IBM Software Services for WebSphere

Project Zero

Matthew Perrins
Executive IT Specialist
Software Group Lab Services

Agenda

- Web SOA
  - Attributes
  - REST
  - Extending Global SOA
- Project Zero
  - Introducing Project Zero
  - Project Zero Overview
  - Core Programming Model
  - RESTful Services
  - Ajax
  - Zero Assemble
- AIM Product Plans
  - WebSphere Web 2.0 Feature Pack
  - WebSphere MQ Bridge
  - DataPower Web 2.0 Appliance
  - Other: WebSphere Commerce
- Across the Brands
  - Lotus: Community
    - Mashup Maker, Portal, Quickr, Connections

Web SOA

Web SOA

- Web SOA is an instance of SOA that uses concepts from the Web as the primary service architecture (WOA is an term from Gartner – we will use “Web Extended SOA” for the same thing)
  - REST to represent and access services
    - Entities are addressed via URL
    - GET, POST, PUT, DELETE are the actions
  - Data is encoded as JSON or XML, and ATOM
  - Rich User Interfaces built using AJAX
- Key aspects of building an effective Web Extended SOA
  - UI runs in any commodity web-server / browser
  - Make content simple and human readable
  - Use well-established, ubiquitous technologies for scalability, performance and security

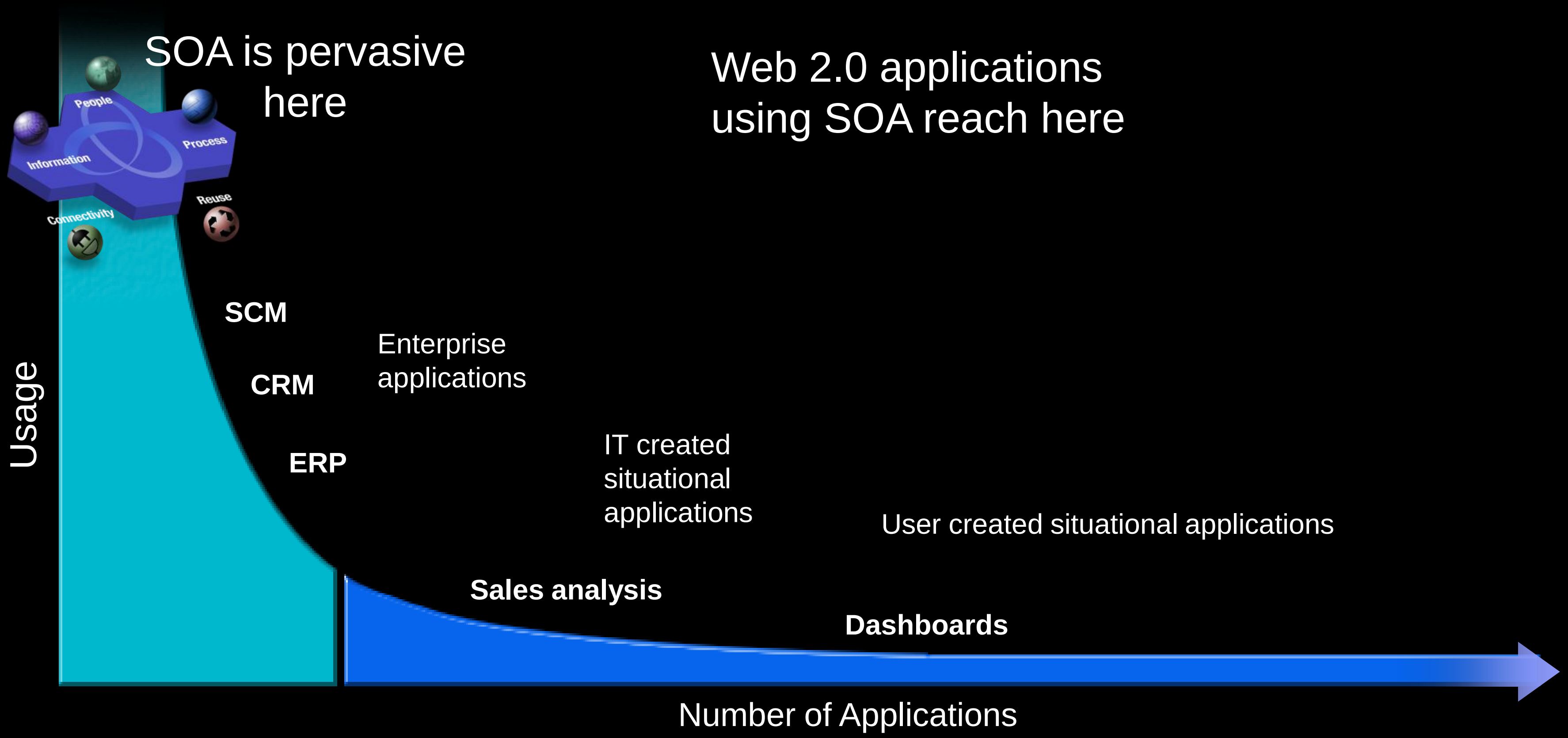
Why a Web SOA

- A Web Extended SOA (or “WOA”) is a SOA because a SOA is an architecture for representing services
  - A Web Extended SOA is just one implementation of a SOA where the Web is the SOA platform
- A Web Extended SOA extends the reach of your enterprise SOA
  - Putting services in the hands of the masses

Simple to Access - Web 2.0 Applications Use SOA

To Serve New Markets with Specific Needs

From dozens of markets of millions of users to millions of markets of dozens of people



What is a Mashup?

- A **mashup** is a web application that combines data or capabilities from more than one source into an integrated experience
  - Very popular on the consumer web, where over 3.45 new mashups/day are appearing:

Zillow

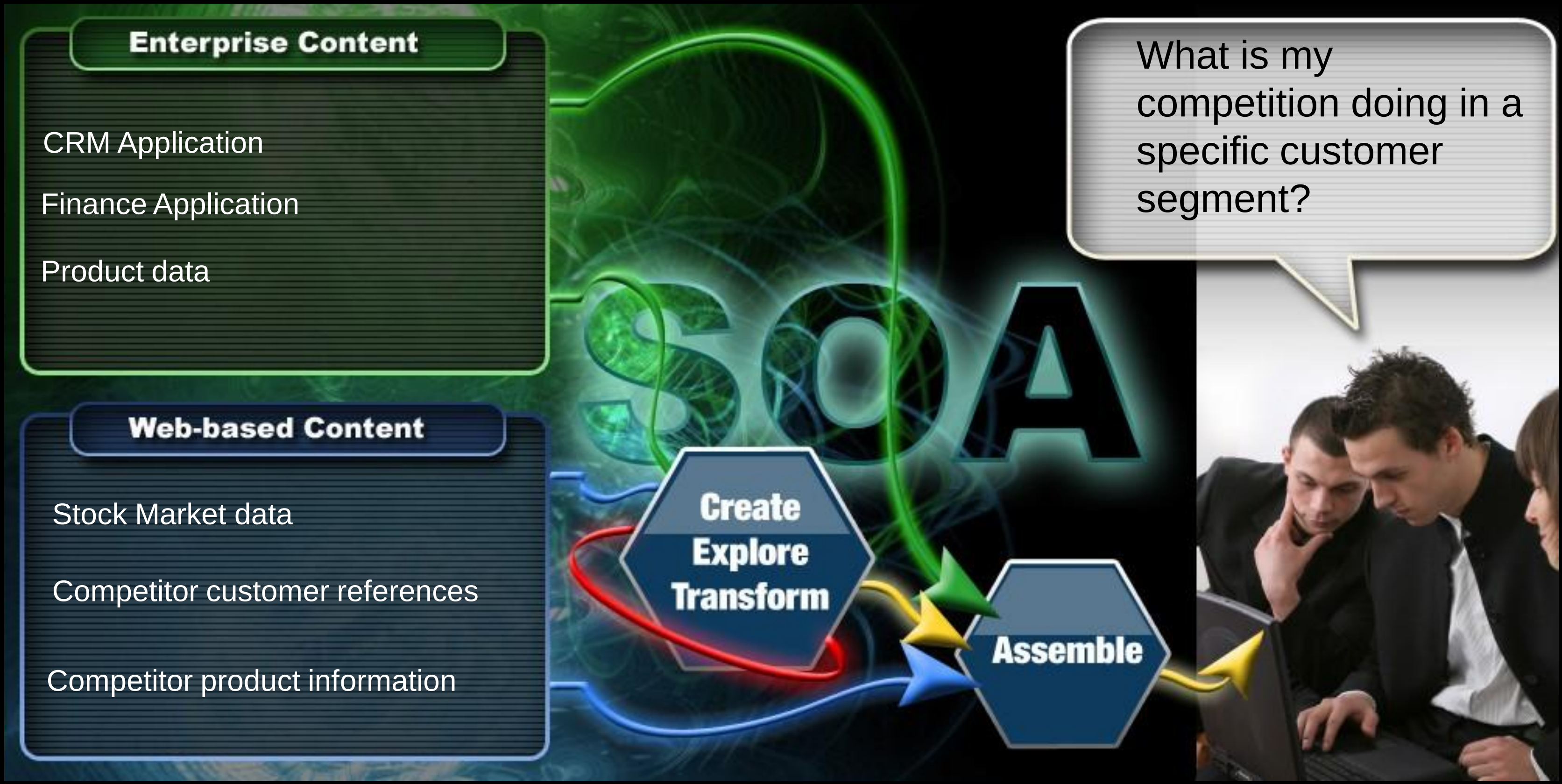
Pageflakes.com

SeatSnapper

- What typically characterizes a mashup?
  - Lightweight integration of applications (enables rapid development)
    - “Widgets” that make up a mashup are often developed and deployed independently without knowledge of each other
    - Widgets can be mashed and wired together in the browser
  - Utilize web technologies like HTTP, JSON, XML, Javascript, ATOM, RSS
  - Often incorporates one or more public API and online services
  - New applications deliver new insights and capabilities (1+1 = 4)
  - Often, mashups can be customized by the end user

Introducing Project Zero

Agility….
Project Zero



Motivation

- Enterprise Customers
  - **Problem**: JEE and Integration products do more than some applications need
  - **Result**: Movement to Open Source Solutions – e.g., Tomcat, Mule
  - **Opportunity**: Leverage investment in Java, introduce agilie development via dynamic scripting and assembly, that can be managed and run with enterprise qualities
- Mid-Market
  - **Problem**: Current “Express” products are indeed Enterprise Software
  - **Result**: Poor adoption, Movement to Open Source Solutions.
  - **Opportunity:** Provide an architect OEM-able mid-market platform that enables the creation of dynamic web applications and simple integration
- Hosting companies and hosted services
  - **Problem**: Current IBM middleware is not designed for multi-tenancy or hosted scale
  - **Result**: No adoption, LAMP, MySQL are the pervasive solution
  - **Opportunity** : Provide a host-able platform, supports multi-tenancy, development environment hosted as a service, applications hosted as a service

What is Project Zero?

- **Project Zero is an Agile Web Application Platform.**
  - **Architected around Dynamic Scripting, REST, Rich Web Interfaces, AJAX, and Feeds.**
  - **Optimized for speed of development, simple deployment, and cost-effective operation.**

Project Zero Overview

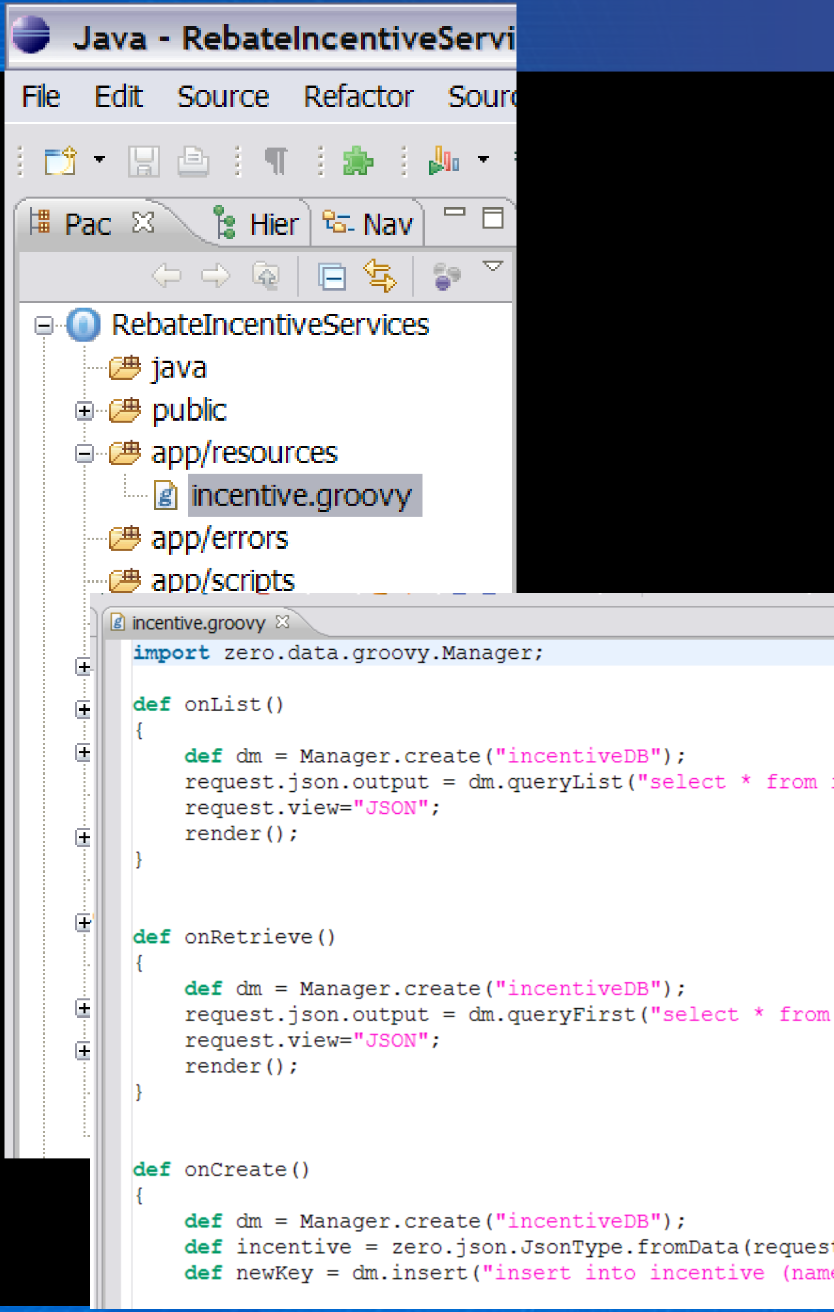
Innovations in Project Zero

- **CREATE:**
  - Web-Oriented Programming Model
- **ASSEMBLE**:
  - Simple Application and Service Assembly
- **EXECUTE:**
  - “New Reality” Runtime

Create

- Simple Conventions to avoid excessive code and configuration
- Dynamic Scripting and Templates
  - PHP language syntax
  - Groovy (Java Language Syntax)
  - Java as System Programming Language
- Effortless creation of Restful Services and Data Feeds (RSS, ATOM)
- Data Access using pureQuery
- Reactive Client integrated Dojo with Zero
- State externalized into a shared memory space (Global Context)
- State-less, Event-driven architecture

- Catalog of Services and Libraries providing useful building blocks



```
import zero.data.groovy.Manager;

def onList()
{
    def dm = Manager.create("incentiveDB");
    request.json.output = dm.queryList("select * from 
    request.view="JSON";
    render();
}


def onRetrieve()
{
    def dm = Manager.create("incentiveDB");
    request.json.output = dm.queryFirst("select * from
    request.view="JSON";
    render();
}


def onCreate()
{
    def dm = Manager.create("incentiveDB");
    def incentive = zero.json.JsonType.fromData(reques
    def newKey = dm.insert("insert into incentive (name
```

Languages and Scripting

- Zero is a dynamic scripting platform
- Application Logic is created in one of two scripting languges
  - *Groovy (for people that prefer Java)*
  - *PHP*

- Java is positioned as the “system” language
  - Mostly used to implement system extensions and application libraries
  - Entire applications can be written in Java, if desired
    - Requires more configuration

PHP Support

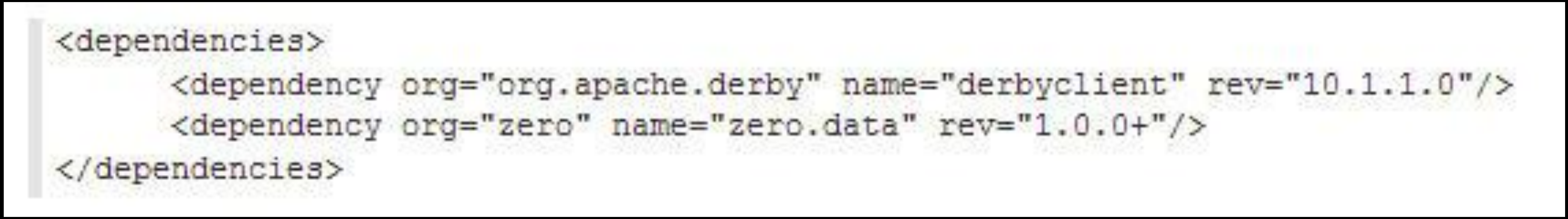
- The Project Zero PHP runtime (P8) is built on top of IBM's J9 JVM
  - Supports use of many PHP Extensions
    - XAPI-C interface allows C-based extensions
    - XAPI-J interface allows Java based extensions
  - Supports bridging between Java and PHP
- All of PHP is not supported
- PHP runtime provided directly by Project Zero
- The goal of P8 in Project Zero is to provide PHP support within the Zero programming model

Application Centric Runtime

- Project Zero is an application-centric runtime
  - You create an application and run it
  - You do not package an application and deploy it to a multi-application server
  - Each application runs in its own process (JVM)
  - Runtime is designed to be short lived
- Project Zero is a full stack runtime
  - Everything needed to run the application is provided by Project Zero
    - Including the HTTP stack
  - No external proxy or web server is required
  - An external proxy is used for clustering and multiapp routing

Modular Architecture

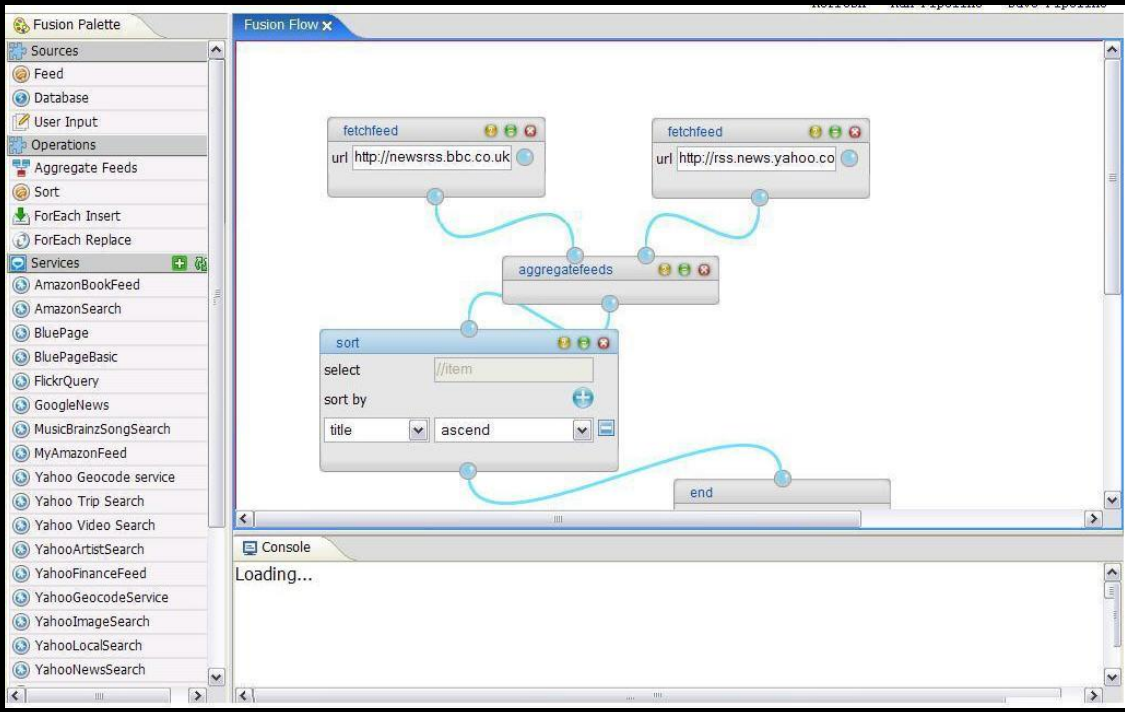
- Zero applications are based on a very small core
  - 4.3 MBytes (includes Groovy).
  - PHP adds additional 5 MBytes
  - Core provides all of the Zero framework and runtime support, including HTTP transport
- Zero tools are provided separately
  - Eclipse tools and Command Line tools are provided
  - Both less than 4 MBytes
- Additional features provided in downloadable modules
  - Applications declare a dependency on desired features (using Ivy)
  - A package management system provides the ability to resolve those dependencies on the local machine or pulls them from a remote catalog server



```
<dependencies>
        <dependency org="org.apache.derby" name="derbyclient" rev="10.1.1.0"/>
        <dependency org="zero" name="zero.data" rev="1.0.0+"/>
</dependencies>
```

Assemble

- Composition of applications by “wiring” REST services using the SPLICE flows. Incorporating both activity and data flows.
- A solution may be rapidly assembled by combining existing feeds and services that enrich, sort, and filter data in a pipeline. Either visually or programmatically.
- Configure templates to alter pipeline routes, log events along the pipeline, as well as transform data
- Adapters to enhance integration with existing systems.



fetchfeed — url: http://newsrss.bbc.co.uk

fetchfeed — url: http://rss.news.yahoo.co

sort — select: //item — sort by: title, ascend

Loading...

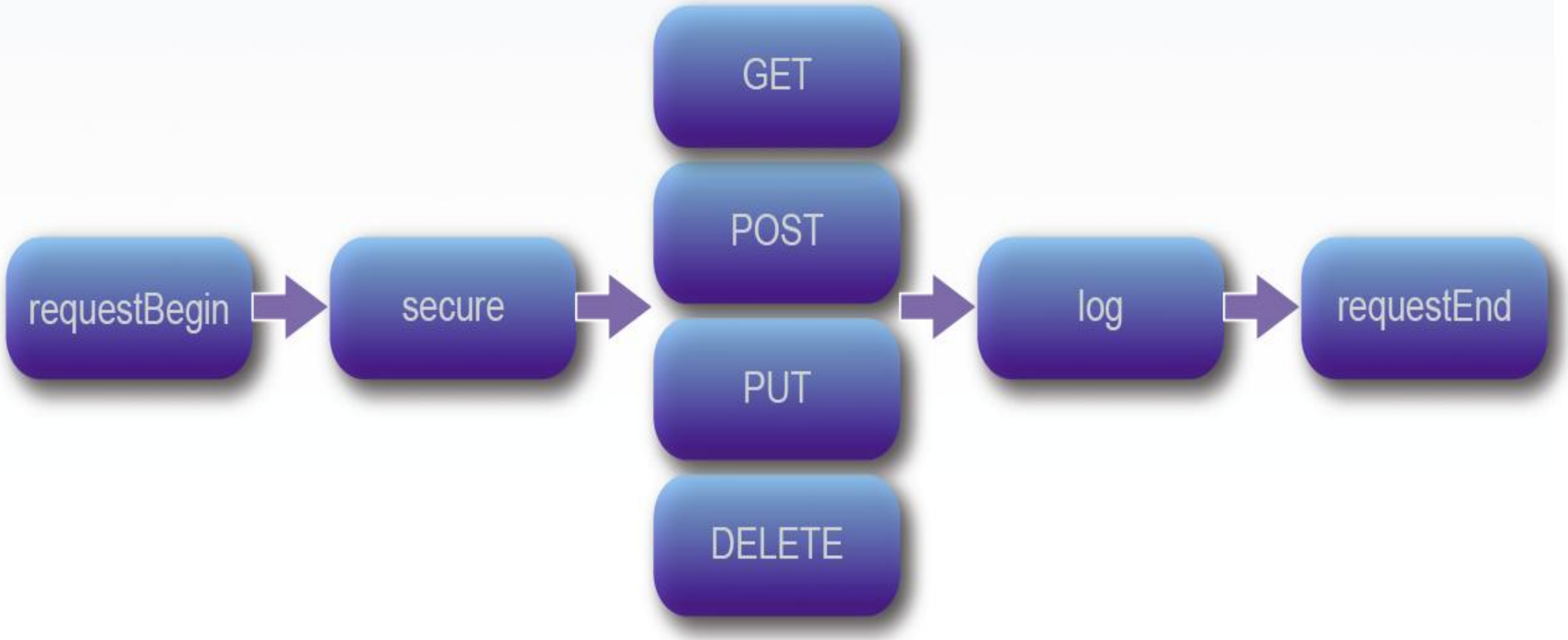
Execute: Runtime Charcateristices

- Desired traits
  - Nimble - Instant On
  - Clean - Graceful recovery, isolation, tolerates “bad” code
  - Cheap - Cost effective to run in small and large quantities
- Supported on “stock” JVM
  - IBM, Sun, Mac, etc - Any JSE 5 JVM
  - Currently Zero takes about 1 second to start and consumes about 20 MBytes of memory
- Working on a “new reality runtime”
  - Modified JVM based on IBM J9 JVM
  - Looking at sharing behavior and startup time
  - Prototype shows startup times with an order of magnitude improvement and 2.4x improvement in memory footprint

Core Programming Model

Events

- All behavior in the system is modeled as a set of event
  - Applications are built by handling these events and providing desired behavior
  - Similar to AJAX model or classic UI programming



Event Handlers

- All handlers are stateless
- Can be implemented in Groovy, PHP, and Java

Groovy

```
println “Hello World”
```

```
def onGET()
{
   println “Hello World”
}
```

```
function onGET()
{
   echo “Hello World”;
}
```

PHP

```
public void onGET()
{
   PrintWriter writer = (PrintWriter)zget(“/request/writer”);
   writer.println(“Hello World”);
}
```

Java

```
/config/handlers += {
   “events” : “GET”,
   “handler” : “HelloWorld.class”,
   “conditions” : [“/request/path matches /hello”]
}
```

Global Context – State Management

- The Global Context (GC) provides access to and management of all application state
  - Conceptually a map of data
- Externalizes all state from the application logic
  - Enables the restartability of the JVM without data loss
  - Enables clustering and scaling to be added transparently
- Simplifies and unifies access to application state and data structures and simplifies state passing within the application
- Contains information provided by both the runtime (such as request parameters) and by the application

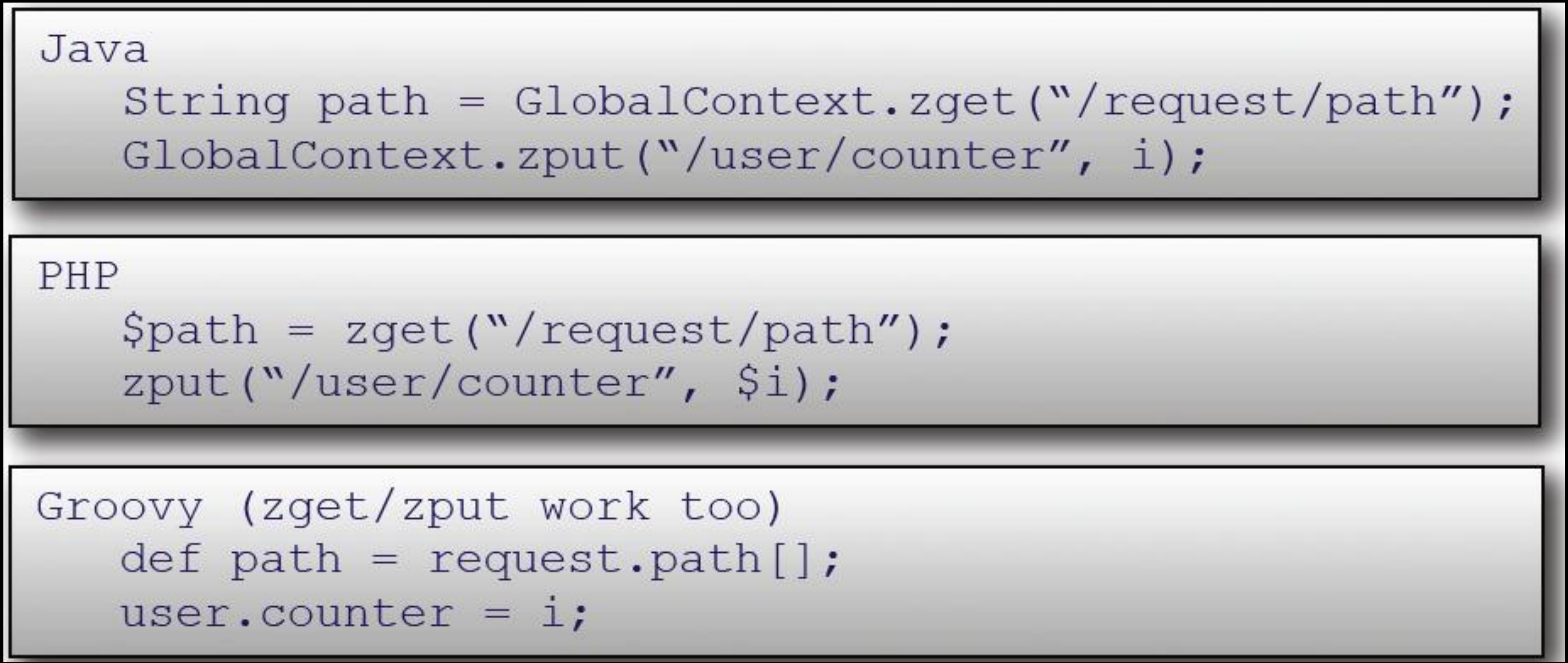
Global Context Zone

- Divided into 6 zones representing different data lifecycles

Zone	Scope/Visibility	Lifecycle
Application	All requests for all users of the application	For the life of the application, but not "persistent"
User	All request for a particular user (HTTP Session equivalent)	For first access until user times out
Request	All handlers along the path of a single request	For the duration of a single request
Event	All handlers for a single event	The duration of a single event
Client	All handlers along the path of a single request, including both browser and server	The duration of a single request
Config	All event handlers, all the time	Entire life of the application. Read only.

Accessing the Global Context

- Data is organized by a URI structure
  - First part of URI is always the Zone name
    - /app, /user, /request, /config, /event, /client
- Access is modeled after REST
  - GET, PUT, POST, DELETE



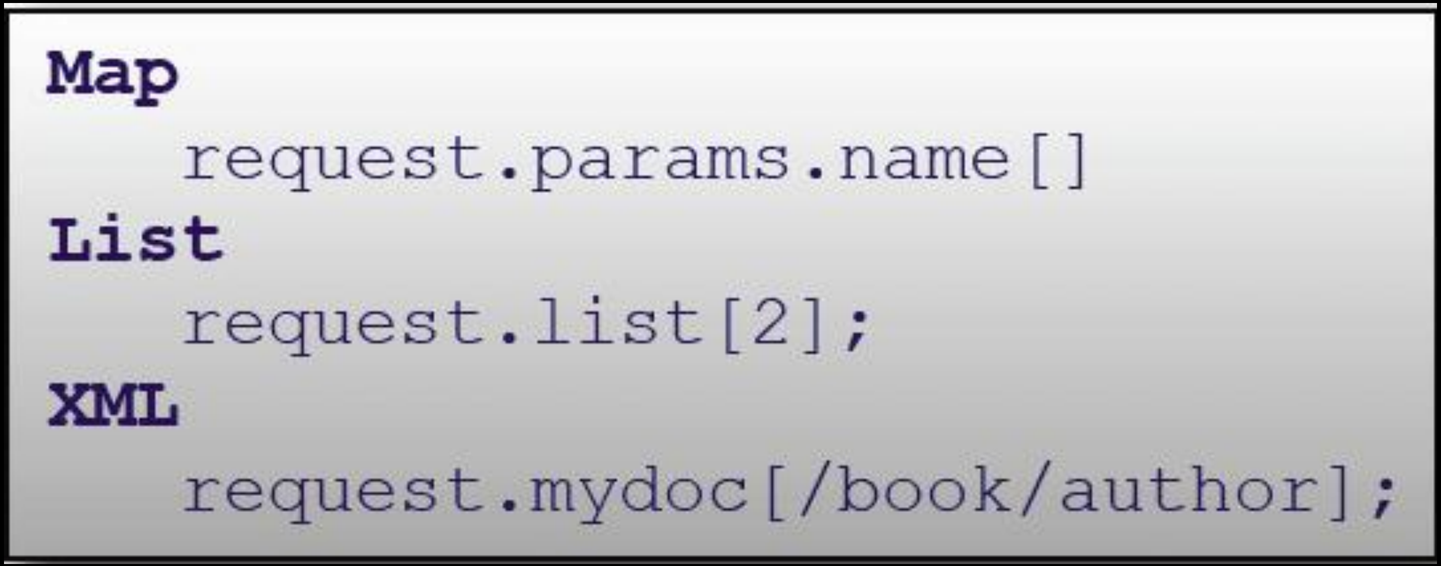
```
Java
    String path = GlobalContext.zget(“/request/path”);
    GlobalContext.zput(“/user/counter”, i);
```

```
PHP
    $path = zget(“/request/path”);
    zput(“/user/counter”, $i);
```

```
Groovy (zget/zput work too)
    def path = request.path[];
    user.counter = i;
```

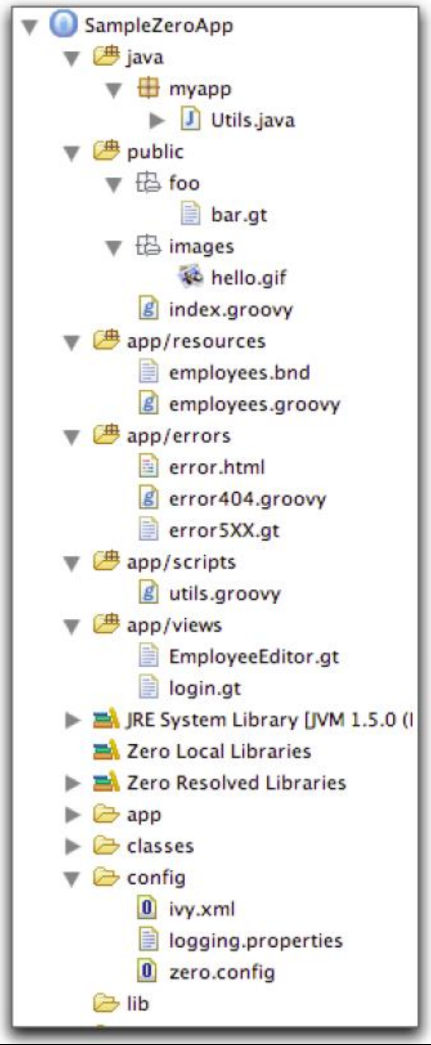
Value Pathing

- The GC provides simplified access to certain data structures
  - Called ***Value Pathing***
- Understands
  - Maps, List, Objects, XML, JSON
- Allows read and write access to internals of the structure through the GC address

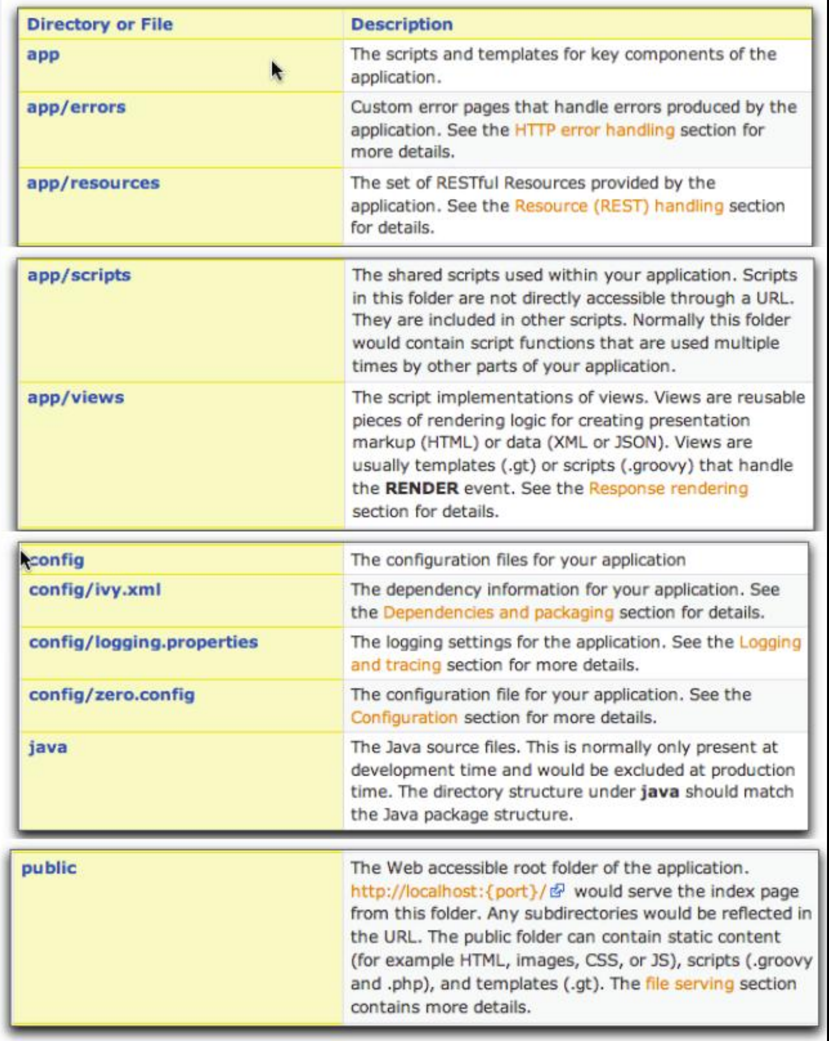


```
Map
    request.params.name[]
List
    request.list[2];
XML
    request.mydoc[/book/author];
```

Application Directory Layout



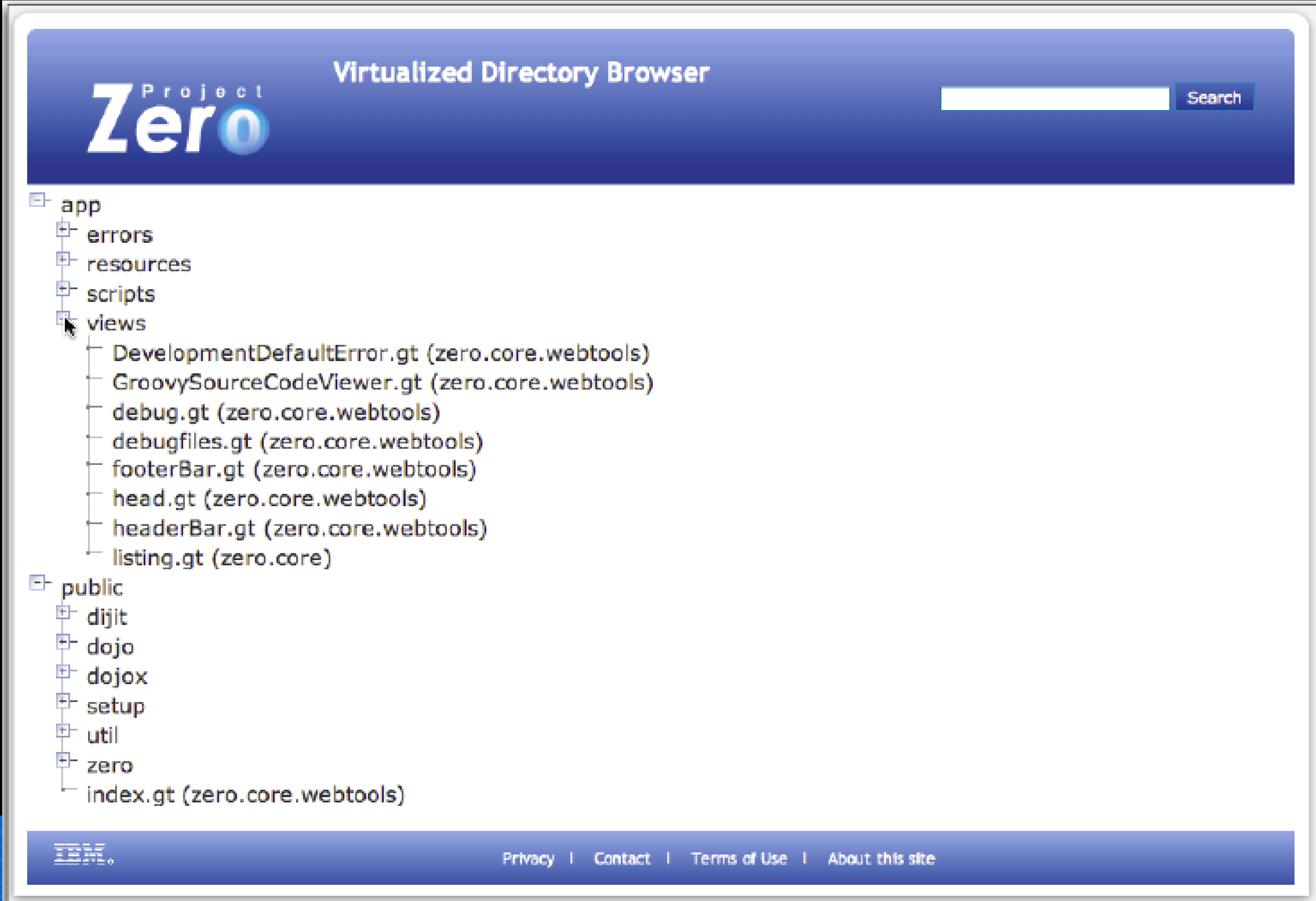
```
▼ SampleZeroApp
    ▼ java
        ▼ myapp
            ▶ Utils.java
    ▼ public
        ▼ foo
              bar.gt
        ▼ images
              hello.gif
          index.groovy
    ▼ app/resources
          employees.bnd
          employees.groovy
    ▼ app/errors
          error.html
          error404.groovy
          error5XX.gt
    ▼ app/scripts
          utils.groovy
    ▼ app/views
          EmployeeEditor.gt
          login.gt
    ▶ JRE System Library [JVM 1.5.0 (I
      Zero Local Libraries
    ▶ Zero Resolved Libraries
    ▶ app
    ▶ classes
    ▼ config
          ivy.xml
          logging.properties
          zero.config
      lib
```



Directory or File	Description
app	The scripts and templates for key components of the application.
app/errors	Custom error pages that handle errors produced by the application. See the HTTP error handling section for more details.
app/resources	The set of RESTful Resources provided by the application. See the Resource (REST) handling section for details.
app/scripts	The shared scripts used within your application. Scripts in this folder are not directly accessible through a URL. They are included in other scripts. Normally this folder would contain script functions that are used multiple times by other parts of your application.
app/views	The script implementations of views. Views are reusable pieces of rendering logic for creating presentation markup (HTML) or data (XML or JSON). Views are usually templates (.gt) or scripts (.groovy) that handle the **RENDER** event. See the Response rendering section for details.
config	The configuration files for your application
config/ivy.xml	The dependency information for your application. See the Dependencies and packaging section for details.
config/logging.properties	The logging settings for the application. See the Logging and tracing section for more details.
config/zero.config	The configuration file for your application. See the Configuration section for more details.
java	The Java source files. This is normally only present at development time and would be excluded at production time. The directory structure under **java** should match the Java package structure.
public	The Web accessible root folder of the application. http://localhost:{port}/ would serve the index page from this folder. Any subdirectories would be reflected in the URL. The public folder can contain static content (for example HTML, images, CSS, or JS), scripts (.groovy and .php), and templates (.gt). The file serving section contains more details.

Virtualized Directory

- Project Zero provides seamless integration of directories across an application and its dependencies, while maintaining each as separate entities.
- All artifacts are searched within both the application and its declared dependencies



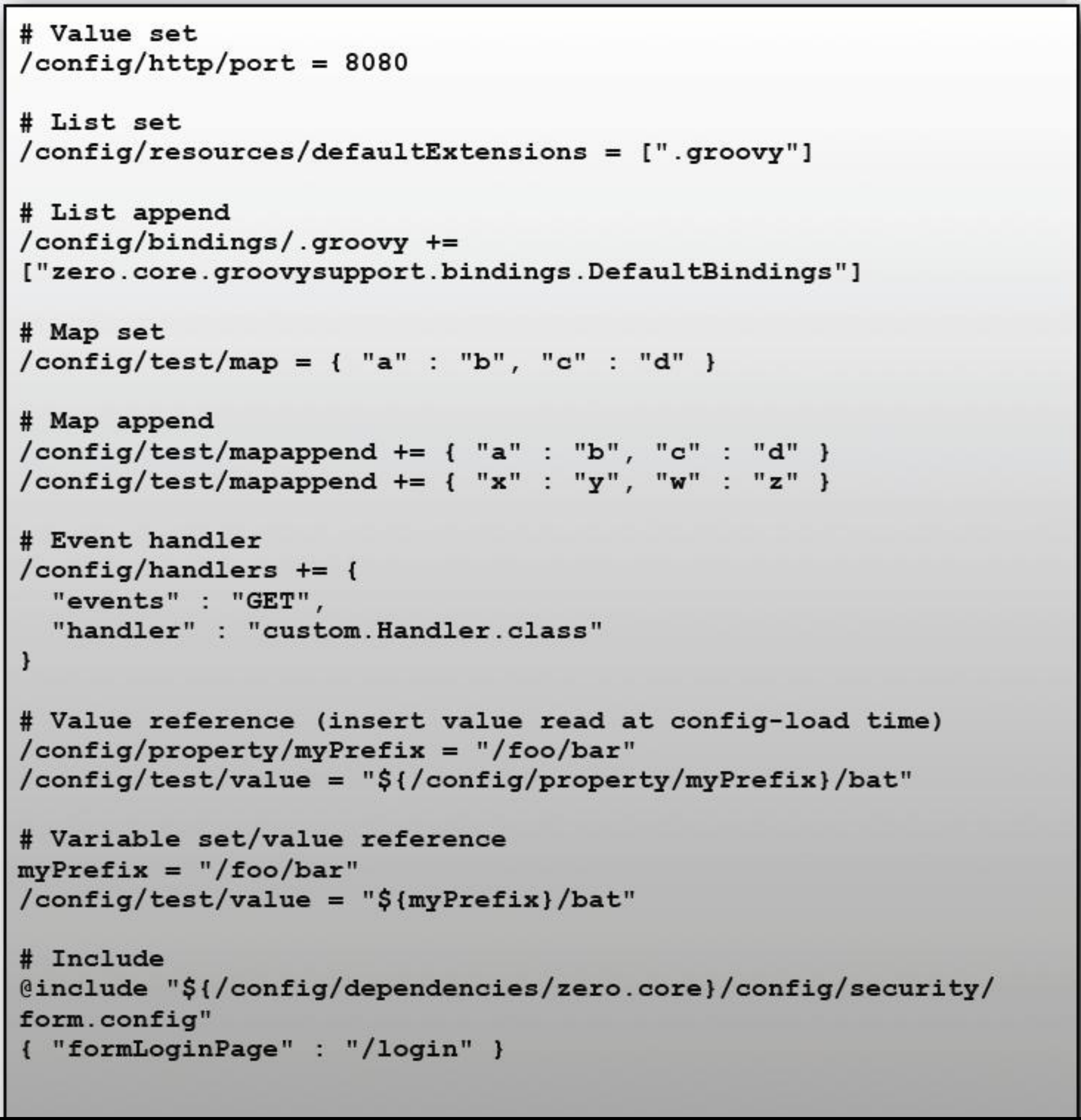
Virtualized Directory Browser
Project Zero
Search
Privacy | Contact | Terms of Use | About this site

```
app
  errors
  resources
  scripts
  views
    DevelopmentDefaultError.gt (zero.core.webtools)
    GroovySourceCodeViewer.gt (zero.core.webtools)
    debug.gt (zero.core.webtools)
    debugfiles.gt (zero.core.webtools)
    footerBar.gt (zero.core.webtools)
    head.gt (zero.core.webtools)
    headerBar.gt (zero.core.webtools)
    listing.gt (zero.core)
public
  dijit
  dojo
  dojox
  setup
  util
  zero
  index.gt (zero.core.webtools)
```

IBM

Configuration

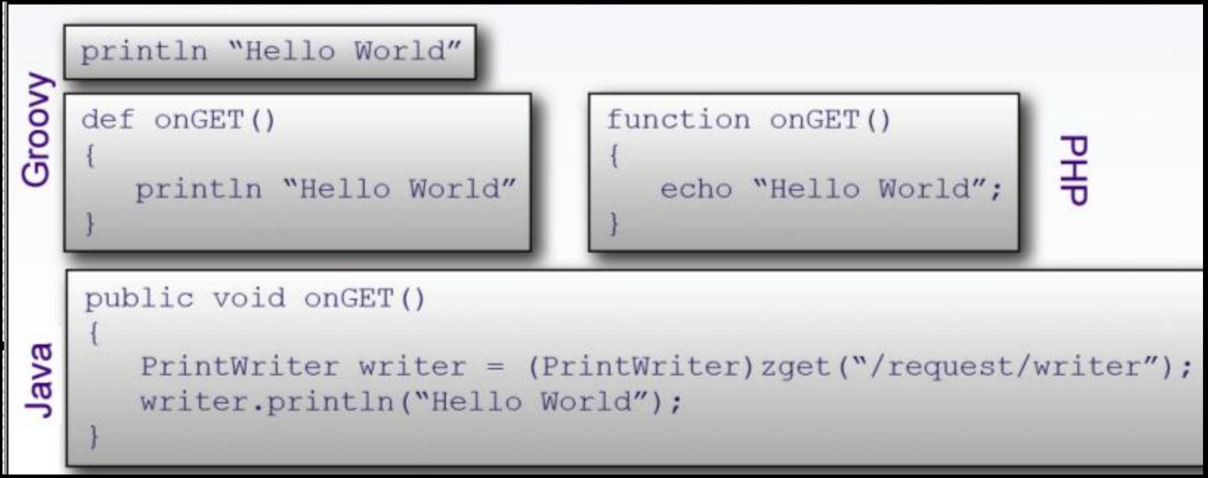
- **Zero configuration file: zero.config**
  - The config/zero.config file is processed at the start of a Zero application.
  - The content of a config/zero.config file is organized into "stanzas" of related key/value pairs. Stanzas are associated with directives, such as "store to the Global Context" and "include another configuration file.”



```
# Value set
/config/http/port = 8080

# List set
/config/resources/defaultExtensions = [".groovy"]

# List append
/config/bindings/.groovy +=
["zero.core.groovysupport.bindings.DefaultBindings"]

# Map set
/config/test/map = { "a" : "b", "c" : "d" }

# Map append
/config/test/mapappend += { "a" : "b", "c" : "d" }
/config/test/mapappend += { "x" : "y", "w" : "z" }

# Event handler
/config/handlers += {
  "events" : "GET",
  "handler" : "custom.Handler.class"
}

# Value reference (insert value read at config-load time)
/config/property/myPrefix = "/foo/bar"
/config/test/value = "${/config/property/myPrefix}/bat"

# Variable set/value reference
myPrefix = "/foo/bar"
/config/test/value = "${myPrefix}/bat"

# Include
@include "${/config/dependencies/zero.core}/config/security/
form.config"
{ "formLoginPage" : "/login" }
```

Rendering

- Rendering
  - Direct
  - Indirect

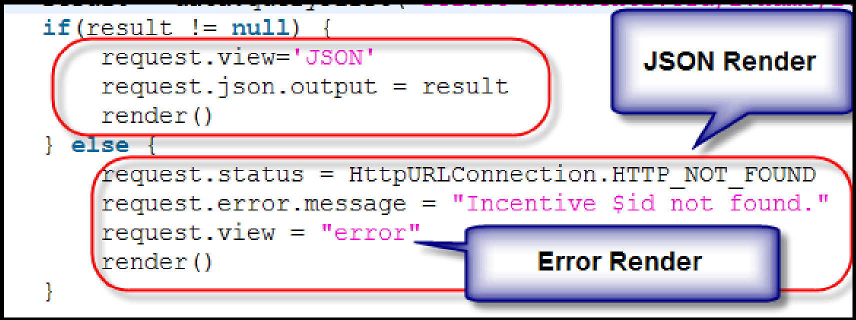


```
println "Hello World"
```

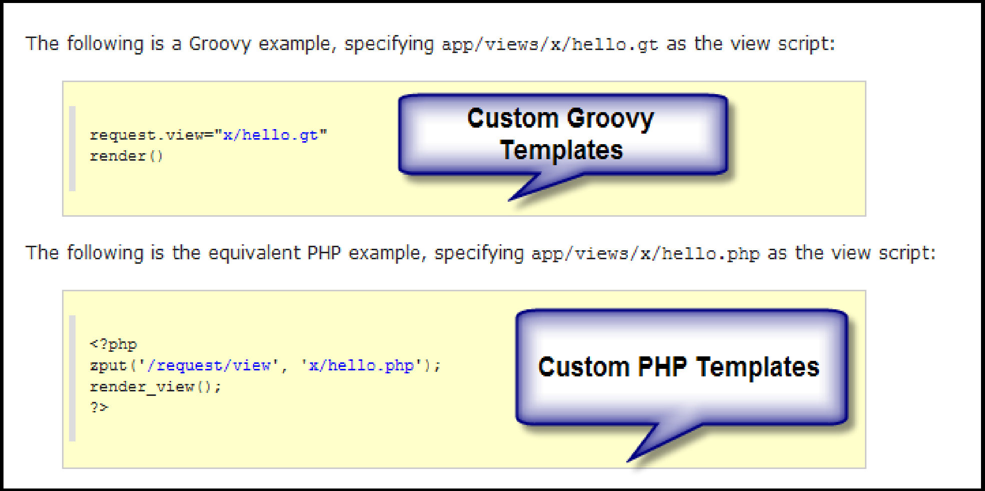
```
def onGET()
{
   println "Hello World"
}
```

```
function onGET()
{
   echo "Hello World";
}
```

```
public void onGET()
{
   PrintWriter writer = (PrintWriter)zget("/request/writer");
   writer.println("Hello World");
}
```



```
if(result != null) {
    request.view='JSON'
    request.json.output = result
    render()
} else {
    request.status = HttpURLConnection.HTTP_NOT_FOUND
    request.error.message = "Incentive $id not found."
    request.view = "error"
    render()
}
```



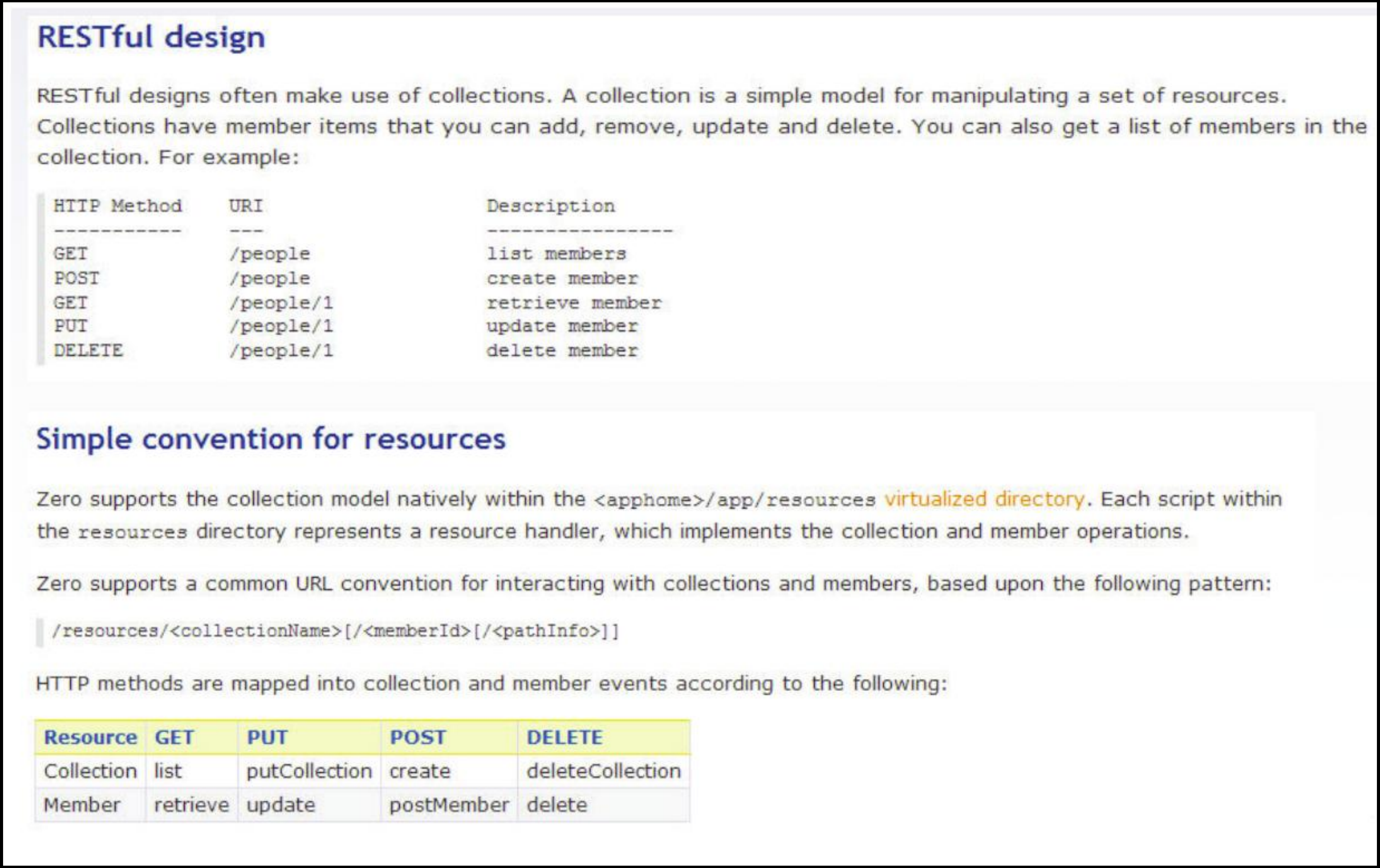
The following is a Groovy example, specifying app/views/x/hello.gt as the view script:

```
request.view="x/hello.gt"
render()
```

The following is the equivalent PHP example, specifying app/views/x/hello.php as the view script:

```
<?php
zput('/request/view', 'x/hello.php');
render_view();
?>
```

RESTful Resources



RESTful design

RESTful designs often make use of collections. A collection is a simple model for manipulating a set of resources. Collections have member items that you can add, remove, update and delete. You can also get a list of members in the collection. For example:

```
HTTP Method    URI                   Description
-----------    ---                   ----------------
GET            /people               list members
POST           /people               create member
GET            /people/1             retrieve member
PUT            /people/1             update member
DELETE         /people/1             delete member
```

Simple convention for resources

Zero supports the collection model natively within the `<apphome>/app/resources` virtualized directory. Each script within the `resources` directory represents a resource handler, which implements the collection and member operations.

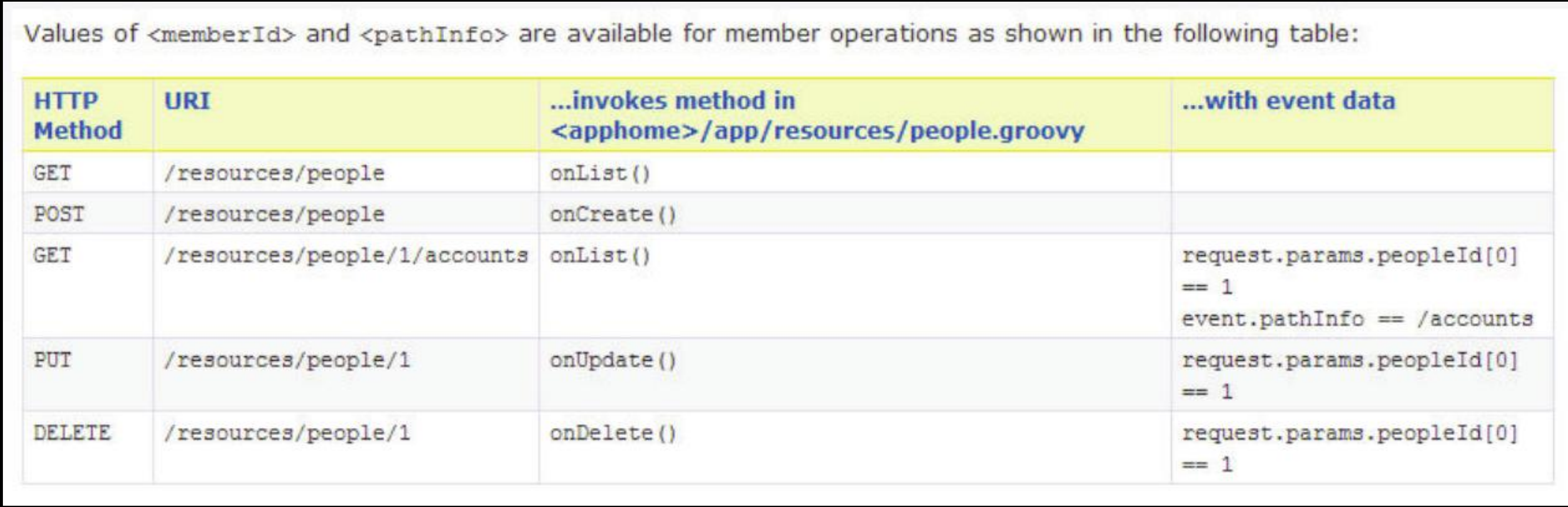
Zero supports a common URL convention for interacting with collections and members, based upon the following pattern:

```
/resources/<collectionName>[/<memberId>[/<pathInfo>]]
```

HTTP methods are mapped into collection and member events according to the following:

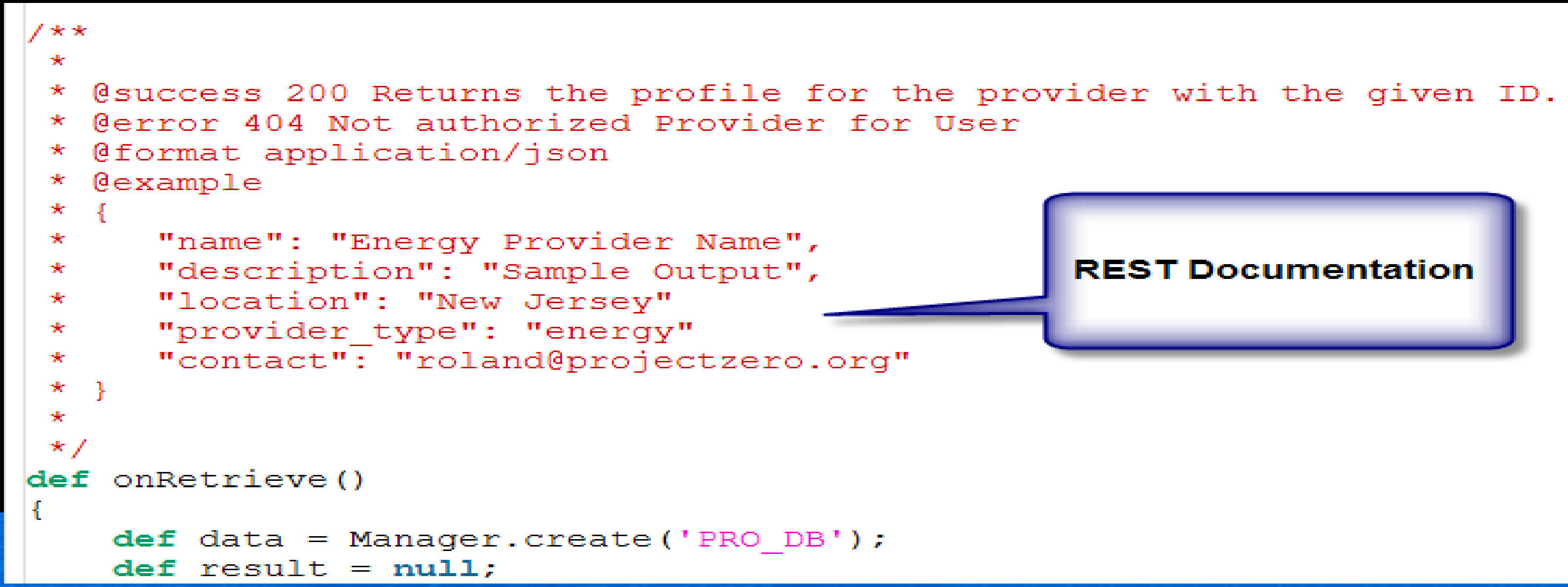
Resource	GET	PUT	POST	DELETE
Collection	list	putCollection	create	deleteCollection
Member	retrieve	update	postMember	delete

RESTful Resources



Values of <memberId> and <pathInfo> are available for member operations as shown in the following table:

HTTP Method	URI	...invokes method in <apphome>/app/resources/people.groovy	...with event data
GET	/resources/people	onList()	
POST	/resources/people	onCreate()	
GET	/resources/people/1/accounts	onList()	request.params.peopleId[0] == 1 event.pathInfo == /accounts
PUT	/resources/people/1	onUpdate()	request.params.peopleId[0] == 1
DELETE	/resources/people/1	onDelete()	request.params.peopleId[0] == 1



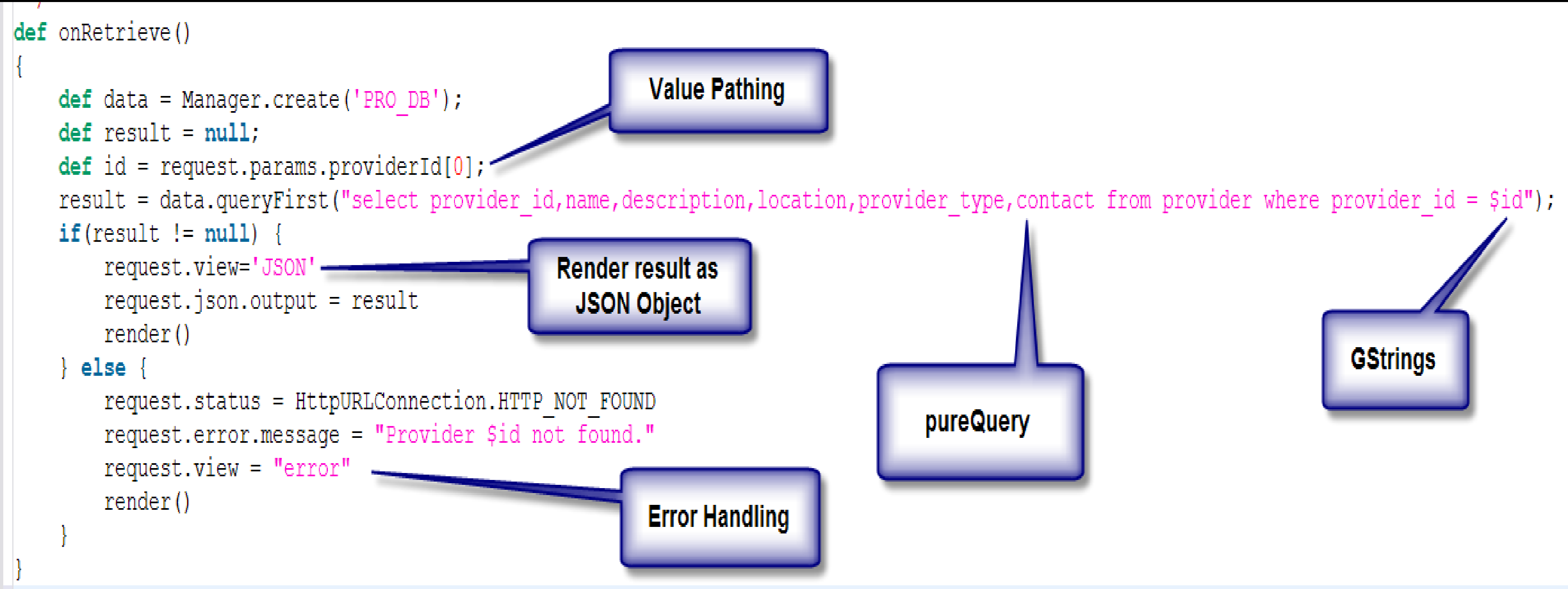
```
/**
 *
 * @success 200 Returns the profile for the provider with the given ID.
 * @error 404 Not authorized Provider for User
 * @format application/json
 * @example
 * {
 *     "name": "Energy Provider Name",
 *     "description": "Sample Output",
 *     "location": "New Jersey"
 *     "provider_type": "energy"
 *     "contact": "roland@projectzero.org"
 * }
 *
 */
def onRetrieve()
{
    def data = Manager.create('PRO_DB');
    def result = null;
```

REST Documentation

IBM

Zero.data (powered by pureQuery)

- Full power of SQL
- Provides inline SQL directly from scripts
- Designed to allow straight SQL while simplifying common query patterns
  - Result set access, output and formatting
  - Parameter passing
  - Paging, etc

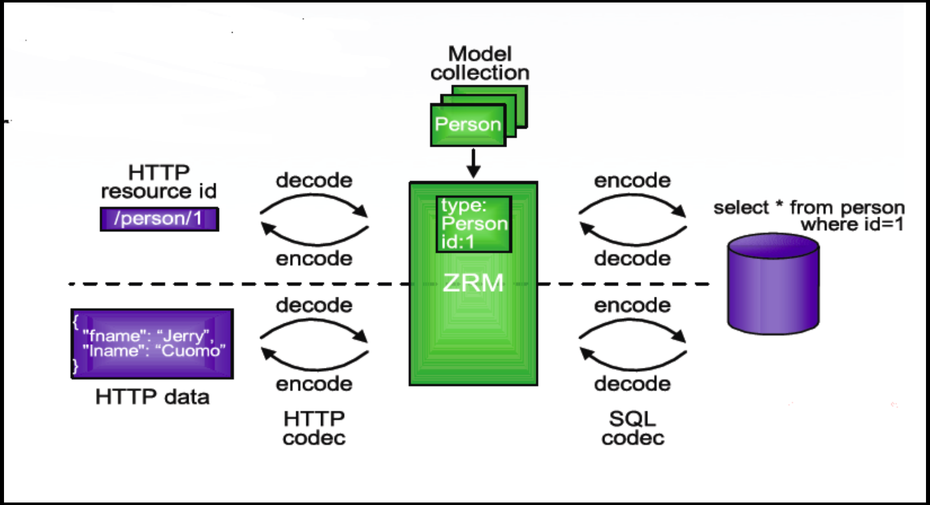


```
def onRetrieve()
{
    def data = Manager.create('PRO_DB');
    def result = null;
    def id = request.params.providerId[0];
    result = data.queryFirst("select provider_id,name,description,location,provider_type,contact from provider where provider_id = $id");
    if(result != null) {
        request.view='JSON'
        request.json.output = result
        render()
    } else {
        request.status = HttpURLConnection.HTTP_NOT_FOUND
        request.error.message = "Provider $id not found."
        request.view = "error"
        render()
    }
}
```

Exposing Relational Data RESTfully

Zero Resource Manager

- Zero Resource Manager (ZRM) extension provides a REST oriented view to data
- A layer of abstraction between the actual data store and REST
  - A constrained set of APIs that encourage RESTful application architecture
  - A data model that maps well to REST and feeds
    - Default top-down data mapping scheme & automatic table creation
- First class support for accessing data in a RESTful fashion both programmatically and via HTTP
  - Resources defined in the data model can be accessed via HTTP with no manual coding
- First class support for accessing data as feeds
- Robust frameworks for
  - Persistence, validation, and serialization

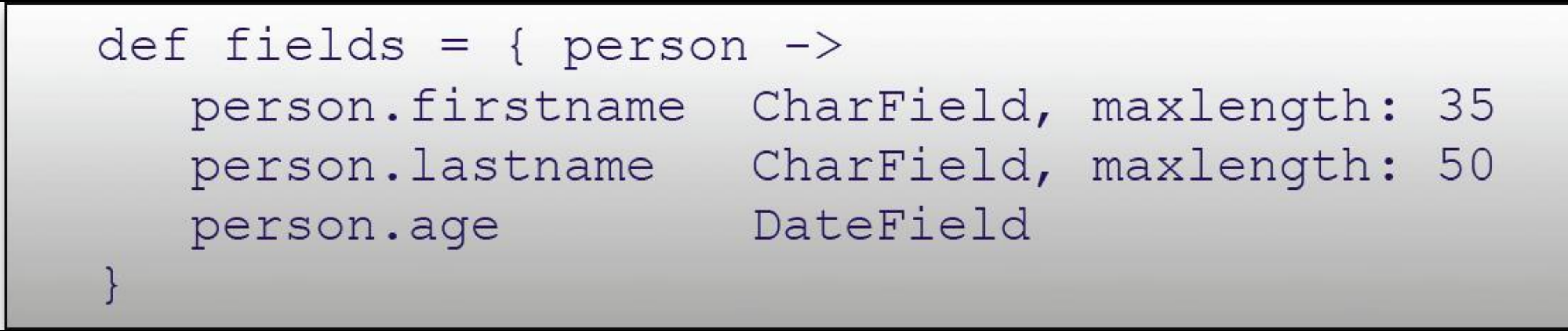


Not an ORM

- Significantly constrained model from ORMs
  - A data model that is geared towards REST rather than an arbitrary POJO model
    - No byte code weaving etc. object management magic needed
  - A default top down mapping scheme rather than mapping an arbitrary db schema to an arbitrary POJO model
    - No mapping tools or annotations required
  - No state maintained per user rather than a stateful POJO model
    - No complex persistence engine needed to keep track of dirtiness

Resource Model

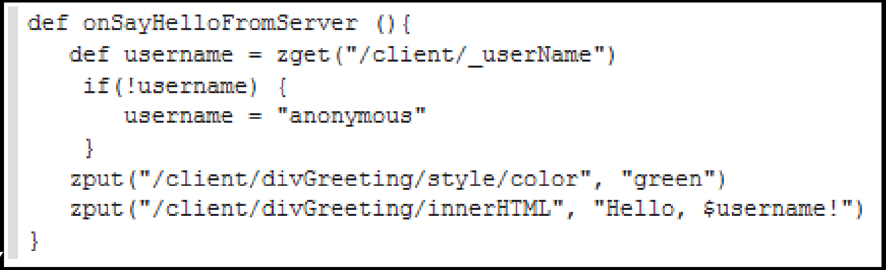
- The resource model is a collection of resource type definitions
  - A resource type definition contains the definitions for fields, constraints,relationships, and collections
- The resource type definitions
  - Are datastore independent
  - Default top-down mapping scheme for RDBs
  - Can be shared across applications
  - Can be created and manipulated either declaratively / programmatically or via
- HTTP
  - Defined in a groovy script in **/app/models**



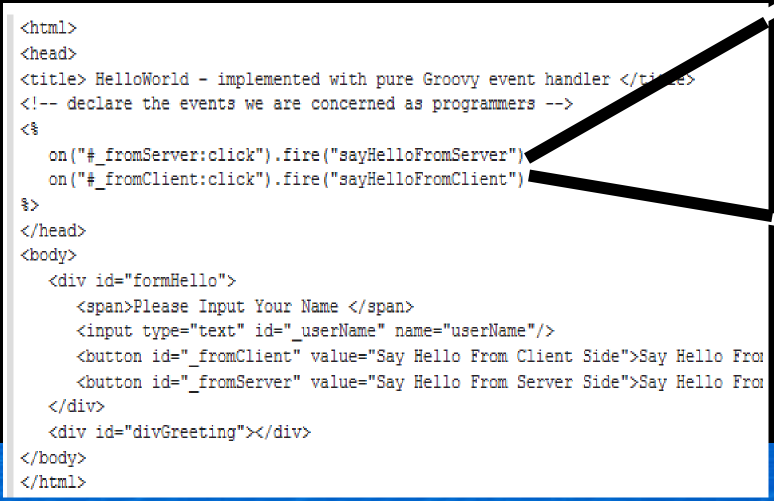
```
def fields = { person ->
    person.firstname  CharField, maxlength: 35
    person.lastname   CharField, maxlength: 50
    person.age        DateField
}
```

Rich Internet Applications

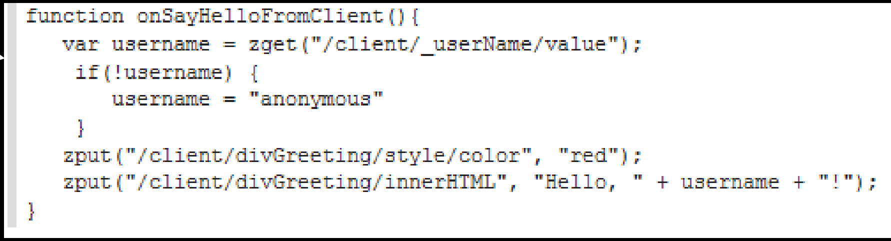
- Solidify model for simplifying the UI side of a web application
  - Service model, Dojo, Reactive Client, Global Context and Events extended to the browser
- GC/Events on the Browser
  - Allows you to map a physical UI event to a logical application event
  - Can handle the application event on either the server or in the browser
  - The DOM of the browser is exposed via the GC to the application on either side



```
def onSayHelloFromServer (){
   def username = zget("/client/_userName")
    if(!username) {
       username = "anonymous"
    }
   zput("/client/divGreeting/style/color", "green")
   zput("/client/divGreeting/innerHTML", "Hello, $username!")
}
```



```
<html>
<head>
<title> HelloWorld - implemented with pure Groovy event handler </title>
<!-- declare the events we are concerned as programmers -->
<%
   on("#_fromServer:click").fire("sayHelloFromServer")
   on("#_fromClient:click").fire("sayHelloFromClient")
%>
</head>
<body>
   <div id="formHello">
      <span>Please Input Your Name </span>
      <input type="text" id="_userName" name="userName"/>
      <button id="_fromClient" value="Say Hello From Client Side">Say Hello Fro
      <button id="_fromServer" value="Say Hello From Server Side">Say Hello Fro
   </div>
   <div id="divGreeting"></div>
</body>
</html>
```

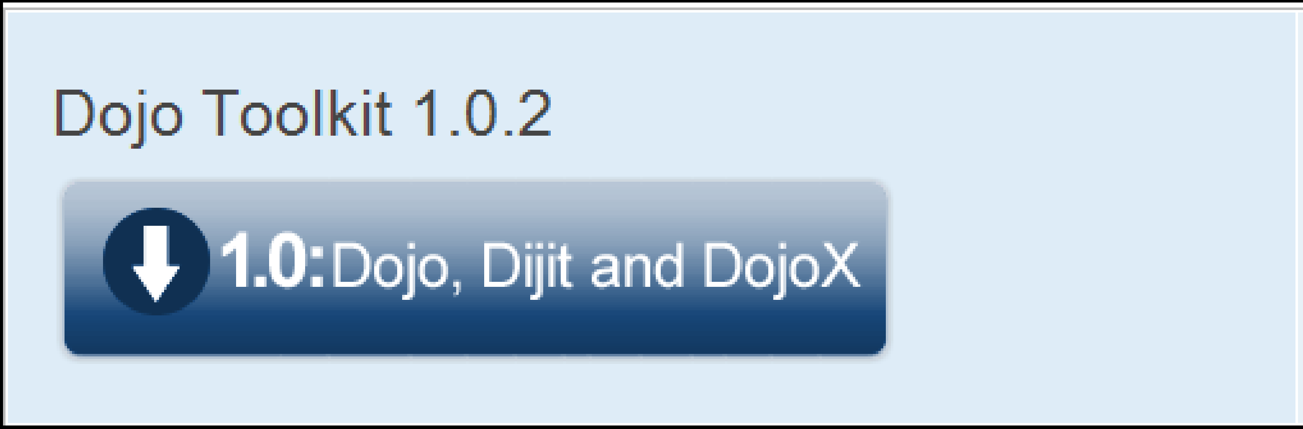


```
function onSayHelloFromClient(){
   var username = zget("/client/_userName/value");
    if(!username) {
       username = "anonymous"
    }
   zput("/client/divGreeting/style/color", "red");
   zput("/client/divGreeting/innerHTML", "Hello, " + username + "!");
}
```

Dojo Broswer Toolkit

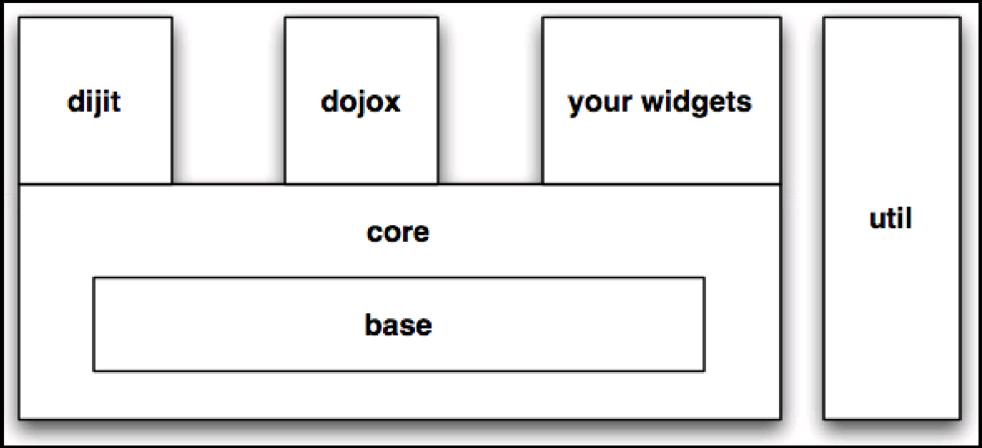
- Dojo is an Open Source DHTML toolkit written in JavaScript. It builds on several contributed code bases.

  - Provides Rich Set of Widgets
  - Web UI Framework
  - Rich Event handling System
  - General Purpose HTML Libraries
  - Several other utilities
  - Math, XML to JS parsing, etc…

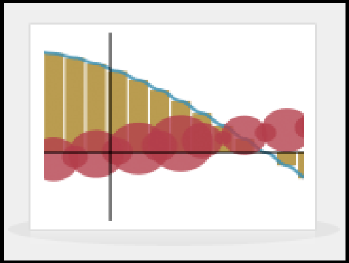


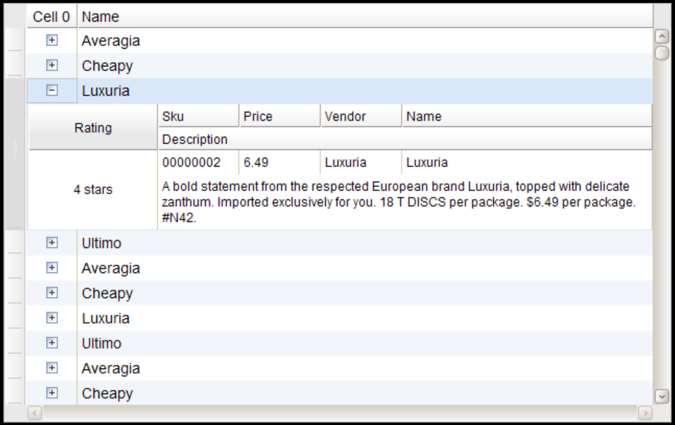
Dojo Architecture

- Base
  - The kernel of the toolkit wrapped into a 25k js file (dojo.js). Base bootstraps the toolkit, includes AJAX utilities, class based inheritance, packaging system and more
- Core
  - Provides addition facilities on top of the base for accessing data stores, effects such as wipes/slides, internationalization (i18n) and back-button handling among other things. Separate package keeps base small
- Dijit
  - Shorthand for "Dojo widget". Could refer to a single Dojo widget (a dijit) or to the entire component of the toolkit containing all of Dojo's widgets (Dijit)
- DojoX
  - “Dojo Experimental" and contains features that stand a chance of one day migrating into Core, Dijit or even a new module. A great proving ground for new features while maintaining standards of core and base.
- Util
  - A collection of Dojo utilities (more later)



Sample Dojo Widgets

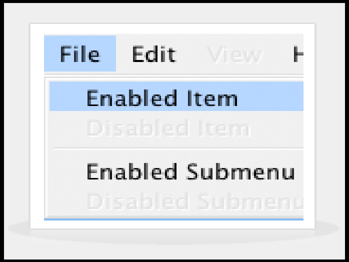


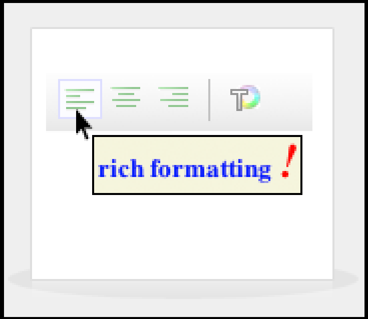


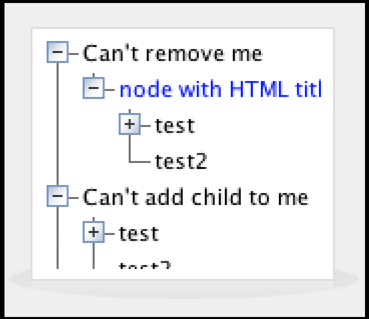
Cell 0	Name
⊞	Averagia
⊞	Cheapy
⊟	Luxuria

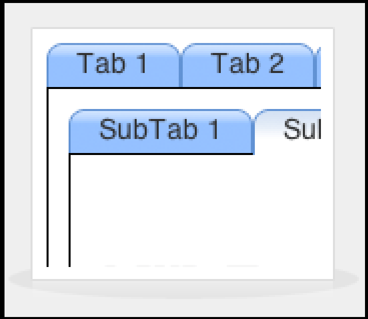
Rating	Sku	Price	Vendor	Name
	Description			
4 stars	00000002	6.49	Luxuria	Luxuria
	A bold statement from the respected European brand Luxuria, topped with delicate zanthum. Imported exclusively for you. 18 T DISCS per package. $6.49 per package. #N42.			

⊞	Ultimo
⊞	Averagia
⊞	Cheapy
⊞	Luxuria
⊞	Ultimo
⊞	Averagia
⊞	Cheapy



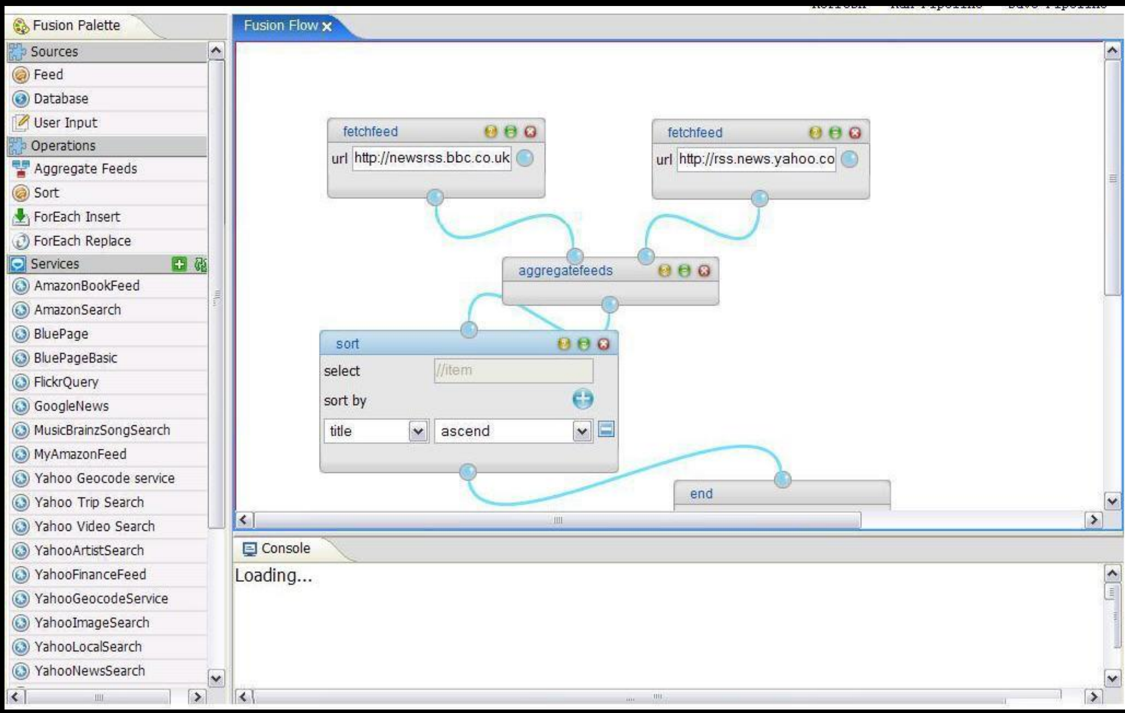






Assemble

- Composition of applications by "wiring" REST services using the SPLICE flows. Incorporating both activity and data flows.
- A solution may be rapidly assembled by combining existing feeds and services that enrich, sort, and filter data in a pipeline. Either visually or programmatically.
- Configure templates to alter pipeline routes, log events along the pipeline, as well as transform data
- Adapters to enhance integration with existing systems.



url http://newsrss.bbc.co.uk

url http://rss.news.yahoo.co

select //item

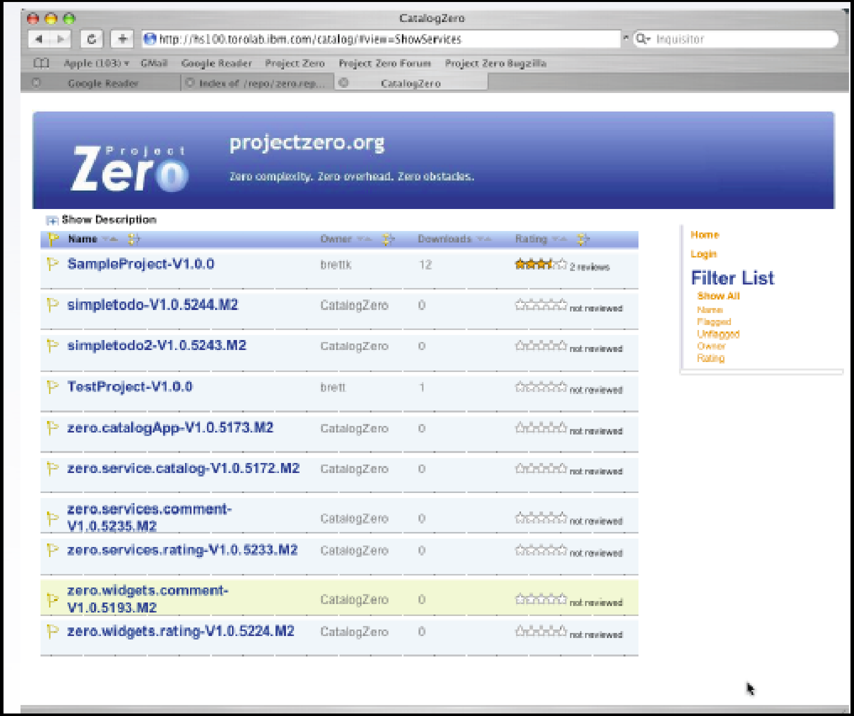
sort by

title ascend

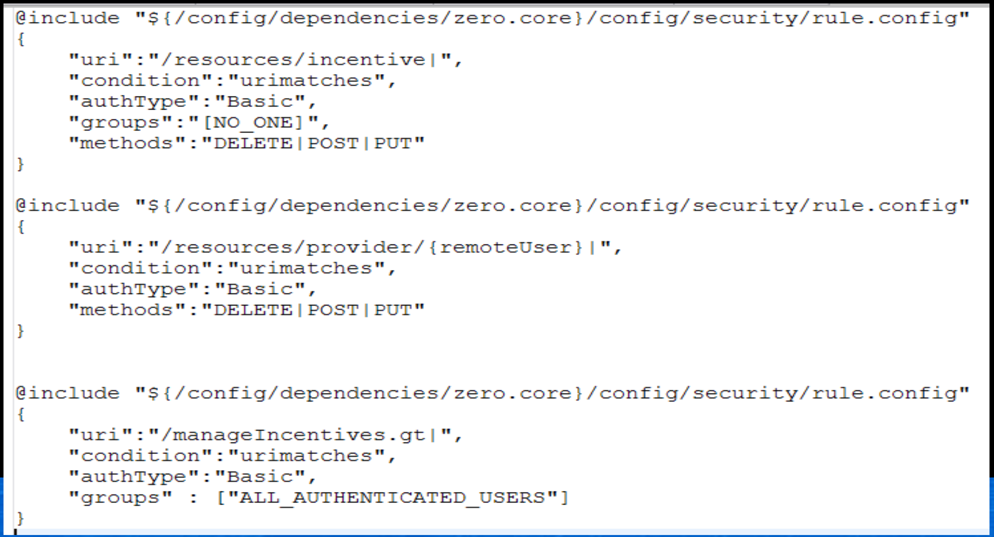
Loading...

Catalog of Content

- Data Formats
  - JSON, XML, ATOM (including APP), RSS
- Flow Engine
  - Feed and Control flows
- Accessing backend resources
  - Data Zero, HTTP, Mail
- Presentation Components
  - Dojo, Reactive Client, Web Flow, Templates
- Services and Widgets
  - tagging, comments, ratings, blogs, profile, news, reviews, ratings, polls, wiki pages, Amazon (ECS, EC2, S3), Google, EBay
- Security
  - Active Content Filtering, OpenID, Authentication, Authorization, LTPA, SSO, SSL
- Misc
  - Apache mod_proxy config
  - Zero Catalog Application

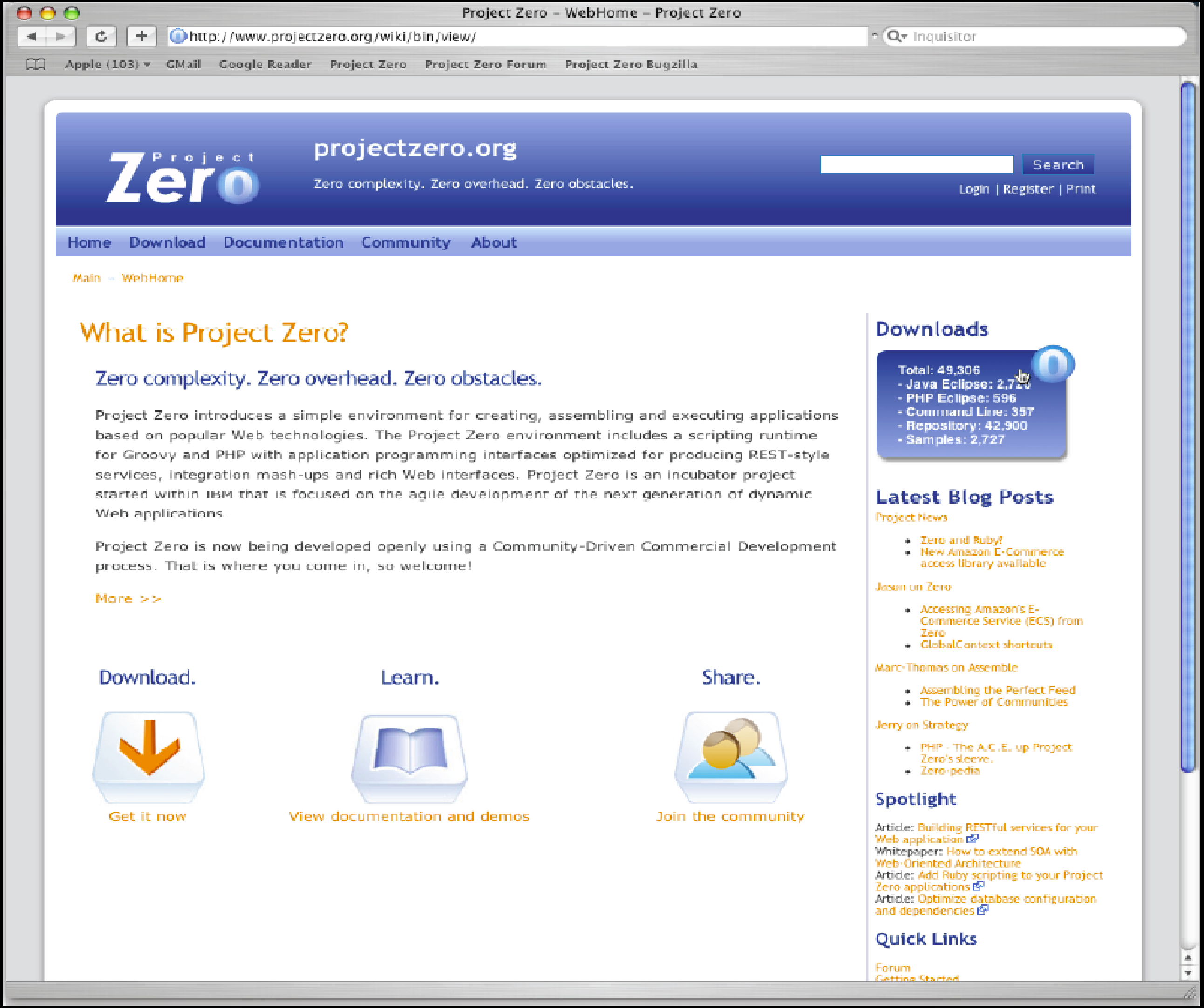

CatalogZero
http://hs100.torolab.ibm.com/catalog/#view=ShowServices
Inquisitor
Apple (103) GMail Google Reader Project Zero Project Zero Forum Project Zero Bugzilla
Google Reader
Index of /repo/zero.rep...
CatalogZero
Project Zero
projectzero.org
Zero complexity. Zero overhead. Zero obstacles.
Show Description
Home
Login
Filter List
Show All
Name
Flagged
Unflagged
Owner
Rating

Name	Owner	Downloads	Rating
SampleProject-V1.0.0	brettk	12	★★★★☆ 2 reviews
simpletodo-V1.0.5244.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
simpletodo2-V1.0.5243.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
TestProject-V1.0.0	brett	1	☆☆☆☆☆ not reviewed
zero.catalogApp-V1.0.5173.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
zero.service.catalog-V1.0.5172.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
zero.services.comment-V1.0.5235.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
zero.services.rating-V1.0.5233.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
zero.widgets.comment-V1.0.5193.M2	CatalogZero	0	☆☆☆☆☆ not reviewed
zero.widgets.rating-V1.0.5224.M2	CatalogZero	0	☆☆☆☆☆ not reviewed



```
@include "${/config/dependencies/zero.core}/config/security/rule.config"
{
    "uri":"/resources/incentive|",
    "condition":"urimatches",
    "authType":"Basic",
    "groups":"[NO_ONE]",
    "methods":"DELETE|POST|PUT"
}

@include "${/config/dependencies/zero.core}/config/security/rule.config"
{
    "uri":"/resources/provider/{remoteUser}|",
    "condition":"urimatches",
    "authType":"Basic",
    "methods":"DELETE|POST|PUT"
}


@include "${/config/dependencies/zero.core}/config/security/rule.config"
{
    "uri":"/manageIncentives.gt|",
    "condition":"urimatches",
    "authType":"Basic",
    "groups" : ["ALL_AUTHENTICATED_USERS"]
}
```

Community: http://www.projectzero.org


Project Zero – WebHome – Project Zero
http://www.projectzero.org/wiki/bin/view/
Inquisitor
Apple (103) GMail Google Reader Project Zero Project Zero Forum Project Zero Bugzilla
Project Zero
projectzero.org
Zero complexity. Zero overhead. Zero obstacles.
Search
Login | Register | Print
Home Download Documentation Community About
Main → WebHome

What is Project Zero?

Zero complexity. Zero overhead. Zero obstacles.

Project Zero introduces a simple environment for creating, assembling and executing applications based on popular Web technologies. The Project Zero environment includes a scripting runtime for Groovy and PHP with application programming interfaces optimized for producing REST-style services, integration mash-ups and rich Web interfaces. Project Zero is an incubator project started within IBM that is focused on the agile development of the next generation of dynamic Web applications.

Project Zero is now being developed openly using a Community-Driven Commercial Development process. That is where you come in, so welcome!

More >>

Download. Learn. Share.

Get it now | View documentation and demos | Join the community

Downloads

Total: 49,306
- Java Eclipse: 2,720
- PHP Eclipse: 596
- Command Line: 357
- Repository: 42,900
- Samples: 2,727

Latest Blog Posts

Project News
- Zero and Ruby?
- New Amazon E-Commerce access library available

Jason on Zero
- Accessing Amazon's E-Commerce Service (ECS) from Zero
- GlobalContext shortcuts

Marc-Thomas on Assemble
- Assembling the Perfect Feed
- The Power of Communities

Jerry on Strategy
- PHP - The A.C.E. up Project Zero's sleeve.
- Zero-pedia

Spotlight

Article: Building RESTful services for your Web application
Whitepaper: How to extend SOA with Web-Oriented Architecture
Article: Add Ruby scripting to your Project Zero applications
Article: Optimize database configuration and dependencies

Quick Links

Forum
Getting Started

Zero Community

- Experiment in a more transparent form of commercial development
- Community Driven Commercial Development
  - *Community-Driven* means that we want feedback, insight, suggestions, criticism, and dialogue with the users of Project Zero
  - *Commercial* means that this is not an open source project. The licensing is very liberal, but not completely open
  - *Development* means that this community is about the technology and how it is developed and evolves.

What is Available

- Wiki containing documentation, project information, roadmaps, design documents, demos, samples, iCal calendar, future plans, etc
- Forum for interactive discussion
  - Help and Feedback for questions from users
  - Developer Alerts to notify users of new features and breaking changes
  - Zero Development for publically accessible discussion amongst the Zero development team
- Blogs for alerts and personal commentary
  - Product blog
  - Personal blogs of the project leaders
- Downloads
  - Eclipse Tools, Command Line tools and Zero components
  - Bug Tracking System (Bugzilla)
  - Source code (Subversion)
- An online catalog of Zero extensions
  - Soon to be open for user contributions
  - Zero Alive! - online Zero playground

Other Resources

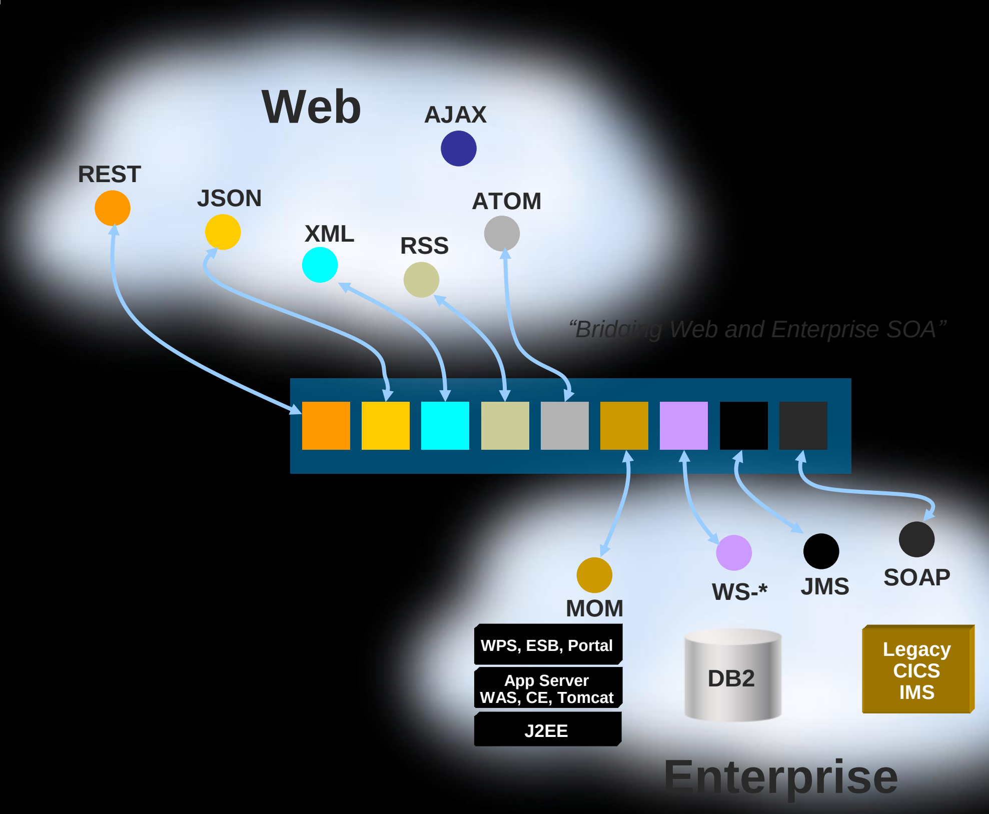
- Zero DeveloperWorks Space
- Web 2.0 and Middleware Blog
- Developer Works Articles

Global SOA

- Exposing Enterprise Services to the Web extends your enterprise globally and simplifies:
  - Development
  - Composition of Services
  - Deployment and accessibility
  - QoS: Performance, Scale, Security

Enterprises are ***exposing*** more services and feeds to the Web

…and ***consuming*** more services and feeds from the Web

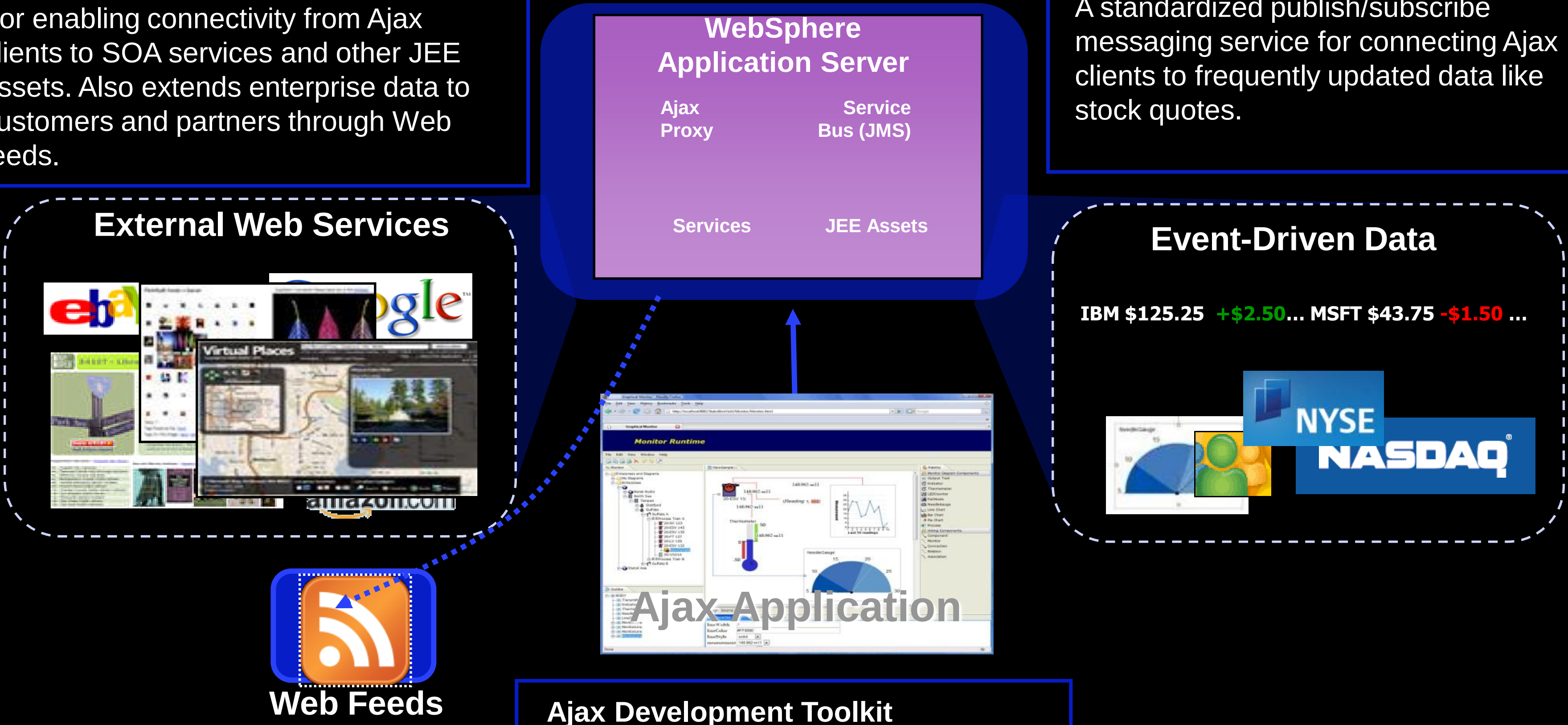


Extending SOA to rich Web 2.0 applications

WAS Web 2.0 Feature Pack

Web 2.0 to SOA Connectivity
For enabling connectivity from Ajax clients to SOA services and other JEE assets. Also extends enterprise data to customers and partners through Web feeds.

AJAX Messaging
A standardized publish/subscribe messaging service for connecting Ajax clients to frequently updated data like stock quotes.



Ajax Development Toolkit
A best-in-class Ajax development toolkit for WebSphere Application Server based on Dojo, an Open Source JavaScript toolkit, with IBM extensions.

Unlocking enterprise content simply

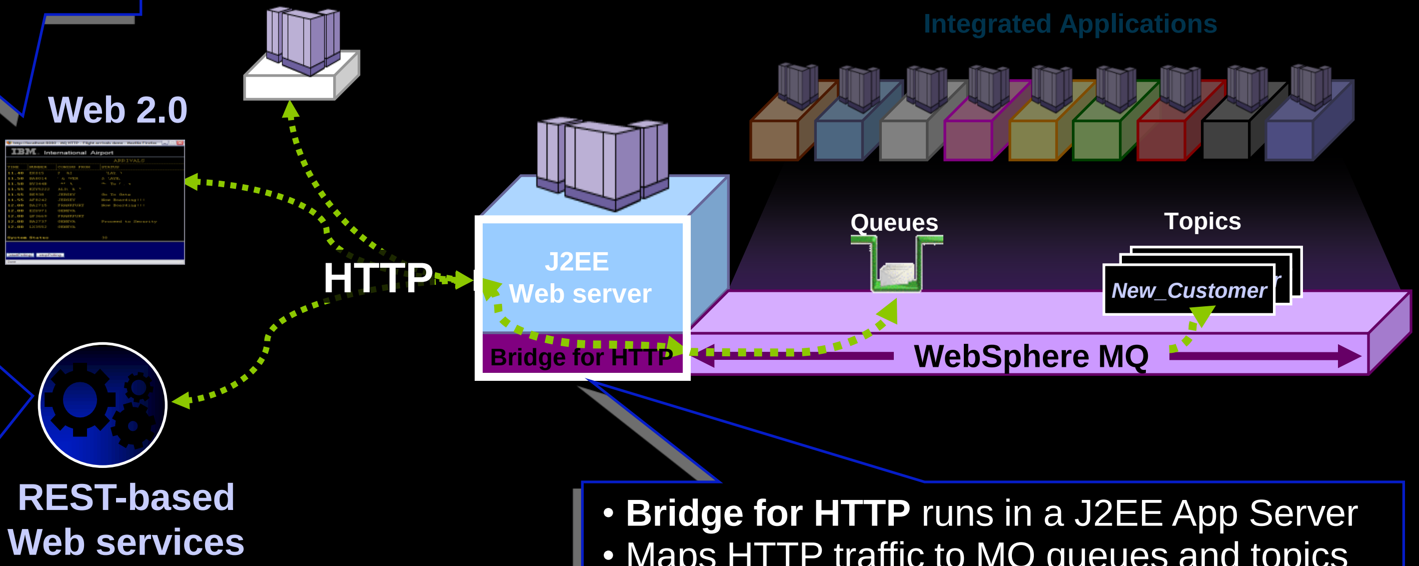
WebSphere MQ HTTP Bridge

Dynamic Web applications

- Speeds and eases integration of new Web apps with enterprise applications and data
- No MQ skills needed

No MQ client footprint

- Simplifies deployment and maintenance of large scale distributed applications

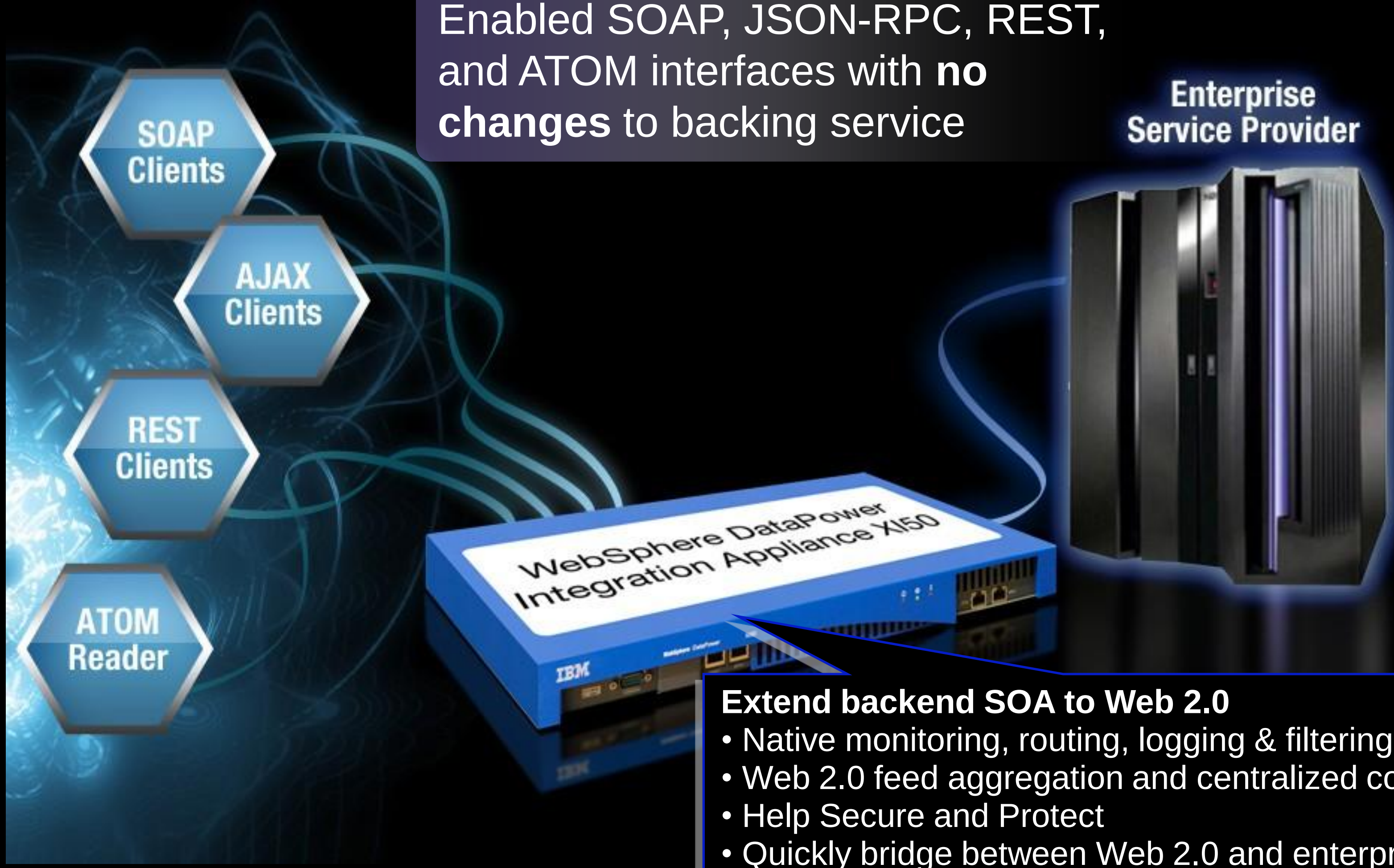


- Easy access to enterprise applications and ESB
- Reliable delivery of SOAP across MQ backbone
- Pub/Sub distribution

- **Bridge for HTTP** runs in a J2EE App Server
- Maps HTTP traffic to MQ queues and topics

Extend SOA simply and securely

WebSphere Datapower SOA Appliance


SOAP Clients
AJAX Clients
REST Clients
ATOM Reader
WebSphere DataPower Integration Appliance XI50
Enterprise Service Provider

Enabled SOAP, JSON-RPC, REST, and ATOM interfaces with **no changes** to backing service

Extend backend SOA to Web 2.0
- Native monitoring, routing, logging & filtering
- Web 2.0 feed aggregation and centralized control
- Help Secure and Protect
- Quickly bridge between Web 2.0 and enterprise SOA

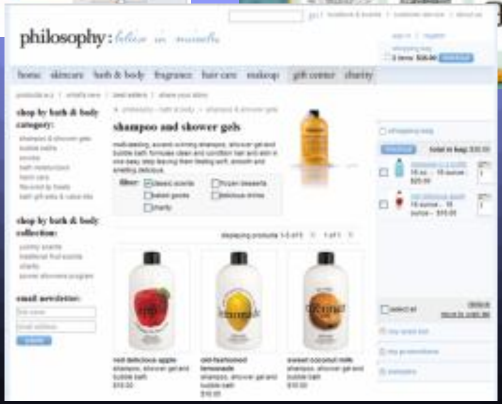
Simply transforming the Shopping Experience

WebSphere Commerce

Dynamic customer driven experience

Interactive AJAX interfaces

Streamline purchasing process



Lotus - Community

Lotus Mashup Maker

Business Users can create Mashups

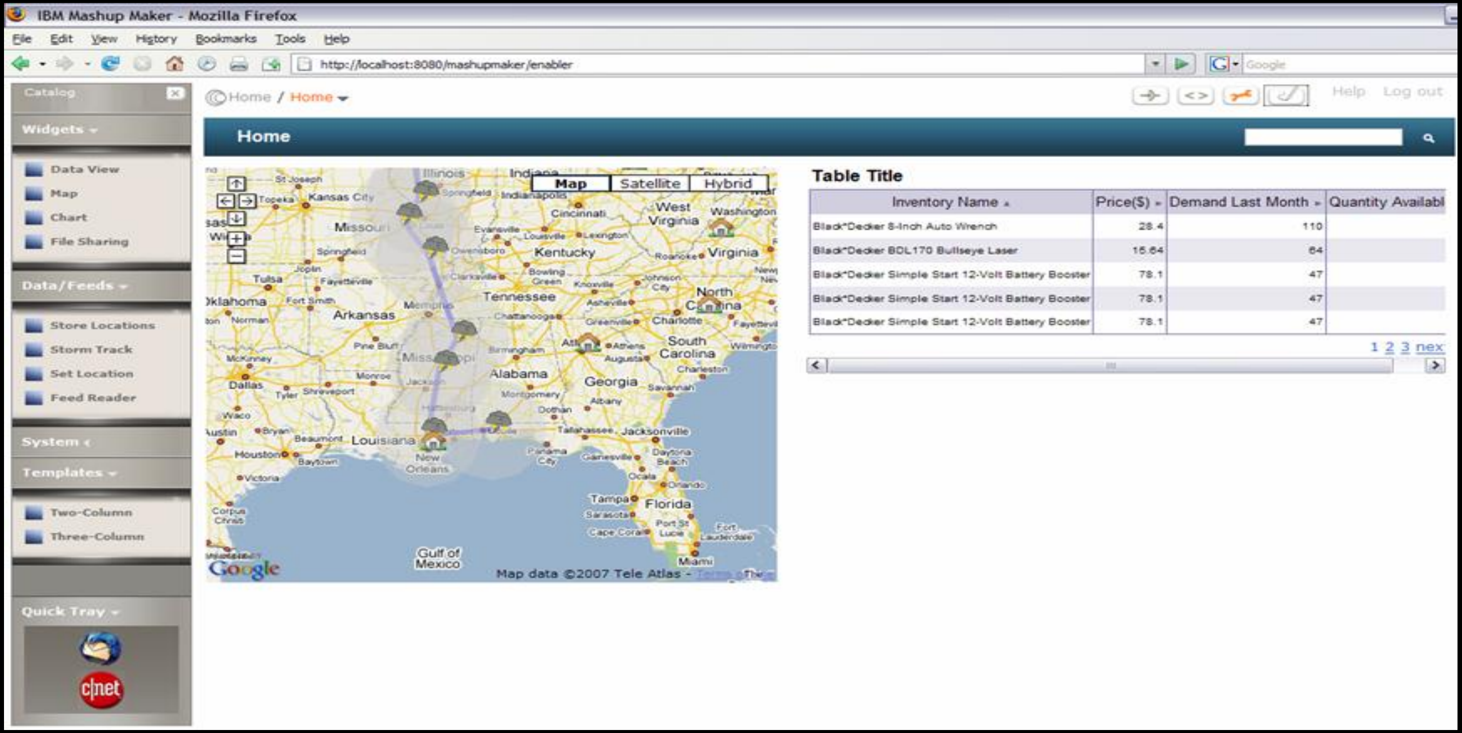


Table Title

Inventory Name	Price($)	Demand Last Month	Quantity Availabl
Black*Decker 8-Inch Auto Wrench	28.4	110	
Black*Decker BDL170 Bullseye Laser	16.64	64	
Black*Decker Simple Start 12-Volt Battery Booster	78.1	47	
Black*Decker Simple Start 12-Volt Battery Booster	78.1	47	
Black*Decker Simple Start 12-Volt Battery Booster	78.1	47	

1 2 3 nex

AJAX based Client Side Aggregation in the Web Browser

REST-accessible Markup Fragments from WP Portlets or any other URL

Atom / RSS Feeds

Services created with Google Gadgets

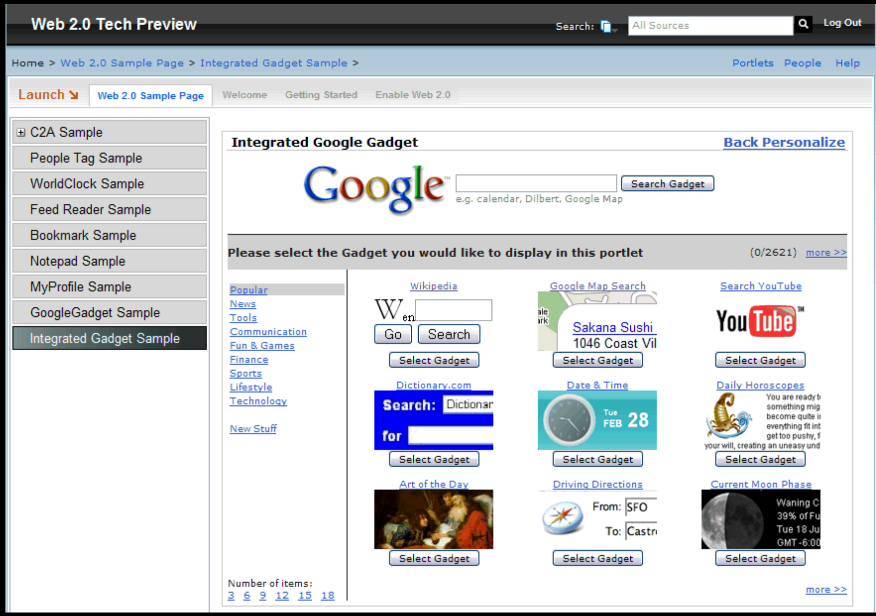
WSRP Services

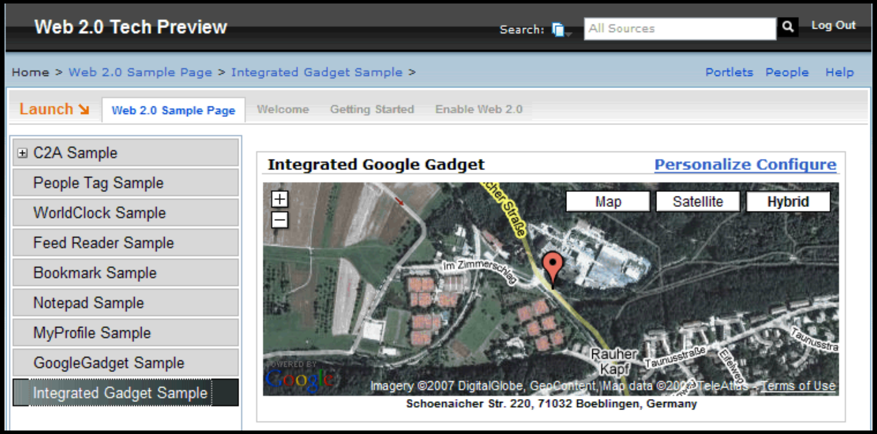
Google Gadget Integration

- Enable customers to easily integrate Google Gadgets into portal pages

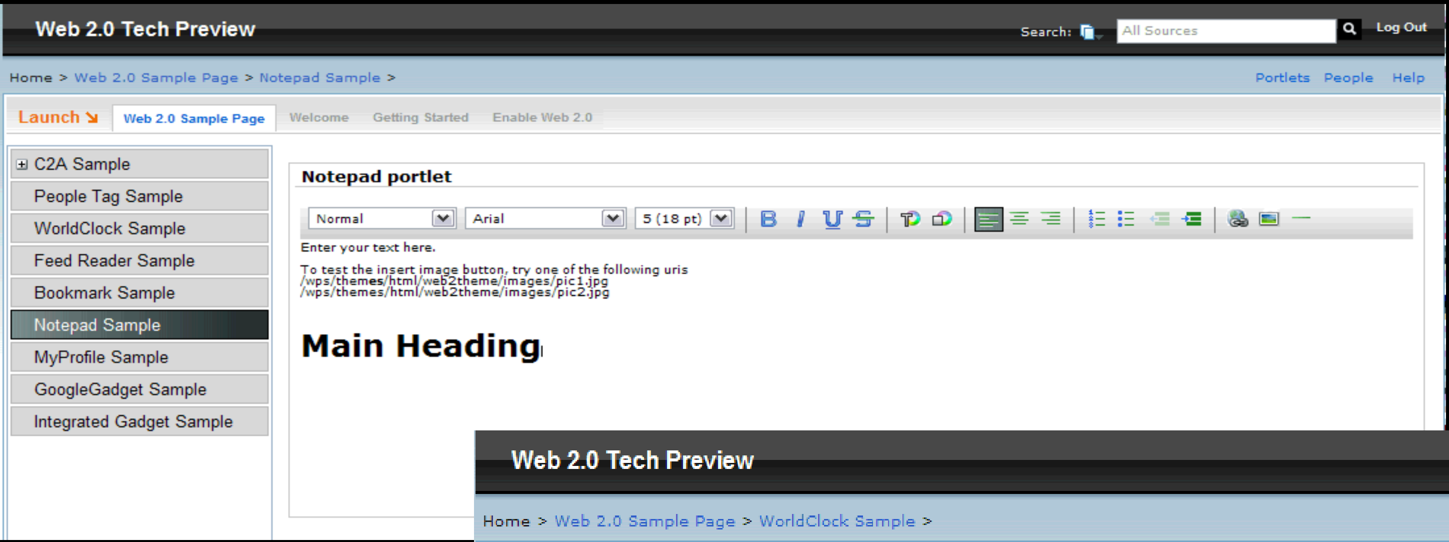
→From an end user perspective, Google Gadgets integrated in WebSphere Portal behave just like local portlets: viewable and customizable like any local portlet

- If allowed by admin, users can drag Generic Gadget Portlets on their pages and select Gadgets to display from the Gadget Catalog
  - Gadget Portlet initially lets user select the Gadget to display from the Gadget Catalog
  - Gadget Portlet then displays the selected Gadget
  - User can view and customize the selected gadget like any local portlet
- Administrators can pre-define Gadget Portlets for the portlet palette
  - Generic Gadget Portlet is pre-configured by the admin to connect it to a certain gadget, e.g. an admin could create a “Map Portlet” by creating a Gadget Portlet and connecting it to the Google Maps Gadget
  - Users can then select such pre-configured Gadget Portlets from the palette and drag them onto their pages like any local portlet





AJAX Portlets using Dojo



Web 2.0 Tech Preview — Search: All Sources — Log Out

Home > Web 2.0 Sample Page > Notepad Sample > — Portlets People Help

Launch — Web 2.0 Sample Page — Welcome — Getting Started — Enable Web 2.0

- C2A Sample
- People Tag Sample
- WorldClock Sample
- Feed Reader Sample
- Bookmark Sample
- Notepad Sample
- MyProfile Sample
- GoogleGadget Sample
- Integrated Gadget Sample

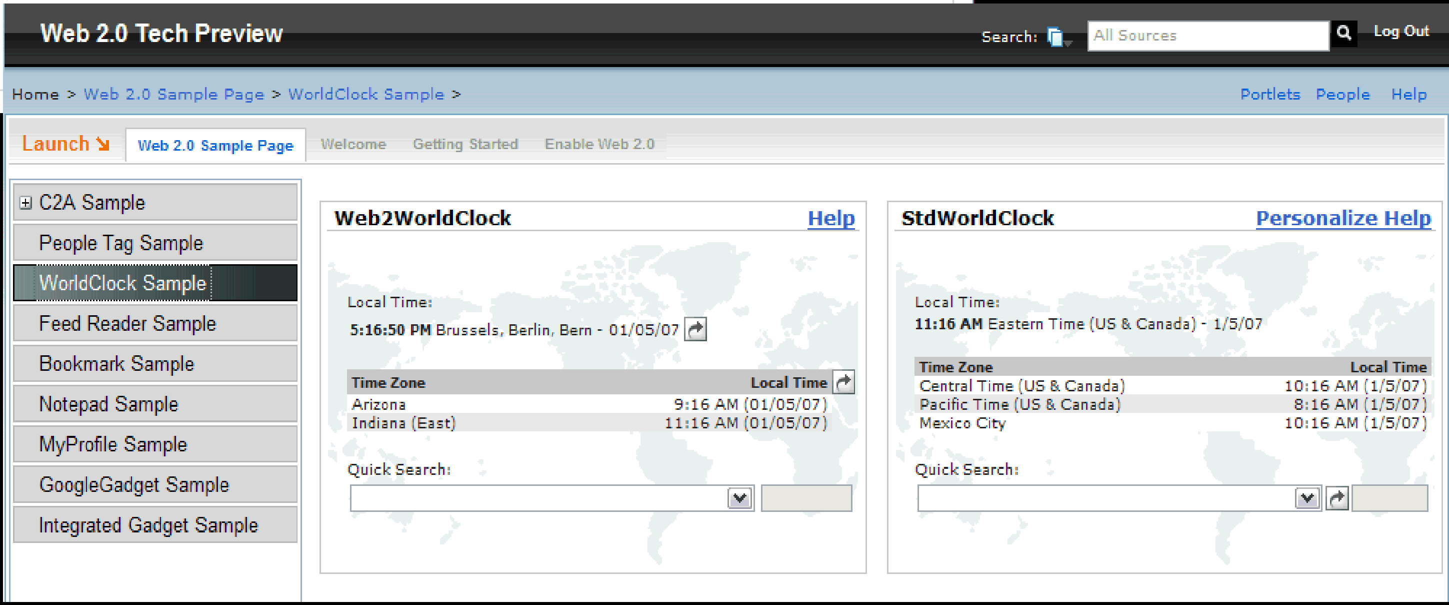
Notepad portlet

Normal — Arial — 5 (18 pt)

Enter your text here.

To test the insert image button, try one of the following uris
/wps/themes/html/web2theme/images/pic1.jpg
/wps/themes/html/web2theme/images/pic2.jpg

Main Heading



Web 2.0 Tech Preview — Search: All Sources — Log Out

Home > Web 2.0 Sample Page > WorldClock Sample > — Portlets People Help

Launch — Web 2.0 Sample Page — Welcome — Getting Started — Enable Web 2.0

- C2A Sample
- People Tag Sample
- WorldClock Sample
- Feed Reader Sample
- Bookmark Sample
- Notepad Sample
- MyProfile Sample
- GoogleGadget Sample
- Integrated Gadget Sample

Web2WorldClock — Help

Local Time:
5:16:50 PM Brussels, Berlin, Bern - 01/05/07

Time Zone	Local Time
Arizona	9:16 AM (01/05/07)
Indiana (East)	11:16 AM (01/05/07)

Quick Search:

StdWorldClock — Personalize Help

Local Time:
11:16 AM Eastern Time (US & Canada) - 1/5/07

Time Zone	Local Time
Central Time (US & Canada)	10:16 AM (1/5/07)
Pacific Time (US & Canada)	8:16 AM (1/5/07)
Mexico City	10:16 AM (1/5/07)

Quick Search:

IBM

Lotus Quickr – Web 2.0 Collaboration
(see www.ibm.com/lotus/quickr)

Content Sharing
Quickly share and manage documents, forms, images, or other media in content libraries.

Team Blogs
Keep a journal or blog of your meetings or creating discussions on different topics effecting your team.

Wikis
A shared editing space that team members can use to create and manage content such as designs, presentations, or other group material.

Team Calendar
Manage a community view of important events and activities that effect your team.

Lists
Lists are simple “databases” of information such as tasks, announcements, contacts, … that can be used to share information simply and quickly.

Lotus Connections – Web 2.0 Social Networking
(see www.ibm.com/lotus/connections)

Profiles
Quickly find the people you need by searching across your organization using keywords that help identify expertise, current projects and responsibilities

Communities
Create, find and join communities of people who share a common interest, responsibility, or area of expertise

Blogs
Use a weblog to present your point of view and get feedback from others; read what others are saying

Dogear
Save, organize and share bookmarks to valued online resources, discover bookmarks that have been shared by others

Activities
Organize your work, plan next steps, and collaborate easily with others to execute on your everyday deliverables

Questions ?