# Introduction to Stream Explorer and Oracle Event Processor

In this hands-on, we will work with Stream Explorer and Oracle Event Processor- along with JMS, REST, SOA Suite and EDN.

In the hands-on you are working with a VM running in Oracle Virtual Box. The VM contains the Linux Ubuntu operating system, an Oracle 11gR2 XE database and JDeveloper 12.1.3 with integrated WebLogic Server 12.1.3 that also contains the SOA Suite 12c Quick Start Installation. The Oracle Event Processor 12c is installed with the Stream Explorer "patch" applied to it. An OEP domain has been created. Note that this domain runs completely outside the integrated WebLogic Server domain started from JDeveloper and running the SOA Suite 12c.

The hands-on leads you through the following steps:

- learn the basics of Stream Explorer: how to get going with events produced from a CSV file and simple event analysis
- learn more advanced Stream Explorer techniques both in terms of the analysis (enrichment and pattern matching) and in terms of the integration and communication (using REST, JMS and EDN)
- learn how to export a StreamExplorer application to an OEP project that can be edited and fine tuned in JDeveloper, then redeployed to OEP

# 1. Getting started

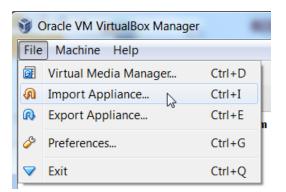
The folder *HandsOnStreamExplorerAndOEP\_May2015* contains the Virtual Machine that you will be using in this hands-on.

Name	Date modified	Туре	Size
StreamExplorerandOEPHandsOnInstructi	16-5-2015 17:32	Microsoft Word D	13.033 KB
🔁 StreamExplorerandOEPHandsOnInstructi	18-5-2015 18:44	Adobe Acrobat D	7.578 KB
💗 SX and OEP - JDev 12.1.3 with Oracle DB	16-5-2015 10:52	Open Virtualizatio	12.311.877
VirtualBox-4.3.26-98988-OSX.dmg	15-4-2015 17:33	DMG File	111.304 KB
💗 VirtualBox-4.3.26-98988-Win	15-4-2015 17:33	Application	108.541 KB

This VM is in a file called SX and OEP - JDev 12.1.3 with Oracle DB XE 11gR2 on

*UbuntuHandsOnSOA\_SIG19may2015.ova*. You can import this file into Virtual Box. If you do not yet have Virtual Box running on your machine, you can use the Virtual Box installer that is also in this directory.

Start Virtual Box. From the File menu, open the option Import Appliance.



The file selection dialog opens. Select file SX and OEP - JDev 12.1.3 with Oracle DB XE 11gR2 on UbuntuHandsOnSOA\_SIG19may2015.ova and press Open.

Import Virtual Appliance
Appliance to import
VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.
\HandsOnStreamExplorerAndOEP_May2015\SX and OEP - JDev 12.1.3 with Oracle DB XE 11gR2 on UbuntuHandsOnSOA_SIG19may2015.ova
Hide Description Next Cancel

Press Next in the Import dialog:

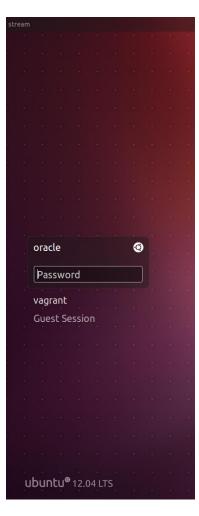
Appliance settings		
	d in the appliance and the suggested settings of the imported VirtualBox machines. You can chan ng on the items and disable others using the check boxes below.	nge ma
or the properties shown by addite circla		
Description	Configuration	
Virtual System 1		
쒛 Name	SX and OEP - JDeveloper 12.1.3 with Oracle Database XE 11gR2 on Ubuntu_1	
🗮 Guest OS Type	🐕 Ubuntu (64 bit)	
🔲 CPU	2	
RAM	6023 MB	
💿 DVD		
🗗 Network Adapter	Intel PRO/1000 MT Desktop (82540EM)	
🛇 Hard Disk Controller (IDE)	PIIX4	
Hard Disk Controller (IDE)	PIIX4	
4 🏈 Hard Disk Controller (SATA)	AHCI	
		-

Review and accept (or modify if you feel like it) the Appliance Settings. Press Import to start the import process.

Importing the appliance will take a few minutes. When done, the VM is shown in the Virtual Box user interface. Select it and press the Start icon to launch the VM.

💗 Oracle VM VirtualBox Manager	
<u>F</u> ile <u>M</u> achine <u>H</u> elp	
New Settings Start Discard	🚱 Details 🛛 💿 Snapshots
SX and OEP - JDeveloper 12.1.3 with Oracle Dat Powered Off	General   Name: SX and OEP - JDeveloper 12.1.3 with Oracle Database XE 11gR2 on Ubuntu   Operating System: Ubuntu (64 bit)   System   Base Memory: 4096 MB   Processors: 2   Boot Order: Hard Disk, CD/DVD   Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX   Display   Video Memory: 32 MB   Remote Desktop Server: Disabled   Video Capture: Disabled   Storage   Controller: IDE Primary Master:   [CD/DVD] VBoxGuestAdditions.iso (55,60 MB)   IDE Primary Master:   [CD/DVD] Empty   Controller:   SATA Port 0:   box-disk1_2.vmdk (Normal, 80,00 GB)   Audio   Nation   Network   Adapter 1:   Intel PRO/1000 MT Desktop (NAT)   VISB
	th.

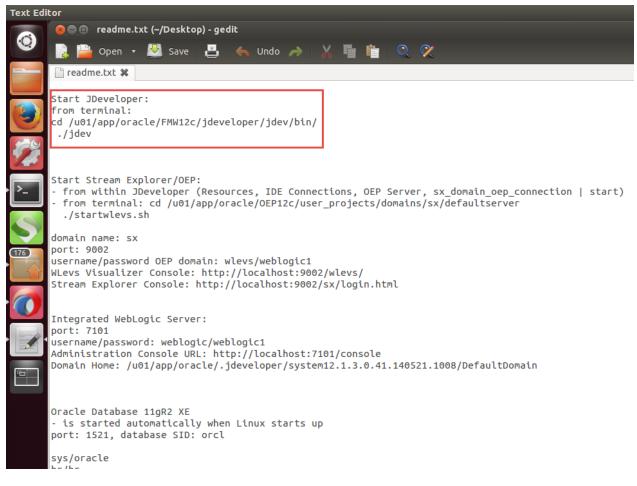
When the VM is running, login with username/password: oracle/oracle



Note: User *vagrant* with password *vagrant* can be used to assume root privileges.



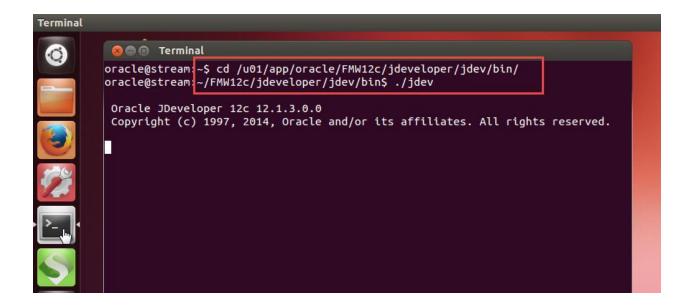
At the end of this document are some details about the configuration of the VM – such as database configuration and host folder mapping. The text document readme.txt that you find on the desktop also contains such details:



The highlighted lines provide the instruction for running JDeveloper.

Start JDeveloper with the following steps:

- open a new terminal (from the launcher icon)
- change the directory: cd /u01/app/oracle/FMW12c/jdeveloper/jdev/bin/
- run jdev: ./jdev

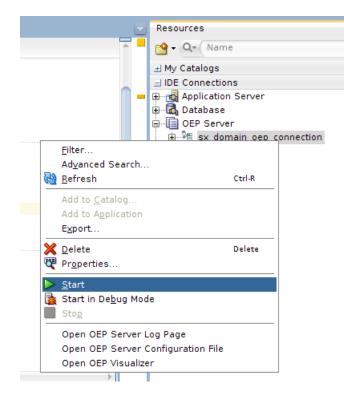


#### The familiar splash window appears:



And after some time, the IDE is opened.

Start the OEP Server using the OEP Server connection in the Resources window as shown in the next figure:



The console displays:

sx\_domain\_oep\_connection - Log × Live Issues: sx\_domain\_oep\_connection - Issues
OEP Server is starting, please wait...

and after some time, the server is running.

Both the StreamExplorer GUI and the Visualizer console can now be accessed in the browser.

Start Firefox from the launcher:



And use the bookmark to open StreamExplorer:

😹 Absolutely Typical	× +			
(Contemporal Contemporal Conte	gy. <b>amis.nl</b> /2011/12/12/absolutely	-typical-the-who	le-story-on-typ	es-and-how-they-pov
Most Visited ▼ ① Oracle	Event Processi 🔲 OEP Stream	Explore	RESTClient	WebLogic Admin C
	TRENI	DING Unle	ash the power	of Java API's on your '

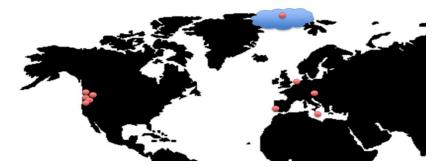
The login page appears. Use wlevs/weblogic1 as the credentials for the login.

OEP Stream Explorer L × 🕂	
Cocalhost:9002/sx/login.html	▼ C ] [Q, Search
🛅 Most Visited 👻 💭 Oracle Event Processi 💭 OEP Stream Explorer 🧧 RESTClient 💭 WebLogic Admin Co	n
	SIGN IN TO STREAM EXPLORER Business Insight into Fast Data

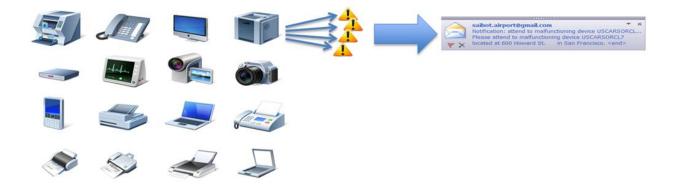
ORACLE

### 2. The Live Device Demo

We assume a large number of devices – such as printers, copiers, sensors, detectors, coffee machines – spread across the globe – and the cloud.



All devices continuously report their status, by sending a message every other second that contains their device identifier, a code that can indicate the healthy status or an error and some additional details. The sheer number of devices combined with the continuous stream of reports they sent in set the challenges perimeters within which we have to implement fast and effective monitoring. Our specific challenge is: "whenever a device reports an error code three times within 10 seconds, we consider that device broken, and action should be taken" (that also means that we do not spring into action on the first or even second fault report from a device). Additionally: we only require a single action for a broken device – once the action is initiated, we do not have to start an action again for that same device – unless of course it is broken again at a much later point in time.



This first step with Stream Explorer uses a CSV file as the source for events – because it is convenient during development, not because it is a realistic scenario for a production deployment. Normally, a live stream such as a JMS destination or an HTTP channel would be used. Note that the Stream Explorer implementation would be exactly the same for these other stream types. Stream Explorer processes the device signals. For signals that satisfy the requirements of a broken device, the information is enriched from a database with device details – such as the physical location of the device

jms C	DRACLE :	tream Explorer	Noto = sitevs =
EDN	Welcome! Build real-time applications with microse	cond latency for any industry in half the tim	e with Oracle Stream Explorer
REST	((ç))		
-	Distributed Intelligence for IOT	Rich and Frend Management	Transmission and I asiation
	Distributed Intelligence for IOT Acquire, analyze, and act on high-volume, high-velocity data from sensors and deuces both at the edge and in the data center in real-time	Risk and Fraud Management Leverage industry's best stream processing patform to assess rule, and powert financial flaud in real-time	Transportation and Logistics Manage feet, track assets, and improve supply chan efficiencies by combering streaming data with Cracle's advanced spatial functions
	<b>2</b>	0	
······································	Customer Experience and Consumer Analytics Know the sentiment of your customers to reduce churit, improve leyalty, make offers, and attract customers in real-time	Telecommunications Proactively monitor your networks, predict network failures, and prevent distributed denial of service type attacks	
****			
CSV file		Devi	ces

Resources: this <u>blog article</u> provides an overview of this device monitoring example and this <u>YouTube</u> <u>video</u> shows what it looks like.

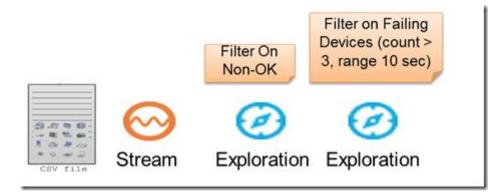
With Stream Explorer - everything starts from a Stream - a source of events or messages such as a JMS Queue or Topic, the SOA Suite Event Delivery Network, an HTTP channel or a CSV file such as in this case. Through one or more Explorations - that each can do filtering, aggregation, enrichment and pattern matching - finally conclusions can be published to a target. Targets can be JMS destinations, HTTP channels, a CSV file and the Event Delivery Network of SOA Suite.

In a number of steps, we will go from the CSV file with device signals to an output file that holds all broken devices. In later practices we will not just send output to a file but to live channels such as JMS, EDN and REST service targets.

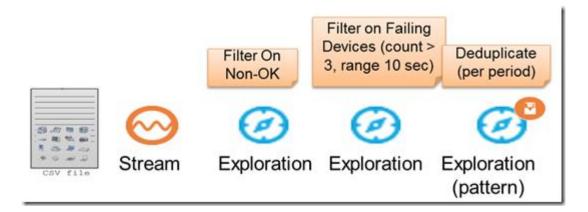
The first step will be an exploration that filters the non-ok signals from the stream:



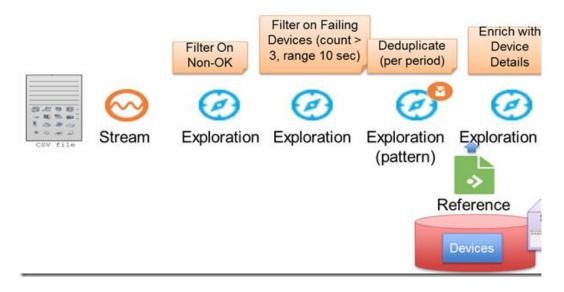
The second step will find failing devices by counting the number of non-ok signals in a 10 second period and filtering on any device with a count greater than or equal to 3:



Next, to prevent any failing device from being reported more than once (in a certain period of time) we perform deduplication, using one of the special patterns shipped out of the box in Stream Explorer:



The remaining messages report a unique failing device and we need to enrich those messages with details about the device location, taken from a Reference defined for a database table:



The enriched messages are routed to a target – in this case a simple csv file.

Open Stream Explore	(http://localhost:9002/s	sx) and log in (	(wlevs/weblogic1):
---------------------	--------------------------	------------------	--------------------

SIGN IN TO STREAM EXPLOREI Business Insight into Fast Dat	58
wievs I •••••	ORACLE"

#### This welcome page greets you:

Bemen	nber Password 🔻	6	Catalog n Home
Welcome!			
Build real-time applications with mi	crosecond latency for any industry in h	alf the time with Oracle Stream Explorer	
((ọ))	1		
Distributed Intelligence for IO	T Risk and Fraud Manageme	nt Transportation and Logist	lics
Acquire, analyze, and act on high-volume, high-velocity data from	Leverage industry's best stream processing platform to assess risk	Manage fleet, track assets, and in and supply chain efficiencies by comb	
sensors and devices both at the edge and in the data center in real-time	prevent financial fraud in real-time		

Open the Catalog with the button in the upper right hand corner.

# DeviceSignals Stream and NonOkSignals Exploration

Create a new item of type Stream. Set the Source Type to CSV.

Catalog Create New Item	Create Stream		Sort By Recent
Explorations	S Back	Source Details Type Properties Straps	Next >
Streams	Name	DeviceSignals	
References	- Description		
	Tags	Enterlag	
	* Source Type	Choose a Source Type	
	L 1	Choose a Source Type CSV File HTTP Subscriber EON JMS REST	
		Cancel	Create

Press Next. Select the CSV file - device\_signals.csv located in the folder StreamExplorer\_Handson under the home [folder] of the oracle user (/u01/app/oracle/StreamExplorer\_Handson):

StreamExplorer_Handson				
Devices	• 👼 Home	StreamExplorer_Handson		
💿 VBOXA  🚔				
Computer	Devic USCAR USCAR			
📠 Home	device signa	ls.csv		
Desktop				
📴 Documents				

Create Stream			×			
Sc Back	Durce Details Type Properties	Shape	Next 📏			
Type: CSV File	* File is not uploaded yet.	😣 File Upload				
* Source Path	Opioau me	🖉 🖣 📷 oracle	StreamExplore	er_Handson		
	Or provide the file URL here	Places	Name	~	Size	Modified
Event Interval	1000	Q Search Recently Used	🔋 🖿 device_sig	nals.csv	108.0 KB	04:49
Initial Delay	0	in oracle				
		Desktop	-			
		-				All Files 💲
					Cancel	Open

You may want to briefly inspect this file:

😣 🗖 🗊 device_signals	s.csv (~/Strea	mExplor	er_Hand	lson) -	gedit			
🔋 📔 Open 🔻 💆	Save 📑	•	Undo 🥖	⇒	X 🖬	١Ì	Q	X
📄 device_signals.csv 🗱								
device_signals.csv <b>%</b> DeviceID,ErrorCode, USCARSORCL8,0000,De USCARSORCL6,0000,De UKLOXXORCL6,0000,De OFMWEMEA02,1313,Som UKLOXXORCL4,0000,De FRAPAREDF01,0000,De ITROMETELI5,0000,De UKLOXXORCL7,0000,De UKLOXXORCL8,0000,De USCARSORCL8,0000,De USCARSORCL6,0000,De UKLOXXORCL6,0000,De UKLOXXORCL6,0000,De UKLOXXORCL6,0000,De UKLOXXORCL4,0000,De ITROMETELI5,0000,De UKLOXXORCL4,0000,De UKLOXXORCL7,0000,De UKLOXXORCL7,0000,De UKLOXXORCL7,0000,De UKLOXXORCL7,0000,De UKLOXXORCL7,0000,De UKLOXXORCL7,0000,De UKLOXXORCL6,001A,Te UKLOXXORCL4,001A,Te UKLOXXORCL6,001A,Te ITROMETELI5,7B90,Dr	vices in H vices in H	Healthy Health	State State	from from from from from from from from	heartB heartB heartB heartB heartB heartB heartB heartB heartB heartB, heartB, heartB, ggered ggered heartB, ggered	,0000 ,0000		
UKLOXXORCL7,001A,Te	mp indi	.cator h	nas bee	n tri	.ggered	,0085		

Back in the wizard, refine the Data Shape definition and give it a name (DeviceSignal):

atalog Cr	eate New Item *	Create Stream				×	Sort By Recent	×
Explorations	~	K Back Sc	O urce Details Type	O O Properties Shape		Next >		
Streams	~	Type: CSV File CSV File Shape						
References	~	Name	DeviceSignal					
		Manual Mapping	DeviceID	String	•	×		
			ErrorCode	Integer	0	×		
			ErrorDescription	String	•	×		
			DataValue	String	•	× +		
		💮 Select Shape	Choose a Shape					
					(Income)			
					Cano	cel Create		

Press Create to complete the Stream definition.

Next, the Exploration wizard is started. Set the name of the Exploration to NonOkSignals:

atalog	Create New Item *			Sort By: Recent	
Exploration	ons 🗸	Create Explor	ration ×	ited by wlevs	*×
Streams	-	* Name	NonOkSignals	/15 5.26 PM	
_		Description	Provide a description that help people understand this Exploration	Show	10 🐨
Referenc	es 🗸				
		Tags	Enter Tag		
		* Source	💮 DeviceSignals 🛪		
			Cancel Create		
			Di		

Press the Create button.

The Exploration editor is showing and the first messages read from the CSV file are produced in the exploration:

ORACLE	Stream Explorer			1	nelp ▼ wiev	/S 🔻 🚥	Ê
Return To Catalog			③ Configure a T	arget 📑 Act	ions 🦻	o e i 🖸	
Welcom the Live		low. In addition, certain typ	out your data stream using analytic tools and f bes of streams can be refined using the Range				E
🗏 NonOkSignals 🧭	Draft						
Sources O Device	eSignals 🛪						O
Summaries + Add a Summary		Group by	Filters				
∡ Live Output Stream	n			R	Properties	Timestamp	
DeviceID	ErrorCode	ErrorDe	escription	DataValue	•		
ITROMETELI5	7B90	Drive M	otor Malfunction detected HDR	8979			
UKLOXXORCL6	001A	Temp	indicator has been triggered	0085			
FRAPAREDF01	5100	Unable	to handl Document Jammed UNIT	0003			
OEMWEMEA02	1313	Devices	s in Healthy State from beartB	OPOP			

Define a filter for the exploration, to only produce messages for non ok error codes:

DRACLE	Stream Explorer				help 🔻	wie	vs v	***	Í
Keturn To Catalog				Configure a Target	C Actions	Þ	n (* )	0	
	rs to narrow-down your output stream ource-specific parameters) using the		or any filter, you can ch	ange the type of value you'	re evaluating (sl	tring, inte	ger,	×	=
🔺 NonOkSignals ⊘	Draft								
Sources O Device	Signals 🕱								O
Summaries		Group by	Filters   Match All	of the following O Match	Any of the follow	ing			
🕂 Add a Summary			ErrorCode + 1	not equais + 0000% 0000	¢	×			
▲ Live Output Stream					Pro	operties	Timestar	mp	
DeviceID	ErrorCode	ErrorDescrip	ption	Dat	aValue				
ITROMETELI1	6A95	Mechn rotatio	n element spike failure	900	0			^	
ITROMETELI1	6A95	Mechn rotatio	n element spike failure	900	0			=	
TRONETCLU	0405	Useka ostatia	a standard and the failt of	000	0				-

Publish the exploration – click on the Actions button and select Publish - and return to the catalog.



# **Exploration for Failing Devices**

Create a new Exploration. Call it *FaultyDevice* and use exploration *NonOkSignals* as the source.

atalog	Create New Item 👻			Sort By: Recent	
Exploratio	ns 🗸	Create Explor	ation ×	ited by wlevs	*×
Streams		* Name	FaultyDevice	/15 5.27 PM	
Reference		Description	Provide a description that help people understand this Exploration	ited by wievs /15 5 26 PM	*×
		Tags	Enler Tag	Show	10 *
		* Source	⊖ DeviceSignals		
			Cancel Create		

Press Create to navigate to the editor for the exploration.

Specify a summary:

```
count DeviceId , group by DeviceId.
```

Next, add a filter, to produce results only when for a DeviceId we have counted 3 or more NonOkSignals. Finally, specify a time window over which to calculate the count. Set the range of the window to 20 seconds and set the evaluation frequency to [once every] 3 [seconds].

Return To Catalog           Windows           Use the Range Window (accessible by the window icon to the right of the screen) to specify time example, aggregate functions in summaries, such as MAX or AVG, will use the Range paramethow often the system will produce the result events.	
Use the Range Window (accessible by the window icon to the right of the screen) to specify tim example, aggregate functions in summaries, such as MAX or AVG, will use the Range paramet	me and event based windows used in processing your stream. For
	R
A FaultyDevice 🖉 Draft	
Sources NonOkSignals x	Signals Range 20 seconds 💽 Evaluation frequency 3
	ich All of the following wiceiD greater than or equals 3
✓ Live Output Stream	Properties Timestamp
COUNT_of_DeviceID	DeviceID

Publish the exploration. Note that if you wait for some time, you will see devices being reported in the Live Output Stream section of this page:

			hei	p ▼ wie	vs v	**
Return To Catalog	ź	Configure a Target	C Action	ns 🖗	n a (	9
Exploration Editor Welcome to the Explorer. This is where you discover interesting things ab the Live Output Stream table and as a graph below in addition, certain typ more about these tools, watch this space as you explore.			Unpublish	Discard	土 Export	¢
✓ FaultyDevice 🖉 Draft						
Sources 🕢 NonOkSignals 🛪						G
Summaries Group by DeviceID COUNT of DeviceID	Filters Match All of th COUNT of DeviceID gre					
⊿ Live Output Stream				Properties	Timestam	p
COUNT_of_DeviceID		DeviceID				
3		FRAPAREDF01				

Return to the Catalog.

#### **Deduplicate Devices**

Create a Pattern - an exploration based on a predefined pattern. Pick the Eliminate Duplicates pattern.

atalog	Create New Item *		1		Sort By: Recent	-
Exploratio	Stream	Top N Bottom N	Eliminate Duplicates Use the Eliminate Duplicates pattern to build an exploration		Updated by wievs 4/8/15 5:28 PM	*×
Streams Reference	Reference Pattern >	Up Trend Down Trend Fluctuation	that eliminates duplicate events in your event stream.		Updated by wievs 4/8/15 5.27 PM	*×
		Eliminate Duplicates			Updated by wlevs 4/8/15 5:26 PM	*×
		Detect Missing Event W Inverse W		1 > 3	Show	10 *

Select *FaultyDevice* as the source for this pattern exploration. Select DeviceId as the key to determine the uniqueness by (eliminate the second and subsequent events from the FaultyDevice stream). Set the window to 1 minute. This means that any subsequent events for a device are blocked during 1 minute after the initial event for the device. After that minute, the slate is cleared for that device.

#### Publish this exploration too.

ORACLE <sup>®</sup> Str	ream Explorer		help 🔻	wievs 🔻 🚥
Return To Catalog		③ Configure a Target	C* Actions	P n n 😶
Eliminate D Use the Elim	uplicates inate Duplicates pattern to build an exploration that eliminates duplicate events in your e	event stream.		×
Exploration4     Orat				
* Event stream	FaultyDevice ×			
* Duplicate key	DeviceID ×			
* Window	1 v A minutes V			
▲ Live Output Stream			Prop	erties Timestamp
	COUNT_of_DeviceID	DeviceID		
	3	FRAPAREDF01		
	3	ITROMETELI1		
	7	ITROMETELI5		

Return to the catalog.

# Enrich the FailingDeviceEvents and Publish to CSV file

Create a Reference:

atalog	Create New Item		Sort By: Recen	nt 💽
Exploratio	Stream	Create a reference Create a reference to static data to enrich your event	Updated by wievs 4/8/15 5:30 PM	*×
Streams	Referense Pattern i S	stream.	Updated by wievs 4/8/15 5/28 PM	*×
			Updated by wlevs 4/8/15 5:27 PM	*×
		Device Signals	Updated by wievs 4/8/15 5:26 PM	*×
		Page 1 of 1 (1-4 of 4 items)	R C 1 2 3	Show 10 *

Note: before doing this, a Data Source to the schema (WC) that contains the database table should have been set up in OEP - either in the configuration file directly or through the WLEvent Visualizer console (<u>http://localhost:9002/wlevs/</u>):

Click on the node *defaultserver*.

Most Visited  Oracle Event Processing V	🗌 OEP Stream Explorer 🧧 RESTClient 📄 WebLogi /isualizer 👫 Home 🥃 Security 🖂 Daathboard 🚕 Va		
Welcome : wievs	Ceptoyment JMX DataSource HT	TTP Server Work Manager SSL Logging Event Inspector	
Deployment Security V Ddfaultserver	Applicati Libraries Extension L Deployed Applications	ibraries	
▶ ♠ Applications	Name	State	Target
V Dervices	sx-2-1-CinemaComingAndGoing	RUNNING	
Console Output	sx-2-10-RoomOccupancy	RUNNING	
📸 Event Repository	com.bea.wlevs.dataservices	RUNNING	
₩ Event Type Repository	sx-1-1-simpleRest	RUNNING	
🚧 Http Pub/Sub Server			
🔳 Log Query			

Open the tab Data Sources and click on the Add button.

🛉 Welcome : wievs

The wizard for creating a [JDBC] Data Source is opened. Provide a name and a JNDI name, as shown in the figure:

📋 DataSource:reference_data	
Data Sou Global Transa	action Protocol Connection Pool
Name	reference_data
JNDI Name	jdbc/ref_data
Global Transaction Protocol	OnePhaseCommit 🔹

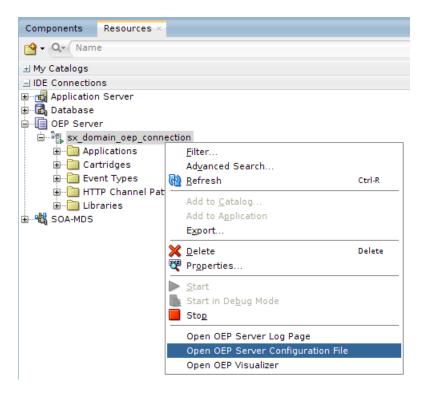
In step two, select Oracle as the Data Type, provide the [JDBC} URL and the user name and password (both are *wc*).

DataSource:WC	Iobal Transaction Protocol Connection Pool	
Database Type	Oracle •	
URL	jdbc:oracle:thin:@localhost:1521:xe	
Driver Name	oracle.jdbc.OracleDriver	ĺ
User Name	wc	
Password	****	
Use XA	false 🔹	

Accept the defaults on step 3 and press Save.

DataSource:reference_data							
Data Source	Glob	al Transaction Protocol	Connection Pool				
Initial Capacity		1					
Max Cap	acity	15					
Capacity Incren	nent	1					
Test							
E Save	Cancel						

This will create the data source and add its definition to the configuration file:



#### Config.xml Q- Find - 📮 🛛 🔍 🎠 🐘 🗆 😁 </cluster> Ξ <domain> <name>sx</name> </domain> <data-source> <name>reference\_data</name> <data-source-params> <jndi-names> <element>jdbc/ref\_data</element> </jndi-names> <global-transactions-protocol>OnePhaseCommit</global-transactions-protocol> </data-source-params> = <connection-pool-params> <credential-mapping-enabled></credential-mapping-enabled> <test-table-name></test-table-name> <initial-capacity>l</initial-capacity> <max-capacity>15</max-capacity> <capacity-increment>l</capacity-increment> </connection-pool-params> = <driver-params> <use-xa-data-source-interface>true</use-xa-data-source-interface> <driver-name>oracle.jdbc.OracleDriver</driver-name> <url>jdbc:oracle:thin:@localhost:1521:xe</url> Ξ <properties> Ξ <element> <value>wc</value> <name>user</name> </element> Ξ <element> <value>{AES}IsLG2oKtvoY3H3RpkL6s6Q==</value> <name>password</name> </element> </properties> </driver-params> </data-source> /ns2:contia>

The table used as a reference is shown here:

• - 🔁 🗶 🔁	🖌 🖈	📑 💥 🛸 🖷	Sort   Filter:					🔻 🔫 Acti	ions
🗟 IDE Connections		() ID	BUSINESS_NAME	ADDRESS	CITY	STATE_PROVINCE	<pre>     POSTAL_CODE </pre>	UNGITUDE	lati
sys_xe_local	1	UKLOXXORCL6	AMIS HQ	Edisonbaan 15	Nieuwegein	NL	94111	-122.401601	37.79
⊡ <mark>wc_local</mark>	2	UKLOXXORCL4	Great Clothing	100 Fourth St.	San Francisco	CA	94107	-122.404296	37.7
Tables (Filtered)     DEVICES	3	FRAPAREDF04	Super Electronics	100 First St.	San Francisco	CA	94107	-122.397632	37.7
	4	ITROMETEL15	Oracle Cloud	Stratosphere	The Sky	UP	94110	-122.404201	37.7
BUSINESS NAME	5	UKLOXXORCL7	Vintage Clothing	200 Mason St.	San Francisco	CA	94108	-122.409679	37.7
ADDRESS	6	UKLOXXORCL8	SOMA Coffee and Tea	600 Howard St.	San Francisco	CA	94106	-122.407175	37.7
CITY	7	FRAPAREDF01	Boscolo Hotel	Erzsébet krt. 9-11	Budapest	HU	94111	-122.420901	37.7
STATE_PROVINCE	8	USCARSORCL9	Smart Electronics	10 Grove St.	San Francsico	CA	94103	-122.415181	37.7
POSTAL_CODE	9	USCARSORCL6	FMW Partner Forum	Boscolo Hotel	Budapest	HU	94112	-122.400815	37.7
	10	ITROMETELI1	The Coffee Sack	2800 Sacramento St.	San Francisco	CA	94101	-122.438968	37.
MOVIES	11	USCARSORCL7	Robin Coffee Shop	600 Howard St.	San Francisco	CA	94106	-122.407175	37.7
WC_FOOTBALL_REGIONS	12	USCARSORCL8	Lloyd Cheese Store	100 Fourth St.	San Francisco	CA	94107	-122.404296	37.7
. WC_MATCH_RESULTS	13	OFMWEMEA01	Hilton Malta	Portomaso	Saint Julian's	i MA	(null)	(null)	(
WC_MATCH_TAGS	14	OFMWEMEA02	Hilton Vilamoura As Cas	Rua da Torre d'Agua, Lote 4.11.1B	Vilamoura	PT	8125615	(null)	()

Define the Reference - set its name and its type:

Catalog	Create New Item	Create Referen	ce ×	ort By Recent	•
Exploration	5	< Back	Source Details Type Properties Shape	5 by wievs 5:30 PM	*×
Streams		* Name Description	DeviceDetails	<b>i by wievs</b> 5.28 PM	*×
				<b>1 by wievs</b> 5 27 PM	*×
			Enter tag Choose a Source Type	<b>1 by wievs</b> 5.26 PM	*×
			Choose a Source Type Database Table	Show	10 💌
			Cancel Create		

Then select the table that provides the data for this reference:

	cplorer		help 🐐 wievs	Home
Catalog Create New Item	Create Reference	×	ort By: Recent	•
Explorations	< Back	O O O Next 3	<b>1 by wievs</b> 5:30 PM	*×
Streams	Type: Database Tal Database Table Shap		<b>5 by wievs</b> 5 28 PM	*×
	Select Shape	Choose a Shape	5 by wievs 5.27 PM	*×
		FU_ARLINE_CARRIERS FU_ARPORTS FU_CONTS FU_FUSTOMERS FU_FUGHTS FU_PASSENGERS ROOMS	<b>1 by wievs</b> 5:26 PM	*×
		Cancel Create	Show	/ 10 *
Copyright © 2015 Oracle and/or its affiliates. Al	I rights reserved.			

And press Create.

Now create a new exploration - our last one in this practice:

	slorer	help • wiew	A Home
atalog Create New Item	(	Sort By: Recent	
Exploration     Exploration     Stream     Reference	DeviceDetails	Updated by wievs 4/8/15 5:31 PM	*×
Streams Reference Pattern	Exploration4	Updated by wievs 4/8/15 5:30 PM	*×
	FaultyDevice Published	Updated by wievs 4/8/15 5:28 PM	*×
	Published	Updated by wlevs 4/8/15 5:27 PM	*×
	DeviceSignals	Updated by wievs 4/8/15.5:26 PM	*×
	Page 1 of 1 (1-5 of 5 items) 8 < 1 > 9	Shov	w 10 w

Set its name and select Exploration4 - the name auto-assigned to the [Eliminate Duplicates] Pattern based exploration - as the source:

atalog	Create New Item 🔻			Sort By Recent	
Explorations		Create Explor	ration	ed by wievs	*×
Streams		* Name	EnrichedFaultyDevices	6 5 31 PM	
References	Description	Provide a description that help people understand this Exploration	ed by wievs 5.5.30 PM	±×	
		Tags	Enter Tag	ed by wievs 5.5.28 PM	*×
		* Source	Output DeviceSignals NonOkSignals FaultyDevice	ed by wievs 5 5 27 PM	*×
			Exploration4	ed by wievs 5 5 26 PM	*×
		Page 1 of	f1 (1-5 of 5 items) 1	Show	10 🐨

Add the Reference *DeviceDetails* as a second source for this exploration:

ORAC			heip 🔻	wievs *	•••		î
Keturn To 0	Catalog	③ Configure a Target	P Actions	• • •	0		
	Sources Sources come in two general types, streams and references. To select a source, open the source into the field to filter the source menu.) if a source is disabled in the menu, it means that it won't wo			ype directly	×		
Enriched	FaultyDevices 🖉 Draft						
Sources	Exploration4 ×					©	
Summari	DeviceDetajis DeviceSignals FaultyDevice NonOKSignals						
I Live Outp	but Stream		Prope	erties Timesta	amp		
	COUNT_of_DeviceID	DeviceID					

Then specify the correlation condition:

Return To Catalog			() Co	nfigure	a Target	Act	ions 👳	50	0	
Multiple sources need to be correlated Now that you've added another source, you need to co stream by specifying equivalent properties. To do this, a be different ) If you have more than one source, you'll in	add a correlation be	elow, and then selec	t an equivalent	propert	y for two strea				×	
EnrichedFaultyDevices 🖉 Draft										
Sources Exploration4 x DeviceOctails x		Correlations	DeviceID		business	• a	×			O
Summaries	Group by	Filters + Add a Filte	r		business_na state_provin address postal_code	ice				
⊿ Live Output Stream					longitude latitude		Properti	es Times	tamp	
COUNT_of_DeviceID DeviceID business_name	id state_pr	rovince add	iress po	stal_co	city		latitu	le cit	y	

Finally, configure this exploration to send its findings to a CSV file. Do so by clicking on the button Configure a Target.

Select CSV file as the target.

Cancel	Configure a Target Finish
Target Typ	e CSV File
* File Nam	No Target
Apper	HTTP Publisher d EDN JMS
	REST

Specify a name for the file – it has to be a valid file name, no spaces, and an extension of .csv. Also check the checkbox to append:

Cancel		Configure a Target	Finish مالی
	Target Type	CSV File	- -
,	File Name	failedDevices.csv	
	Append		

Press Finish. Next, Publish the exploration.



The file failedDevices.csv will be created and as the device signals continue to be processed, at some point reports on failed devices are written to the file. Note that the location of the file is:

/u01/app/oracle/OEP12c/user\_projects/domains/sx/defaultserver/FileStorage/wlevs

#### 3. Cinema Control - Using REST Services and Database Driven Enrichment



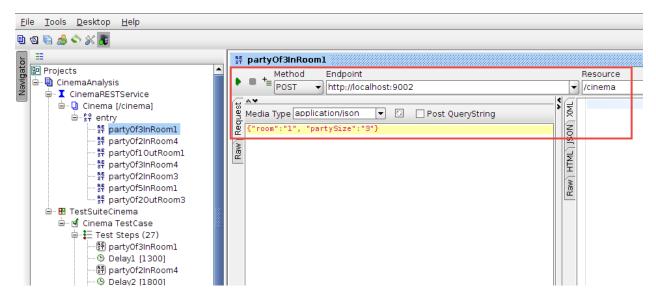
In this practice, we will look at a movie theater. More specifically, we will monitor the visitors entering and leaving the various rooms in the theater so we keep track of the number of people currently present in each room. When rooms are almost full, we may have to stop selling tickets or perhaps open up an additional room showing the same movie. When a room is [almost] empty, perhaps we should cancel the show altogether. When people keep coming and going, there may be a problem we should attend to.

In this case, the events are received by Stream Explorer in the form of JSON messages posted to a REST service. We use a SoapUI test case with requests to send in these events. To get an idea about the events, please start SoapUI – from the launcher:



SoapUI should start with the CinemaAnalysis project already loaded. If it is not, you can open this project from the SmartBear/SoapUI-5.1.3/user-projects directory under the home folder (/u01/app/oracle).

Open the test request called aprtyOf3InRoom1, as shown in the figure:



The request is configured to be sent to the endpoint <u>http://localhost:9002</u>. That is the endpoint for Stream Explorer (and OEP and more specifically the Jetty server running the OEP domain on top of which Stream Explorer was applied). The (REST) Resource is specified as /cinema. Together this means that this request is sent as a POST request to the end point <u>http://localhost:9002/cinema</u>. So that is where our StreamExplorer application will have to consume the message.

The message itself has a JSON payload with two simple properties: room (to identify a room in the movie theater) and partySize (to stipulate the number of people involved in an observation). Note: the number of people is positive when a party enters the room and negative when it leaves the room.

The TestSuite *TestSuiteCinema* contains a single test case with a number of steps that simulate events on a slow night in the movie theater.

#### **Create a Stream and a First Exploration for Handling Cinema Events**

Create a new Stream. The name is not crucial. The Source Type should be REST.

Create Stream				×
Back	Source Details	Type Properties	Shape	Next 🕽
* Name	CinemaComingAndGoi	ng		
Description	events describing group	os entering and leaving ro	ooms in a <u>movietheatr</u>	ê
Tags	Enter tag			
* Source Type	REST			•
	Create Exploration w	ith this source (Launch E	xploration Editor)	
				Cancel Create

#### Specify the Context Path as /cinema.

Create Stream				×
<b>K</b> Back	Source Details	Type Properties	Shape	Next 🕽
Type: REST				
* Context	Path /cinema			
				Cancel Create

Define the REST Shape. Set the name to CinemaEntryOrExitEvent. Define two properties: room – of type String – and partySize – of type Integer.

Create Stream			×
<b>K</b> Back	Ource Details Type Propertie	es Shape	Next 📏
Type: REST REST Shape			
Name	CinemaEntryOrExitEvent		
Manual Mapping	room	String	• ×
	partySize	Integer	• × +
○ Select Shape	Choose a Shape		•
		(	Cancel Create

Click on Create to create the Stream.

The Exploration wizard opens next. Set a name for the exploration – for example RoomOccupancy. You may provide a description as well.

Create Exploration			
RoomOccupancy			
Description Aggregation of people assembled in each room			
CinemaComing ×			
Cancel Cre	ate		
	RoomOccupancy Aggregation of people assembled in each room Enter Tag CinemaComing ×		

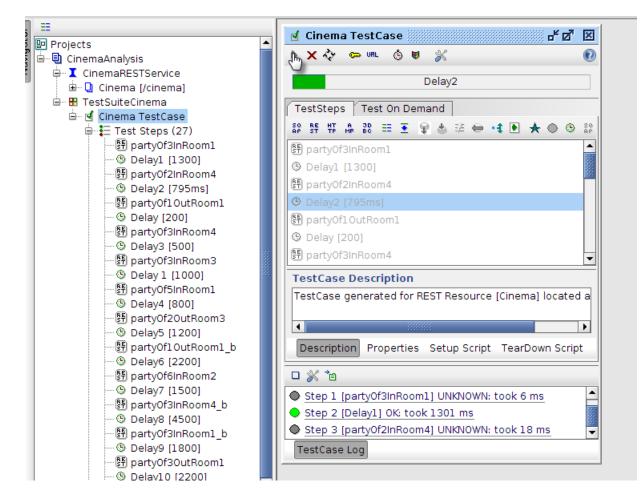
Click on Create.

Configure the Exploration for example like this:

		help v wlevs v				
Return To Catalog		② Configure a Target Configure a Target C Actions	0			
Windows Use the Range Window (accessible by the window icon to the right of the screen) to specify time and event based windows used in processing your stream. For example, aggregate functions in summaries, such as MAX or AVG, will use the Range parameter for calculation. The Evaluation Frequency parameter defines how often the system will produce the result events.						
RoomOccupancy 🖉 Draft						
Sources 🖂 CinemaComing 🗴		AndGoing Range: 8 hours 1 Evaluation frequency: 5	seconds			
SUM of partySize	Group by room Filters					
▲ Live Output Stream		Properties Timestan	np			
room	RoomOccupancy					

The events are aggregated grouped by *room*. The aggregation to be calculated is a sum over the *partySize*. This should produce the total number of people in every room. In this case, I have also chosen to have the summary calculated once every 5 seconds and to include in the results only the events from the last 8 hours (which is quite arbitrary).

At this point, return to SoapUI. Open the test case and run it:



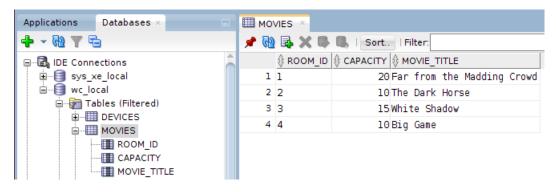
The exploration should start reporting its findings:

▲ Live Output Stream	
room	RoomOccupancy
4	0
4	10
4	8
1	6
3	1
2	6
1	2
4	5
3	3

Feel free to edit the test case in SoapUI, influencing the data fed into StreamExplorer. By running the test case again [and again] from SoapUI, you can keep the events flowing into the stream and exploration.

#### **Enrich the Movie Theater Findings**

Configure a new Reference called MovieDetails based on table Movies:



Note: we can leverage the Data Source that was set up in the previous section.

Create Reference				×
<b>K</b> Back	Source Details	Type Properties	Shape	Next 🕽
Type: Database T * Data source n	ame reference_da	ource name		•

Select the reference\_data Data Source. Press Next.

Select table MOVIES as the shape for this Reference.

Create Reference	e			×
<b>&amp;</b> Back	Source Details	O Type Properties	Shape	Next 📏
Type: Database Database Table Sh				
Select Shape	MOVIES			-
				Cancel Create

Click on Create.

	Stream Explorer			Catalog	A Home
Catalog	Create New Item 🔻			Sort By: Recent	
<ul> <li>Explorations</li> </ul>	~	MovieTheaterDetails	Reference data on the rooms and movies in the movie theater	Updated by wlevs 5/8/15 7:43 PM	±Χ
Streams	*	RoomOccupancy Draft	Aggregation of people assembled in each room	Updated by wievs 5/8/15 5:51 PM	*×
		CinemaComingAndGoing	events describing groups entering and leaving rooms in a movietheatre	Updated by wlevs 5/8/15 4:35 PM	*×

Using this reference, we can now enrich the results from the *RoomOccupancy* exploration.

Create a new exploration, for example called *EnrichedRoomOccupancyReports* and based on the *RoomOccupancy* exploration.

Click on the sources area and add the *MovieTheaterDetails* reference.

Return To Catalog	
Sources Sources come in two general types, streams and references. To select a source, open the source menu menu, it means that it won't work with your current Explorer configuration.	by clicking the Source box. (You ca
EnrichedRoomOccupancyReports      Draft	
Sources RoomOccupany ×	
😔 CinemaComingAndGoing	
Summari MovieTheaterDetails	
+ Add a Summary	Add a Filter
▲ Live Output Stream	
SUM of partvSize	

Next, define the correlation between the two sources in this exploration – based on the room:

	help v wlevs v	
turn To Catalog 🔊 Configure a Target C Actions F In C Configure a		
Now that you've added another source, you need to correlate all the sources. In effect, what you're doing	pere is linking all the sources together into one big stream by specifying equivalent properties. To do this, add a correlation	×
EnrichedRoomOccupancyReports     Orat	[]	
Sources 😥 RoomOccupany X	Correlations room v = room_id v X	O
	eik	

After configuring the correlation, run the test case in SoapUI to push some events into the Stream Explorer. Check whether the results in this exploration are enriched alright:

Live Output Stre	am					Properties	Timestam
	SUM_of_partySize	room	movie_title	capacity	room_id		
	0	4	Big Game	10	4		
	10	4	Big Game	10	4		
	18	2	The Dark Horse	10	2		
	8	4	Big Game	10	4		
	3	3	White Shadow	15	3		
	18	1	Far from the Madding Crowd	20	1		
	14	1	Far from the Madding Crowd	20	1		
	5	4	Big Game	10	4		

You may want edit the properties, for a better result in the Live Output Stream.

You may also want to configure a target – for example a CSV file – to somehow publish the results from this exploration.

# 4. Publish Events from PL/SQL in the Database [to a REST Service]

Publishing events to a REST service is easy to today from within many environments. All it takes is sending an HTTP request – and that is something we can easily do from JavaScript, Java, SOA Suite and also from PL/SQL.

We will look now at how events can be pushed to a REST service from PL/SQL running in the Oracle Database 11gR2 XE in the hands-on VM. A PL/SQL procedure publish\_cinema\_event will be created, after we have made some preparations as DBA concerning network security.



Open the file publishEventsFromDatabase.txt.

The file contains a section to be executed as SYS in order to allow database user WC to call out through UTL\_HTTP to localhost at port 9002. Note: these actions are already performed.

Next, a PL/SQL procedure is created that takes a room identifier, a party size and an optional waiting time (in seconds). This procedure creates the JSON message and sends it to the http://localhost:9002/cinema REST resource as a POST request. Note: this procedure is also already created.

Finally, the procedure can be invoked once or multiple times to push events to the StreamExplorer stream – just like we have done from SoapUI. Statements such as:

begin

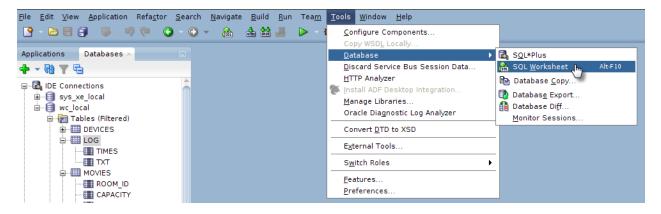
```
publish_cinema_event('1', 3);
```

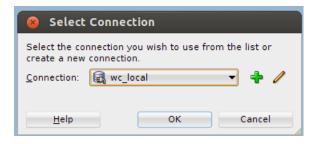
end;

suffice to publish an event to the REST service exposed by the Stream Explorer stream. This event tells the story of 3 people entering room 1. The next PL/SQL fragment corresponds roughly to the SoapUI test case – with the 3<sup>rd</sup> parameter the delay in seconds before pushing the event.

```
begin
    publish_cinema_event('2', -4, 1.3);
    publish_cinema_event('1', 4, 0.5);
    publish_cinema_event('3', -1, 2.6);
    publish_cinema_event('4', 2, 1);
    publish_cinema_event('4', 3, 2.3);
    publish_cinema_event('2', -1, 1.7);
    publish_cinema_event('1', 5, 0.4);
| publish_cinema_event('1', 5, 0.4);
| publish_cinema_event('1', -1, 2.2);
    publish_cinema_event('3', 6, 3.3);
    publish_cinema_event('2', -2, 1.6);
    publish_cinema_event('4', -1, 0.8);
end;
```

Open JDeveloper and open a SQL Worksheet for the wc\_local database connection.





Run the PL/SQL block with the repeated calls to publish\_cinema\_event

🗟 wc_local 🕨 🕞	× * 🍓 🗔   🔯 🗟   🏯 🍇 🥔 🗔
Worksheet	Query Builder
put put put put put put put put put	<pre>lish_cinema_event('2', -4, 1.3); lish_cinema_event('1', 4, 0.5); lish_cinema_event('3', -1, 2.6); lish_cinema_event('4', 2, 1); lish_cinema_event('4', 3, 2.3); lish_cinema_event('2', -1, 1.7); lish_cinema_event('1', 5, 0.4); lish_cinema_event('1', 5, 0.4); lish_cinema_event('1', -1, 2.2); lish_cinema_event('1', -1, 2.2); lish_cinema_event('3', 6, 3.3); lish_cinema_event('2', -2, 1.6); lish_cinema_event('4', -1, 0.8);</pre>
Script Out	out ×
* / 8 /	ScriptRunner Task 🛛 🔞

- and verify the effects in Stream Explorer. Note that the movie showing in room 2 must be quite awful;-)

A RoomOccupanyReports 🖉 Draft			
Sources 💮 CinemaComing 🗙			
SUM of partySize	Group by room *	Filters + Add a Filter	
▲ Live Output Stream			room
4			4
5			3
-7			2
-5			2
10			1
-1			3
5			4
-4			2
4			1

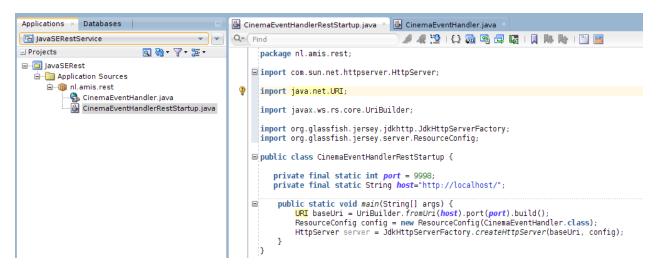
Feel free to experiment with the parameters in PL/SQL calls to simulate a different spread of people over the rooms in the theater.

# 5. Publish Exploration Outcomes to a REST Service

Findings from Stream Explorer explorations can be published to a CSV file as we have seen. This is not often useful, as CSV files are not really an appropriate mechanism in a real time environment. Other, more useful publication channels include JMS, EDN and HTTP Channel (for consumption by Dojo based Web Applications directly from a browser).

We will next look at a yet another way of publishing findings from an SX exploration: push to a REST service. This service can be implemented in a number of ways – that does not concern StreamExplorer. In this case we will use an implementation in a Java Servlet (JAX-RS). Alternatives are a SOA Suite service or even a Stream Explorer application with a REST based input stream.

In JDeveloper, open application JavaSERestService with project JavaSERest. This project has dependencies on several Jersey libraries (release 2.17) that allow it to publish a REST service from a Java SE environment, so without starting a Java EE container.



When you run the class CinemaEventHandlerRestStartup, a REST service is published at http://localhost:9998/movieevent, supporting both GET and a POST operations.

REST services can be a target for StreamExplorer – so we should be able to quickly configure this service as a target. It will not do much with any messages it receives – it will just show in the console output that it *did* receive them.

There turns out to be a problem with StreamExplorer publishing an aggregated value to a REST target (a temporary bug probably) so in order to demonstrate the capability to publish to a REST target, we will create a new exploration and configure it with this REST target.

From the Catalog, create a new Exploration based on the stream CinemaComingAndGoing.

Create Explo	ration ×
* Name	CinemaExploration
Description	Provide a description that help people understand this Exploration
Tags	Enter Tag
* Source	CinemaComing ×
	Cancel Create

Then run either the SoapUI test case or the PL/SQL script to have some cinema events published. Verify that these events lead to results in the new exploration.

				help 🔻	wley	/S 🔻		
K Return To Catalog			② Configure a Target	Actions	Þ	<u>י</u> מי ו	0	
Exploration Editor Welcome to the Explorer. This is where you discover interesting things as a graph below. In addition, certain types of streams can be refined u	about your data str sing the Range Wir	eam using analytic tools and filters. Incom dow drawer to the right. To learn more abo	ning data appears both in the out these tools, watch this s	Live Output Stre bace as you expl	am table ore.	and	×	
CinemaExploration 🖉 Draft								
Sources CinemaComing X								©
Summaries	Group by	Filters						
🕂 Add a Summary		Add a Filter						
▲ Live Output Stream				Pro	operties	Timestar	np	
room		p	artySize					
3			3					
1			-1					
4			3					
4			2					
1			3					

Next, click on Configure a Target. Select REST as the target type

CLE <sup>®</sup> Stream Explorer				help 🔻 wlevs 🔻	-
o Catalog			Sconfigure a Target	Actions 🗗 🖒	~   6
Exploration Editor Welcome to the Explorer. This is as a graph below. In addition, ce	where you discover interesting things rtain types of streams can be refined t	about your data stream using analytic tools and fi using the Range Window drawer to the right. To lea	Iters. Incoming data appears both in th m more about these tools, watch this a	ie Live Output Stream table and space as you explore.	×
Exploration 🖉 Draft					
CinemaComing ×	Cancel	Configure a Target	Finish		
	Target Type No	) Target	-		
<b>ies</b> a Summary	CS				
	RE	ST			
utput Stream				Properties Tim	estam
			3		

and set the resource URL to http://localhost:9998/movieevent .

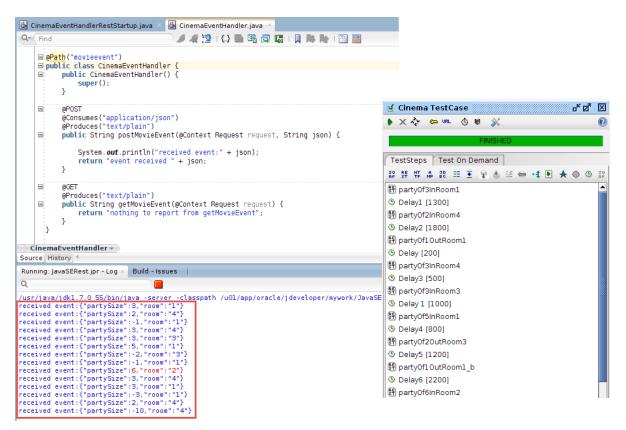
CinemaExploration 🖉 Draft			
Sources CinemaComing X	Cancel	Configure a Target	Finish
		Target Type REST	-
Summaries	*	Target URL http://localhost:9998/movieevent	
+ Add a Summary			
Live Output Stream			
4			
4			
1			
			3
			3

Click on Finish. Next, publish the Exploration.



Then: either run the SoapUI test case or the PL/SQL script. New rows should appear in the Live Output Stream *and* output should appear in the log window in JDeveloper, showing the reception by the REST service of the messages sent from the exploration.

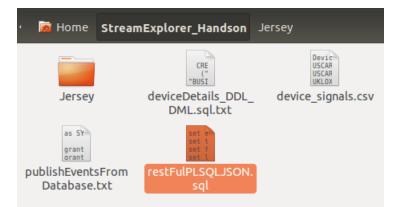
For example the SoapUI test case publishing to Stream Explorer results in the following outcome in the Java REST service:



In preparation for the next practice, please also configure this REST service as the target for the exploration *EnrichedRoomOccupancyReports*. At the present, this will not actually work correctly as a result of the naming of property SUM\_of\_partySize. In practice 6 we will address this problem.

#### Send Events from Java REST Service to PL/SQL

One of the things we could do from the *CinemaEventHandler* is pass events to the database. From Java there is of course the option of using JDBC to talk to a database. However, let's use HTTP as our protocol. Using MOD\_PLSQL – see the script restFulPLSQLJSON.sql in the StreamExplorer\_Handson folder



- a PL/SQL procedure called *movieevents* has been set up to be invoked on the URL

http://localhost:8080/api/movieevents with an HTTP POST request that contains a parameter p\_json\_payload. The PL/SQL procedure will do little more than reply with a response indicating that it did indeed receive a request with said parameter.

```
-- this procedure can be invoked over the following URL with an HTTP POST request
-- <u>http://localhost:8080/api/movieevents</u>
create or replace procedure movieevents
( p_json_payload in varchar2 default '{}'
)
is
begin
htp.p('call received p_json_payload='||p_json_payload);
htp.p('REQUEST_METHOD='||owa_util.get_cgi_env(param_name => 'REQUEST_METHOD'));
end movieevents:
```

To send the HTTP POST from Java, class *CinemaEventRouter* was included in the project. From this class, method *sendPost* can be invoked to make the HTTP POST call to the url handled by the PL/SQL procedure. If you uncomment the following line in method *postMovieEvent* in class *CinemaEventHandler*, then the *CinemaEventRouter* will be called upon to deliver the goods.

```
@POST
@Consumes("application/json")
@Produces("text/plain")
public String postMovieEvent(@Context Request request, String json) {
    Svstem.out.println("received event:" + json);
    //CinemaEventRouter.sendPost(json);
    return event received + json;
}
@GET
@Produces("text/plain")
public String getMovieEvent(@Context Request request) {
    return "nothing to report from getMovieEvent";
}
```

The output from running the SoapUI Testcase to make the REST calls to StreamExplorer is now as follows – composed of the initial reception in the Java REST service and the output from the interaction with the PL/SQL procedure (via HTTP POST call).

Running: JavaSERest.jpr - Log × Build - Issues
٩
<pre>/usr/java/jdkl.7.0_55/bin/java -server -classpath /u01/app/oracle, received event:{"partySize":3,"room":"1"} call received p_json_payload={"partySize":3,"room":"1"} REQUEST_METHOD=POST received event:{"partySize":2,"room":"4"} call received p_json_payload={"partySize":2,"room":"4"} REQUEST_METHOD=POST received event:{"partySize":-1,"room":"1"} call received p_json_payload={"partySize":-1,"room":"1"} REQUEST_METHOD=POST received event:{"partySize":3,"room":"4"} call received p_json_payload={"partySize":3,"room":"4"} REQUEST_METHOD=POST received event:{"partySize":3,"room":"3"} call received p_json_payload={"partySize":3,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":5,"room":"1"} call received p_json_payload={"partySize":5,"room":"1"} REQUEST_METHOD=POST received event:{"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-1,"room":"1"} call received p_json_payload={"partySize":-2,"room":"3"} REQUEST_METHOD=POST received event:{"partySize":-1,"room":"1"}</pre>
REQUEST METHOD=POST

## 6. Refine Stream Explorer application as OEP project in JDeveloper

At the present, StreamExplorer has a number of shortcomings that limit its usefulness. We cannot for example perform any calculations, derive or manipulate the value of properties or leverage built in functions. The built in patterns are handy, but only when there is a direct fit. There is no way for us in StreamExplorer to manipulate the CQL that is behind the pattern, to make it appropriate for our use case.

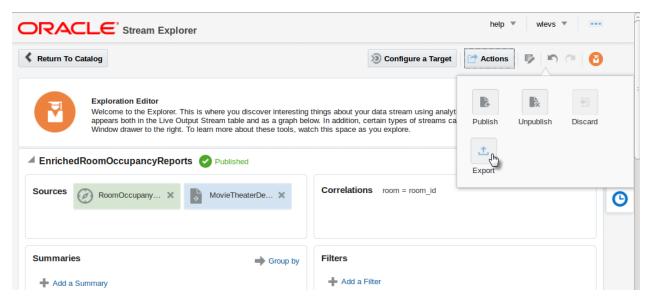
StreamExplorer as it turns out is [just] a pretty face on top of Oracle Event Processor. By creating streams and explorations in the StreamExplorer browser based user interface we are actually creating OEP projects that consist of adapters, channels and processors using [fairly simple] CQL queries. And we can export these OEP project from StreamExplorer and subsequently import them into JDeveloper – the IDE for OEP 12c – to fine tune them to our particular need.

Continuing from the MovieTheater example discussed in the previous two sections, we can now add the logic to spot rooms in the theater that are close to filling up (occupancy > 90%) so we can take appropriate action. Note that StreamExplorer at the present does not allow us to check for 90% occupancy unless we resort to tricks such as create a (Virtual) Column in the MOVIES table that contains the value for 90% occupancy and use it in a filter on the exploration.

### **Export StreamExplorer Exploration**

An exploration can be exported in StreamExplorer to a JAR file that can subsequently be imported into JDeveloper as a new OEP project. We will use this mechanism to refine the StreamExplorer application.

Open the EnrichedRoomOccupancyReports exploration. Open the Actions menu and click on Export.



The Export popup window appears. Click on Export.

xport List			
Resource		Туре	
EnrichedRoomO	ccupancyReports	② Exploration	
RoomOccupany	Reports	Ø Source	
MovieTheaterDe	tails	Source	

The browser's file download dialog appears.

Opening EnrichedRoomOccupancyReports.jar
You have chosen to open:
EnrichedRoomOccupancyReports.jar
which is: Java archive (4.9 KB) from: http://localhost:9002
What should Firefox do with this file?
Open with
Save File
Do this <u>a</u> utomatically for files like this from now on.
Cancel OK

Click on OK to save the Jar file.

## Import OEP bundle into JDeveloper

Return to JDeveloper. Create a new OEP application.

😣 Create OEP App	lication - Step 1 of 4	
Name your applica	tion	F
Application Name	Application Name:	
]	RoomOccupancy	
🔶 Project Name	Directory:	
<ul> <li>Project Java Settings</li> </ul>	/u01/app/oracle/jdeveloper/mywork/RoomOccupancy	B <u>r</u> owse
o Project OEP technolo	Application <u>P</u> ackage Prefix:	
i i		
i l		
<u>H</u> elp	< Back Next > Einish (h)	Cancel

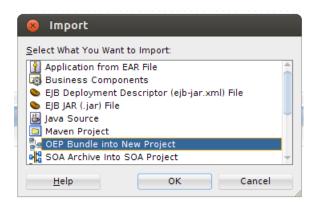
Click Finish.

The Project1 that is created by default can be deleted. We do not need it.

Click on File | Import



we ge



Click on OK.

The Import wizard appears. Enter a name for the new project to be created:

😣 Import OEP Bur	ndle as Project - Step 1 of 2	
Create Project		<b>8</b> .9
Create Project	Project Name:	
Import OEP Bundle	RoomOccupancy	
	Directory Name:	
	/app/oracle/jdeveloper/mywork/RoomOccupancy/RoomOccupancy	Bro <u>w</u> se
L L L L L L L L L L L L L L L L L L L		
ä		
a		
: 		
: ::::::::::::::::::::::::::::::::::::		
<u>H</u> elp	< <u>B</u> ack <u>N</u> ext >	Cancel

then press Next.

Click on Browse and locate the jar file that was exported from StreamExplorer (probably in the Home | Downloads folder):

🗴 Import OEP Bun	dle as Project - Step 2 of 2	
Import OEP Bundle	: ଅନ୍ତ୍ର କୁଜୁ	
<u>Create Project</u> <b>Import OEP Bundl</b>	OEP Bundle: Browse	
	Open     Location: 1/001/app/oracle/Downloads	000 🗟 👌 📰
	Work Home	
Help	Desktop Documents	
'4"} ize":3,"room":"4"} '1"} ize":3,"room":"1"}	File Name:       EnrichedRoomOccupancyReports.jar         File Type:       OEP bundles filter: .jar, .zip	
:"1"} ize":-3,"room":"1"}	Help	Den 🖿 Cancel

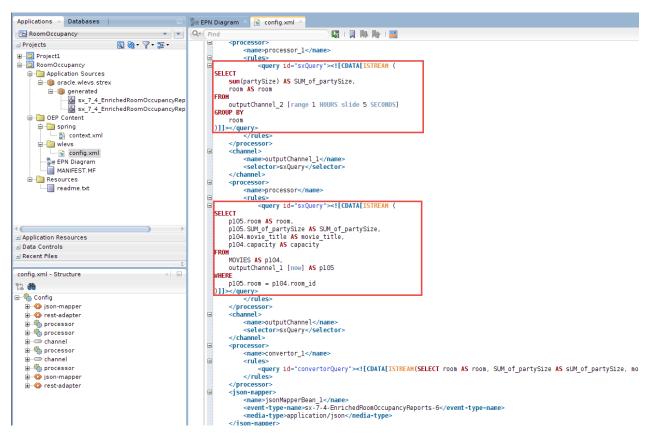
Press Open, then press Finish.

A new OEP project is created from the sources in the jar file:

Applications - Databases	Se EPN Diagram	Components	Resources	
🖼 RoomOccupancy 🔹 💌	🗞 Q, 100% - 5# Full EPN - 100 100	Q+(		
」Projects 図 線・マ・海・		Base EPN Co	mponents	
Project1     RoomOccupancy		- Base EPN C	omponents	
Application Sources	MOVIES	Basic Components		
		Adapter	Sean	¢ Cache
B Sx_7_4_EnrichedRoomOccupancyRep		Event Bean	<b>%</b> Processor	1 Table
B Context xml	rest-inbound-adapter convertor outputChannel_2 processor_1 outputChannel_i	Advanced Adapters		
wievs		S C5Vinbou	CSVOutb	HTTP Publish
ANNEST.MF	jponMapperBean_1	jMS Inbound	JMS Outbound	
		Big Data Exte	nsions	
	<b>%</b>	Hadoop	MoSQLDB	
⊴ Application Resources ⊴ Data Controls	jsonMapperBean	+ C	ons	

This project contains the sources that were created through our manipulations in the StreamExplorer GUI. In the EPN (the visual Event Processing Network) it is easy to see the inbound rest adapter that together with the first channel and processor (convertor) is based on the Stream definition in StreamExplorer. The second channel and processor *processor\_1* represent the first exploration – the one that aggregates the party sizes. The Movies cache is based on the Reference that was created for the MOVIES database table. The second processor is created for the *EnrichedRoomOccupancyReports* 

exploration and the final channel and outbound rest adapter are created from the target was that configured for this exploration.



Open file config.xml to inspect the CQL queries that were generated by StreamExplorer:

The first one performs the aggregation, the second one the enrichment (based on the correlation or join between two channels).

We can change the logic of the second processor to have it produce only events in case the aggregated number of people in the room (sum\_of\_partySize) is larger than 90% of the room capacity – a piece of processing logic we currently cannot define through the StreamExplorer UI.

First however, we have to correct the results of an unfortunate characteristic of StreamExplorer – the naming of the property SUM\_of\_partySize. This name is not conform bean property naming standards and as such causes several problems including our inability to publish the outcomes from the (Enriched)RoomOccupancy to either CSV file or REST service.

In the file config.xml please replace every occurrence of SUM\_of\_partySize with sum\_of\_partySize. Do the same in the file context.xml. Finally, edit the Java Class that was generated for the event type to be reported to the outbound REST adapter – probably called something like

sx\_7\_4\_EnrichedRoomOccupancyReports\_6. The bean property should be called sum\_of\_partySize and the getter and setter methods should be named accordingly, as is shown in the next screenshot.

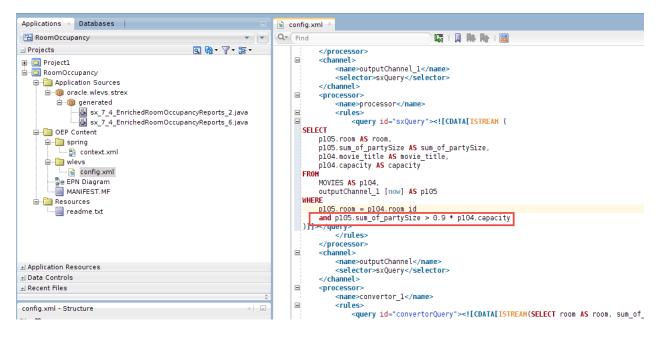
EPN Di	agram 🛛 📓 config.xml 🗐 🔮 context.xml 👋 遇 sx_7_4_EnrichedRoomOccupancyReports_6.java
Find	) 🖉 🖉 I 🖓 🐻 🗔 🖓 I 💭 🔤 🖾
pá	ackage oracle.wlevs.strex.generated;
🗆 pı	ublic class <pre>sx_7_4_EnrichedRoomOccupancyReports_6 {</pre>
	<pre>private String room; private int sum_of_partySize;</pre>
	private String movie_title; private int capacity;
	<pre>public sx_7_4_EnrichedRoomOccupancyReports_6() {     super(); }</pre>
=	<pre>public String getRoom() {     return room; }</pre>
	<pre>public void setRoom(String room) {     this.room = room; }</pre>
=	<pre>public int getSum_of_partySize() {     return sum_of_partySize; }</pre>
=	<pre>public void setSum_of_partySize(int sum_of_partySize) {     this.sum_of_partySize = sum_of_partySize; }</pre>
	<pre>public String getMovie title() {</pre>

With these changes, the publication of the outcomes from *EnrichedRoomOccupancyReports* to the REST service will now be successful. Before we go an deploy the OEP project, let's first add a little bit of CQI logic.

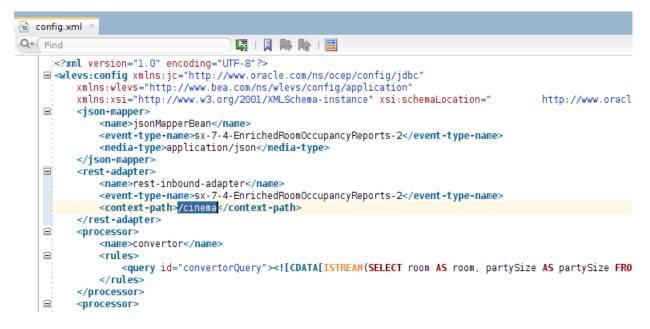
Open file config.xml as shown below. Add the following line to the Where clause of the CQL query of processor *processor*:

#### and p105.sum\_of\_partySize > 0.9 \* p104.capacity

Note: the alias p105 and p104 could be different in your particular case.



At this point we are ready to deploy the OEP project, resulting in an event processor that will publish findings to the (Java) REST service whenever a room has an occupancy of more than 90% of its capacity. We have one last issue to deal with: only one OEP project can expose an inbound REST adapter on a certain context path. Right now, both the StreamExporer stream CinemaComingAndGoing and the OEP project we are currently working on have an inbound REST adapter configured to listen at context path /cinema. There are two solutions: delete the StreamExplorer stream (and its dependent explorations) before deploying the OEP project or change the context path at which the inbound REST adapter in the OEP project listens - in file config.xml:



Note: if you opt for this second approach, you need to ensure that the SoapUI test case also is adjusted to publish to the new context path.

For example:

update context path in OEP application to /cinemaEventsSink

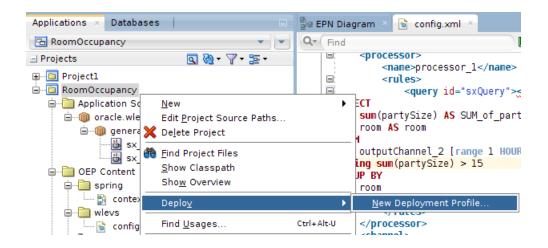
S config.:	kmi ×	
Q. Find		
	<pre>kml version="1.0" encoding="UTF-8"?&gt; Levs:config xmlns:jc="http://www.oracle.com/ns/ocep/config/jdbc" xmlns:wlevs="http://www.bea.com/ns/wlevs/config/application" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=" <json-mapper> <name>jsonMapperBean</name> <event-type-name>sx-7-4-EnrichedRoomOccupancyReports-2</event-type-name> <media-type>application/json</media-type> </json-mapper></pre>	http://www.
	<pre><rest-adapter></rest-adapter></pre>	

and adjust SoapUI resource accordingly:

SoapUI 5.1.3			
Eile Iools Desktop Help			
Image: Second state st	Cinema	EventsSink Value	Style

Now let's deploy the OEP project, with these steps:

Open the context menu for the project node and select Deploy | New Deployment Profile



Select the OEP Project Deployment Profile and click on OK

😣 Create Deployment Profile
Click OK to create your new deployment profile and immediately open it to see its configuration.
Profile <u>T</u> ype:
OEP Project Deployment Profile
Deployment Profile <u>N</u> ame:
oep_profile-1
Description:
Creates an OSGi Bundle that can be deployed to an OEP Server.
Help OK Cancel

Accept the default definition of the Deployment Profile and click OK.

😣 OEP Project Deployn	😣 OEP Project Deployment Profile						
Q		Deployment	Properties				
Deployment Properties         Package Exports         Contributors         Filters         Package Imports         Contributors         Filters         Project Output         Contributors         Filters         Project Output         Contributors         Filters         Project Output         Contributors         Filters         End Contributors         Filters         META-INF_wlevs         Contributors         Filters         META-INF_sorian         New       Delete	<ul> <li></li> <li></li> </ul>	<u>C</u> onnection to OEP Server: <u>S</u> ymbolic Name: B <u>u</u> ndle Name: Bundle <u>V</u> ersion: OSGi <u>I</u> AR File:	sx_domain_oep_connection (127   EnrichedRoomOccupancyReports   EnrichedRoomOccupancyReports   1.0.0   omOccupancy/deploy/oep_profile-1.jar     Save to MANIFEST.MF				
Help			OK Cancel				

The deployment profile is now created.

Open again the context menu for the project node and under Deploy there is now the new option oep\_profile-1, based on the deployment profile that was just created.

Applications × Databas	es 📄	😂 EPN Di	agram × 🗟 config.xml ×
🔁 RoomOccupancy	• •	Q. Find	
_ Projects ⊕ ☐ Project1	G & · ▼ · № ·		<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
RoomOccupancy     Application S	New		<query id="sxQuery"></query>
	Edit <u>P</u> roject Source Paths Delete Project		sum(partySize) AS SUM_of_par room AS room DM
s 🗰	<u>F</u> ind Project Files <u>S</u> how Classpath		outputChannel_2 [range 1 HOU ving sum(partySize) > 15
in the spring	Sho <u>w</u> Overview		DUP BY room
👘 🛐 conti	Deploy	۱.	oep_profile-1.
🖨 👘 wlevs 👘 confi	Find <u>U</u> sages	Ctrl+Alt-U	New Deployment Profile
EDN Diad B			schannetz

Select this option.

The deployment wizard appears:

8 Deploy oep_profile-1				
Deployment Action	1			
Deployment Action     Summary	<u>S</u> elect a deployment action from the list below. Deploy OSGi bundle to target platform Deploy to OSGi Bundle			
	Creates an OSGi bundle with a Manifest that conforms to OSGi specifications and deploys it to a target platform.			
	< Back Next > 0 Finish Cancel			
<u>H</u> elp	< <u>Back</u> <u>Next &gt; الس </u> <u>F</u> inish Cancel			

Click on Next.

😣 Deploy oep_pro	file-1
Summary	
<u>Deployment Action</u> Summary	Deployment Summary: Deployment Details Archive Details Output file: /u01/app/oracle/jdeveloper/mywork/RoomOccupancy/RoomO Include manifest: Yes 
20 20	4
<u>H</u> elp	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel

Click on Finish. Deployment will now take place:



The OEP project has been deployed and is ready for action.

Run the SoapUI test case (set to publish JSON messages to the updated context path) and see the results streaming in to the (Java) REST Service. Note: you may have to adjust the partySize values in the test case or run the test case multiple times in order to get output in the REST service.



This ouput tells us that both rooms 4 and 2 have an occupancy that is larger than 90% of their capacity. We also know that rooms 3 and 1 have an occupancy that is lower than 90% of their capacity. By changing the requests in the test case it is easy to overload room 1 as well:

<u>File T</u> ools <u>D</u> esktop <u>H</u> elp			
U 🕲 🛍 🍰 🔷 💥 🐻			
۲	<pre>§F partyOf3InRoom1</pre>		
Projects     CinemaAnalysis     CinemaAnalysis     CinemaKESTService     Cinema[cinemaZventsSink]     Ginema[cinemaZventsSink]     GinemaZventsSink]     Ginema[cinemaZventsSink]     GinemaZventsSink]     GinemaZventsSink]     GinemaZventsSink]     GinemaZventsSink]     GinemaZventsSink]     GinemaZv	Method Endpoint     Endpoint     Post      Name Value Style	Resource	Parameters
- ⑤ Delay1 [1300] - 쩐 Delay2 [1300] - 쩐 Delay2 [1800] - 쩐 Delay2 [1800] - 쩐 Delay2 [200] - 쩐 Delay2 [200] - 쩐 Delay2 [300] - 쩐 Delay3 [500] - 쩐 Delay1 [1000] - 쩐 Delay1 [1000] - 쩐 Delay4 [800]	Required: Sets if parameter is required Type: Options: Media Type application/ison Post Querys {"room": "1", "partySize": "S45"}	Add.	

The effect is visible within a few seconds:

Running: JavaSERest.jpr - Log ×	Build - Issues
Q	
	10,"movie_title":"Big_Game","room":"4","sum_of_partySize":10} d={"capacity":10,"movie_title":"Big_Game","room":"4","sum_of_partySize":10}
call received p_json_payloa REQUEST METHOD=POST	.0, movie_title":"Big Ga,"room":"4","sum_of_partySize*:10} d={"capacity":10,"movie_title":"Big Game","room":"4","sum_of_partySize":10} 20 "movie_title":"Ear from the Madding Crowd" "room":"1" "sum of partySize":363}

received event:{"capacity":20,"movie\_title":"Far from the Madding Crowd","room":"1","sum\_of\_partySize":363} call received p\_json\_payload={"capacity":20,"movie\_title":"Far from the Madding Crowd","room":"1","sum\_of\_partySize":363} REQUEST\_METHOD=POST

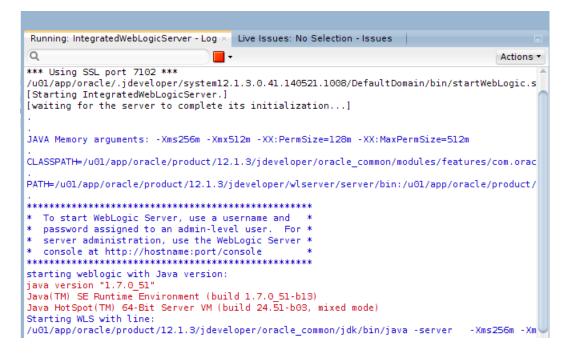
## 7. Using JMS as Source and Target for Stream Explorer

In this section, we will look at the very common interaction between Stream Explorer and JMS. JMS is a commonly used channel for decoupled exchange of messages or events. Stream Explorer can both consume messages from a JMS destination (through Stream) and publish findings to a JMS destination (with a target). The use case we discuss here is about temperature sensors: small devices distributed over a building, measuring the local room temperature every few seconds and reporting it over JMS. The Stream Explorer application has to look out for rooms with quickly increasing temperatures and report those over a second JMS queue.

In order to run the JMS container and the SOA Suite engine, start the Integrated WebLogic Server, using Run | Start Server Instance:

<u>R</u> u	n Tea <u>m T</u> ools <u>W</u> indow <u>H</u> elp	
	Choose Active Run Configuration	•
	<u>R</u> un View.jpr	F11
	<u>D</u> ebug View.jpr	Shift-F9
	Profile Project	
Upp	<u>A</u> ttach Profiler Profi <u>l</u> ing	•
۰	Toggle Breakpoint	F5
	<u>S</u> uspend All Breakpoints	
00	Pause	
⊪	<u>R</u> esume	F9
	Detac <u>h</u>	
	Restart Current Method from <u>B</u> eginning	
6	Step O <u>v</u> er	F8
4	Step Into	F7
1	Step <u>O</u> ut	Shift-F7
	Step to End of <u>M</u> ethod	
	Step Into Method <u>a</u> t Cursor	
	<u>C</u> ontinue Step	Shift-F8
	Pop <u>F</u> rame	
۰Ň	R <u>u</u> n to Cursor	F4
+	Find Execution Point	Alt-F3
Û	<u>G</u> arbage Collect	
в	Start Server Instance (IntegratedWebLogicServer)	N
	Debug Server Instance (IntegratedWebLogicServer)	3
	Terminate	×.

It may take a few minutes to bring up the WebLogic Server.



When the message appears "IntegratedWebLogicServer started", the server has been started and you can proceed.

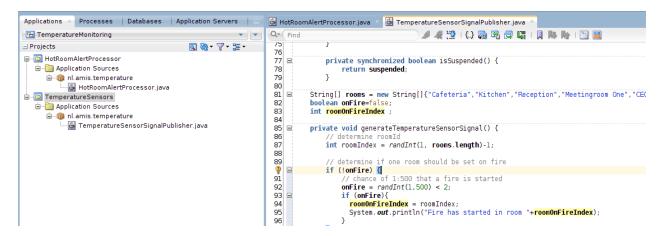


In this WebLogic domain, a number of JMS resources has been configured. For this practice we will use the queues jndi/temperatureMeasurements and jndi/hotRooms

	Settings for hand	Settings for handson-jms-module							
efaultDomain B- Environment	Configuration	Subdeployments	Targets	Security	Notes				
Deployments Services Messaging Data SourcesData Sources	This page disp	olays general informa	ation about	a JMS syste	em module	and its resources. It also allows you	to configure n	ew resources and access existing resources	
Persistent Stores Foreign JNDI Providers	Name:			handson-jms-module		т	The name of this JMS system module. More Info		
Work Contexts XML Registries XML Entity Caches	Descriptor File	Descriptor File Name:			on-jms-moo	lule-jms.xml	т	he name of the JMS module descriptor file.	More Info
JCOM Mail Sessions						this JMS system module, including store-and-forward parameters.	queue and to	pic destinations, connection factories, JMS te	mplates, destination sort
File T3	1								
low do I 🖃	Customize this	is table							
	Customize this								
iow do I 🖃		esources						Showi	ng 1 to 4 of 4 Previous   N
ow do L.  No task help found.  ystem Status	Summary of Re	esources lete		Туре		JNDI Name		Showi	ng 1 to 4 of 4 Previous   N
w do L  No task help found.	Summary of Rev	esources lete	ory	Type Connectio	on Factory	JNDI Name jms/handson-jms-connectionFi	ctory		
w do L.  No task help found.  stem Status ealth of Running Servers  Failed (0) Critical (0)	Summary of Rev	esources ete	ory		on Factory		ctory	Subdeployment	Targets
w do L  No task help found.  ystem Status Falled (0)	Summary of Ro New Dela Name & handson- hotRoome	esources ete		Connectio	on Factory	jms/handson-jms-connectionFa		Subdeployment Default Targetting	Targets DefaultServer

The JDeveloper application TemperatureMonitoring contains two projects that each contain a single class. One project is HotRoomAlertProcessor with class HotRoomAlertProcessor that registers as a listener to the HotRooms queue. Any message received on that queue is reported to the console.

The second project is TemperatureSensors. It contains class TemperatureSensorSignalPublisher. This class generates temperature values (in Celsius!) for a number of rooms, and publishes these to the queue temperatureMeasurements. At some random point, the class will start a fire in a randomly selected room. In this room, temperatures will soon be over 100 degrees.



Our objective in this practice is to read the temperature measurements from the JMS Queue into a Stream Explorer application, calculate the average value per room and then detect the room on fire. This *hot room* should then be reported to the JMS Queue.

Run class TemperatureSensorSignalPublisher to start producing messages on the JMS queue jndi/ temperatureMeasurements.

From the Stream Explorer Catalog page, create a new item of type Stream. Select JMS as the source type.

Create Stream				×
Back	Source Details	Type Properties	Shape	Next 🕻
* Name	TemperatureSignals			
Description				
Tags	Enter tag			
* Source Type	JMS			-
	Create Exploration w	ith this source (Launch E	xploration Editor)	
				Cancel Create

Press Next.

Configure the URL for the WebLogic domain (<u>http://localhost:7101</u>), the WebLogic Admin's username and password (weblogic/weblogic1) and the JNDI Name for the JMS Queue (or Topic): jndi/ temperatureMeasurements

Create Stream				>
So	O urce Details	Type Properties	Shape	Next 📏
Гуре: JMS				
* URL	t3://localhost:	7101		
* Username	weblogic			
* Password	•••••			
		ureMeasurements		
				Cancel Create

Press Next.

Define a new Shape. The properties in the JMS (Map)Message produced by the Java Class TemperatureSensorSignalPublisher are called RoomId (of type String) and Temperature (of type Float).

Create Stream			×
So So	O O O O O O O O O O O O O O O O O O O	ies Shape	Next 📏
Type: JMS JMS Shape			
Name	TemperatureMeasurement		
Manual Mapping	RoomId	String	- ×
	Temperature	Float	• × +
○ Select Shape	Choose a Shape		•
			Cancel Create

Press Create.

The Exploration editor appears to create an exploration based on the Stream.

Define a Name. Then click on Create.

Create Explo	ration	×
1	TemperatureMonitor	
Description	Provide a description that help people understand this Exploration	
Tags	Enter Tag	
* Source	TemperatureSig 🗙	
	Cancel Cre	ate

The temperature measurement events start streaming in:

		help 🔻 Wievs 🔻
Return To Catalog		🔊 Configure a Target 🛛 🖓 Actions 🛛 🖗 🍋 🦳 🔞
Summaries Use Summaries to perform simple creating thresholds and organizing i	operations on your data and/or to group inform nformation in a meaningful way.	$\mbox{\scriptsize \times}$ nation by one or more properties, potentially from multiple sources. This is useful for
TemperatureMonitor 🖉 Draft		
Sources TemperatureSig X		
Summaries	Group by	Filters
+ Add a Summary		Add a Filter
Live Output Stream		Properties Timestamp
Roomld	Temperature	
Cafeteria	18	ĥ
Cafeteria	18	U
CEO Office	156	
Reception	15	
Reception	15	
Office Floor A	14	
Reception	16	
Reception	16	

The first step is the definition of a Summary: calculate the average temperature per room. Also set the time range for the aggregation to 10 seconds (determine the temperature using the most recent 10 seconds worth of data) and the evaluation frequency to 5 seconds.

Return To Catalog					Onfigure a Target	Actions	Þ	<b>n</b> 0	
example,	Range Window (accessible by the	s, such as MAX or AVG, will i			and event based windows used in for calculation. The Evaluation Fre				×
Sources	-			🔗 Tempera	atureSignals Range: 10	seconds	- Evi	aluation	frequenc
Temper	atureSig 🗙		Θ				_		·
AVG of Temperature		Group by RoomI	d	Filters	r				
		Group by Rooml	d		r	Pro	operties	Time	estamp
AVG of Temperature	AVG_of_Temperature	Group by Roomi	d		r Roomld	Pro	operties	Time	
AVG of Temperature		Group by Room!	d			Pro	operties	Time	stamp
AVG of Temperature	AVG_of_Temperature	Group by Rooml	d		Roomld	Pro	operties	Time	
AVG of Temperature	AVG_of_Temperature	Group by Room	d		RoomId Reception	Pro	operties	Time	
AVG of Temperature	AVG_of_Temperature 17.8 18	Group by Room	d		RoomId Reception Office Floor A	Pro	operties	Time	
AVG of Temperature	AVG_of_Temperature 17.8 18 18	Group by Room	d		RoomId Reception Office Floor A CEO Office	Pro	operties	Time	

Fewer events are shown in the Live Output Stream – and with less variation.

Next, add a filter: we are going to hunt for the room on fire. Only records with an average temperature higher than 80 degrees should be reported. Also change the name of the property AVG\_of\_Temperature to AverageTemperature.

DRACLE' Stream Explorer		o ▼ wlevs ▼ ····
Return To Catalog	2 Configure a Target	ns 📝 🔊 (~ 🙆
	ble by the window icon to the right of the screen) to specify time and event based windows used in processing yo summaries, such as MAX or AVG, will use the Range parameter for calculation. The Evaluation Frequency paran e result events.	
TemperatureMonitor		
Sources TemperatureSig X	O TemperatureSignals Range: 10 seconds	Evaluation frequency
Summaries AVG of Temperature	Group by RoomId Filters Match All of the following AVG of Temperature greater than 80	
AVG of Temperature		Properties Timestamp
AVG of Temperature	AVG of Temperature greater than 80	Properties Timestamp
AVG of Temperature	AVG of Temperature greater than 80	Properties Timestamp
Live Output Stream	AVG of Temperature greater than 80	Properties Timestamp

The screenshot shows that in this case, it is the Cafeteria where there is a fire. If you stop class TemperatureSensorSignalPublisher and then start it again, it will take some time for it to start a fire again and when the fire was started, the Live Output Stream will show it.

Finally, click on Configure Target.

Configure a JMS Target, as shown in the figure. The URL is the familiar one (t3://localhost:7101), username and password are weblogic and weblogic1 and the JNDI Name of the JMS target is jndi/hotRooms.

	Cancel	Configure a Target	Finish
iç	Target Type	JMS	•
l	* URL	t3://localhost:7101	
	* Username	weblogic	
	* Password	•••••	
	* JNDI name	jndi/hotRooms	
l			
1			

Click on Finish. Publish the Exploration.

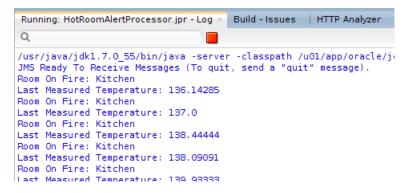
When there is now a room discovered with temperatures in the hot zone, a message will be published to the JMS Queue, in the form of a MapMessage with properties RoomId and AverageTemperature.

Stop and start class TemperatureSensorSignalPublisher. Run class HotRoomAlertProcessor to have it start listening to the jndi/hotRooms queue.

The former writes:



And the latter will report hot rooms by writing a message to the console:



While the Stream Explorer browser interface shows:

Sources Temper	ratureSig 🗙			
Summaries AVG of Temperature + Add a Summary		Group by RoomId x	All of the following ture greater than 80	
4 Live Output Stream				Properties Timestamn
-			Promid	Properties Timestamp
Time (Date) 🔟 🗙	AverageTemperature		Roomid Kitchen	Properties Timestamp
Time (Date)	AverageTemperature			Properties Timestamp
Live Output Stream Time (Date) X 5/16/15 10:17:55 AM 5/16/15 10:17:50 AM 5/16/15 10:17:45 AM	AverageTemperature		Kitchen	Properties Timestamp

## 8. Integrating with SOA Suite 12c through the Event Delivery Network

We assume here that you have started the Integrated WebLogic Server and therefore the SOA Suite 12c run time environment is up and running.

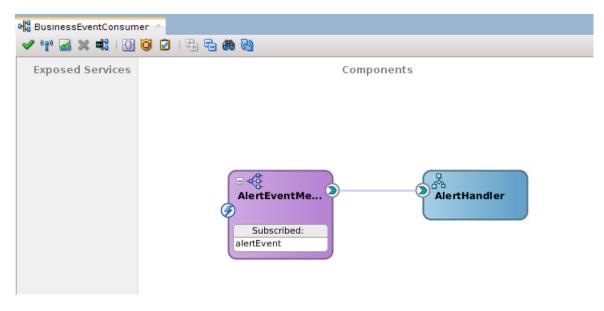
Two simple SOA Composites have been deployed to the SOA Suite runtime:



The BusinessEventPublisher contains a Mediator that can be invoked to have a very simple EDN event – operationalEvent - published (that contains one element called payload of type String):

📲 BusinessEventPublisher 🐣	
🖌 🕼 🔀 💥 🖏 I 🖸 🧕 🖓 🖓	5 🙃 🔞
Exposed Services	Components OperationBu Published: operationalEvent

The BusinessEventConsumer also contains a Mediator. This one will consume a different EDN event – alertEvent. When it does, it invokes a BPEL component that does absolutely nothing.



With this simple configuration as starting point, we will create the following StreamExplorer application:

From SoapUI we invoke the BusinessEventPublisher to have a number of operationalEvents published to the EDN. The StreamExplorer application will use an inbound EDN adapter to capture this operationalEvent in a Stream. A simple exploration does some form of processing – for example aggregation or duplicate detection. Its results are published as alertEvent to an EDN target and consumed in SOA Suite by the BusinessEventConsumer.

The SoapUI project that sends requests to the BusinessEventPublisher is shown here:

_		
E	ile <u>T</u> ools <u>D</u> esktop <u>H</u> elp	
Q	) 🕲 院 🍰 🔷 💥 🐻	
Navioator		<pre> Provide a state of the st</pre>

It sends payload values Green, Blue and Red.

In StreamExplorer, create a new Stream.

ORACLE <sup>®</sup> Stream Explorer								
Catalog	Create New Item	•						
Evplorations	Exploration	Stream						
<ul> <li>Explorations</li> </ul>	Stream	Create a stream of events to						
Streams	Reference	analyze.						
Ŭ	Pattern	•						
References		-						

Define a name and set the type to EDN.

Create Stream				×
Back	Source Details	Type Properties	Shape	Next 💙
	EDNOperationalEvents			
Description				
Tags	Enter tag			
* Source Type	Choose a Source Type			Ţ
	Choose a Source Type CSV File HTTP Subscriber			
	EDN JMS REST			
				Cancel Create

Click on Next.

Specify the URL (t3://localhost:7101) and the username and password (weblogic/weblogic1). Press Next.

Create Stream		×
So	O Type Properties Shape	Next 💙
Type: EDN		
* URL	t3://localhost:7101	
* Username	weblogic	
* Password	•••••	
		Cancel Create

Select the EDN Event Type – operationalEvent – and click on Create.

Create Stream		×
<b>K</b> Back	Source Details Type Properties Shape	Next 📏
Type: EDN EDN Shape		
Select Shape	operationalEvent Choose a Shape operationalEvent alertEvent	
		Cancel Create

The Stream is created and subscribed to the EDN for the operationalEvent.

Next, the exploration is created, based on this Stream. Add a summary – count on payload grouped by payload. Also specify a time range – for example one hour – and an evaluation frequency.

Exploration1 Oraft				
Sources EDNOperational X	O	SEDNOperationalEvents Range:	1 hours 👤	Evaluation frequency: 1
Summaries COUNT of payload		iters + Add a Filter		
▲ Live Output Stream			Pr	operties Timestamp
COUNT_of_payload		payload		

Send some requests to the BusinessEventPublisher composite, for example from SoapUI, to have operationlEvents published to EDN. These events are consumed by the Stream and reported in aggregated form by the exploration.

		EDNOperationalEvents Range: 1 h	ours  Evaluation frequency
Sources EDNOperational X	O	Convoperationalevents Range 1	
Summaries	Group by payload	ters	
COUNT of payload	4	Add a Filter	
Live Output Stream			Properties Timestamp
COUNT_of_payload		payload	
3		Blue	Â
		Blue	
7			
		Blue	
3		Blue Red	
3			
7 3 1 4 3		Red	

In EM FMW Control, we can see that instances of the BusinessEventPublisher have been created; these instances were responsible for publishing the EDN events.

BusinessE SOA Compos	ventPublisher [1.0] () site •			Page Refres	00	s weblogic   loca 5:39:39 PM CEST
	Composite Definition Flow Instan	ces Unit Tests Polic		nt Instances I Instances With Fi	with Decouverbl	Prelated Links
Actions -	ults - Instances Created (24 F	iours)		nt instances   instances with Fa		le Details
Flow ID	Initiating Composite	Flow State	Created	Last Updated	Partition	Na
130004	BusinessEventPublisher [1.0]	Completed	May 15, 2015 4:22:46 PM	May 15, 2015 4:22:46 PM	default	<b>A</b>
130003	BusinessEventPublisher [1.0]	Completed	May 15, 2015 4:22:46 PM	May 15, 2015 4:22:46 PM	default	
130002	BusinessEventPublisher [1.0]	Completed	May 15, 2015 4:22:23 PM	May 15, 2015 4:22:23 PM	default	
		A Completed	May 45, 0045 4:00:00 DM	May 15, 2015 4:22:22 PM		
130001	BusinessEventPublisher [1.0]	Completed	May 15, 2015 4:22:20 PM	Way 15, 2015 4.22.22 FW	default	
130001 120011	BusinessEventPublisher [1.0] BusinessEventPublisher [1.0]	Completed	May 15, 2015 4:22:20 PM May 15, 2015 4:11:29 PM	May 15, 2015 4:11:29 PM	default	
	1 A	* 1		1 N N		

Next, define a target for this exploration:

	Configure a Targ	et Actions	r 🖻 🕅	
			×	
cify time and event based windo or calculation. The Evaluation F				
➢ EDNOperationalEvents	Range: 1	hours 🔳	Evaluation frequency: 1	

Select EDN as the target type and define the URL and username/password.

Cancel	Configure a Target	Finish
Target Type	EDN	•
* URL	t3://localhost:7101	
* Username	weblogic	
* Password	•••••	
	Get Event Definitions	

Click on Get Event Definitions – to read the EDN event definitions that can published.

Select the alertEvent from the dropdown list:

Cancel	Configure a Target	Finish
Target Type	EDN	•
* URL	t3://localhost:7101	
* Username	weblogic	
* Password		
	* This field is required	
* Select Shape	Choose a Shape	
	Choose a Shape operationalEvent	
	alertEvent	

Accept the proposed property mapping:

Cancel	Configure a	<b>Target</b> Finish
Target Type	e EDN	•
* URI	- t3://localhost:7101	
* Usernam	e weblogic	
* Password	i	
	Get Event Definitions	
* Select Shap	e alertEvent	•
	Target Property	Output Stream Property
	payload	payload

And click on finish.

Finally, publish the exploration:

		h	elp 🔻 w	levs 🔻	•••
Ocr	nfigure a Tarç	get 📝 Acti		<b>ا</b> م	0
Incoming data a ubout these tool	Publish	Unpublish	Discard	1 Export	×

Return to SoapUI and make a number of calls to the BusinessEventPublisher.

Let's check the results of these actions in the EM FMW Control; a number of new instances is created of the BusinessEventConsumer composite.

ORACLE Enterprise Manager Fu	sion Middleware Co	ontrol 12c			
📑 WebLogic Domain 👻 🚔 SOA Infrastructure 👻					
Target Navigation	The BusinessEv	ventConsumer [1.0] 🗉			
View 🗸	SOA Compos	ite 🔻			
		e   Shut Down Test Composite Definition Flow Instan ults - Instances Created (24 H	ces Unit Tests Pol	cles	Q
BusinessEventPublisher [1.0]	Actions -				
HelloWorld [1.0]	Flow ID	Initiating Composite	Flow State	Created	Li
	130015	BusinessEventConsumer [1.0]	Completed	May 15, 2015 4:26:05 PM	M
WebLogic Domain	1300	BusinessEventConsumer [1.0]	Completed	May 15, 2015 4:26:04 PM	M
	130007	BusinessEventConsumer [1.0]	Completed	May 15, 2015 4:25:26 PM	Μ

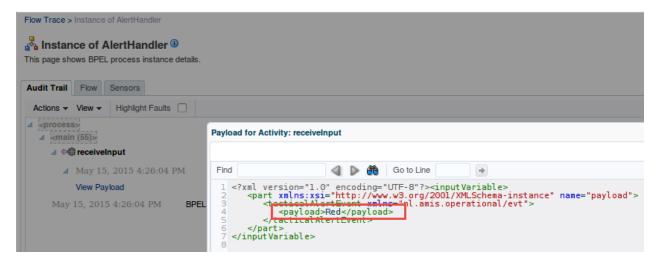
Drilling down on one of them shows that the instance as started from an EDN alertEvent:

#### Flow Trace

This page shows the flow of the message through various composite and component instances.

Faults Composite Sensor Values Composites			
Recover - View -			
Error Message	Fault Owner		
No faults found. Columns Hidden 8			
Trace			
Actions - View - Show Instance IDs			
Instance	Туре	Usage	State
A 4 alertEvent >> 4 AlertEventMediator	Mediator		Completed
AlertHangler	BPEL		Completed

and when we drill down on the BPEL component in this instance, we will find the payload value for this event:



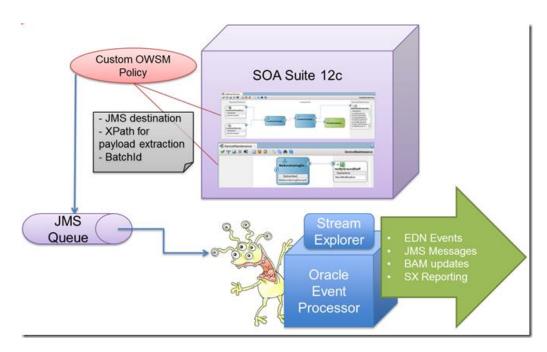
Clearly we have established a route from EDN to Stream Explorer and from Stream Explorer to EDN for decoupled interaction between SOA Suite 12c and Stream Explorer (& OEP).

Note: one limitation in V1 of Stream Explorer is that all properties of the EDN event have to be set in an Exploration's target mapping in order to have the EDN event successfully published. Additionally, all elements in the EDN event type definition must have a value for an EDN event to be successfully consumed by a Stream Explorer stream.

# 9. SOA Suite Service Execution Monitoring

One of the interesting opportunities offered to us by StreamExplorer and OEP is the ability to monitor the execution of services exposed by SOA Composite applications from SOA Suite. Through such monitoring we can keep an eye on response times, volume, trends in number of requests, fault percentage and more business oriented aspects derived from the contents of messages flowing into or out from service operations.

The next figure visualizes the setup: using a custom OWSM policy, we can publish a message for every execution of each operation in a service to a JMS destination. This policy can be bound to service bindings on selected SOA Composites. Each call to these decorated service bindings results in the policy being triggered and a message being published to the JMS Queue or Topic. Using a StreamExplorer Stream based on that JMS destination, we can collect all service execution reports and use them for real time analysis.



Also see blog articles for more details on the custom OWMS policy and on how to leverage the service execution reports to monitor business events:

- Oracle SOA Suite 12c Create, Deploy, Attach and Configure a Custom OWSM Policy to report on service execution
- Live Monitoring of SOA Suite Service Execution with Stream Explorer leveraging Custom OWSM Policy and JMS
- <u>Use Oracle Stream Explorer and the Service Execution Reporter policy to analyze service</u> <u>behavior – find too-late-closing flights on Saibot Airport</u>

## **Explore the HelloWorld Composite**

The custom policy is installed in the SOA Suite running in the Integrated WebLogic Server. This has been done by:

- copying the JAR file that holds the relevant Java classes to the DOMAIN\_HOME\lib directory
- configuring the policy in Enterprise Manager

A binding for the policy has been created for the HelloWorld SOA composite application. This causes every call to the service exposed by this composite to be reported to the JMS destination. Let's verify how this policy binding has been configured.

Select the HelloWorld SOA composite in the default partition:

ORACLE' Enterprise Ma	nager Fusion Middleware Control 12c
📲 WebLogic Domain 👻 🚔 SOA Infr	astructure -
Target Navigation View ▼	ThelioWorld [1.0] ③ SOA Composite ▼
<ul> <li>Application Deployments</li> <li>SOA</li> <li>Soa-infra (DefaultServer)</li> <li>Soa-infra (DefaultServer)</li> <li>Gefault</li> <li>HelloWorld [1.0]</li> <li>WebLogic Domain</li> <li>DefaultDomain</li> <li>DefaultServer</li> <li>Metadata Repositories</li> <li>User Messaging Service</li> </ul>	Active Retire Shut Down Test Settings  Dashboard Composite Definition Flow Instances Unit Tests Policies Components Name HelloWorld Services and References Name HelloWorldServite_ep

Click on the HelloWorldService\_ep service binding. Open the Policies tab.

You will find the custom policy attached to this service binding:

#### TelloWorld [1.0] (1)

SOA Composite 🔻

Dashboard	Policies	Properties				
			nt dropdown to erences are rec		onding effective policy references. Fo	r poli
Constraint	None					
Globally A	ttached F	Policies				
Category/F	Policy Name					
no rows ye						
• •	t					
no rows ye	ttached P		Inable	💥 Disable	💫 Override Policy Configuration	
no rows ye Directly A View -	ttached P	<b>olicies</b> h/Detach		💥 Disable	Sverride Policy Configuration	
no rows ye Directly A View -	t ttached P Attac Policy Name	<b>olicies</b> h/Detach	✓ Enable	💥 Disable	Sverride Policy Configuration	

Select the policy and click on Override Policy Configuration to inspect the properties that apply to the policy binding.

You will find a JMS Connection Factory and JMS Destination configured – used to publish the service execution from the custom policy to the JMS destination.

The Hello World [1.0] (a)				
💖 🕈 HelloWorldService_ep (Web	Service)			
Dashboard Policies Properties				
attached/detached, effective policy references	own to view the corresponding effective po Security Configuration Details	licy references. For policy set flagged as "Not Valid", click		ty policy refe ×
Constraint None Globally Attached Policies	Name	Value	Original Value	
Category/Policy Name	JMSConnectionFactory	jms/handson-jms-connectionFactory	jms/ServiceExecutionReportingCF	cy Set
no rows yet	JMSDestination	jms/ServiceExecutionReportingTopic	jms/ServiceExecutionReportingQueue	.,
	serviceAttribute			
	operationsMap	{ "HelloWorldRequestMessage" : { "d	<pre>{ "getFlightDetailsRequest" : { "operation" : "</pre>	
	reference.priority			
Directly Attached Policies				
View 🔻 🥔 Attach/Detach 🛛 🔗 E				
Category/Policy Name				
▲ management amis/monitoring				

#### **Use StreamExplorer to Consume and Analyze Service Execution Reports**

In StreamExplorer, we will leverage the service execution reports published to the JMS topic. We will do so using a new stream and a new associated exploration.

Create a new stream. Set an appropriate name and select JMS as the Source Type.

Create Stream				×
& Back	Source Details	Type Properties	Shape	Next 📏
* Name	ServiceExecutionReport	s		
Description				
20cT	E. t t			
	Enter tag			
* Source Type	<u>L.</u>			T
	Create Exploration wi	ith this source (Launch Ex	ploration Editor)	
			[	Cancel Create

Press Next.

Configure the same JMS Topic that was configured for the Custom Policy. Use the url t3://localhost:7101 to connect to the WebLogic domain; username and password are weblogic and weblogic1.

Create Stream		×
So So	urce Details Type Properties Shape	Next 📏
Type: JMS		
* URL	t3://localhost:7101	
* Username	weblogic	
* Password	••••••	
* JNDI name	jms/ServiceExecutionReportingTopic	
	Car	Create

Press Next.

Configure the shape as shown in the next figure:

Create Stream			×
Souther Southern	O O O O O O O O O O O O O O O O O O O	s Shape	Next 🔪
Type: JMS JMS Shape			
Name	ServiceExecutionReport		
Manual Mapping	service	String	• ×
	operation	String	• ×
	ecid	String	• ×
	stage	String	• ×
	executionTimestamp	String	• × +
○ Select Shape	Choose a Shape		
			Cancel Create

#### Press Create.

The Exploration editor opens. Enter a name:

Create Explo	ration ×
* Name	ServiceExecutions
Description	Provide a description that help people understand this Exploration
Tags	Enter Tag
* Source	ServiceExecutio ×
	Cancel Create

Press Create.

The Exploration appears.

	cplorer				help 🔻 wie	evs 🔻 🚥
K Return To Catalog				② Configure a Target	🗁 Actions	n a   🖸
Exploration Editor Welcome to the Explo types of streams can	rer. This is where you discover interesting things a be refined using the Range Window drawer to the r	bout your data stream using analytic ight. To learn more about these tools,	tools and filters. Incoming data appea watch this space as you explore.	irs both in the Live Output Stream table and as a	graph below. In addition, c	×
ServiceExecutions Oraft						
Sources ServiceExecutio	×					(
Summaries		Group by	Filters			
Add a Summary			+ Add a Filter			
▲ Live Output Stream					Properties	Timestamp
service	operation	ecid	stage	executionTimestamp		

Publish the Exploration, from the Actions menu.

Switch to SoapUI. Open the HelloWorld project and send a request to the process operation in the HelloWorldBinding service:



This call triggers the OWSM policy that reports the service execution to the JMS destination from which the StreamExplorer stream consumes it, feeding it to the exploration as is shown below.

DRACLE <sup>®</sup> Stre	am Explorer				help 🔻 wlev	vs 🔻 🔳
Return To Catalog				② Configure a	Target 📑 Actions 🞼	n (*   (
Exploration E Welcome to th types of stream	e Explorer. This is where you discover inter	resting things about your data stream using analytic drawer to the right. To learn more about these tools,	cools and filters. Incoming watch this space as you e	data appears both in the Live Output Stream table a xplore.	and as a graph below. In addition, ce	> ertain
ServiceExecutions	Draft					
Sources ServiceExec	utio 🗙					
Summaries		Group by	Filters			
+ Add a Summary			Add a Filter			
▲ Live Output Stream					Properties	Timestamı
service	operation	ecid	stage	executionTimestamp		
default/HelloWorld /HelloWorldService_ep	processResponse	9134022d-1b48-416c- b7fe-6ec7ca332f62-00000087,0	response	1431270406892		
default/HelloWorld		9134022d-1b48-416c-				

Make a few additional calls in SoapUI – to see more results produced in the Exploration.

## **Create an Exploration to Report the Number of Calls Per Service Operation**

One step beyond what we have done so far would be an aggregation exploration that reports the number of calls made to all services and operations. Using StreamExplorer, that is of course easy to achieve.

New Exploration, using previous as source

Add a Summary: count(ecid) group by service, operation. Add a Range – say one hour. You could play a little with the evaluation frequency – once per second is pretty often.

Note: now both request and response are included; add a filter to block events with stage equal to *response*.

ources ServiceExecutions X	ServiceExecutions Range: 1 hours I Evaluation frequency: 1 second
ummaries	Group by service and operation Filters Match All of the following stage equals request

make some calls from SoapUI.

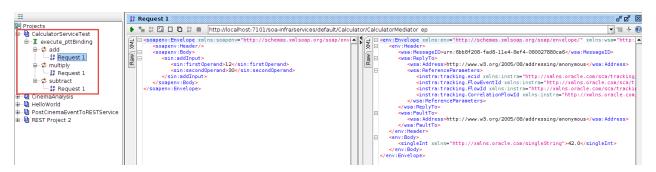
। 🕲 🖺 🦽 🔷 💥 👪			S
	F Request 1	4 a X	
📴 Projects 🖮 🖶 CinemaAnalysis	Request 1	1	్లి 🗵
🖶 🚇 HelloWorld	🗼 🐂 🕼 🖸 🗖 🛱 🐘 http://localhost:7101/soa-infra/services/defau	It/HelloWorld/HelloWorldService ep	▼ 1% + 10
i Hellewordbinding 	<pre>State in the loge in lision pervent the i/schemes.kml soap.org/sources is soapen i bodys</pre>	<pre>intrelowonoservce ep ap/envi</pre>	<pre>780Cca6c/vss:Hessag2D&gt; ting/anonymous /vsalns.oracle.com/sca/tracking. http://xslns.oracle.com/sca/t //xslns.oracle.com/sca/t //xslns.oracle.com/sca/trackii stra="http://xslns.oracle.com ing/anonymous </pre>

The service calls are counted and reported:

AggregatedServiceExecutions						
Sources ServiceExecutions X		ServiceExecutions Ra	ange: 1 hours	Evaluation	frequency: 1	seconds
Summaries COUNT of ecid	Group by service and operatio	Filters Match All of the follow stage equals request	wing			
Live Output Stream					Properties	Timestamp
-	service		operation		Properties	Timestamp
COUNT_of_ecid		elloWorld/HelloWorldService_ep	operation process		Properties	Timestamp
COUNT_of_ecid	default/He	elloWorld/HelloWorldService_ep elloWorld/HelloWorldService_ep			Properties	Timestamp
Live Output Stream COUNT_of_ecid 4 3 2	default/Hu default/Hu		process		Properties	Timestamp

## The Calculator Composite added into the mix

There is another SOA Composite deployed that also exposes a service that has the policy attached to it: the Calculator composite. It offers three operations – add, subtract and multiply. SoapUI has a test project for the service exposed by this composite:



The policy is attached to the service binding in this composite, meaning that every call made from SoapUI (or any other consumer) to the service results in messages being published to the JMS target and being consumed by the StreamExplorer application. You may want to briefly inspect this policy attachment:

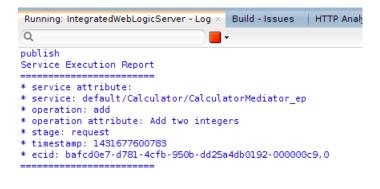
Application Deployments So A So A Contract Mediator - ep (Web Service) Attached Policies Calculator [1.0] Calculator Mediator - ep (Web Service) Calculator Mediator - ep (Web Service) Calculator - ep (Web Service) Calculator - ep (Web Service)	t Navigation ▼	T Calculator [1.0] SOA Composite →	Page Refr	Logged in as eshed May 15, 2015 10	weblogic 🔲 localho
	Application Deployments SOA service-bus (DefaultServer) soa-Infra (DefaultServer) () default () default () default	Dashboard         Policies         Properties           Select an expression from the constraint dropdown to view the corresponding effective policy references. For policy set flagg details. For security policy references, click the violations count link to view violation details. When policies are attached/details			lculated.
User Messaging Service     Effective Only @ All @ Detach     Imagement   amis/monitoring     Page Refreshed N     Calculator [1.0]     Page Refreshed N     Imagement        Page Refreshed N     Page Refreshed N     Imagement        Page Refreshed N     Imagement           Page Refreshed N        Imagement           Page Refreshed N                    Imagement   amis/monitoring	WebLogic Domain	Globally Attached Policies Category/Policy Name Policy Set		Enabled	Total Violation
Security Configuration Details       X         Name       Value       Original Value         JMSConnectionFactory       jms/handson-jms-connectionFactory       jms/ServiceExecution         JMSDestination       jms/ServiceExecutionReportingTopic       jms/ServiceExecution         serviceAttribute			ective Only 💿 All	M Detach	
Name         Value         Original Value           JMSConnectionFactory         jms/handson-jms-connectionFactory         jms/ServiceExecution           JMSDestination         jms/ServiceExecutionReportingTopic         jms/ServiceExecution           serviceAttribute             operationsMap         { "addInput" : { "operation" : "add", "operationAttribute" : "Add two integers", { "getFlightDetailsReq	SOA Composite	<ul> <li>management amis/monitoring</li> <li>Image: Image: Image</li></ul>	Effective ✓	~	
JMSConnectionFactory       jms/handson-jms-connectionFactory       jms/serviceExecution         JMSDestination       jms/serviceExecutionReportingTopic       jms/serviceExecution         serviceAttribute	SOA Composite	management amis/monitoring       Image: Constraint of the service of the servi	Effective	~	l Refreshed <b>Ma</b>
serviceAttribute	Calcula Societies Composite	management amis/monitoring       Image: Constraint of the service of the servi	Effective V	Page	×
operationsMap { "addInput" : { "operation" : "add", "operationAttribute" : "Add two integers", { "getFlightDetailsReq	SOA Composite	management     amis/monitoring	Effective V	✓ Page Original Value	Refreshed Ma
	SOA Composite	management     amis/monitoring	Effective	Page Original Value jms/ServiceEx	Refreshed Ma
reference.priority	SOA Composite Calcula Security Configuration Name JMSConnectionFa JMSDestination	management     amis/monitoring	Effective	Page Original Value jms/ServiceEx	Refreshed Ma
	SOA Composite Calcula Security Configuration JMSConnectionFa JMSDestination serviceAttribute			Page Original Value jms/ServiceEx jms/ServiceEx	Refreshed Ma

Make a few calls to the Calculator service from SoapUI. Verify that StreamExplorer also starts reporting on these calls:

Sources ServiceExecutions	×				
Summaries COUNT of ecid Add a Summary	Group by service x operation x	Filters Match All of the following stage equals request	ng		
Live Output Stream				Properties	Timestamp
	service		operation	Properties	Timestamp
COUNT_of_ecid		HelloWorld/HelloWorldService_ep	operation process	Properties	Timestamp
COUNT_of_ecid	default			Properties	Timestamp
COUNT_of_ecid 6 5	defaultu	HelloWorld/HelloWorldService_ep	process	Properties	Timestamp
<b>COUNT_of_ecid</b> 5 2	default default default	HelloWorld/HelloWorldService_ep HelloWorld/HelloWorldService_ep	process process	Properties	Timestamp
<b>COUNT_of_ecid</b> 6 5 2 6	default default default default default	HelloWorld/HelloWorldService_ep HelloWorld/HelloWorldService_ep Calculator/CalculatorMediator_ep	process process substract	Properties	Timestamp
Live Output Stream COUNT_of_ecid 6 5 2 6 4 1	default default default default default	HelloWorld/HelloWorldService_ep HelloWorld/HelloWorldService_ep Calculator/CalculatorMediator_ep Calculator/CalculatorMediator_ep	process process substract multiply	Properties	Timestamp

You can probably come up with some ideas as to how this information about service invocations can be used in monitoring scenarios – remember that we also have access in Stream Explorer to the timestamps for start of request [processing] and end of response [creation].

Note: the console for the Integrated WebLogic Server also shows the output from the OWMS Policy:



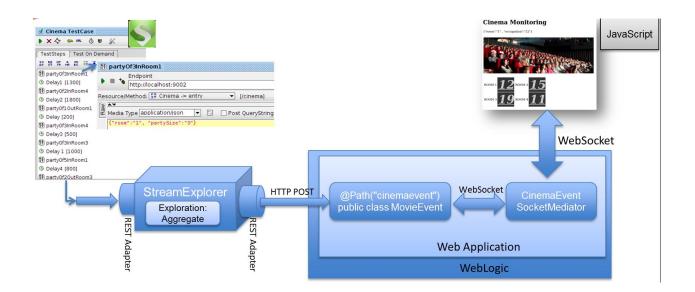
# 10. Stream Explorer, Web Socket and Dashboard

A common desire when doing real time event processing with Stream Explorer and/or Oracle Event Processor is the ability to present the findings from Stream Explorer in a live dashboard. This dashboard should hold a visualization of whatever information we have set up Stream Explorer to find for us – and it should always show the latest information.

User interfaces are commonly presented in web browsers and created using HTML(5) and JavaScript. As part of the HTML5 evolution that brought today's browsers, we now have the ability to use Web Sockets through which we can push information from server to browser to have the user interface updated based on messages pushed from the server. This allows us to create a dashboard that listens from the browser to a Web Socket and use whatever messages appear on the web socket to actualize the user interface.

We will extend the Stream Explorer application last discussed in section 6 (on events in a movie theater) (the one that exposes a REST interface to which publish JSON messages). These messages report on groups of people entering or leaving a specific room in a movie theater.

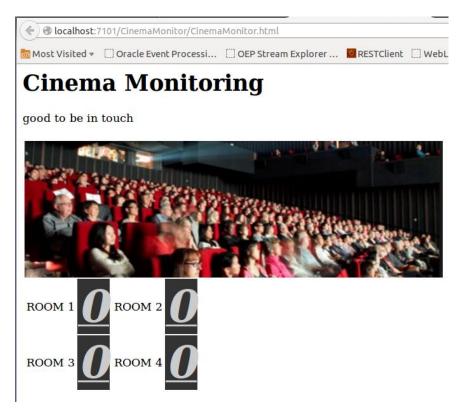
The exploration we add will aggregate the information from the messages – providing us with a constant insight in the total number of people in each room. This information is subsequently pushed to the REST service exposed by a Java EE application that routes that information across the web socket to the HTML5 client. The next figure illustrates the application architecture



See blog articles <u>https://technology.amis.nl/2015/05/14/java-web-application-sending-json-messages-through-websocket-to-html5-browser-application-for-real-time-push/</u> and <u>https://technology.amis.nl/?p=36008</u> for background and details.

The steps are:

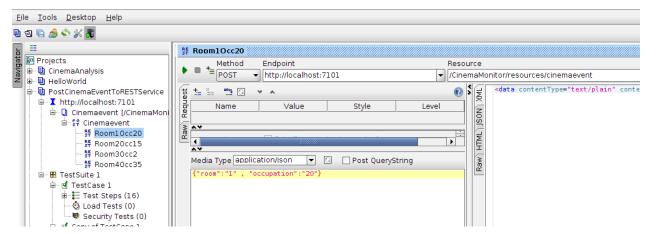
Open URL <u>http://localhost:7101/CinemaMonitor/CinemaMonitor.html</u> - or use the bookmark in the browser toolbar. This will show the monitor web application – running in WebLogic, based on the application CinemaMonitorWebApplication in folder /u01/app/oracle/jdeveloper/mywork.



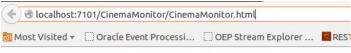
This monitor connects to a web socket channel at ws://localhost:7101/CinemaMonitor/cinemaSocket that is exposed by a class in the web application called *CinemaEventSocketMediator*.

The same web application also publishes a REST service. The JSON payload for any POST message sent to http://localhost:7101/CinemaMonitor/resources/cinemaevent will be forwarded to the WebSocket channel and from *CinemaEventSocketMediator* also to the web client.

Open SoapUI. Use the PostCinemaEventToRESTService project to make some test calls to this REST service.



After posting one or more requests from SoapUI, the monitor will show the latest numbers received:



# **Cinema Monitoring**

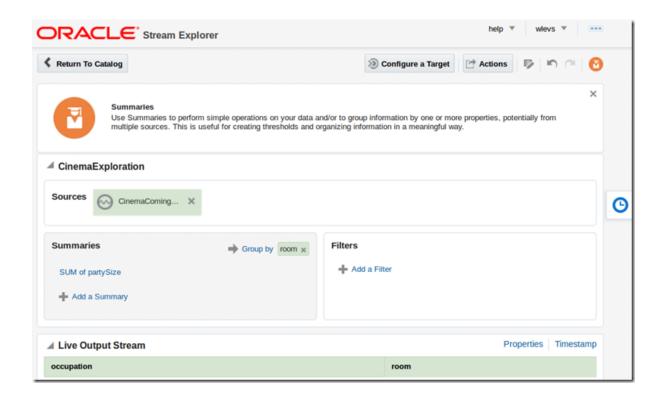
{"room":"3" , "occupation":"2"}



Note: after a period of idleness, the web socket channel may time out. That is currently not visible in the Cinema Monitor. If you have the impression that the dashboard is not updated although it really should, please reload the web page in the your browser to create a fresh web socket connection.

The next step is to create a StreamExplorer exploration that uses the same incoming events about parties entering or leaving the rooms in the theater that we used before and turns them into summation messages that are forwarded to the REST service called from SoapUI. That exploration will then start feeding the live dashboard Cinema Monitor.

Create an exploration, for example called CinemaExploration. Create a Summary of type SUM based on the property partySize and group by room. Edit the Properties and change the name of property SUM\_of\_partySize to *occupation*. The exploration will look like this:



Next, click on Configure a Target.

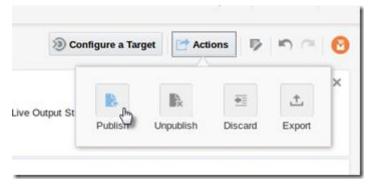
Exploration Editor Welcome to the Explorer. This is where you discover interesting things about your data stream using analytic tools and filters. Incoming data appears both in the Live Output Stream table and as a graph below. In addition, certain types of streams can be refined using the Range Window drawer to the right. To learn more about these tools, watch this space as you explore.	Exploration Editor Welcome to the Explorer. This is where you discover interesting things about your data stream using analytic tools and filters. Incoming data appears both in the Live Output Stream table and as a graph below. In addition, certain types of streams can be	Return To Catalog	Sconfigure a Target C Actions P n n 0
	refined using the Range Window drawer to the right. To learn more about these tools, watch this space as you explore.	Welcome to the Explorer. This is where you di	iscover interesting things about your data stream using analytic tools and filters.
	CinemaExploration 🖉 Draft		

#### Select type REST and set the URL

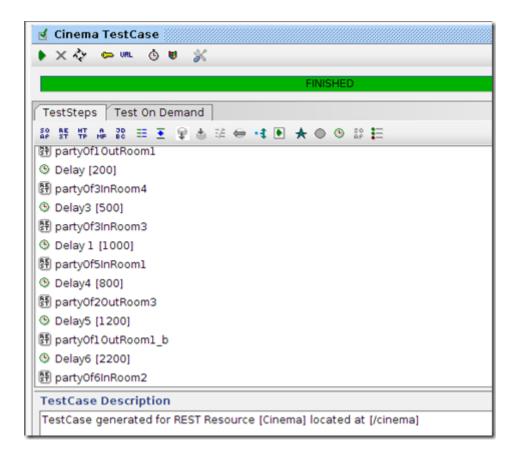
Cancel		Configure a Target	Finish
	Target Type	REST	•
	* Target URL	http://localhost:7101/CinemaMonitor/resources/cinemaevent	

Click on Finish.

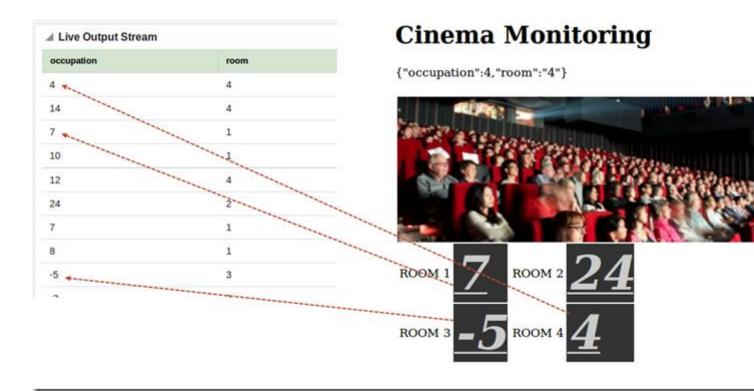
Publish the Exploration.



Now we can run a test case in SoapUI to send test messages to the Stream Explorer application:



Here is what the live output stream in the Stream Explorer UI shows next to a screenshot taken of the Cinema Monitor dashboard:



The dashboard is constantly updated with the most recent finding published by Stream Explorer.

# **Appendix: VM configuration details**

The local Oracle XE database (SID = XE, port =152, host: localhost)

The SYS user: sys/oracle

The WC account that holds several database resources: wc/wc

APEX can be used as well: http://localhost:8080/apex. The administrator credentials: admin/admin and the default workspace is called INTERNAL.

Application Express × 🗃 AMIS Technology Bl	× 🛔
< 🕘 localhost:8080/apex	
	Enter Application Express workspace and credentials. Workspace INTERNAL Username ADMIN Password Click here to learn how to get started
	development tool that lets you share data and create custom applications. xperience, you can develop and deploy powerful applications that are both fast

To run JDeveloper:

```
cd /u01/app/oracle/FMW12c/jdeveloper/jdev/bin/
```

./jdev

The Integrated WebLogic Server (start from within JDeveloper) : weblogic/weblogic1 – running at port 7101. Administration Console URL: http://localhost:7101/console

To run the OEP domain:

cd /u01/app/oracle/OEP12c/user\_projects/domains/sx/defaultserver

. /startwlevs.sh

(or start from within JDeveloper from the Resources palette: IDE Connections, OEP Server, sx\_domain\_oep\_connection | start)

username/password OEP domain: wlevs/weblogic1

WLevs Visualizer Console: http://localhost:9002/wlevs/

Stream Explorer Console: http://localhost:9002/sx

### **Host Folder Mapping**

The VM Guest folder *host\_temp* is mapped by default to the folder *c:\temp* on the host

You can change this mapping in the Virtual Box machine settings:

General	Shared Folders			
System	Eolders List			
Display	Name	Path	Auto-mount	Access
Storage	Machine Folders			
Audio	etc_puppet_files	D:/GitHub/streamexplorer-eventprocessing/files		Full
Network	host_temp	C:\temp	Yes	Full
Serial Ports	tmp_vagrant-puppet-3_manifests	D:/GitHub/streamexplorer-eventprocessing/manifests		Full
USB	tmp_vagrant-puppet-3_modules-0	D:/GitHub/streamexplorer-eventprocessing/modules		Full
Shared Folders	vagrant	D:\GitHub\streamexplorer-eventprocessing\files	Yes	Full
Shared Folders				