Master Project Software Engineering: Team-based Development WS 2010/11

Implementation, September 27th , 2011

Glib Kupetov

Glib.Kupetov@iese.fraunhofer.de

Tel.: +49 (631) 6800 2128

Sebastian Weber

Sebastian.Weber@iese.fraunhofer.de

Tel.: +49 (631) 6800 2116

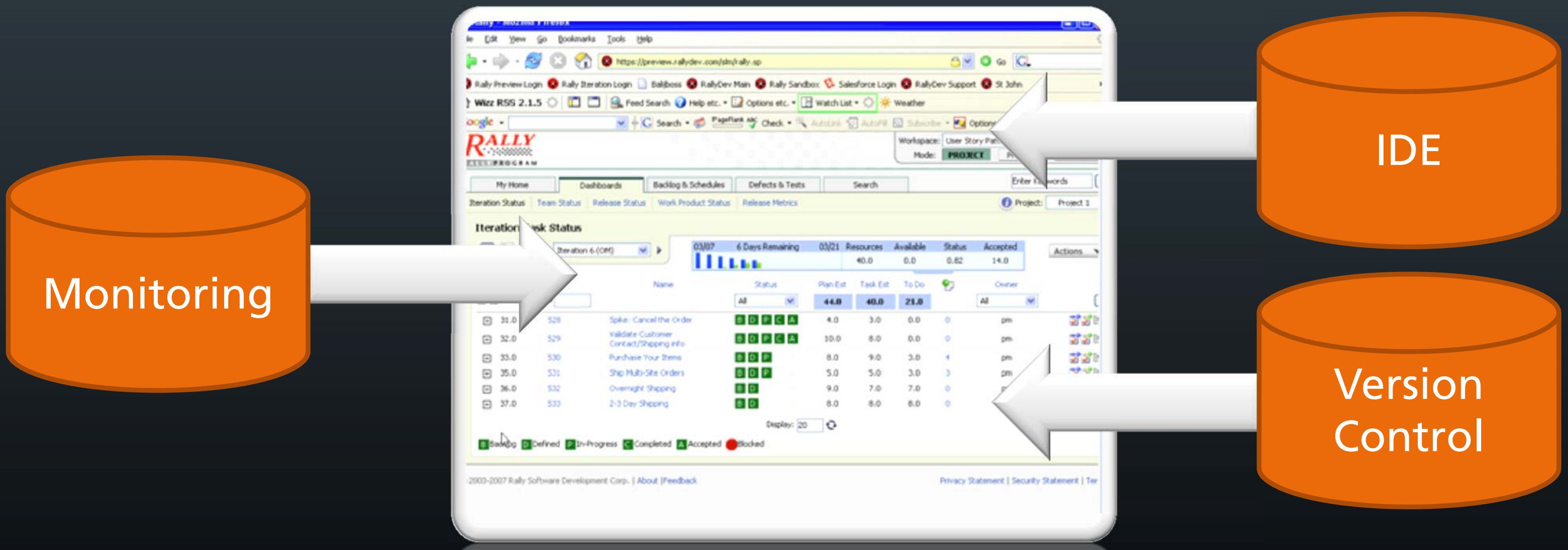
Contents

- Rally Web Services API
- REST
- Development Technologies
- Development Environment
- Tasks

Rally Web Services API

Capabilities of the Rally Web Services API

- Integrate Agile development life cycle with external systems
- Extend core functionality of Rally by generating business specific reports



https://preview.rallydev.com/slm/rally.sp

Iteration Task Status

Iteration 6 (OM)

03/07	6 Days Remaining	03/21	Resources	Available	Status	Accepted
			40.0	0.0	0.82	14.0

		Name	Status	Plan Est	Task Est	To Do		Owner
			All	44.0	40.0	21.0		All
31.0	528	Spike: Cancel the Order	B D P C A	4.0	3.0	0.0	0	pm
32.0	529	Validate Customer Contact/Shipping info	B D P C A	10.0	8.0	0.0	0	pm
33.0	530	Purchase Your Items	B D P	8.0	9.0	3.0	4	pm
35.0	531	Ship Multi-Site Orders	B D P	5.0	5.0	3.0	3	pm
36.0	532	Overnight Shipping	B D	9.0	7.0	7.0	0	
37.0	533	2-3 Day Shipping	B D	8.0	8.0	8.0	0	

Display: 20

B Backlog D Defined P In-Progress C Completed A Accepted ● Blocked

2003-2007 Rally Software Development Corp. | About | Feedback   Privacy Statement | Security Statement | Ter

Web Service API

- Versioned Web Services API

  Updates and changes to the core Rally application will not impact any custom code, applications or extensions

- Supports REST and SOAP

  Standards-based API allows access to Rally data from any language

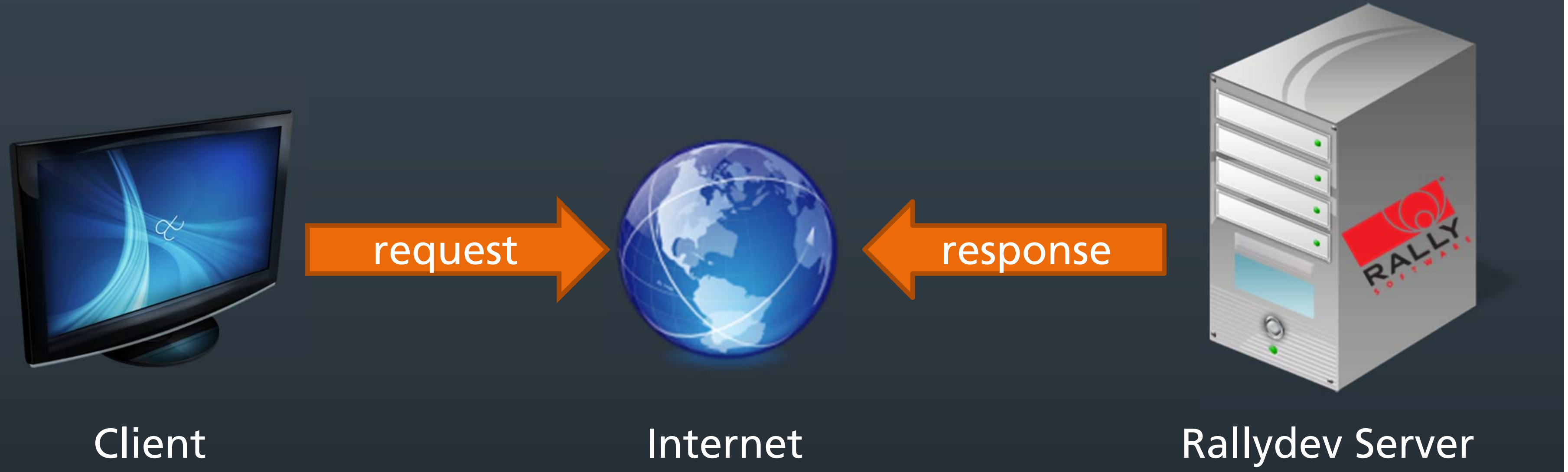
- Interactive Web Service Browser

  Use a standard web browser to display Rally data in either HTML, XML, or JSON

- High Performance API

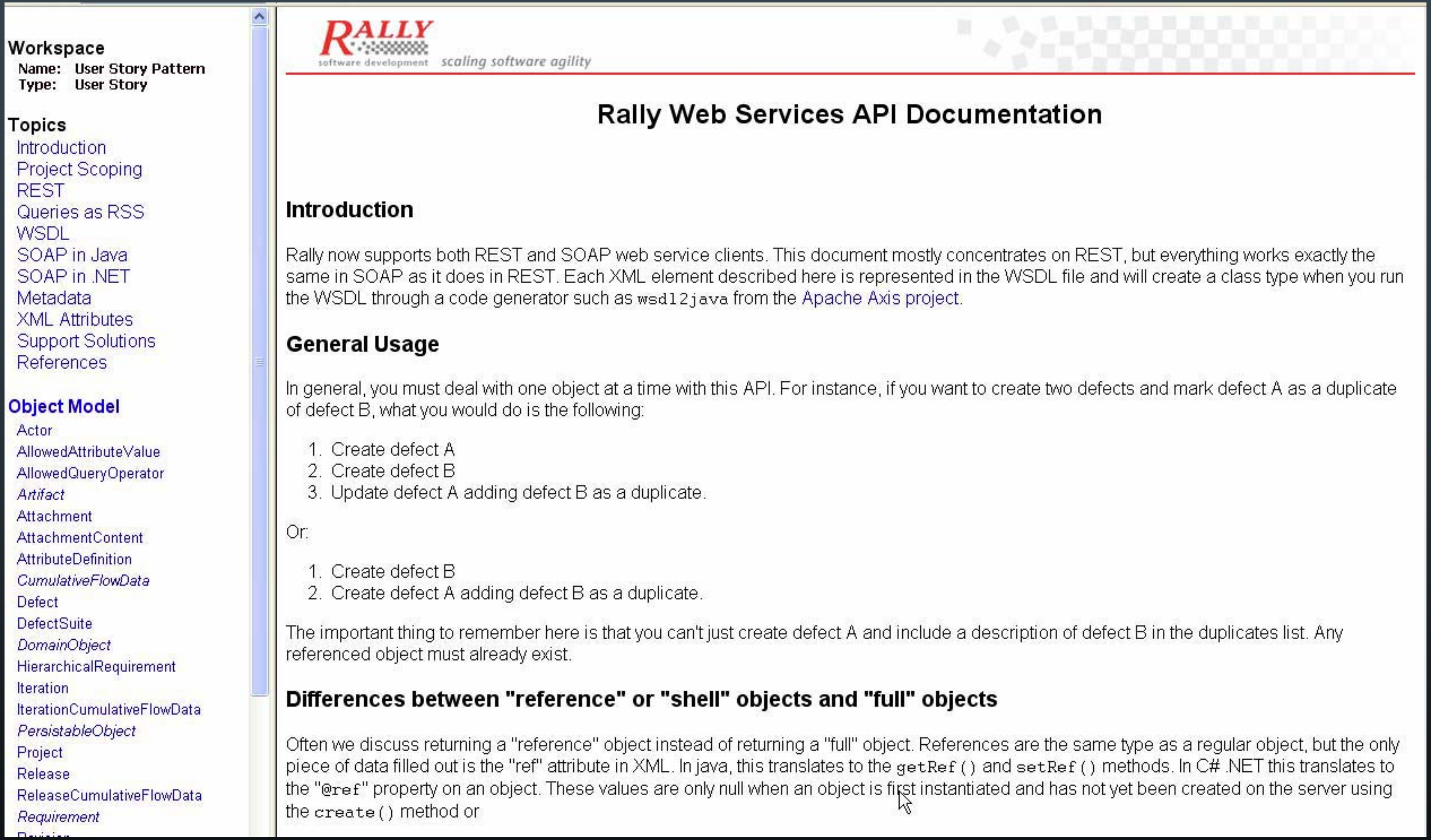
  API supports compound queries that return just the data necessary

How does the WS API work?



- Request-response model
- Client makes the request

Rally Web Services API



Workspace
Name: User Story Pattern
Type: User Story

Topics
- Introduction
- Project Scoping
- REST
- Queries as RSS
- WSDL
- SOAP in Java
- SOAP in .NET
- Metadata
- XML Attributes
- Support Solutions
- References

Object Model
- Actor
- AllowedAttributeValue
- AllowedQueryOperator
- *Artifact*
- Attachment
- AttachmentContent
- AttributeDefinition
- *CumulativeFlowData*
- Defect
- DefectSuite
- *DomainObject*
- HierarchicalRequirement
- Iteration
- IterationCumulativeFlowData
- *PersistableObject*
- Project
- Release
- ReleaseCumulativeFlowData
- *Requirement*
- Revision

Rally Web Services API Documentation

Introduction

Rally now supports both REST and SOAP web service clients. This document mostly concentrates on REST, but everything works exactly the same in SOAP as it does in REST. Each XML element described here is represented in the WSDL file and will create a class type when you run the WSDL through a code generator such as `wsdl2java` from the Apache Axis project.

General Usage

In general, you must deal with one object at a time with this API. For instance, if you want to create two defects and mark defect A as a duplicate of defect B, what you would do is the following:

1. Create defect A
2. Create defect B
3. Update defect A adding defect B as a duplicate.

Or:

1. Create defect B
2. Create defect A adding defect B as a duplicate.

The important thing to remember here is that you can't just create defect A and include a description of defect B in the duplicates list. Any referenced object must already exist.

Differences between "reference" or "shell" objects and "full" objects

Often we discuss returning a "reference" object instead of returning a "full" object. References are the same type as a regular object, but the only piece of data filled out is the "ref" attribute in XML. In java, this translates to the `getRef()` and `setRef()` methods. In C# .NET this translates to the "`@ref`" property on an object. These values are only null when an object is first instantiated and has not yet been created on the server using the `create()` method or

--- REST ---

REST’ful Web Services

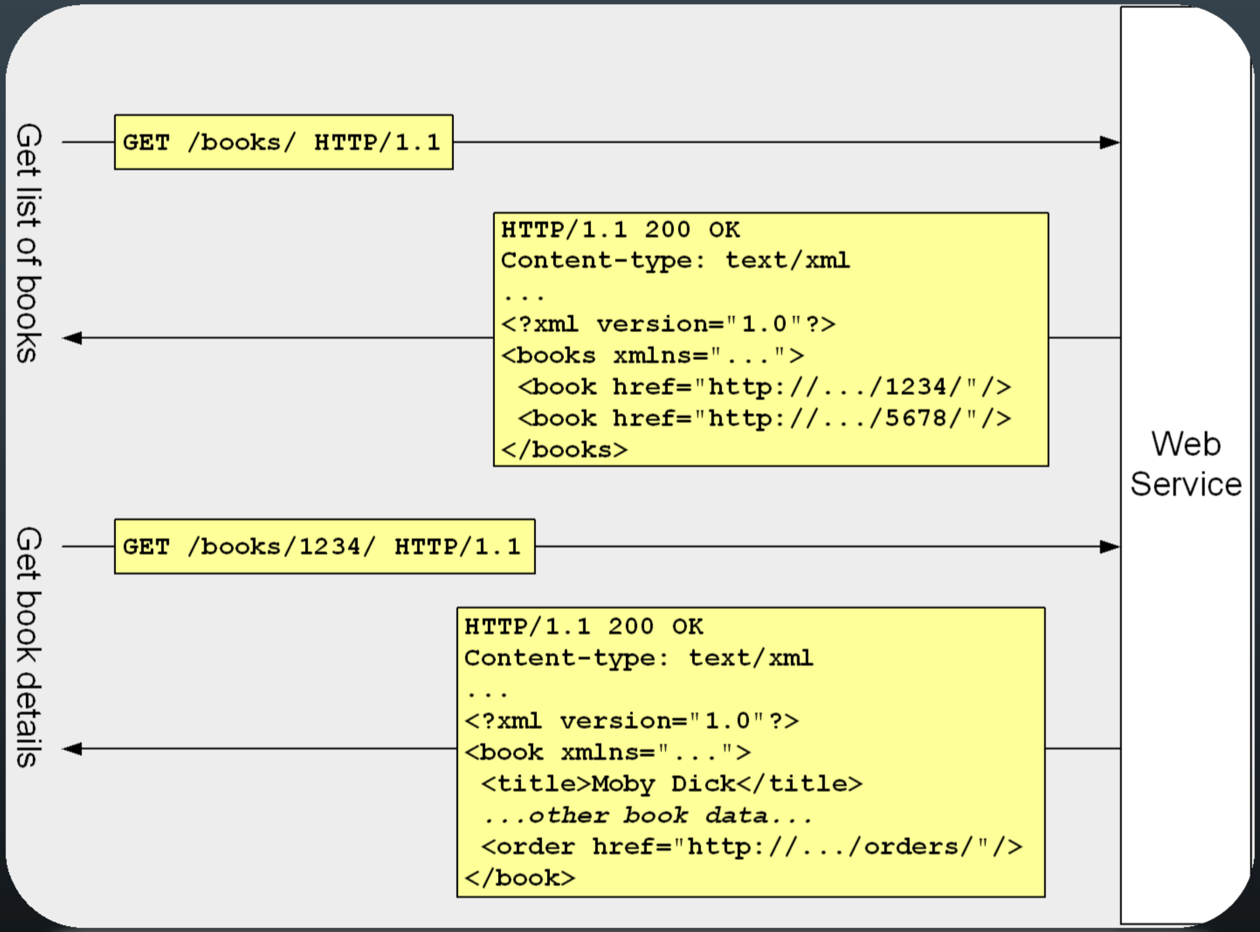
- Communication between client and server via HTTP
- Base URI for the web service, such as http://example.com/resources/
- Internet media type of the data supported by the web service. This is often JSON or XML.
- The key characteristic of a RESTful Web service is the explicit use of HTTP methods to denote the invocation of different operations

- The major advantages of REST-services are:
  - Reusable across platforms (Java, .NET, PHP, etc) since they rely on basic HTTP protocol
  - Basic XML or JSON instead of the complex SOAP XML and are easily consumable

HTTP Methods

- POST
  - CREATE
  - Create a new resource
- GET
  - RETRIEVE
  - Retrieve a representation of a resource
- PUT
  - UPDATE
  - Update a resource

REST HTTP Example (XML)



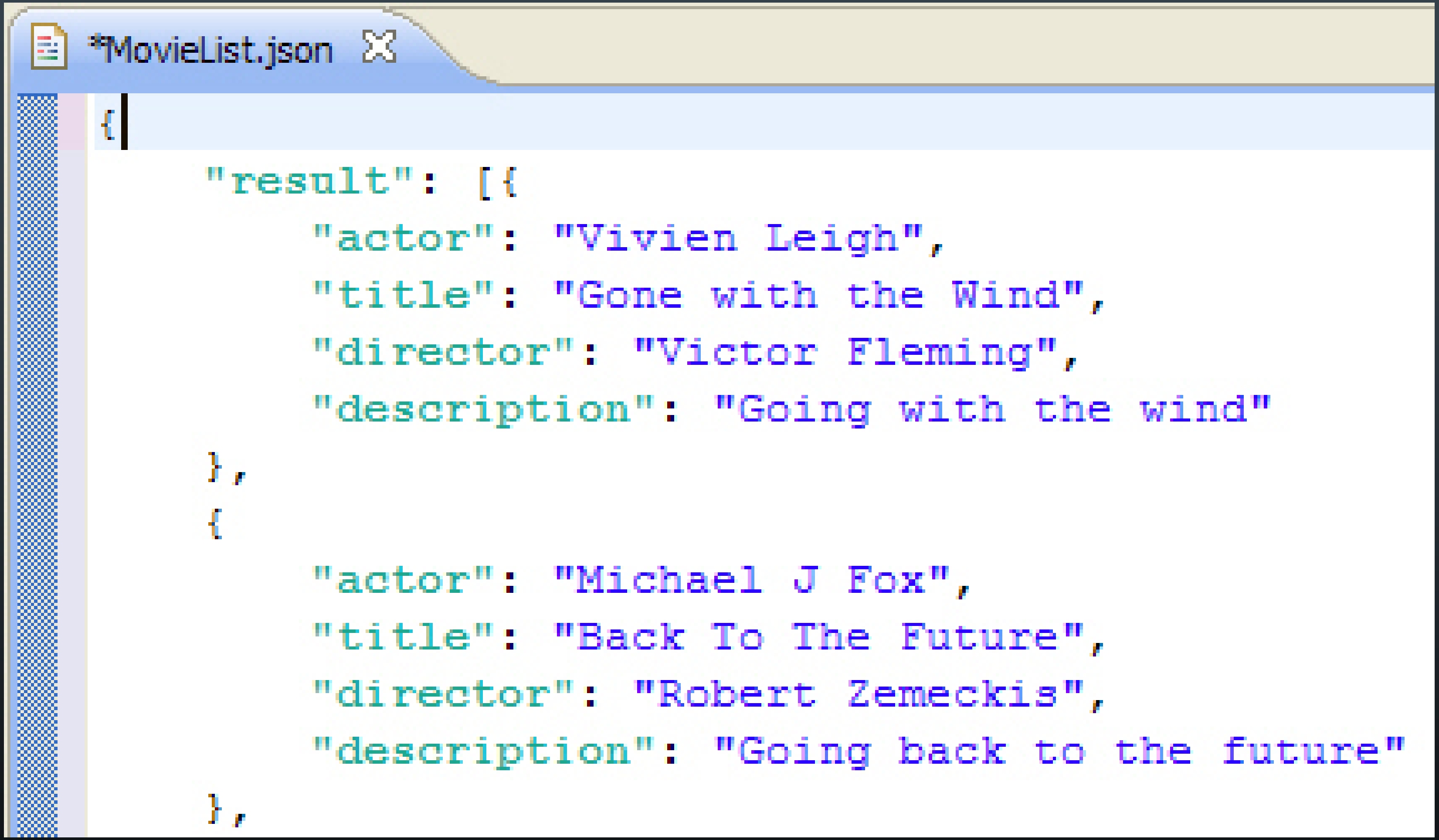
```
GET /books/ HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-type: text/xml
...
<?xml version="1.0"?>
<books xmlns="...">
 <book href="http://.../1234/"/>
 <book href="http://.../5678/"/>
</books>
```

```
GET /books/1234/ HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-type: text/xml
...
<?xml version="1.0"?>
<book xmlns="...">
 <title>Moby Dick</title>
 ...other book data...
 <order href="http://.../orders/"/>
</book>
```

REST HTTP Example (JSON)



```
{
    "result": [{
        "actor": "Vivien Leigh",
        "title": "Gone with the Wind",
        "director": "Victor Fleming",
        "description": "Going with the wind"
    },
    {
        "actor": "Michael J Fox",
        "title": "Back To The Future",
        "director": "Robert Zemeckis",
        "description": "Going back to the future"
    },
```

--- Development Technologies ---

HTML5

- Latest revision of HTML specified by W3C
- Still in progress
- Important thing, major browser support of HMTL5 features is good
- The good thing in this project:
  - We do not bother with IE6 and other outdated, buggy browsers
  - It's a design decision that the Web-based application will run in a modern Firefox environment
- Use CSS for styling purposes

New features of HTML5 in a nutshell

- Backward compatibility and progressive enhancement
- New semantic elements, e.g., section, header, footer, etc.
- New types of form controls, e.g., date and time, email, etc.
- New attributes, e.g., charset on meta and async on script
- Drop of deprecated elements, e.g., font, center, applet, etc.
- In general, many things can be done more easily with HTML5
  - With former version, many things had have to be written with additional javascript code (e.g., validating stuff, remoting, etc.)

JavaScript Framework

jQuery (http://jquery.com/)

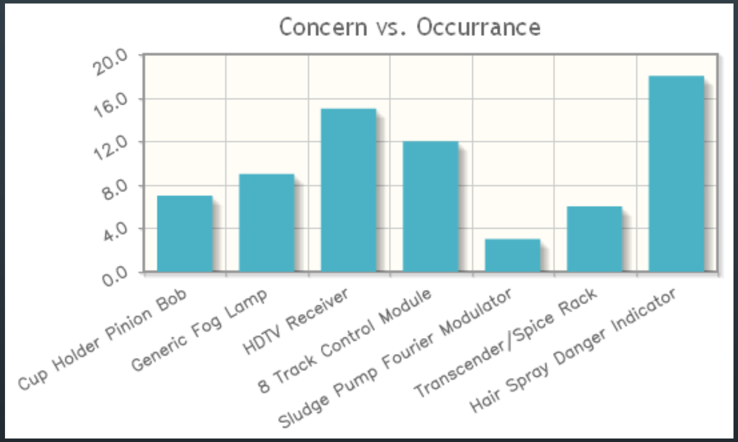
- jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript. --jQuery.com
- Cross-Browser: latest version supports IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome
- Extensible: Tons of plugins for common (and uncommon) tasks

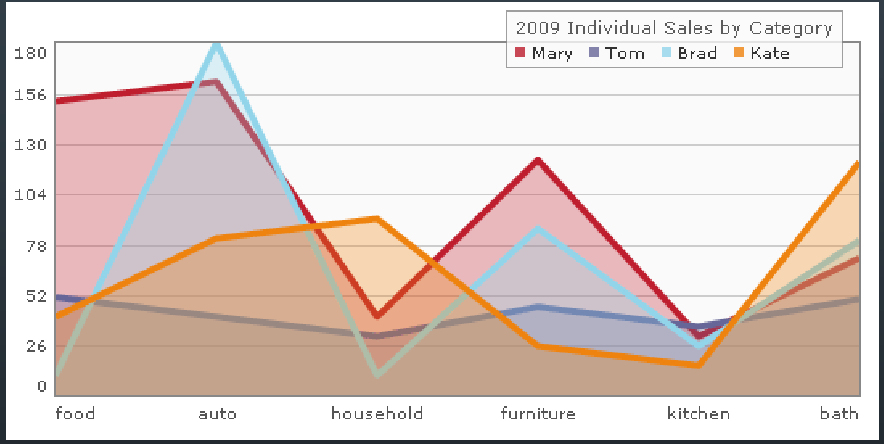
jQuery features

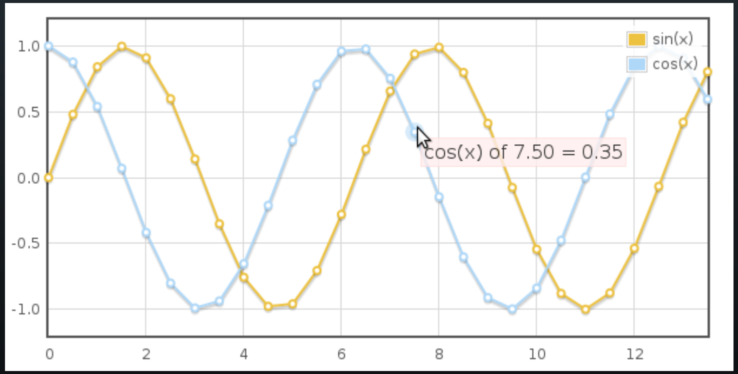
- Cross browser support and detection
- AJAX functions
- CSS functions
- DOM manipulation
- Attribute manipulation
- Event detection and handling
- Hundreds of plugins for pre-built user interfaces, advanced animations, form validation, etc http://plugins.jquery.com/projects/plugins?keywords=&type=All&sort_by=title
- Expandable functionality using custom plugins

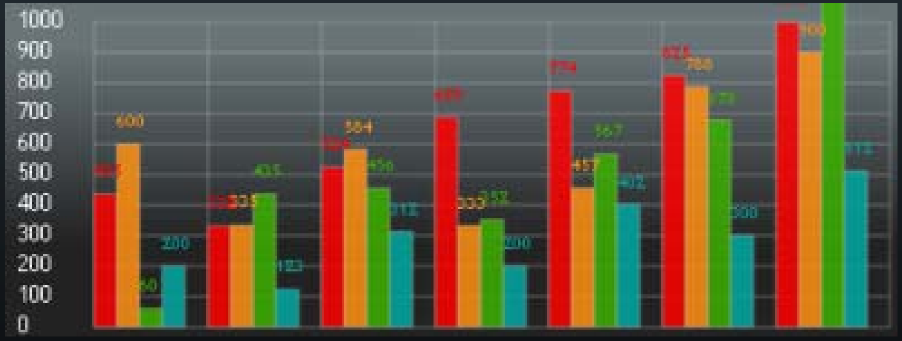
jQuery charting plugins

jQuery write less, do more.

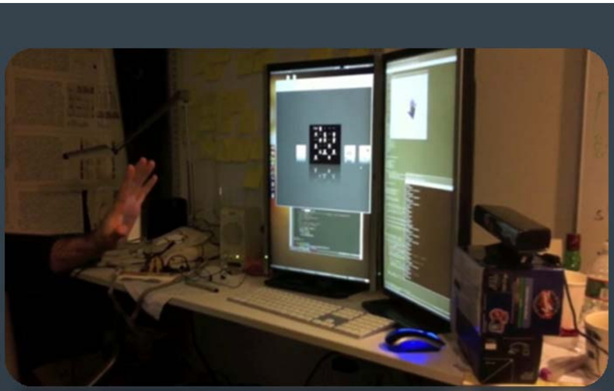








Kinect Interaction



DepthJS (http://depthjs.media.mit.edu/) - ALPHA Version!

- Browser Extension that you lets control Webpage Navigation With Hand Gestures
- MIT media lab have built this extension which uses Microsoft's gesture based controller
- Only for Safari and Chrome with MacOSX or Linux
- Video illustration: http://vimeo.com/17180651

Kinect Interaction (II)

- No keyboard, no mouse - you can just use hand gestures to perform common web actions e.g clicking links, scrolling windows, switching tabs and so on
- Until now, the DepthJS can recognize the following physical movements:
  - Presence of hand (registration)
  - Removal of hand (unregistration)
  - Hand movement
  - Quick open and close of hand (“hand click”)
  - Large swipe up/down/left/right

--- Development Environment ---

IDE

Aptana (http://aptana.com/)

- Standalone IDE or plug it into eclipse
- Support for all the main JS libraries built in
- Dont forget if you are going to use Aptana to write jQuery to enable the jQuery Code Assist :

  Aptana: Window->Preferences->Aptana->Editors->JavaScript->Code Assist-><Check jQuery>

Additional Technologies

- Use Subversion
  - E.g., „Subversive“ Eclipse Plugin (http://www.eclipse.org/subversive/)
- Use Firefox as target environment (in fullscreen mode)
- Use Firebug for debugging (http://getfirebug.com/)

Next Steps & Tasks

Next Steps

- Let’s meet next week and talk about things in more detail
  - Talk about concepts, e.g., how to write clean code, coding guidelines, patterns, anti-patterns, etc.
  - Define rules and guidelines to stick to throughout implementing the system
- In the meantime, start getting familiar with the tools and technologies mentioned in this presentation
  - therefore, it is a wonderful idea to do some home work ;-) → tasks (next slides)

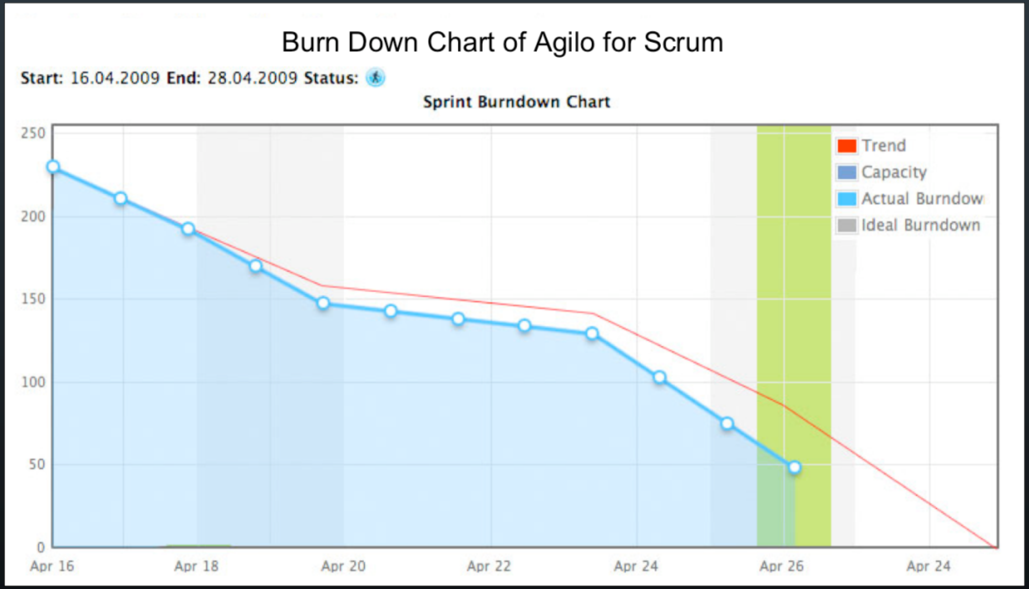
Tasks

Split-up in 3 Teams with different tasks:

- jQuery
    - UI / Charts
    - REST
- Kinect
    - Interaction Gestures

Tasks – jQuery: UI / Charts

- Choose a suitable extension for charting (jQuery plugin)
- Develop an example showing the technical feasibility and the different needed charts for the scrum visualization



Burn Down Chart of Agilo for Scrum

Start: 16.04.2009 **End:** 28.04.2009 **Status:**

Tasks – jQuery: REST

- Choose a suitable extension for REST with JSON payload (jQuery plugin) or code it self
- Exercise:
  - Develop an example that calls (GET) a REST service with a JSON response and create a JSON object with the result. After that output the object.
- Example REST service (JSON)
  http://api.geonames.org/citiesJSON?north=44.1&south=-9.9&east=-22.4&west=55.2&lang=de&username=demo

  Also available with XML output
  http://api.geonames.org/cities?north=44.1&south=-9.9&east=-22.4&west=55.2&username=demo

Tasks – Kinect: Interaction Gestures

- MacBook and Kinect available in the devlab
- Try out the possibilities of the DepthJS browser extension
- Create a demo application that recognize the swipe left / right gesture and webpage zoom in/out
- Check if this ALPHA version is stable enough