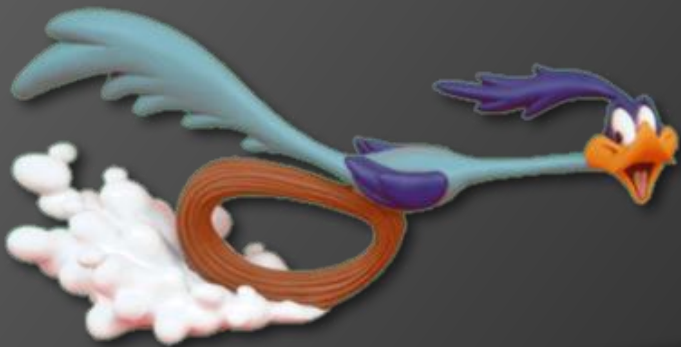


Life in the fast lane with

VELOCITY POWER



VELOCITY POWER

Patrick Talbot / Servoy Stuff

- ⦿ Java/Servoy Freelance
- ⦿ Open Source developer
- ⦿ Servoy Valued Professional
- ⦿ Co-Admin of ServoyForge
 - 50+ projects as Project Leader
 - 10+ as Contributor



VELOCITY POWER

The Velocity “Multiverse”

What is this? (And why should I care?)



VELOCITY POWER

In the beginning...



VELOCITY POWER

In the beginning...

- ⦿ HTML AREA

- Sucks at rendering real world HTML

- ⦿ Browser Suite

- Great but heavy (+ Native issues)

VELOCITY POWER

In the beginning...

- ⦿ xHTMLRenderer plugin (Jeff Bader)
 - Commercial plugin
 - Based on Flying Saucer Java lib (aka xHTMLRenderer)
 - Primary focus was better HTML AREA
 - Open sourced in Feb. 2010

VELOCITY POWER

In the beginning...

- No need to duplicate Browser Suite
- But xHTMLRenderer has strength
 - Static xHTML rendering
 - xHTML to PDF

VELOCITY POWER

In the beginning...



- Shortcomings of JasperReports:
 - iReport complexity
 - Fixed layout
 - Quirky sub reports
 - Difficult/impossible to create multi structured reports

VELOCITY POWER

Entering Apache Velocity...

- Awesome templating tool

- Fast
- Easy
- Fast
- Easy
- ...
- Did I say Fast?



VELOCITY POWER

Apache Velocity + xHTMLRenderer

- ⦿ Merging data into xHTML
- ⦿ Rendering the result
- ⦿ Make it fast and easy!

VELOCITY POWER

VelocityReport

- ⦿ Reporting engine based on xHTML/CSS
- ⦿ PDF creation
- ⦿ HTML or any text based merging
 - Emails
 - CSV
 - XML
- ⦿ Better HTML AREA
- ⦿ Reports/Dashboards
 - Graphs
 - HTML
 - Tables



VELOCITY POWER

Servoy Web client

- ⦿ Great for form based apps but:
 - Lack of layout flexibility
 - Heavy
 - Weird HTML
 - Quirky to integrate CSS and 3rd party JavaScript framework
 - Slow

VELOCITY POWER

Web server

Various approaches:

- Component oriented
 - Tapestry / Wicket / Click / ...
- MVC based
 - Struts
- Ad-hoc templates
 - ASP/PHP/JSP

VELOCITY POWER

Velocity 'Web client'



VELOCITY POWER

Velocity

- ⦿ Better REST WS plugin +
- ⦿ Fast Velocity templating engine +
- ⦿ VelocityReport awesome integration with Servoy

VELOCITY POWER

Velocity

- ⦿ Easy to create web pages and sites
- ⦿ Using ANY client framework or plain HTML
- ⦿ ANY layout
- ⦿ Huge array of applications (web services)
- ⦿ JSON or XML or HTML or PDF
- ⦿ Did I say fast?

VELOCITY POWER

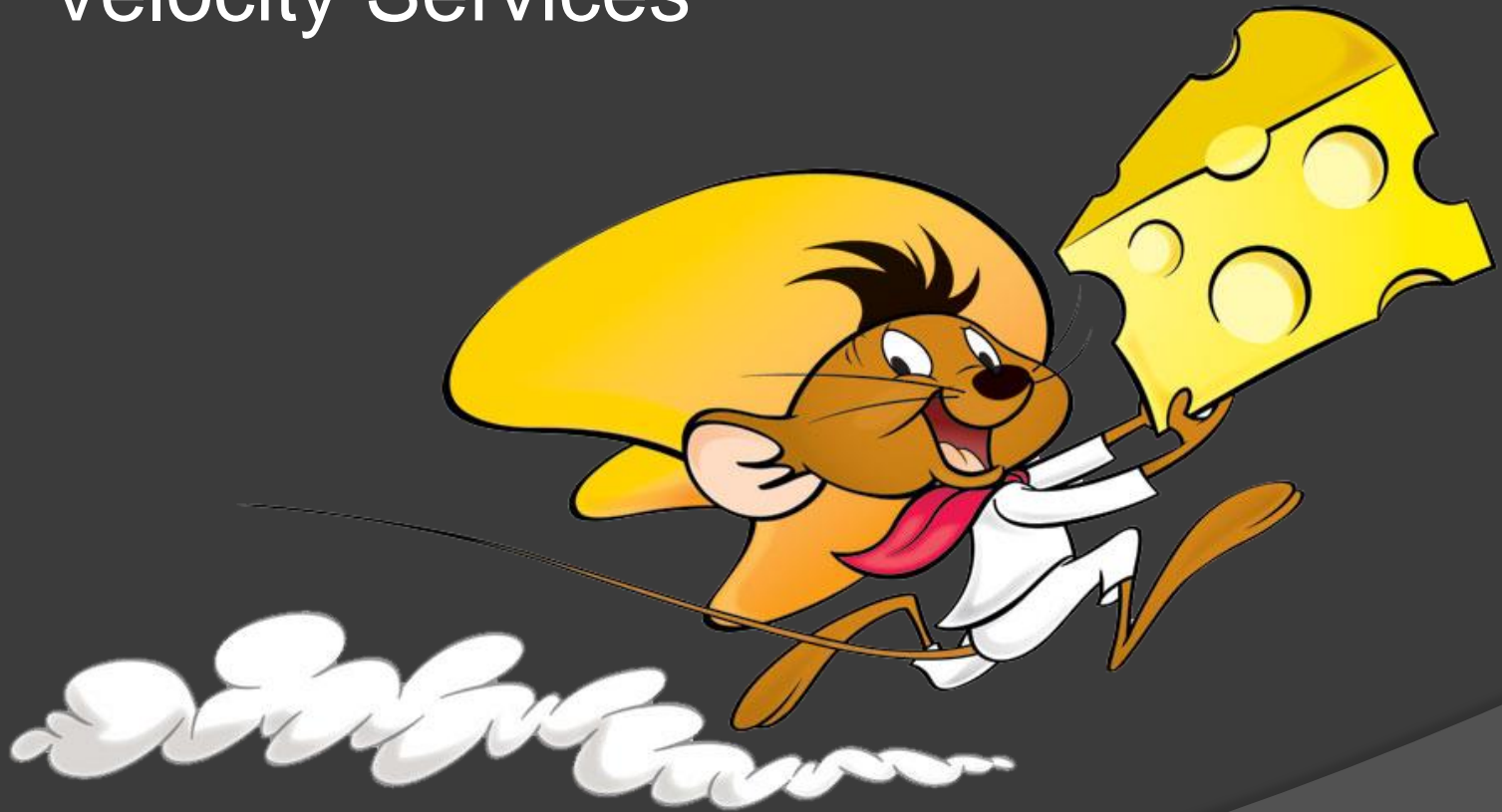
Velocity

Can expose web services that client will call to get data from Servoy (in JSON or XML format)

But how to better consume web services in Servoy???

VELOCITY POWER

Velocity Services



VELOCITY POWER

Velocity Services

- Easy declarative way to integrate JSON/XML web services
- Invocation directly in Servoy (invokeService)
- Or integrated pre or post requests to a Velocity web service/page

VELOCITY POWER

Velocity Services

- Easier than using HTTP plugin
- Can work asynchronously (with a callback)
- On the fly declaration of new services (in config.json)

VELOCITY POWER

In summary: the Velocity multiverse

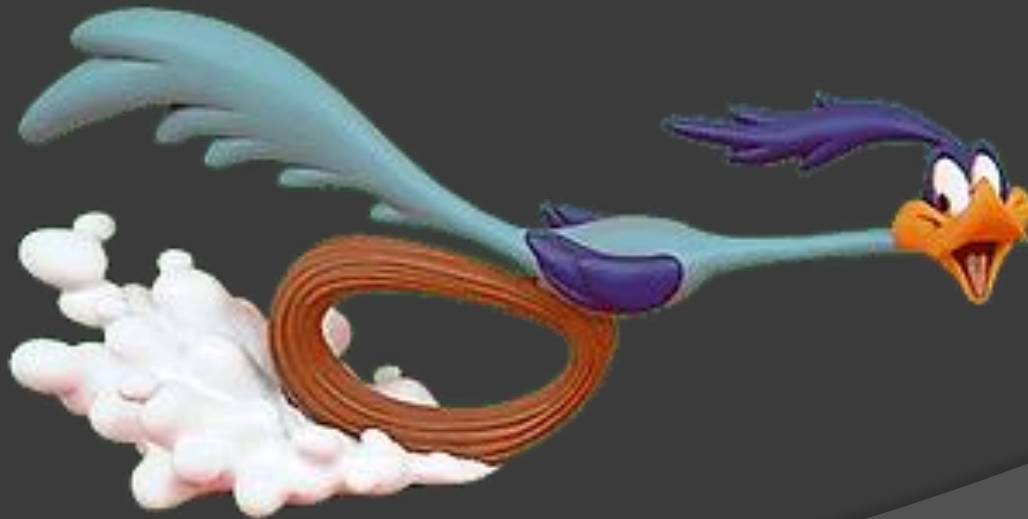
- ⦿ VelocityReport for:
 - Reporting / Merging / Dashboards / Rendering
- ⦿ Velocity for:
 - Serving Web pages / Exposing Web services
- ⦿ Velocity Services for:
 - Consuming Web services

VELOCITY POWER

Questions?

VELOCITY POWER

Velocity Plugin
A fast as hell Web client!



VELOCITY POWER

A REST servlet (on steroid)

+

Fast and furious Velocity
templating engine

+

VelocityReport capabilities

VELOCITY POWER

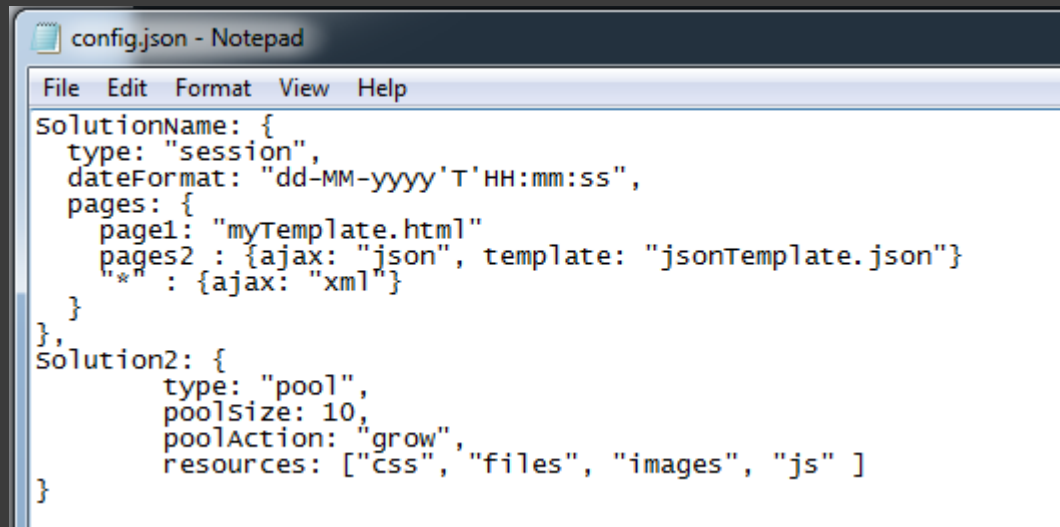
A REST servlet (on steroid)

- ⦿ 3 levels of headless client instantiation (per solution):
 - application (one headless client to rule them all!)
 - pool (just like the REST plugin, but much better!)
 - session (one headless client per HTTP session)
- ⦿ In session mode, your session is stateful
- ⦿ (Actually it can be stateful also in any mode... :)

VELOCITY POWER

A REST servlet (on steroid)

Easy to configure: the config.json file

A screenshot of a Notepad window titled 'config.json - Notepad'. The window has a menu bar with 'File', 'Edit', 'Format', 'View', and 'Help'. The text content is a JSON configuration file for Velocity Power. It defines 'SolutionName' with a 'session' type, a date format, and a list of pages. It also defines 'Solution2' with a 'pool' type, a pool size of 10, a 'grow' action, and a list of resources: 'css', 'files', 'images', and 'js'.

```
config.json - Notepad
File Edit Format View Help
SolutionName: {
  type: "session",
  dateFormat: "dd-MM-yyyy'T'HH:mm:ss",
  pages: {
    page1: "myTemplate.html"
    pages2 : {ajax: "json", template: "jsonTemplate.json"}
    "*" : {ajax: "xml"}
  }
},
Solution2: {
  type: "pool",
  poolSize: 10,
  poolAction: "grow",
  resources: ["css", "files", "images", "js" ]
}
```

VELOCITY POWER

config.json

```
solution1: { }, solution2: {  
  ○ type (application|session|pool)  
  ○ poolSize / poolAction  
  ○ ajax / mimeType  
  ○ dateFormat / numberFormat / jsonDateFormat  
  ○ defaultForm  
  ○ defaultTemplate  
  ○ errorPage  
  ○ extractBody  
  ○ serializeNull  
  ○ authentication  
  ○ defaultUser  
  ○ loginPage  
  ○ userParam / passParm  
  ○ resourceHeaders {}  
  ○ resources: [ ]  
  ○ deny: [ ]  
  ○ pages: { }  
  ○ services: { }  
}
```

VELOCITY POWER

config.json

```
solution1: { }, solution2:
{
  pages : {
    • formName (can use * wildcards): {
      - ajax
      - jsonp
      - template
      - mimeType
      - form
    }
  }
}
```

VELOCITY POWER

Apache Velocity language

\$ # ! . () []

if / else / elseif / end

foreach / end / break

set / define

parse / include / evaluate

macro

VELOCITY POWER

VelocityReport capabilities

- Use ANY Servoy objects
- Extend with build-in charts and barcodes
- Create JSON/XML response on the fly
- Easily return dynamic PDF/Excel to a request

VELOCITY POWER

The one function

```
vr_getContext(request) {  
    // analyse request object  
    // do anything based on that  
    // return a ResponseObject  
}
```

VELOCITY POWER

request object

- A javascript object
 - **request.method**:
 - GET
 - POST
 - DELETE
 - PUT
 - HEAD
 - OPTIONS
 - TRACE
- request contains collections:
 - **request.parameters** {}
 - request.cookies {}
 - request.headers {}
 - **request.session** {}
 - request.parts {}[]
 - request.userGroups []
 - **request.extraPaths** []
- request also contains additional values:
 - request.extension
 - request.pathInfo
 - request.protocol
 - request.scheme
 - request.characterEncoding
 - request.locale
 - request.queryString
 - **request.requestedForm**
 - request.sessionID
 - request.clientID
 - request.multipart
 - request.body
 - request.headlessType
 - request.authentication
 - request.userName
 - request.userID

VELOCITY POWER

Manipulate the response:

- **createReponse**
- createRedirectResponse
- createForwardResponse
- createBytesResponse
- createFileResponse
- createErrorResponse

VELOCITY POWER

ResponseObject properties:

- context
- template
- Inline
- mimeType
- URL
- bytes
- file
- deleteAfterServe
- errorCode
- errorMessage
- errorLink
- errorException

VELOCITY POWER

ResponseObject methods:

- addHeader
- getHeader
- removeHeader
- addCookie
- getCookie
- removeCookie
- addSessionAttribute
- getSessionAttribute
- removeSessionAttribute
- invalidate
- invokeService

VELOCITY POWER

Velocity plugin extra methods:

- createCookie
- getConfig
- invokeService
- createMimeMultipart

VELOCITY POWER

Cookie object:

- domain
- expiryDate
- maxAge
- path
- sessionBased
- value
- getName()

VELOCITY POWER

MimeMultipart object:

- related
- addBodyPart
- addFilePart

VELOCITY POWER

The Velocity workflow

1. HTTP Request
2. Velocity plugin servlet
3. Headless client
4. `function vr_getContext(request)`
5. Do anything you want in Servoy!
6. Return a `ResponseObject`
7. Rendering: template gets filled
8. Browser interprets `html+css+js`
9. Takes no time = Happy user!

VELOCITY POWER

Installing Velocity

- <https://www.servoyforge.net/projects/velocity-report/files>
- Unzip, put /plugins in /plugins
- Put eastwood.war in /server/webapps
- Create a reports folder somewhere
- Put VM_global_library.vm in there
- Create a www/ folder + a config.json file
- In servoy-admin/plugins-setting:
 - set velocityreport.reportfolder
 - and velocityreport.serverURL

VELOCITY POWER

Questions?

VELOCITY POWER

DEMO

**A functional webstore with
Bootstrap/jQuery/Servoy/Velocity**

<https://www.servoyforge.net/projects/velocity-samples>

VELOCITY POWER

Demo: Things to note

- Pick your favorite client framework or roll your own
- Flexible layout, any HTML, even HTML5
- Debugging in Velocity
- Error management
- Tomcat Manager
- config.json is entirely dynamic
- Templates/includes are entirely dynamic
- Form changes are mostly dynamic
- If not, kill that client!

VELOCITY POWER

In the template

- Integrating charts/barcodes in your pages
- Creating PDF on the fly
- Creating JSON/XML on the fly (using the extension)
- No need for extensions (cleaner URLs / REST style)
- Using `$i18n` in the templates
- Using `$globals` and `$scopes` (even functions) in the templates
- Using `$valueLists` in the templates
- `$now`, `$date` and `$number`
- use `.toJSON()` to pass objects to JavaScript
- `$httpSession`
- Silence null and exceptions !
- `$null.isNull()` and `$null.isNotNull()`

VELOCITY POWER

In the template

- `foundset == dataset == Array<Object>`
- `$fs.get() / $fs.getRecord() / $fs[]`
- `$ds.get() / $ds.getRow() / $ds[]`
- `[] == .get()`
- `$function.call()`
- `#foreach > object/array/foundset/dataset`
- In loops, `$foreach.count / $foreach.hasNext`
- `$record` has relationships
- `$record` has calculations/aggregates
- Use dot (.) syntax, just like in Servoy
- Use `${}` to delimit a variable in edge cases
- Manipulate ANY text (html / js / css / xml / json) at rendering time

VELOCITY POWER

A little quiz: what does this do?

// IN SERVVOY:

```
globals.const_TABLE_TO_DETAIL = {  
    products: { related: 'product', ref: 'id' },  
    orders: { related: 'order', ref: 'filter' } // etc.  
}  
  
function vr_getContext(request) {  
    var context = {};  
    context._page: request.requestedForm;  
    // some more treatment here  
    return plugins.Velocity.createResponse(context);  
}
```

// IN THE TEMPLATE:

```
$(document).ready(function () {  
    #set($rel = $globals.const_TABLE_TO_DETAIL[$_page])  
    #if($rel)  
        $('table#$_page tbody tr').click( function(e) {  
            var id = $($ (this).children()[0]).text();  
            window.location.href = '$rel.related?$rel.ref='+id;  
        } );  
    #end  
} );
```

VELOCITY POWER

In config.json

- Automatic reload
- solutionName: {} – default for everything
- type: application vs session vs pool
- dateFormat / numberFormat / jsonDateFormat
- authentication
- defaultUser (users.properties) / loginPage + userParam/passParam
- resources = static files / deny = 403
- defaultForm / defaultTemplate/ errorPage
- extractBody / jsonDateFormat / serializeNull
- resourcesHeaders
- Services (pre/post/client/server?)
- pages – implicit definition {template, form, ajax, jsonp, mimeType}
- Valid extensions for template: html/xhtml/htm/tmpl/tpl/vtl/xml/txt
- Access templates from another solution ../ (shared templates)
- use #include() in config.json – always 'touch' config.json
- JSON need to be well formed! (<http://jsonviewer.stack.hu/>
<http://www.freeformatter.com/json-validator.html>)

VELOCITY POWER

In Servoy

- request gives access to everything!
- use request.method for REST, along with extraPaths []
- Ajax request == http request
- request.parameters for GET/POST/PUT/DELETE
- ResponseObject = createResponse()
- Put anything you want in the context
- Redirect != Forward
- Bytes != File
- Form enctype=multipart/form-data
- Input=file received as JSFile (server-side)
- request.multipart / request.parts
- request.body

VELOCITY POWER

In Servoy

- `request.cookies` = limited storage, browser side
- Cookies can be persistent accross sessions
- Cookies can be disabled by users!
- `request.session` == http globals! (but better)
- Store anything in the session = stateful
- session cannot be disabled by users
- session variables can be spied upon! ☺
- session objects persist even after a server restart!
- `ResponseObject.addSessionAttribute()`
- `ResponseObject.removeSessionAttribute()`
- `ResponseObject.invalidate()`
- `createErrorResponse()` -> `errorPage`

VELOCITY POWER

Questions?

VELOCITY POWER

Want to unleash the power of Velocity?

**I'm offering consulting services/training/help
building your solutions/modules/templates**



Servoy Stuff

admin@servoy-stuff.net

VELOCITY POWER

Brought to you by:



Servoy Stuff

Find it on ServoyForge:

<https://www.servoyforge.net/projects/velocity-report>

Sponsor it!

Send me a note:

admin@servoy-stuff.net