
- MobileFirst Server
 - Can be updated/changed without the customer needing to download an update
 - Runs on traditional WAS or WAS Liberty
 - Administration Panels, Audit Logs
 - Web-based management GUI used to deploy and monitor enterprise applications in real-time
 - Adapters are installed here
 - Gateway between apps and back-end systems

Creating an App

- Written in HTML5/CSS/Javascript
 - Allows use of any responsive web frameworks
 - Each “page” is another app screen
 - Allows for rapid prototyping and changes
- Plugins for RDZ and Eclipse
 - Mobile Browser Simulator
 - Allow you to quickly test and see what your app will look like on a device
 - Gives you the ability to run a MobileFirst Development Server locally
- Front-end app, interfaces with adapters on the MobileFirst Server
 - Adds layer between your app and data
 - App does not have to directly access and format data

HTML



CSS



JS



Creating adapters

- Written in JavaScript
- Invoked within the application
- Adapter methods accept parameters and call the appropriate backend service
- Receive and format data from backend

```

/*****
 *  DETAIL FUNCTION
 *
 *  @param isbn
 *  @returns
 *****/
function detail(isbn) {
    var input = {
        method : 'get',
        returnedContentType : 'json',
        path : 'MobileBook/Details',
        parameters : {

            'isbn' : isbn
        }
    };

    return WL.Server.invokeHttp(input);
}

```

Mobile Bookstore User Flow

Initialize

Search for
Book

View Book
Details

Add To Cart

Checkout

Logout

No
associated
user
interface for
this call

Mobile Bookstore User Flow

Initialize

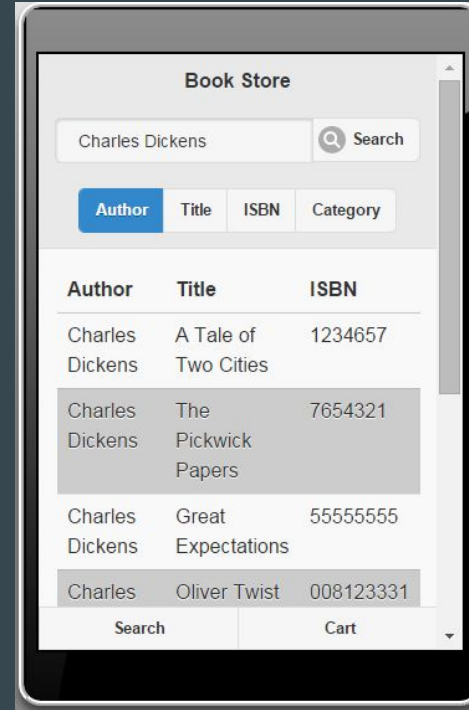
Search for
Book

View Book
Details

Add To Cart

Checkout

Logout



Mobile Bookstore User Flow

Initialize

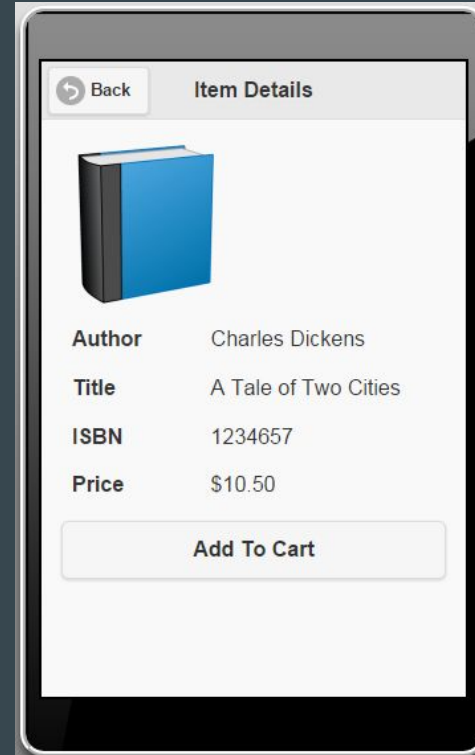
Search for
Book

View Book
Details

Add To Cart

Checkout

Logout



Mobile Bookstore User Flow

Initialize

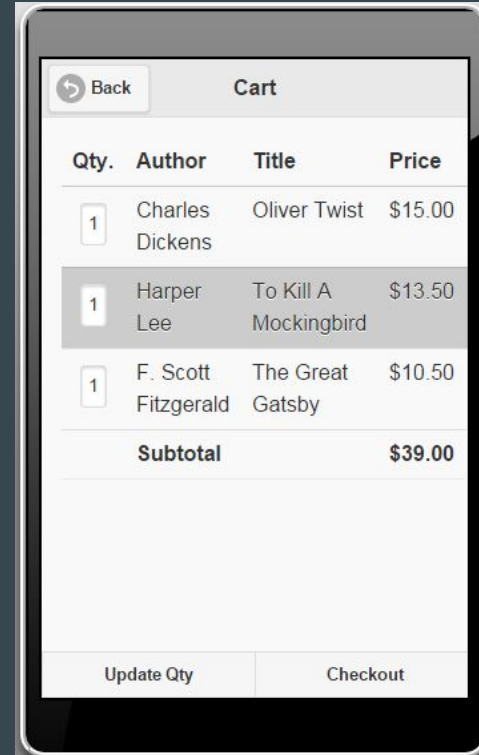
Search for
Book

View Book
Details

Add To Cart

Checkout

Logout



Mobile Bookstore User Flow

Initialize

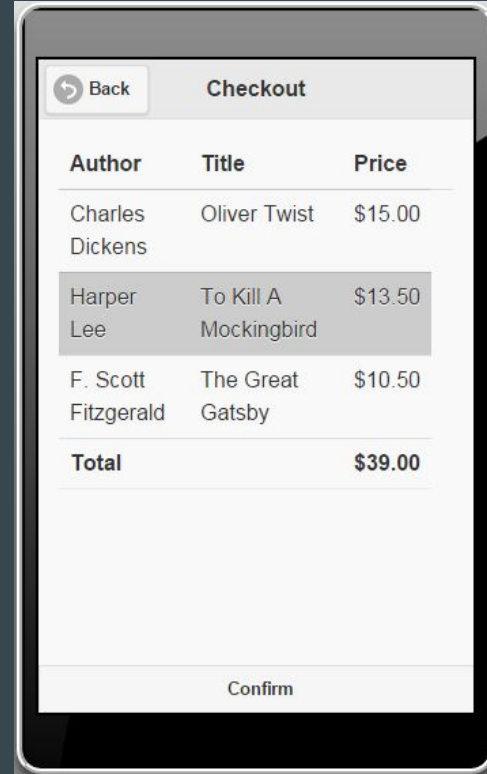
Search for
Book

View Book
Details

Add To Cart

Checkout

Logout



Mobile Bookstore User Flow

Initialize

Search for
Book

View Book
Details

Add To Cart

Checkout

Logout

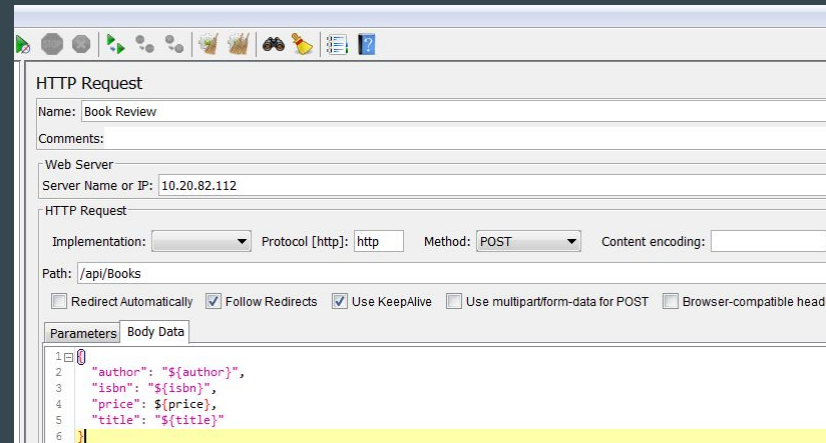
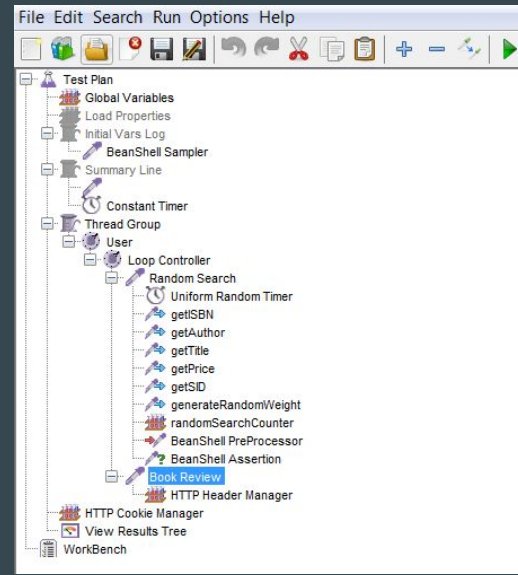
No
associated
user
interface for
this call

Driving This User Flow At Scale



Apache JMeter

- Open source load testing tool
- Can send all types of HTTP Requests
- GUI allows for easy creation of basic scripts
- Advanced use through scripting and ReGex
- Error reporting
 - Built in HTTP error reporting
 - Custom defined error reporting

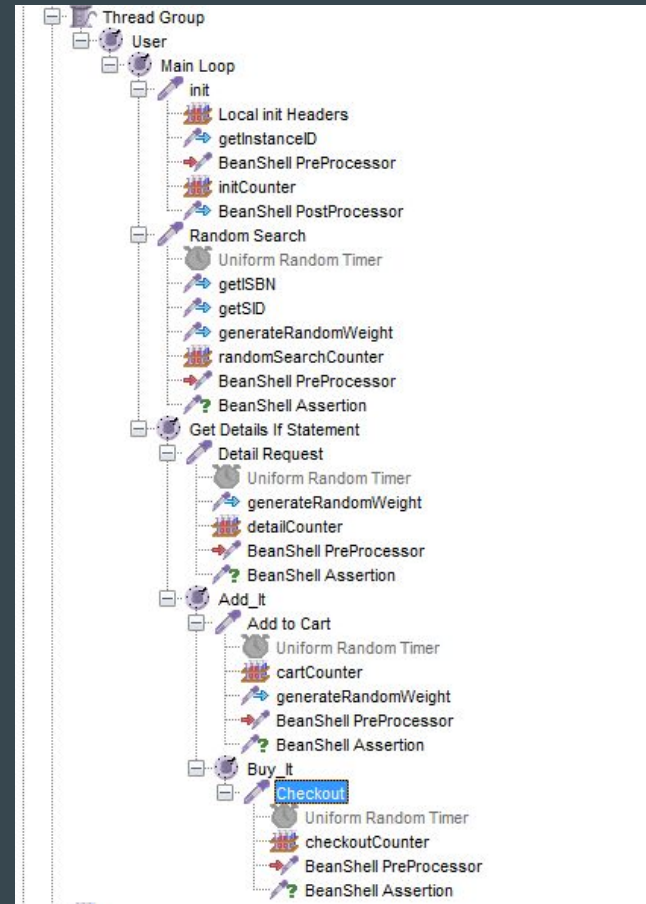


JMeter to MobileFirst Adapters

- JMeter simulates HTTP traffic from multiple synchronous users
- MobileFirst adapters were maintaining a single instance across all invocations.
 - Needed minor change to adapter configuration to function as “endUser”
- Invocation requires a unique instance id
 - id is obtained through init request
 - Must be added to all future request headers
 - Ensures it is actually a MobileFirst device making the calls
- MobileFirst adapters were limited to two simultaneous users by default
 - Needed minor change to adapter configuration “maxConcurrentConnectionsPerNode” property

Our Script

- Basic User Flow
- Listeners to determine success of requests
- Weighted Conditionals
- JMeter uses a simple, tree-based interface with “fill in the blank” test generation



Driving at scale: Bugs, Bugs, Bugs

- Found bugs not seen at previous testing scale
- Character encoding issues
 - Initially used XML
 - Switched to more modern/open JSON
- Not an experience learned in a classroom

JMeter on Windows, Linux, and z/OS

- Linux and Windows
 - Easy to set up
 - Designed and tested on these platforms
- z/OS
 - Write Once Run “Anywhere”
 - Experienced performance issues driving at scale
 - Heap size
 - Requires tuning

Future Plans

What's In Store

- Redundant HA solution for WAS/DB2
- NodeJS + Spark for Linux on Z ~ sentiment analysis
- z/OS Connect (API Endpoints for DB2)
 - CRUD Architecture
- JMeter 3



2016

Questions?