Aug 21st 2012

Decoupling Content Management with Create.js and PHPCR

Henri Bergius

Director of R&D
Nemein Berlin

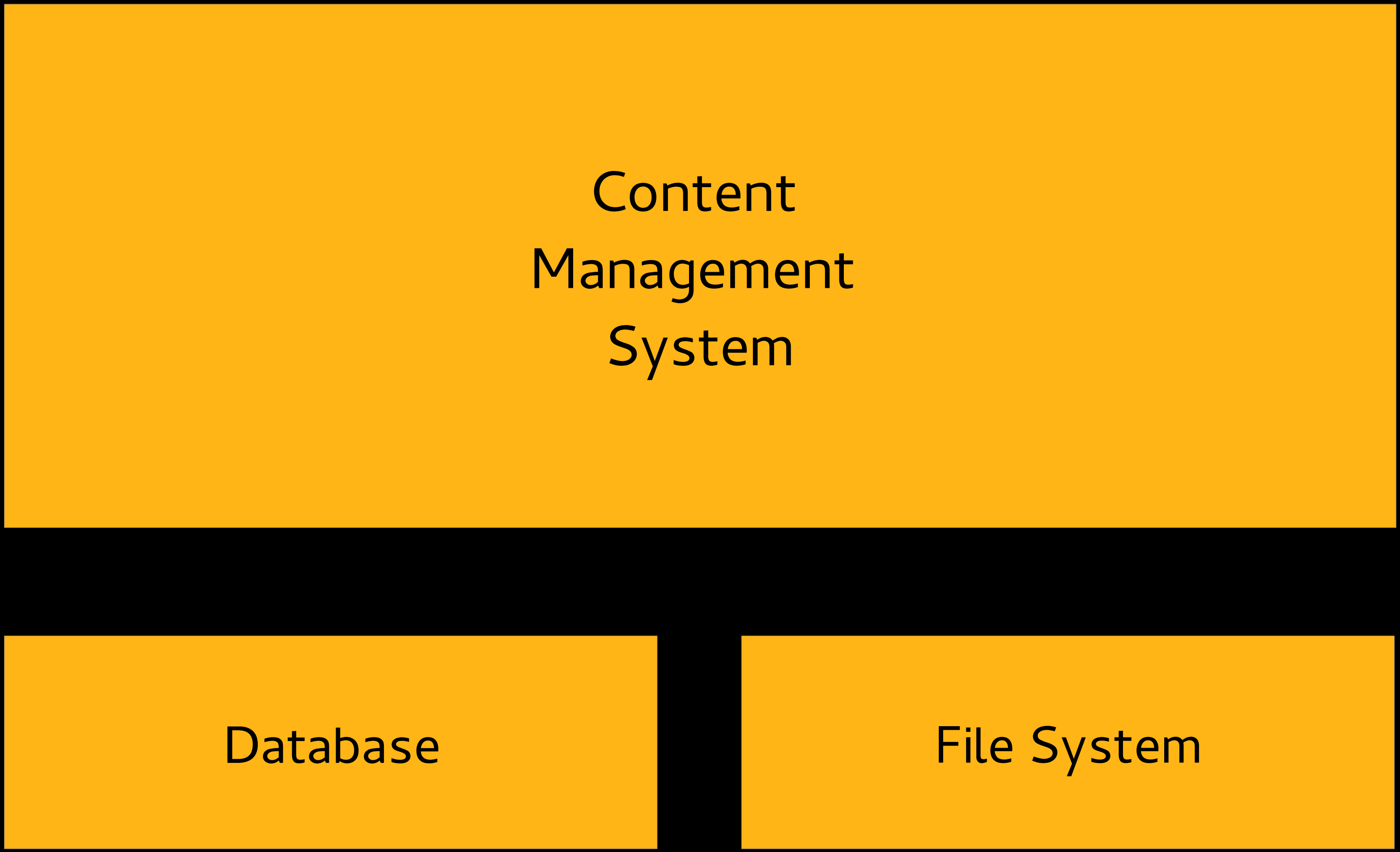
@bergie

bergie.iki.fi

"The Midgard Project is born."

Launching our first Open Source CMS effort
bergie.iki.fi, April 25th 1998

Your CMS is a monolith



Content Management System

Database

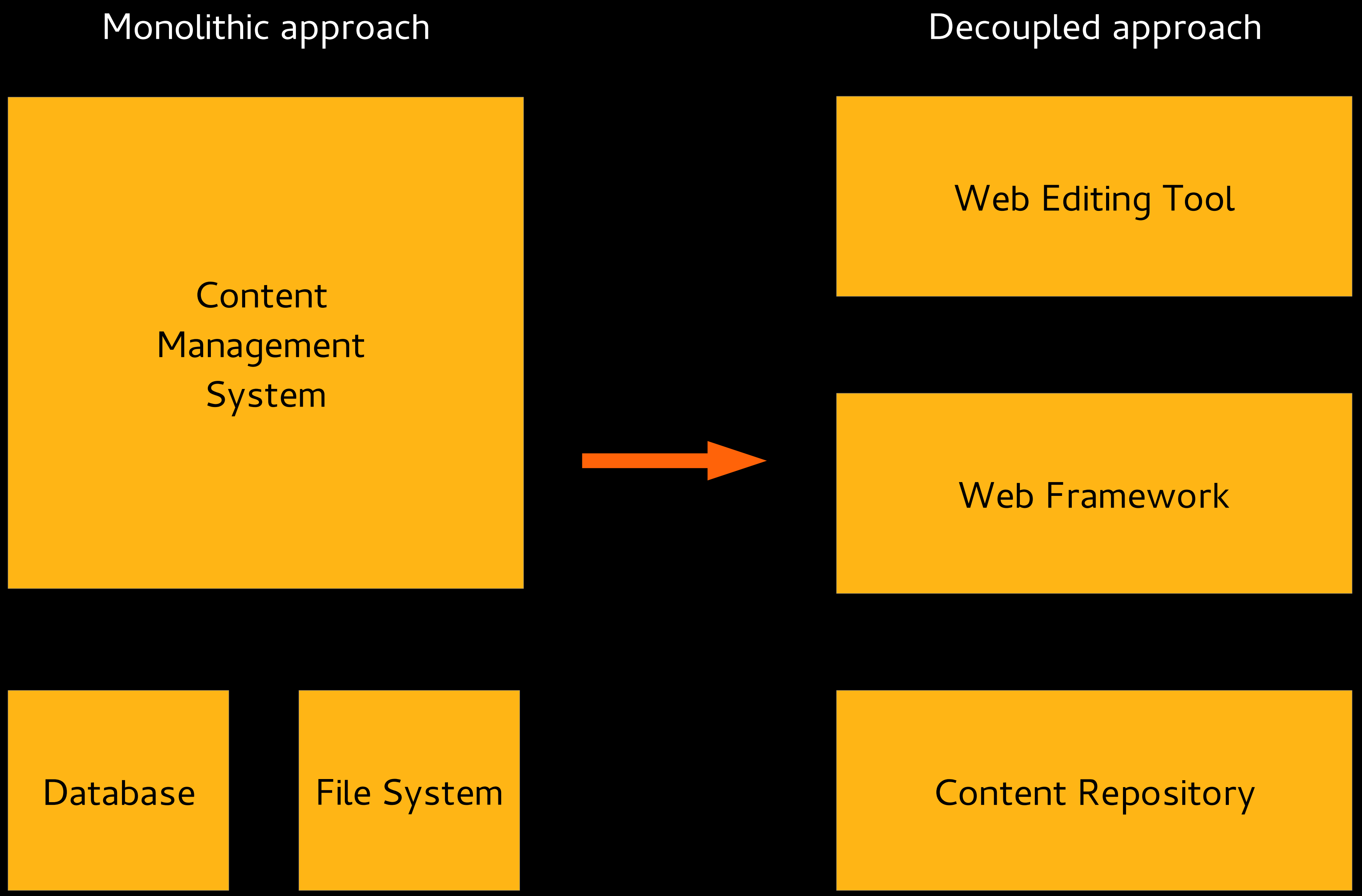
File System

"We like the user interface, but our IT doesn't want to run PHP"

*"We wanted to have a .Net CMS,
but our users hate the UI"*

"Marketing wants this **CMS**, *but we can't develop our app on it"*

Decoupling Content Management



Monolithic approach

Content Management System

Database

File System

Decoupled approach

Web Editing Tool

Web Framework

Content Repository

Clean separation of concerns

Web Editing Tool

Tools that user needs to create, manipulate, and publish content

Web Framework

Rendering and serving content and handling business logic

Content Repository

Data storage and retrieval. Access controls, validations, full-text search

IKS

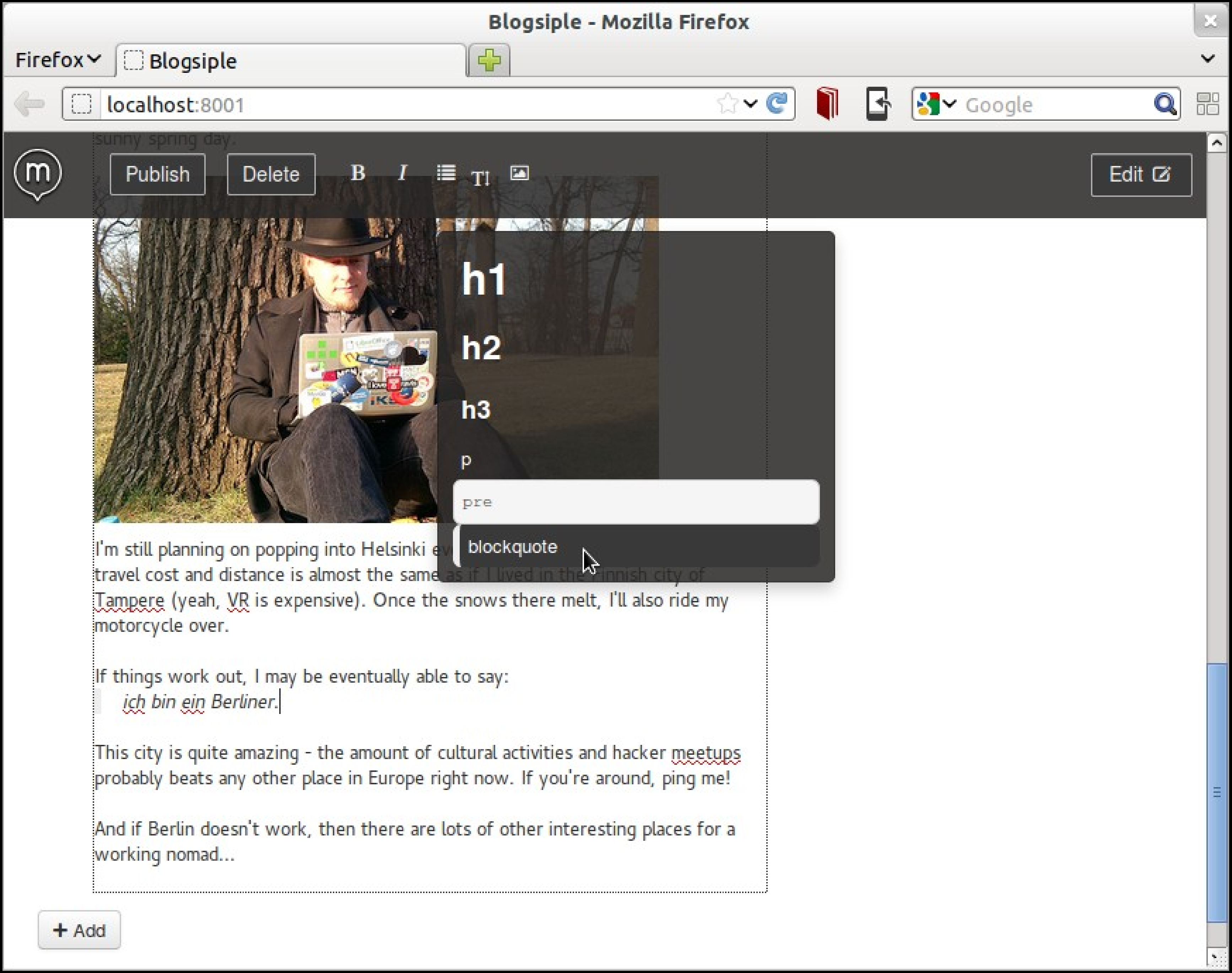
Semantic CMS Community

www.iks-project.eu

Create.js

Create.js is a comprehensive web editing interface for Content Management Systems. It is designed to provide a modern, fully browser-based HTML5 environment for managing content. Create can be adapted to work on almost any content management backend.

*“Build a CMS,
no forms allowed”*



sunny spring day.

I'm still planning on popping into Helsinki ev travel cost and distance is almost the same as if I lived in the Finnish city of Tampere (yeah, VR is expensive). Once the snows there melt, I'll also ride my motorcycle over.

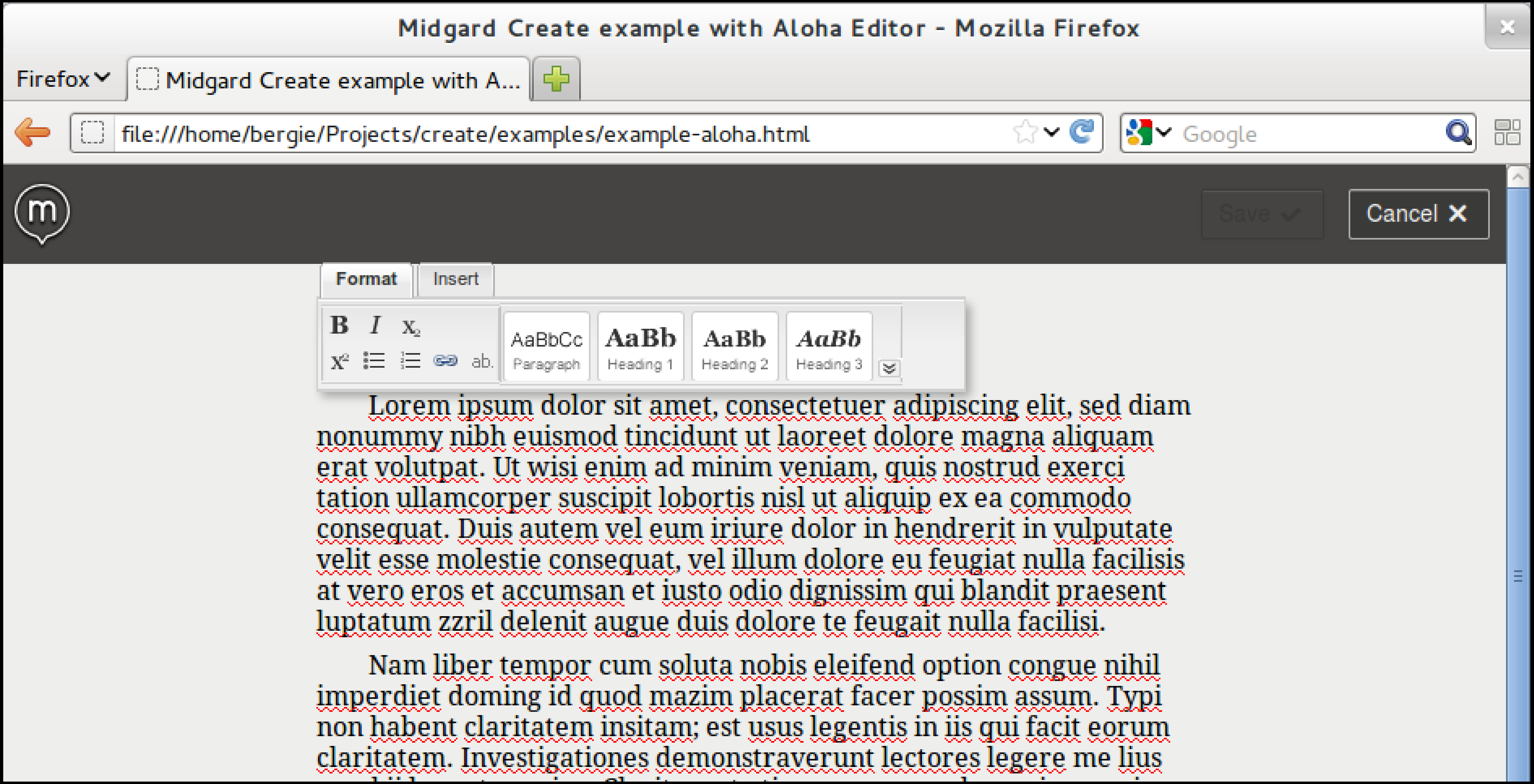
If things work out, I may be eventually able to say:

> *ich bin ein Berliner.*

This city is quite amazing - the amount of cultural activities and hacker meetups probably beats any other place in Europe right now. If you're around, ping me!

And if Berlin doesn't work, then there are lots of other interesting places for a working nomad...

http://hallojs.org & http://github.com/bergie/blogsiple

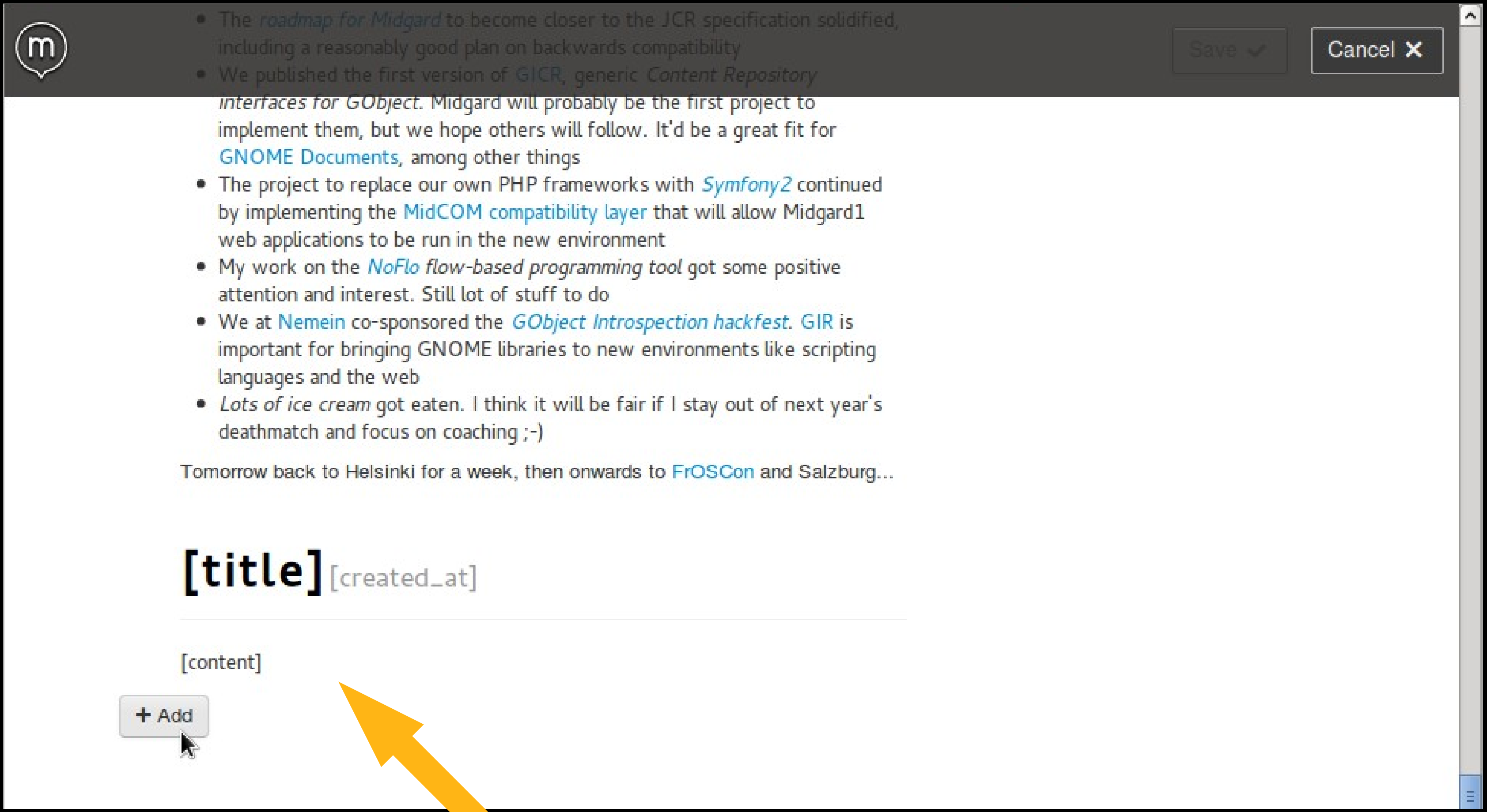

Midgard Create example with Aloha Editor - Mozilla Firefox
Firefox
Midgard Create example with A...
file:///home/bergie/Projects/create/examples/example-aloha.html
Google
m
Save
Cancel ✕
Format
Insert
B
I
ab.
AaBbCc
Paragraph
AaBb
Heading 1
AaBb
Heading 2
AaBb
Heading 3

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius

http://aloha-editor.org

Manage Collections



- The *roadmap for Midgard* to become closer to the JCR specification solidified, including a reasonably good plan on backwards compatibility
- We published the first version of GICR, generic *Content Repository interfaces for GObject*. Midgard will probably be the first project to implement them, but we hope others will follow. It'd be a great fit for GNOME Documents, among other things
- The project to replace our own PHP frameworks with *Symfony2* continued by implementing the MidCOM compatibility layer that will allow Midgard1 web applications to be run in the new environment
- My work on the *NoFlo flow-based programming tool* got some positive attention and interest. Still lot of stuff to do
- We at Nemein co-sponsored the *GObject Introspection hackfest*. GIR is important for bringing GNOME libraries to new environments like scripting languages and the web
- *Lots of ice cream* got eaten. I think it will be fair if I stay out of next year's deathmatch and focus on coaching ;-)

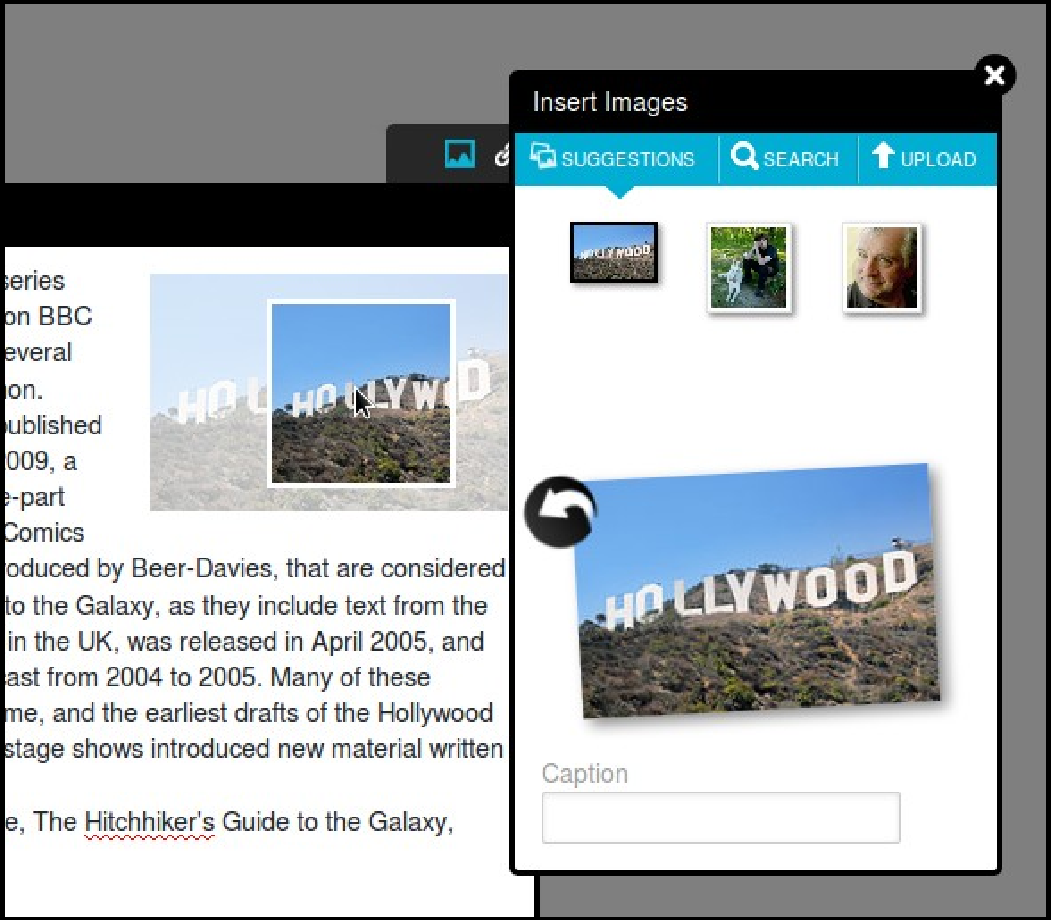
Tomorrow back to Helsinki for a week, then onwards to FrOSCon and Salzburg...

[title] [created_at]

[content]

Content can be added to any collections

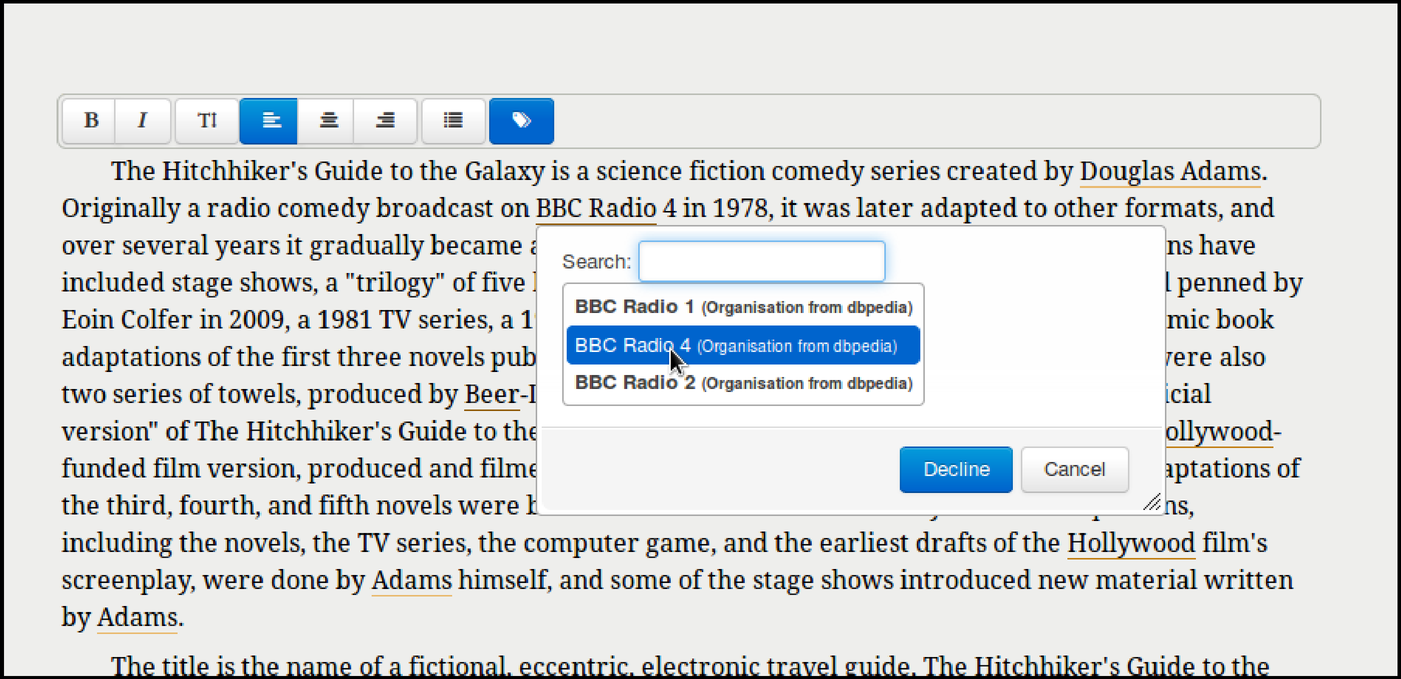
Image handling



eries
on BBC
everal
on.
ublished
009, a
-part
Comics
oduced by Beer-Davies, that are considered
to the Galaxy, as they include text from the
in the UK, was released in April 2005, and
ast from 2004 to 2005. Many of these
me, and the earliest drafts of the Hollywood
stage shows introduced new material written

e, The Hitchhiker's Guide to the Galaxy,

Tagging and annotations

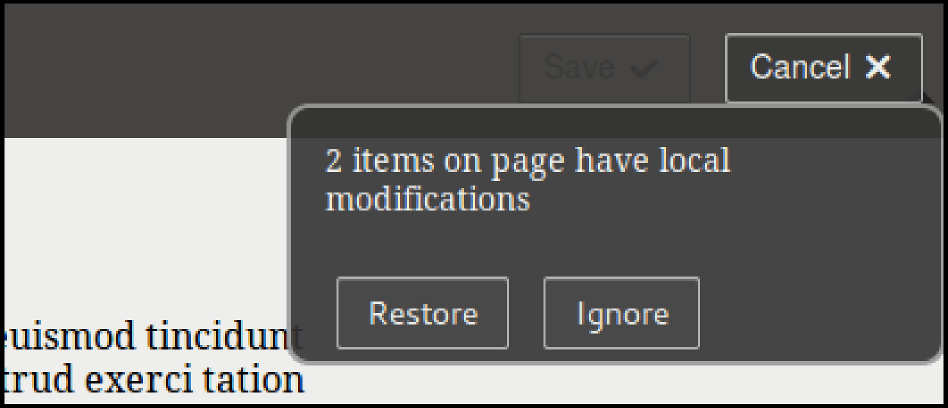

B
I
TI
Search:
BBC Radio 1 (Organisation from dbpedia)
BBC Radio 4 (Organisation from dbpedia)
BBC Radio 2 (Organisation from dbpedia)
Decline
Cancel

The Hitchhiker's Guide to the Galaxy is a science fiction comedy series created by Douglas Adams.
Originally a radio comedy broadcast on BBC Radio 4 in 1978, it was later adapted to other formats, and
over several years it gradually became a ns have
included stage shows, a "trilogy" of five l penned by
Eoin Colfer in 2009, a 1981 TV series, a 1 mic book
adaptations of the first three novels pub vere also
two series of towels, produced by Beer-I icial
version" of The Hitchhiker's Guide to the ollywood-
funded film version, produced and filme aptations of
the third, fourth, and fifth novels were b ns,
including the novels, the TV series, the computer game, and the earliest drafts of the Hollywood film's
screenplay, were done by Adams himself, and some of the stage shows introduced new material written
by Adams.

The title is the name of a fictional, eccentric, electronic travel guide. The Hitchhiker's Guide to the

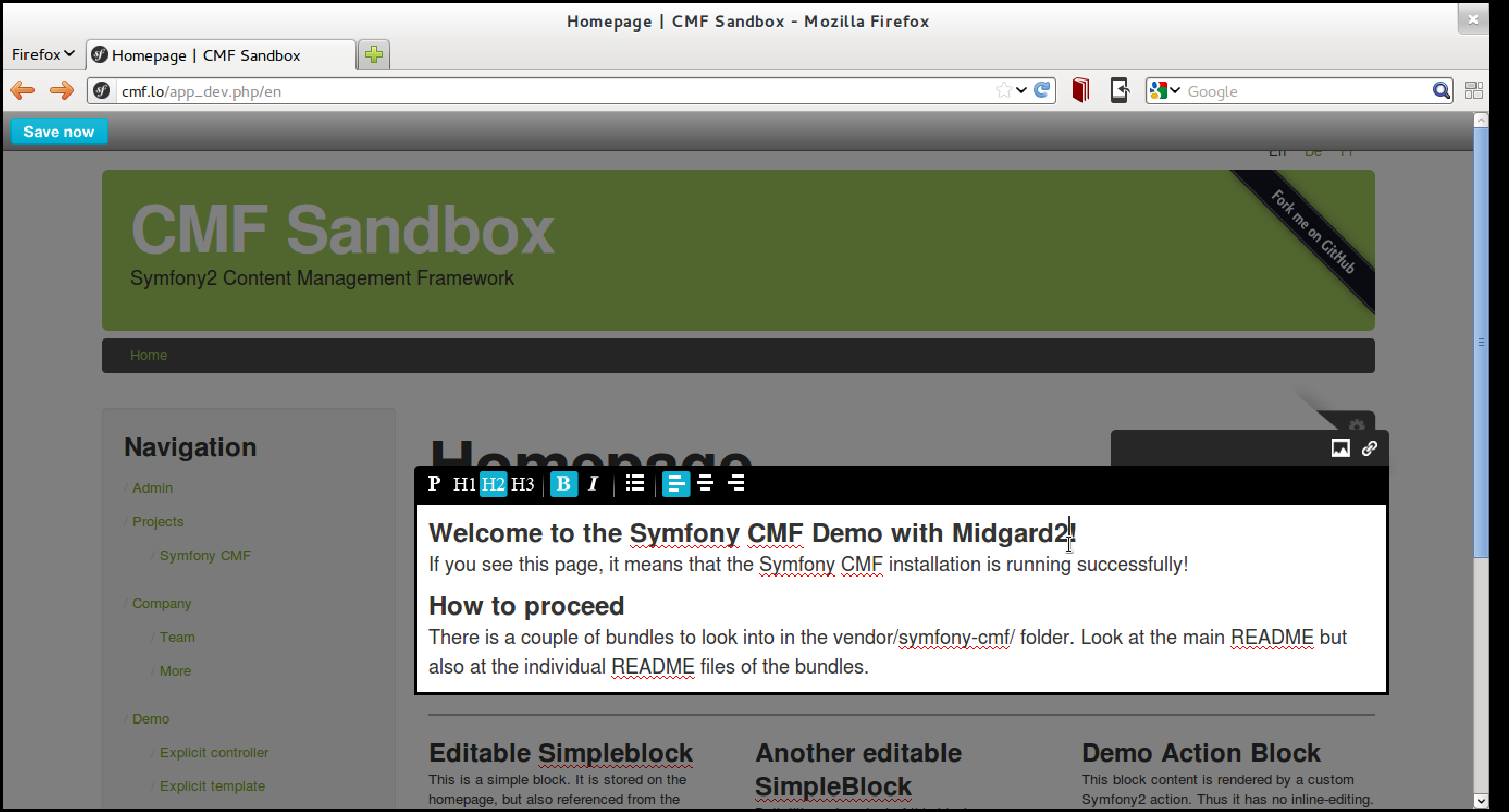
http://szabyg.github.com/annotate.js/

Never lose content



2 items on page have local modifications

Symfony CMF



CMF Sandbox

Symfony2 Content Management Framework

Navigation

- Admin
- Projects
  - Symfony CMF
- Company
  - Team
  - More
- Demo
  - Explicit controller
  - Explicit template

Homepage

Welcome to the Symfony CMF Demo with Midgard2!

If you see this page, it means that the Symfony CMF installation is running successfully!

How to proceed

There is a couple of bundles to look into in the vendor/symfony-cmf/ folder. Look at the main README but also at the individual README files of the bundles.

Editable Simpleblock

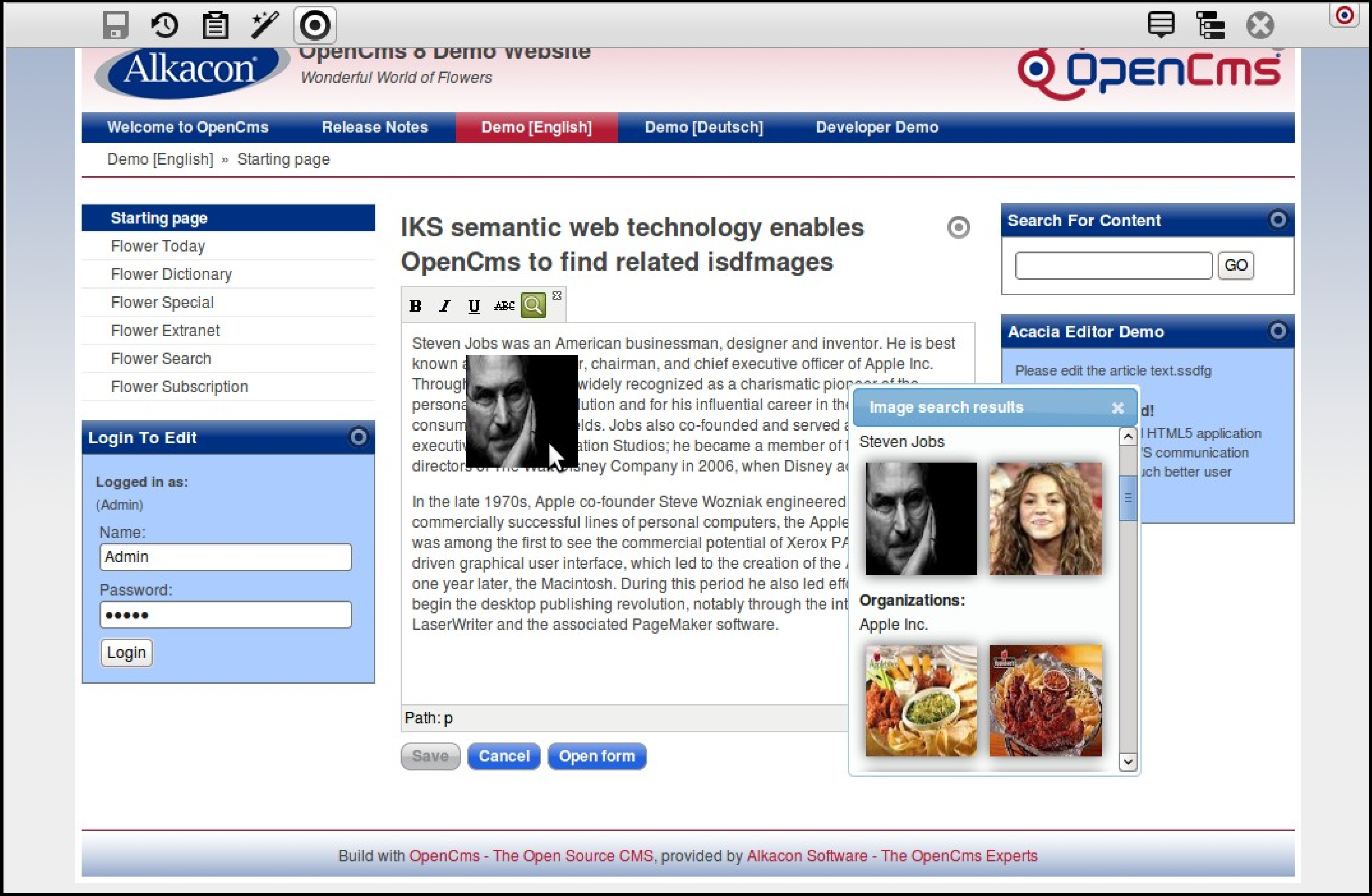
This is a simple block. It is stored on the homepage, but also referenced from the

Another editable SimpleBlock

Demo Action Block

This block content is rendered by a custom Symfony2 action. Thus it has no inline-editing.

OpenCms



Logged in as:
(Admin)

Name: Admin

Password: •••••

IKS semantic web technology enables OpenCms to find related isdfmages

Steven Jobs was an American businessman, designer and inventor. He is best
known a r, chairman, and chief executive officer of Apple Inc.
Throug widely recognized as a charismatic pioneer of the
persona lution and for his influential career in the
consum elds. Jobs also co-founded and served a
executiv ation Studios; he became a member of t
directors of The Walt Disney Company in 2006, when Disney ac

In the late 1970s, Apple co-founder Steve Wozniak engineered
commercially successful lines of personal computers, the Apple
was among the first to see the commercial potential of Xerox PA
driven graphical user interface, which led to the creation of the
one year later, the Macintosh. During this period he also led eff
begin the desktop publishing revolution, notably through the int
LaserWriter and the associated PageMaker software.

Please edit the article text.ssdfg

d!
HTML5 application
S communication
ch better user

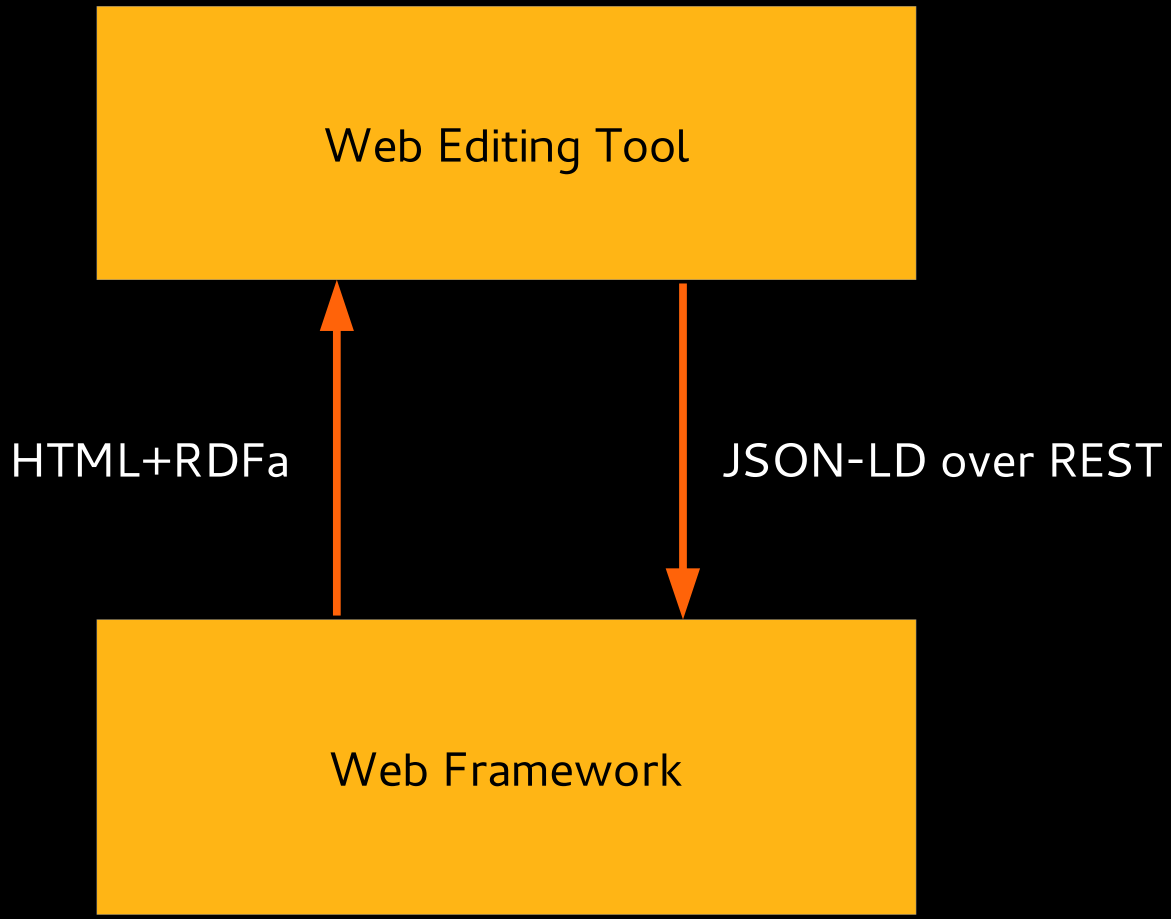
Steven Jobs

Organizations:

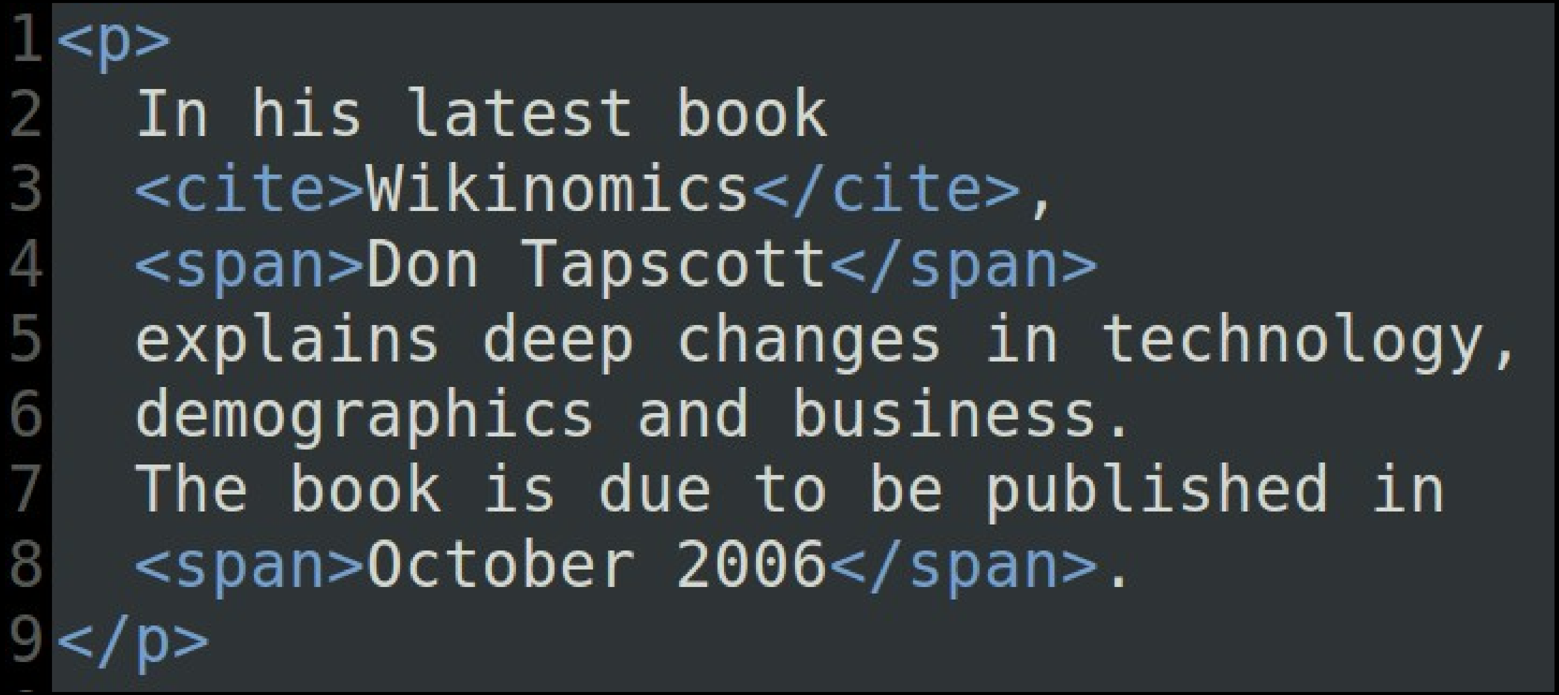
Apple Inc.

Build with OpenCms - The Open Source CMS, provided by Alkacon Software - The OpenCms Experts

Interfacing with Create.js

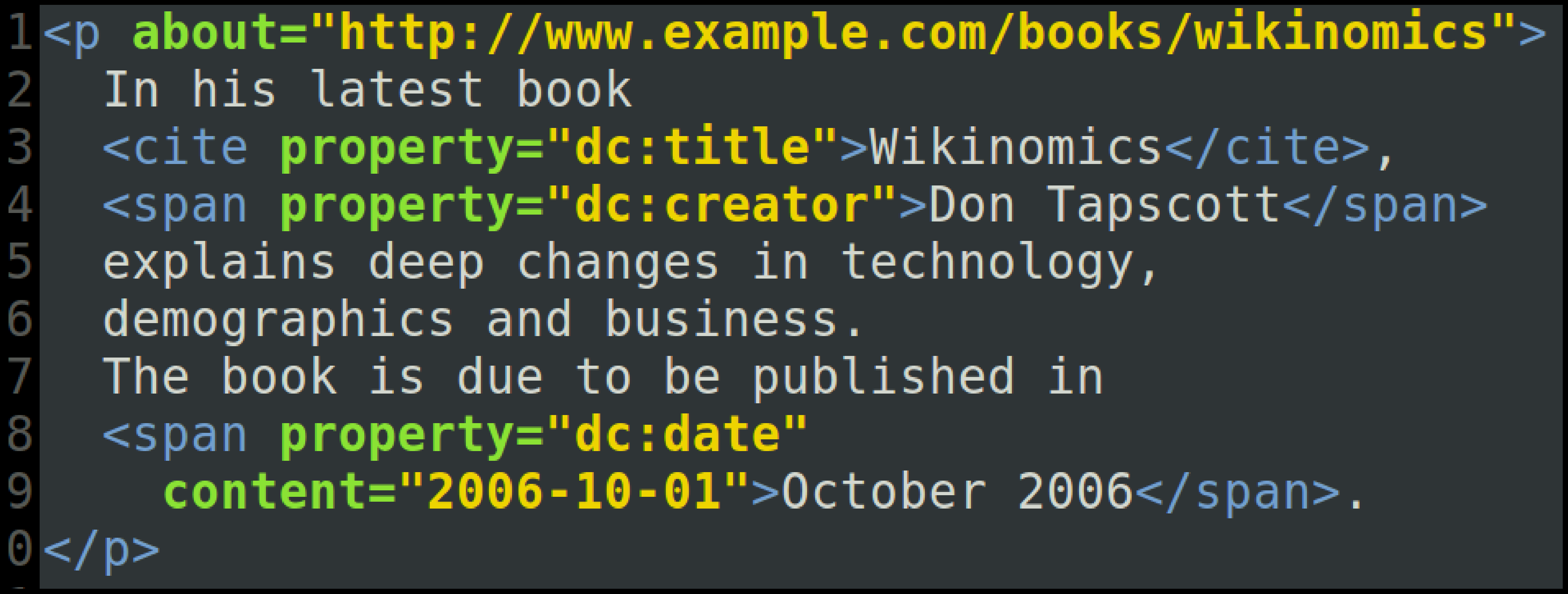


RDFa: Your Content, Explained



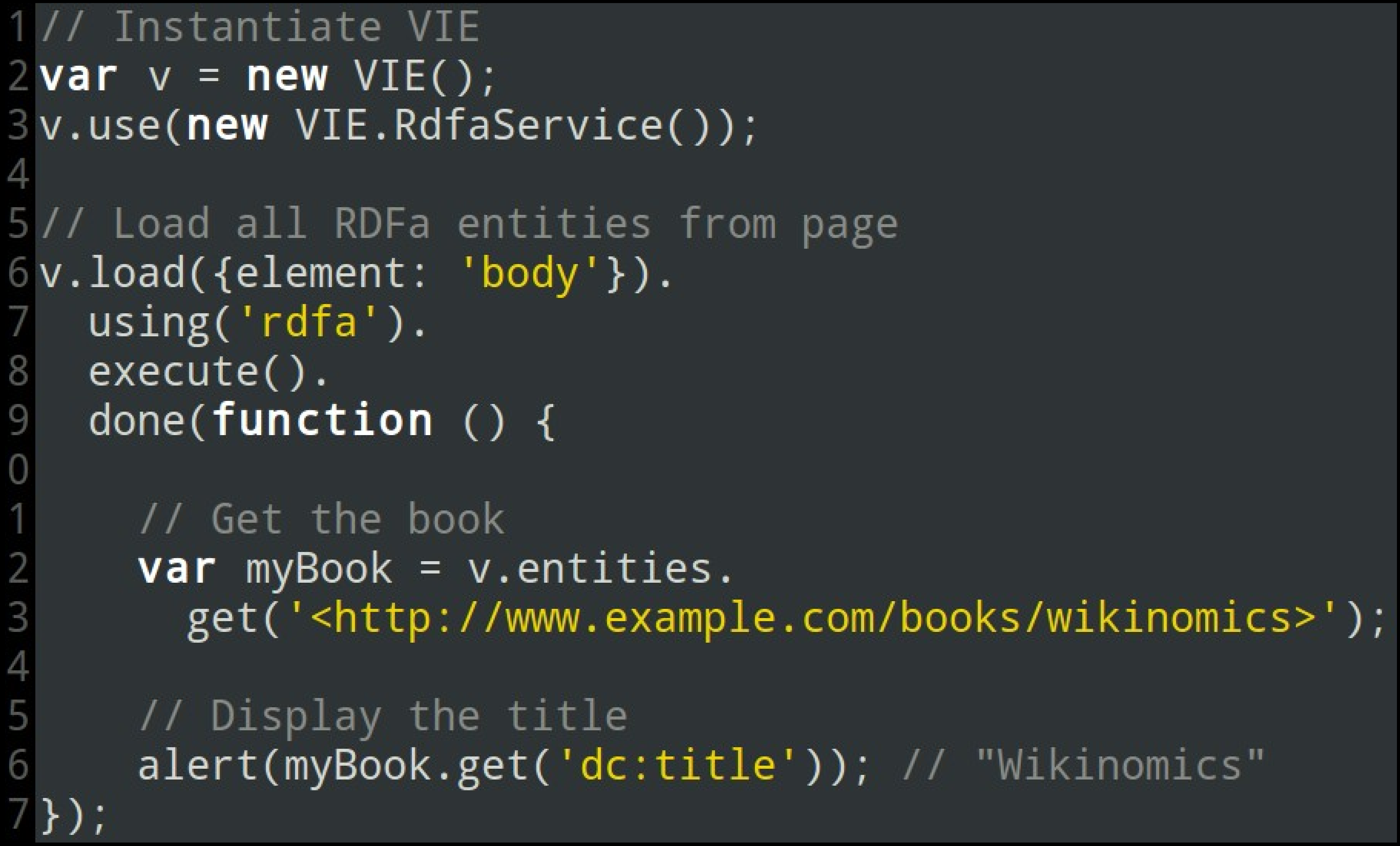
```
<p>
  In his latest book
  <cite>Wikinomics</cite>,
  <span>Don Tapscott</span>
  explains deep changes in technology,
  demographics and business.
  The book is due to be published in
  <span>October 2006</span>.
</p>
```

RDFa: Your Content, Explained



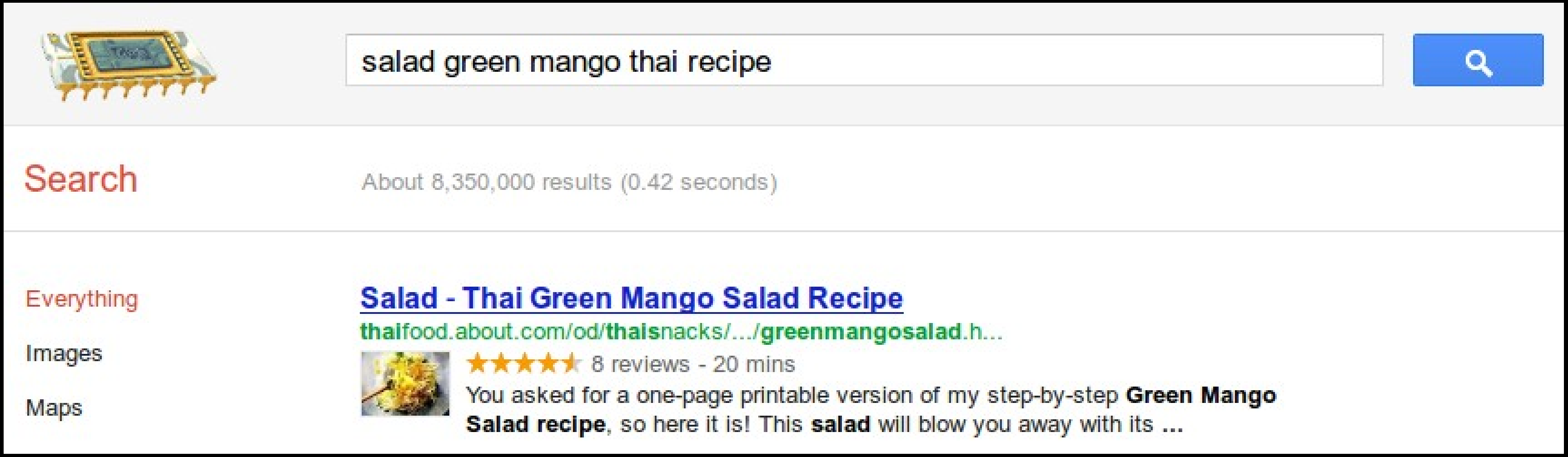
```
<p about="http://www.example.com/books/wikinomics">
  In his latest book
  <cite property="dc:title">Wikinomics</cite>,
  <span property="dc:creator">Don Tapscott</span>
  explains deep changes in technology,
  demographics and business.
  The book is due to be published in
  <span property="dc:date"
    content="2006-10-01">October 2006</span>.
</p>
```

Suddenly JavaScript can understand



```
// Instantiate VIE
var v = new VIE();
v.use(new VIE.RdfaService());

// Load all RDFa entities from page
v.load({element: 'body'}).
  using('rdfa').
  execute().
  done(function () {

    // Get the book
    var myBook = v.entities.
      get('<http://www.example.com/books/wikinomics>');

    // Display the title
    alert(myBook.get('dc:title')); // "Wikinomics"
});
```

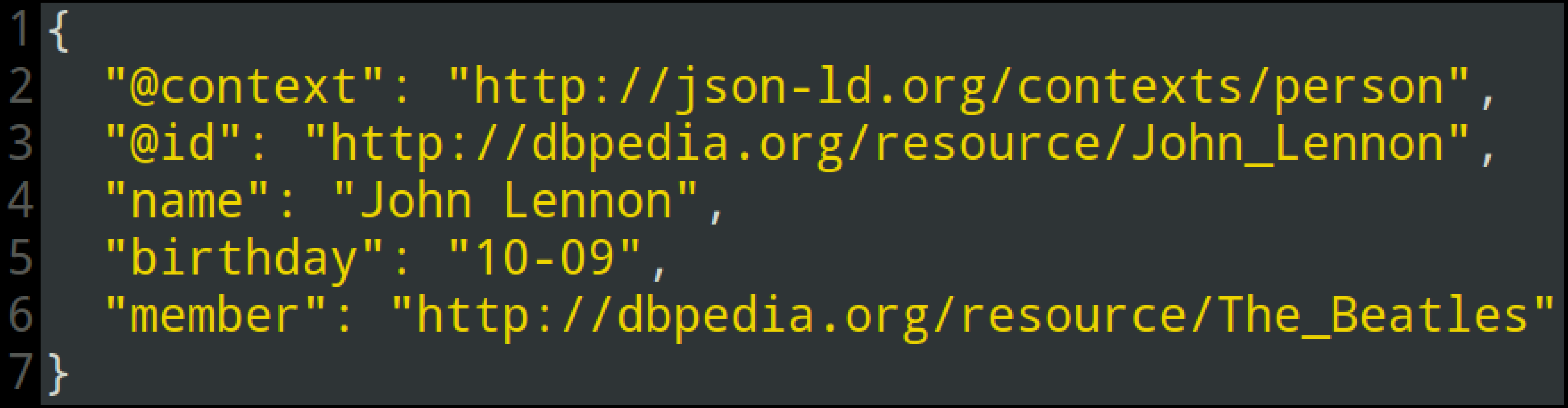
Bonus: SEO



About 8,350,000 results (0.42 seconds)

Salad - Thai Green Mango Salad Recipe
thaifood.about.com/od/thaisnacks/.../greenmangosalad.h...
★★★★★ 8 reviews - 20 mins
You asked for a one-page printable version of my step-by-step **Green Mango Salad recipe**, so here it is! This **salad** will blow you away with its **...**

JSON-LD

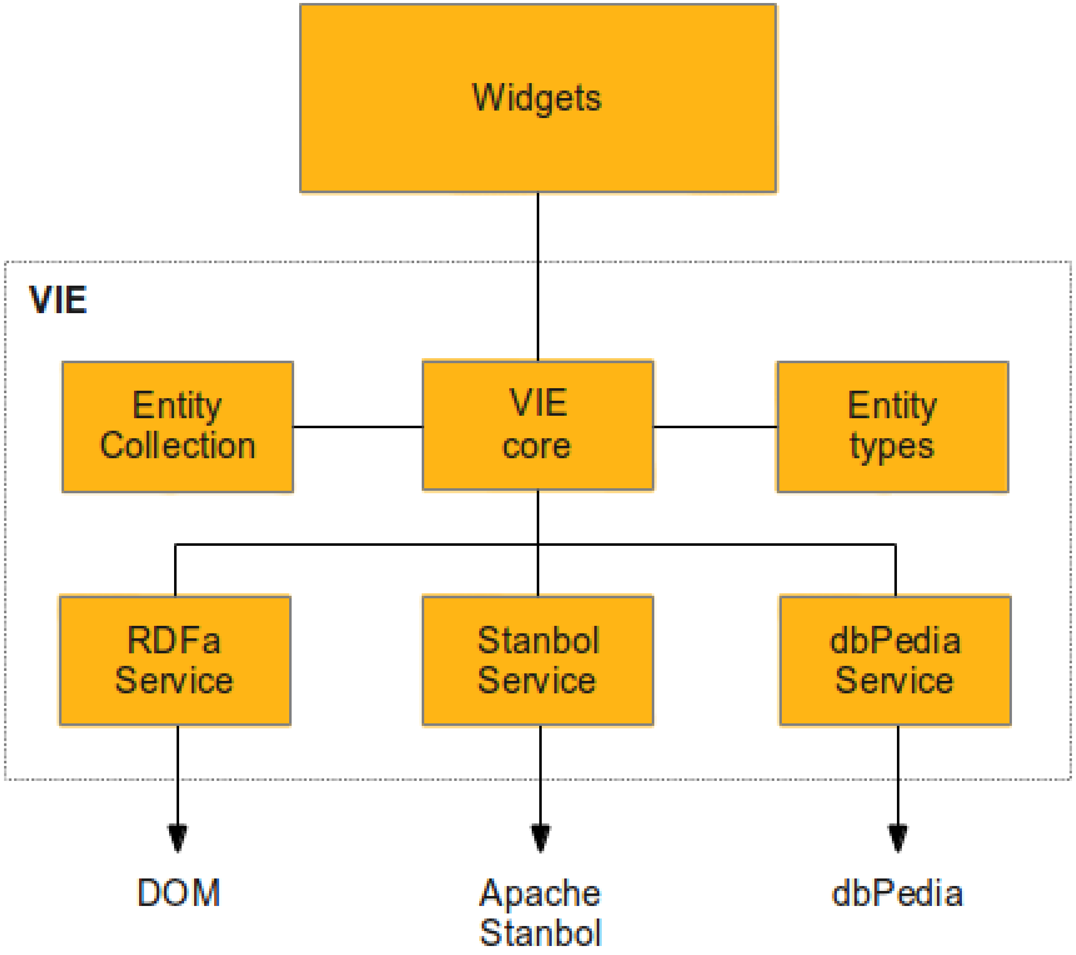


```
{
  "@context": "http://json-ld.org/contexts/person",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "birthday": "10-09",
  "member": "http://dbpedia.org/resource/The_Beatles"
}
```

"Our recommendation at this time is to make JSON-LD our primary supported web services data format. It is quite flexible, and supports the self-discovery capabilities that we want to support."

Drupal WSCCI Web Services Format Sprint Report
http://groups.drupal.org/node/237443

Framework for Semantic Interaction - viejs.org



Widgets

VIE

Entity Collection

VIE core

Entity types

RDFa Service

Stanbol Service

dbPedia Service

DOM

Apache Stanbol

dbPedia

Create.js and PHP

- CreatePHP

  Generate RDFa annotations from your content model

  https://github.com/flack/createphp

- Drupal Create

  Drupal 7 integration of Create.js

  http://drupal.org/project/create

- LiipVieBundle

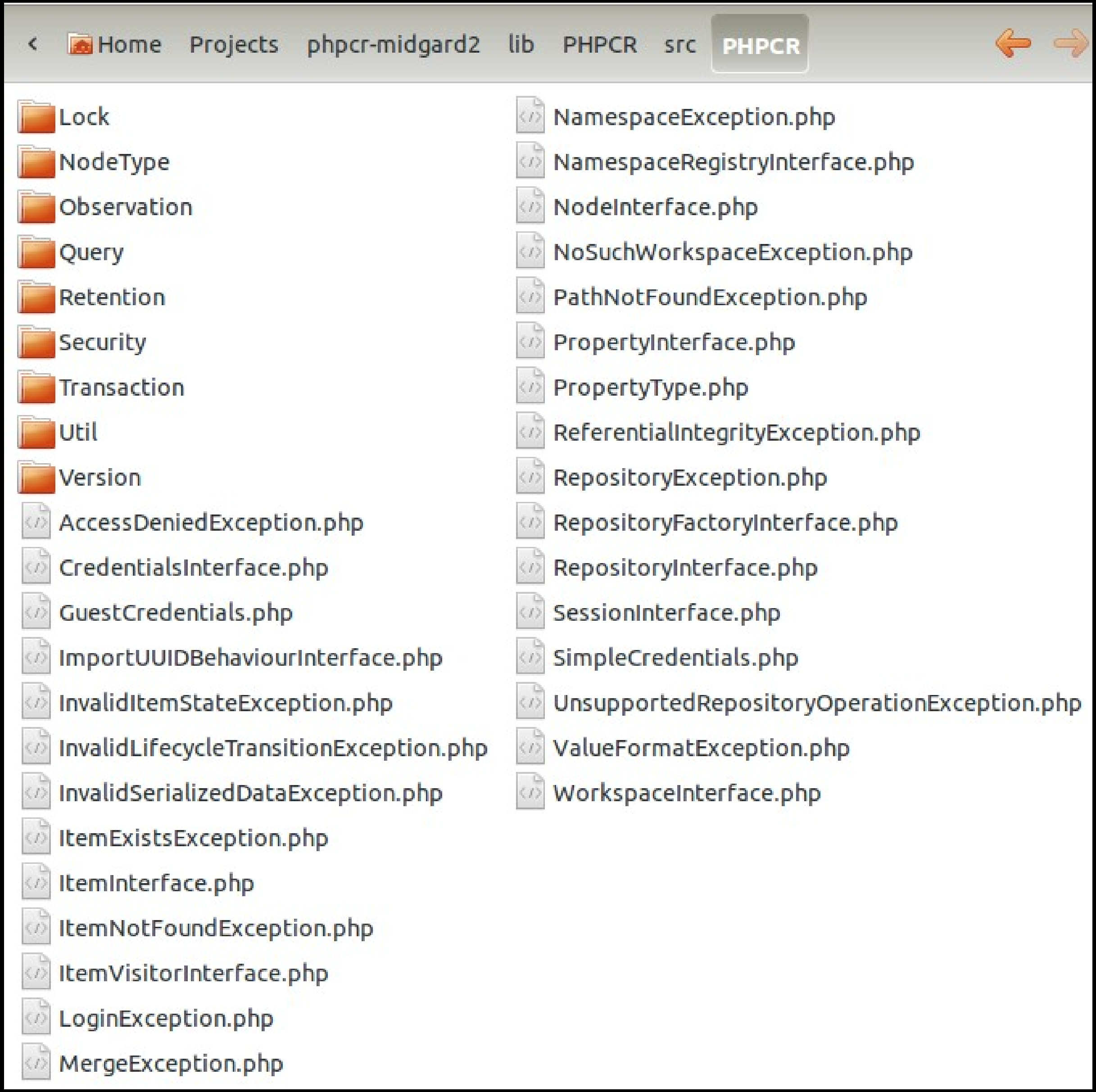
  Symfony2 integration of Create.js

  https://github.com/liip/LiipVieBundle

PHPCR

The PHP Content Repository is an adaption of the Java Content Repository (JCR) standard, an open API specification defined in JSR-283. The API defines how to handle hierarchical semi-structured data in a consistent way.

PHPCR is a collection of interfaces



Lock
NodeType
Observation
Query
Retention
Security
Transaction
Util
Version
AccessDeniedException.php
CredentialsInterface.php
GuestCredentials.php
ImportUUIDBehaviourInterface.php
InvalidItemStateException.php
InvalidLifecycleTransitionException.php
InvalidSerializedDataException.php
ItemExistsException.php
ItemInterface.php
ItemNotFoundException.php
ItemVisitorInterface.php
LoginException.php
MergeException.php
NamespaceException.php
NamespaceRegistryInterface.php
NodeInterface.php
NoSuchWorkspaceException.php
PathNotFoundException.php
PropertyInterface.php
PropertyType.php
ReferentialIntegrityException.php
RepositoryException.php
RepositoryFactoryInterface.php
RepositoryInterface.php
SessionInterface.php
SimpleCredentials.php
UnsupportedRepositoryOperationException.php
ValueFormatException.php
WorkspaceInterface.php

Same API, multiple repositories

- Jackalope
  - Apache Jackrabbit: Java-based repository server, accessible through WebDAV
  - Doctrine DBAL: pure-PHP repository library
- jackalope.github.com

- Midgard2
  - GObject-oriented repository library accessible through PHP extension
  - Python, JavaScript, Java, ...
- midgard-project.org/phpcr

"Developers write against a single API

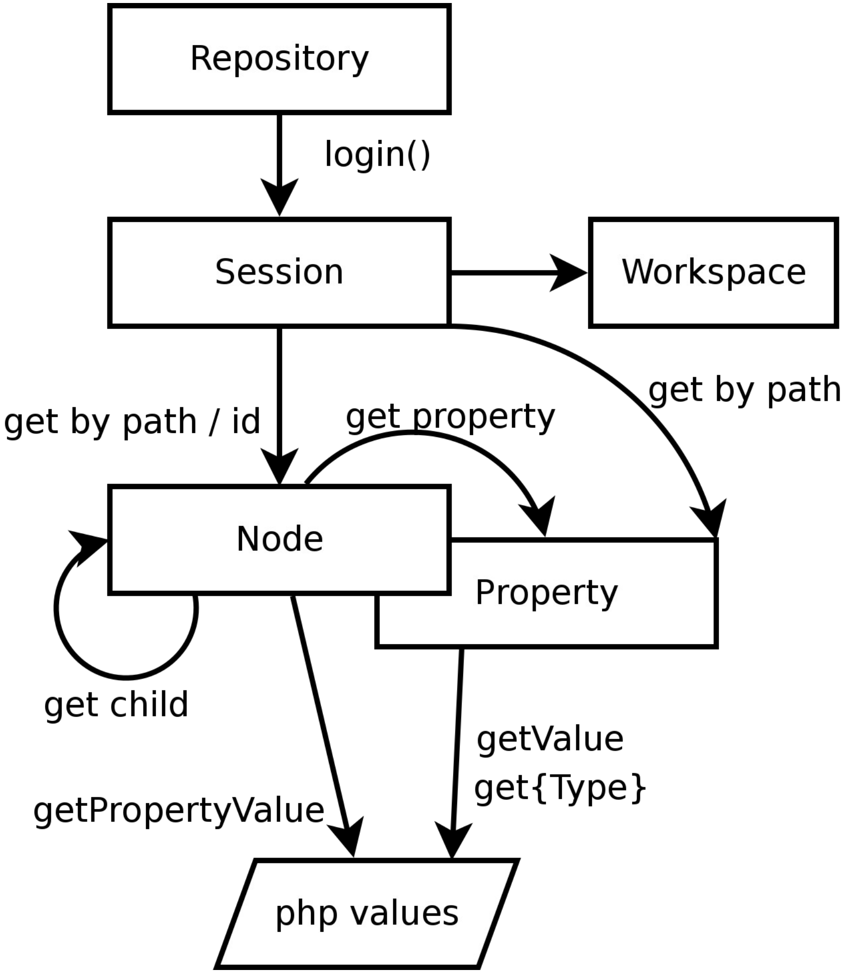
Implementations can be chosen based on deployment needs"

Repository capabilities

- Tree access
- Access by UUID
- Workspaces
- Versioning
- Multi-value properties
- Queries: SQL2, QOM
- XML import and export
- Permissions
- Capability discovery
- Observation
- Locking & Transactions

“Simple stuff should be easy, but complex stuff should still be possible”

PHPCR


Repository
login()
Session
Workspace
get by path / id
get property
get by path
Node
Property
get child
getValue
get{Type}
getPropertyValue
php values

In PHPCR, all content is stored in a tree of nodes

Nodes have a name, a type, and a set of properties

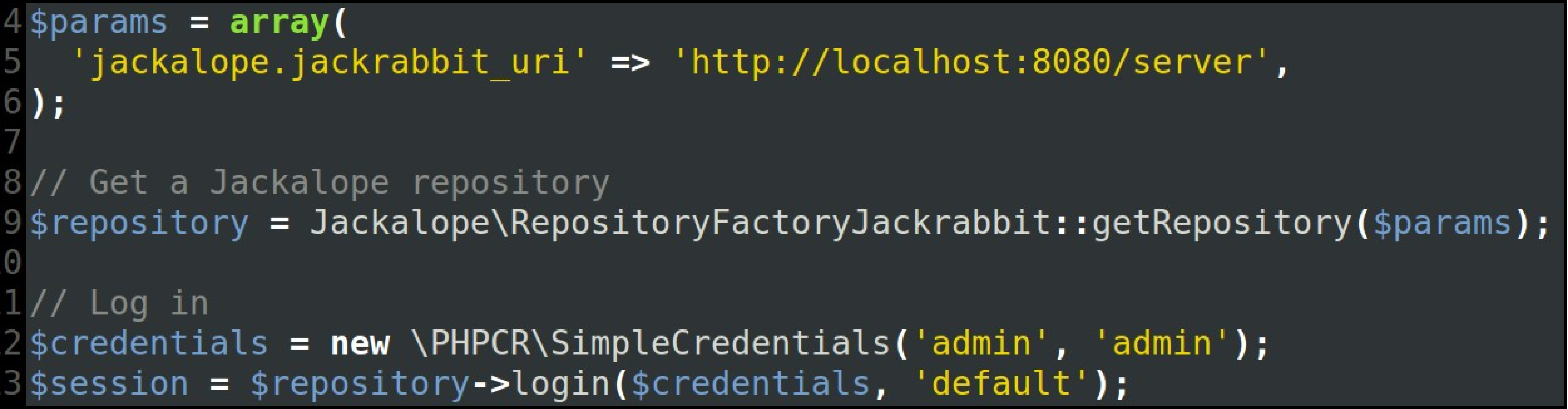
Nodes can also have child nodes

Connecting to a repository: Jackrabbit

- Start Jackrabbit:

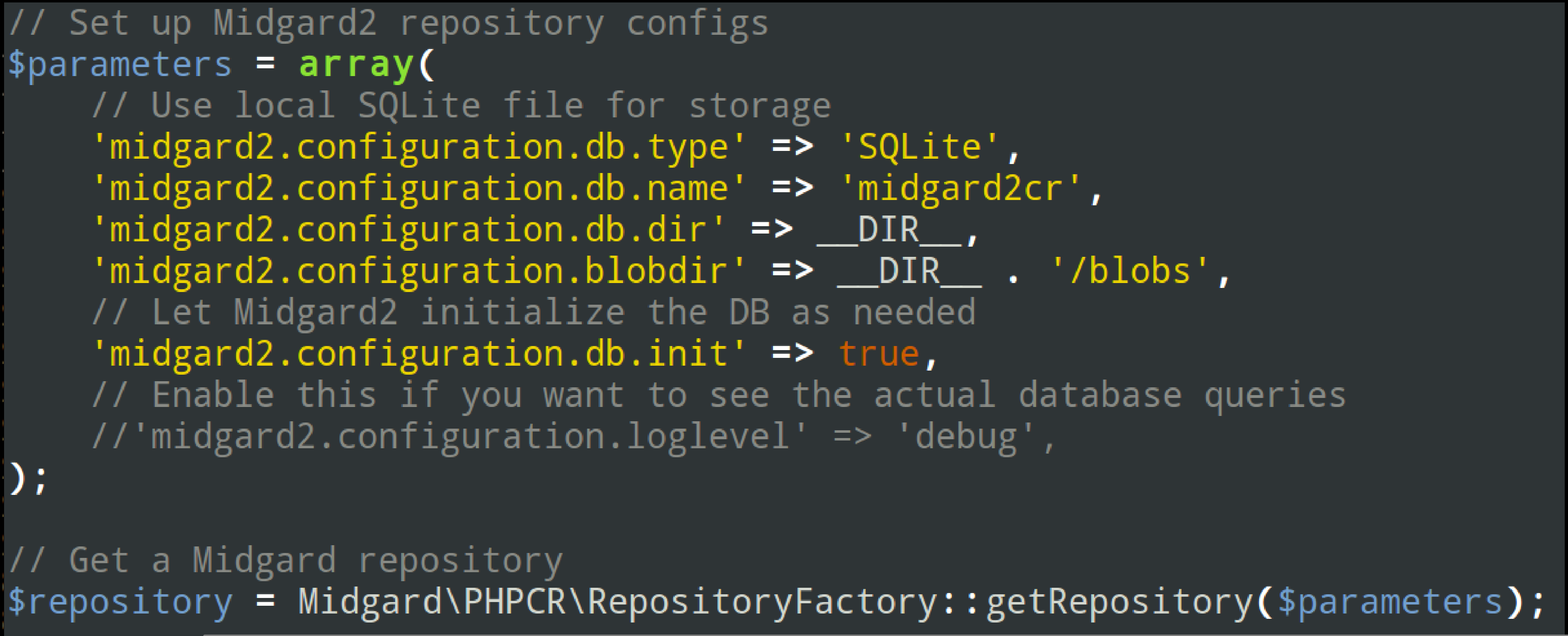
```
$ java -jar jackrabbit-standalone-2.4.1.jar
```

- Connect to repository:

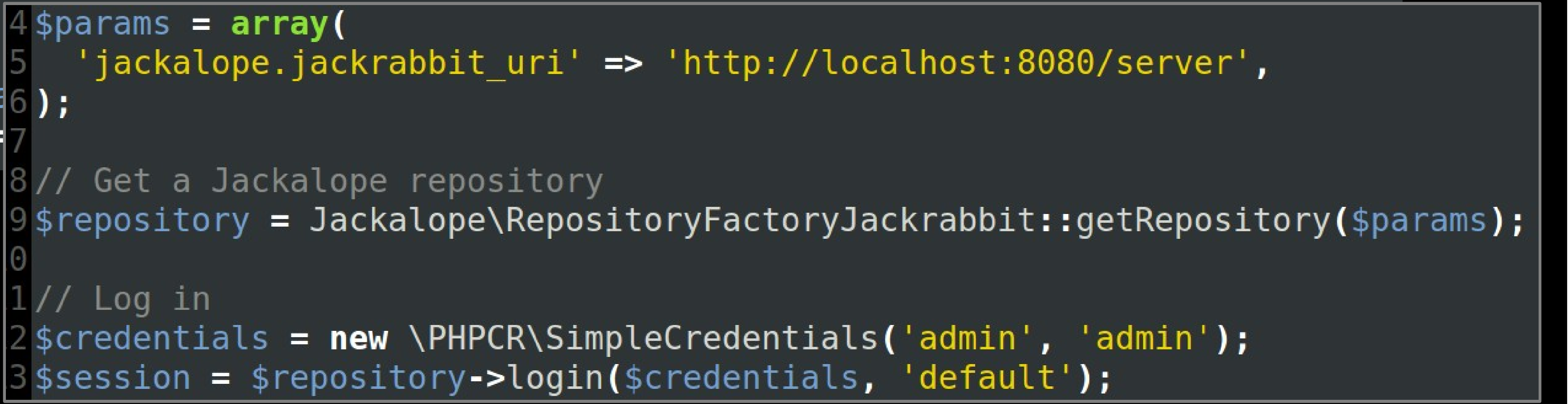


```php
$params = array(
  'jackalope.jackrabbit_uri' => 'http://localhost:8080/server',
);

// Get a Jackalope repository
$repository = Jackalope\RepositoryFactoryJackrabbit::getRepository($params);

// Log in
$credentials = new \PHPCR\SimpleCredentials('admin', 'admin');
$session = $repository->login($credentials, 'default');
```

This is the only repository-specific part

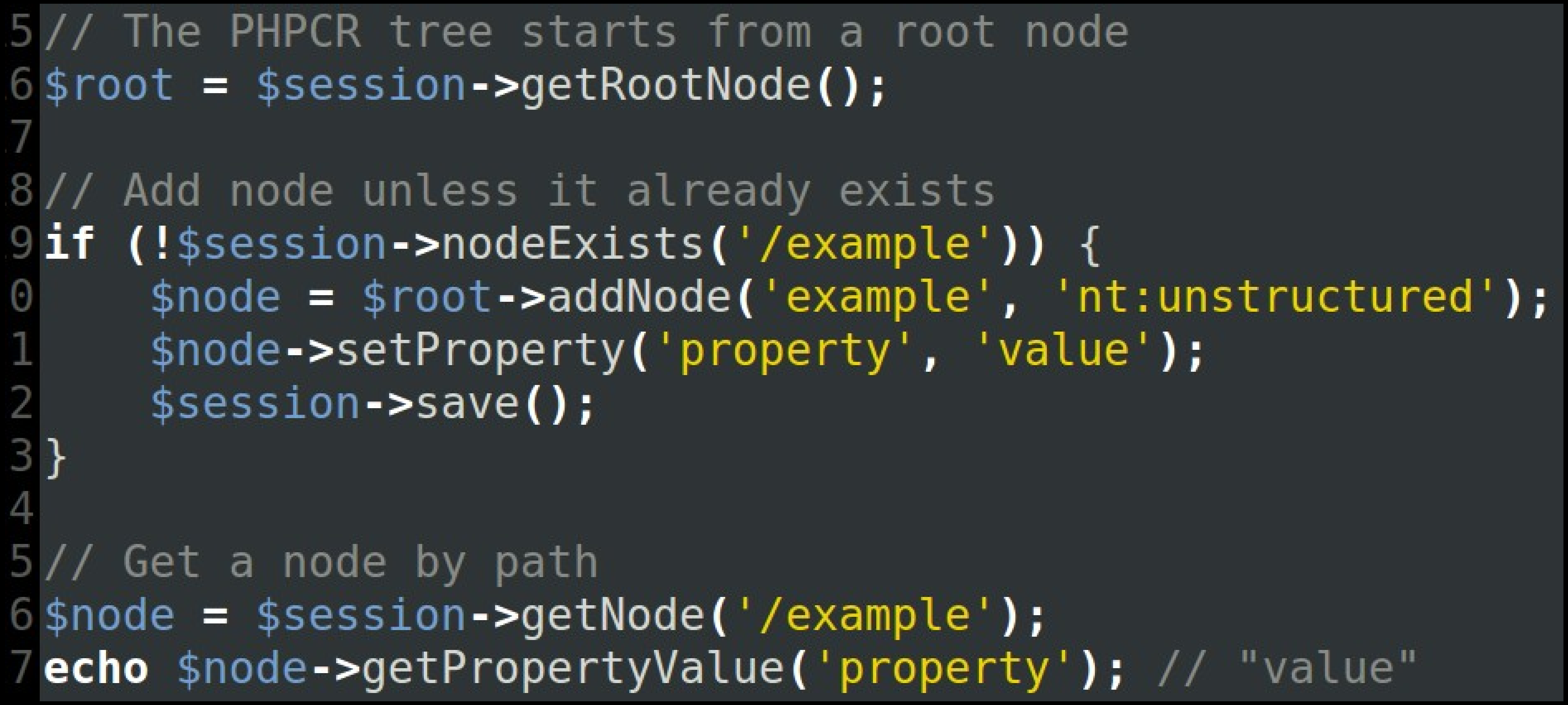


```
// Set up Midgard2 repository configs
$parameters = array(
    // Use local SQLite file for storage
    'midgard2.configuration.db.type' => 'SQLite',
    'midgard2.configuration.db.name' => 'midgard2cr',
    'midgard2.configuration.db.dir' => __DIR__,
    'midgard2.configuration.blobdir' => __DIR__ . '/blobs',
    // Let Midgard2 initialize the DB as needed
    'midgard2.configuration.db.init' => true,
    // Enable this if you want to see the actual database queries
    //'midgard2.configuration.loglevel' => 'debug',
);

// Get a Midgard repository
$repository = Midgard\PHPCR\RepositoryFactory::getRepository($parameters);

// Log in
$credentia
$session =
```



```
$params = array(
  'jackalope.jackrabbit_uri' => 'http://localhost:8080/server',
);

// Get a Jackalope repository
$repository = Jackalope\RepositoryFactoryJackrabbit::getRepository($params);

// Log in
$credentials = new \PHPCR\SimpleCredentials('admin', 'admin');
$session = $repository->login($credentials, 'default');
```

Working with the tree



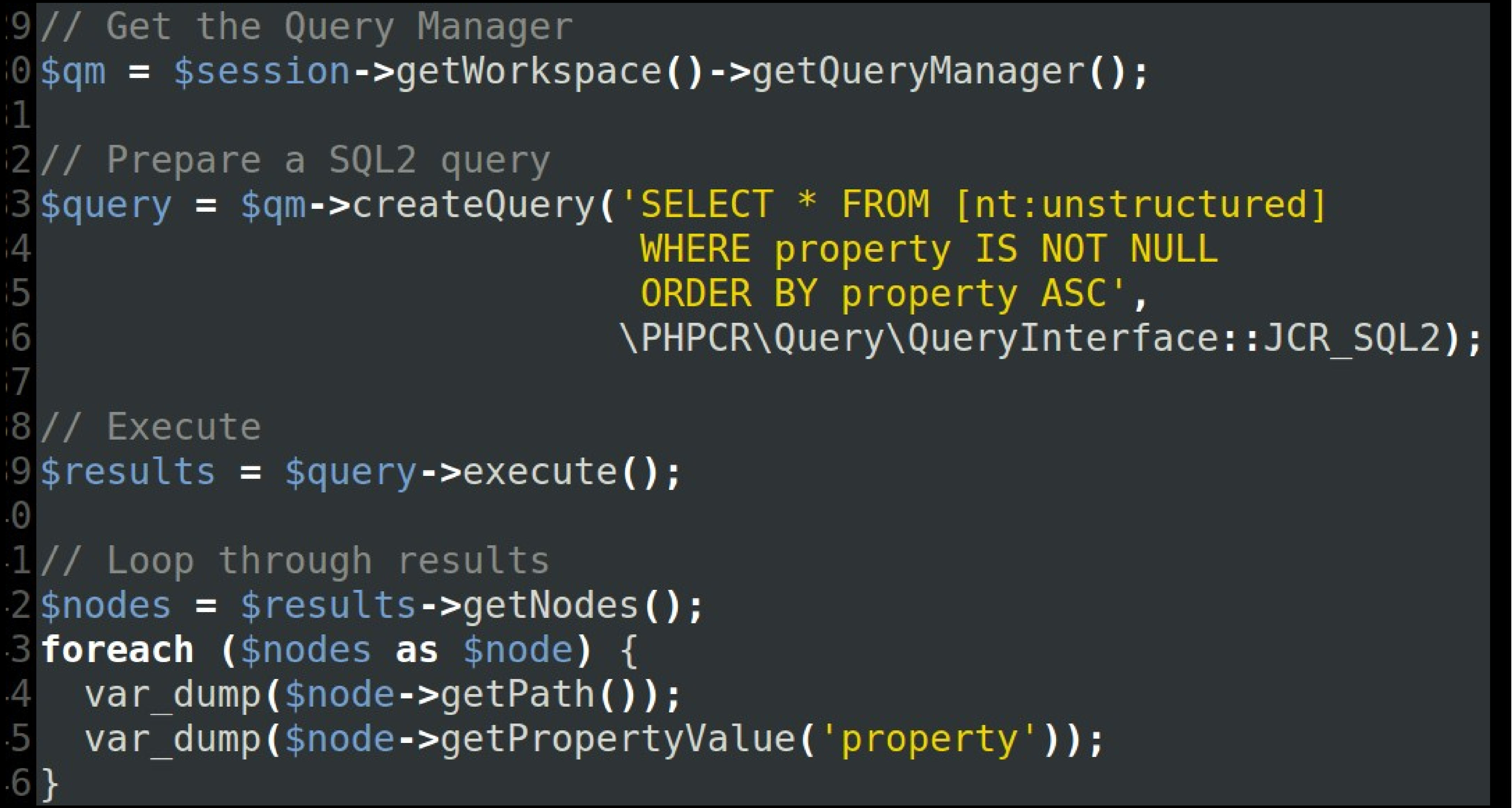
```php
// The PHPCR tree starts from a root node
$root = $session->getRootNode();

// Add node unless it already exists
if (!$session->nodeExists('/example')) {
    $node = $root->addNode('example', 'nt:unstructured');
    $node->setProperty('property', 'value');
    $session->save();
}

// Get a node by path
$node = $session->getNode('/example');
echo $node->getPropertyValue('property'); // "value"
```

About Node types

- Node types determine what properties and what kind of child nodes a node can have
- Some built-in types: `nt:file, nt:folder, nt:resource, nt:unstructured`
- Node types can be amended with "mixin" types, like `mix:lastModified, mix:language`
- Most repositories allow you to define your own types
- Recommendation:

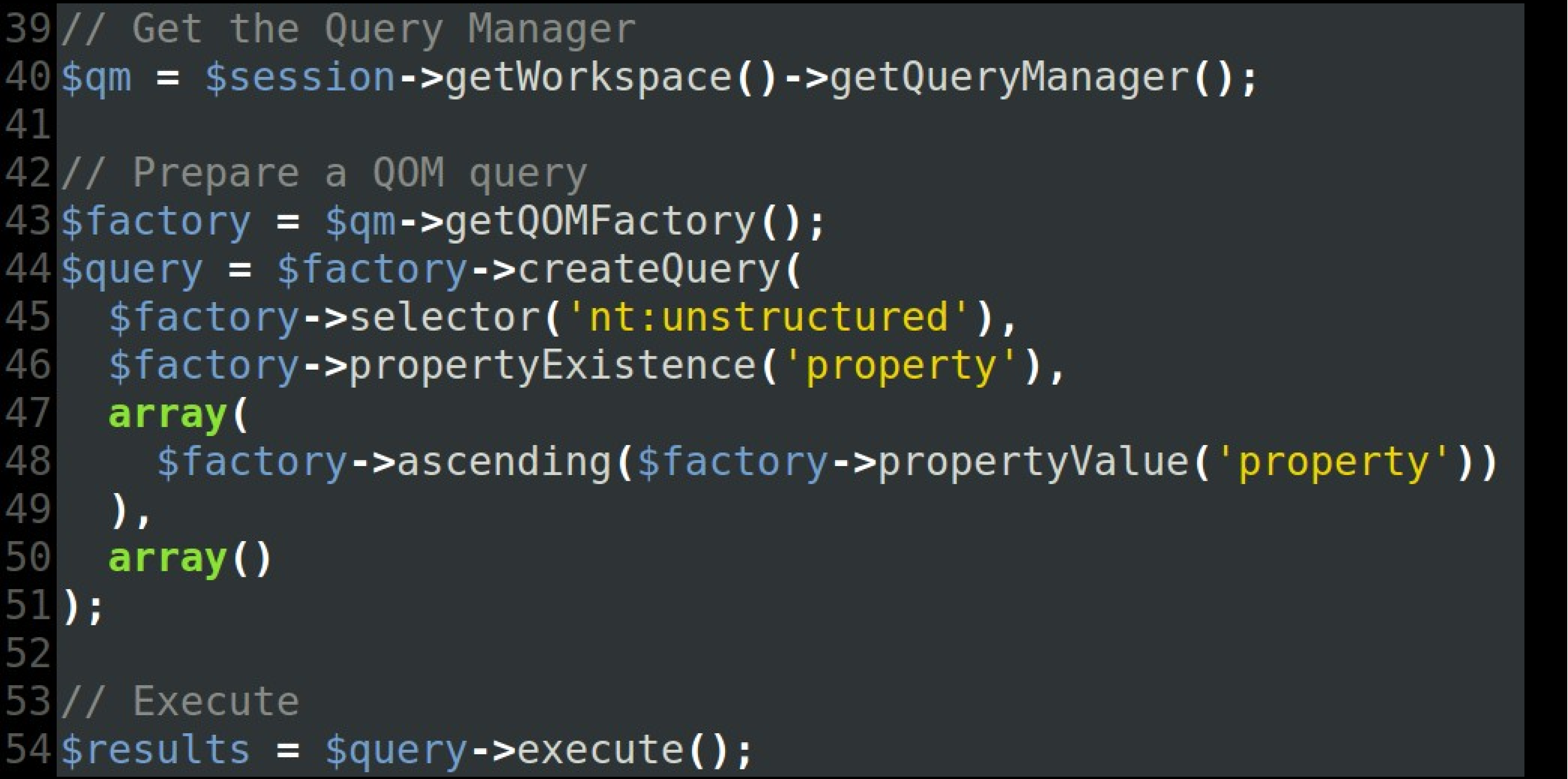
  Start with `nt:unstructured`, add more strict definitions when you know your data model

Queries: SQL2



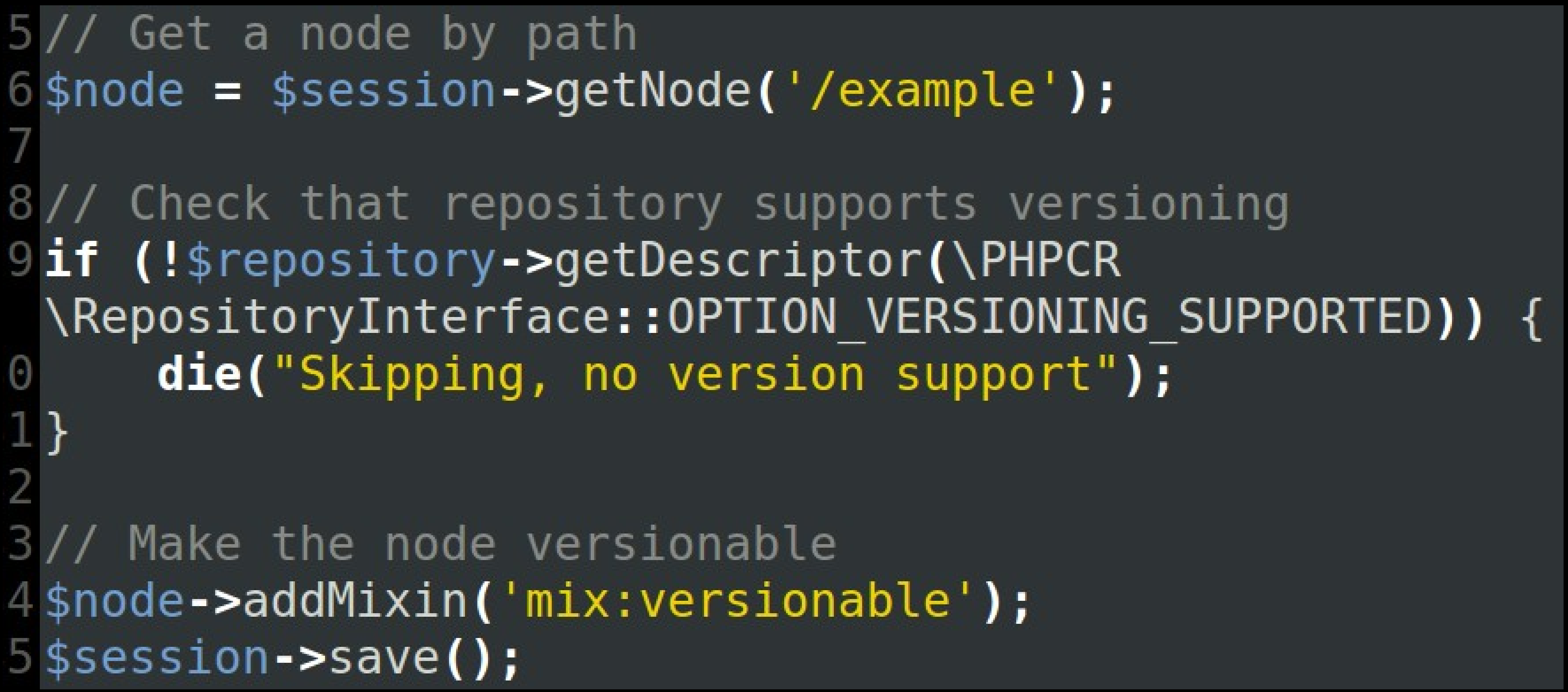
```
// Get the Query Manager
$qm = $session->getWorkspace()->getQueryManager();

// Prepare a SQL2 query
$query = $qm->createQuery('SELECT * FROM [nt:unstructured]
                           WHERE property IS NOT NULL
                           ORDER BY property ASC',
                          \PHPCR\Query\QueryInterface::JCR_SQL2);

// Execute
$results = $query->execute();

// Loop through results
$nodes = $results->getNodes();
foreach ($nodes as $node) {
  var_dump($node->getPath());
  var_dump($node->getPropertyValue('property'));
}
```

Queries: Query Object Model



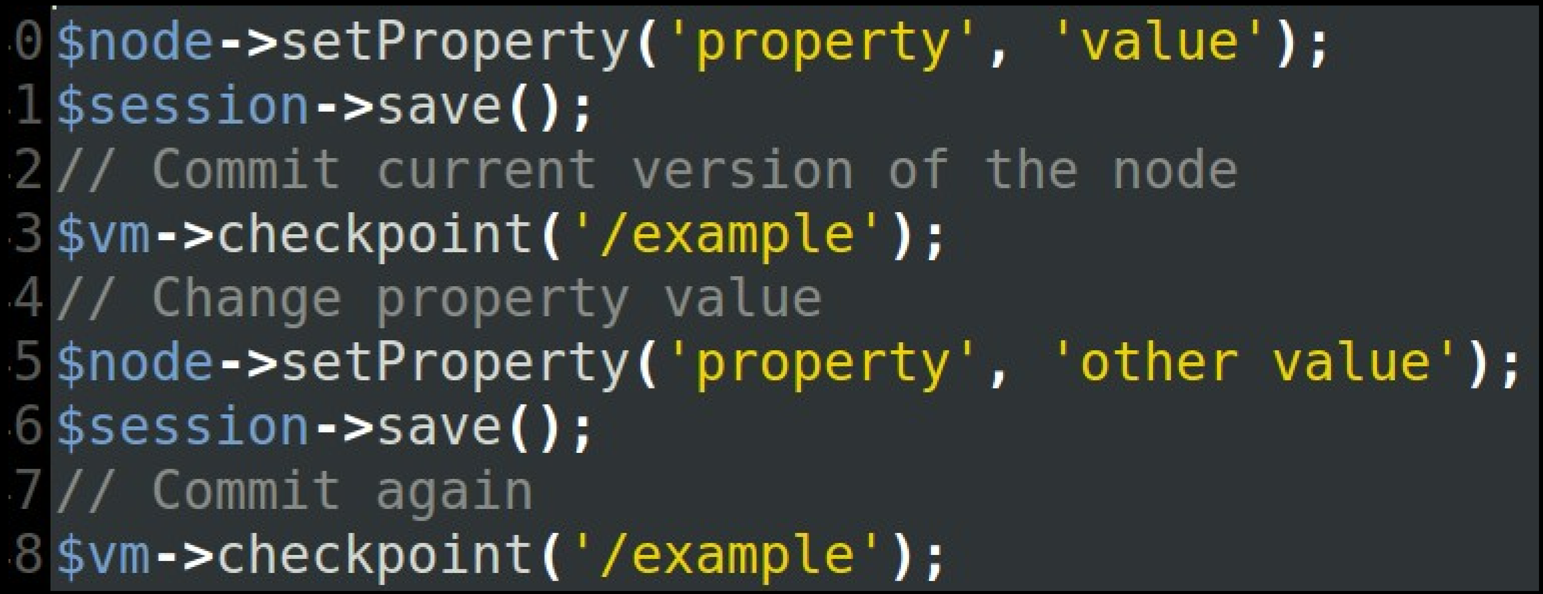
```php
// Get the Query Manager
$qm = $session->getWorkspace()->getQueryManager();

// Prepare a QOM query
$factory = $qm->getQOMFactory();
$query = $factory->createQuery(
  $factory->selector('nt:unstructured'),
  $factory->propertyExistence('property'),
  array(
    $factory->ascending($factory->propertyValue('property'))
  ),
  array()
);

// Execute
$results = $query->execute();
```

Versioning: enable



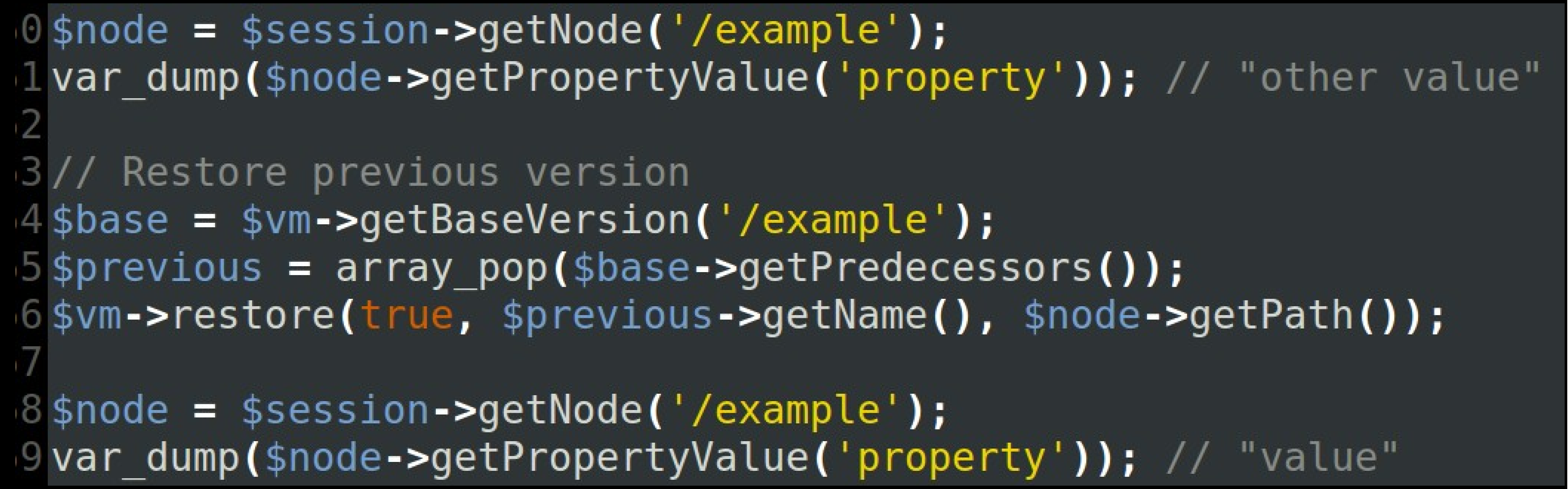
```php
// Get a node by path
$node = $session->getNode('/example');

// Check that repository supports versioning
if (!$repository->getDescriptor(\PHPCR\RepositoryInterface::OPTION_VERSIONING_SUPPORTED)) {
    die("Skipping, no version support");
}

// Make the node versionable
$node->addMixin('mix:versionable');
$session->save();
```

Versioning: commit



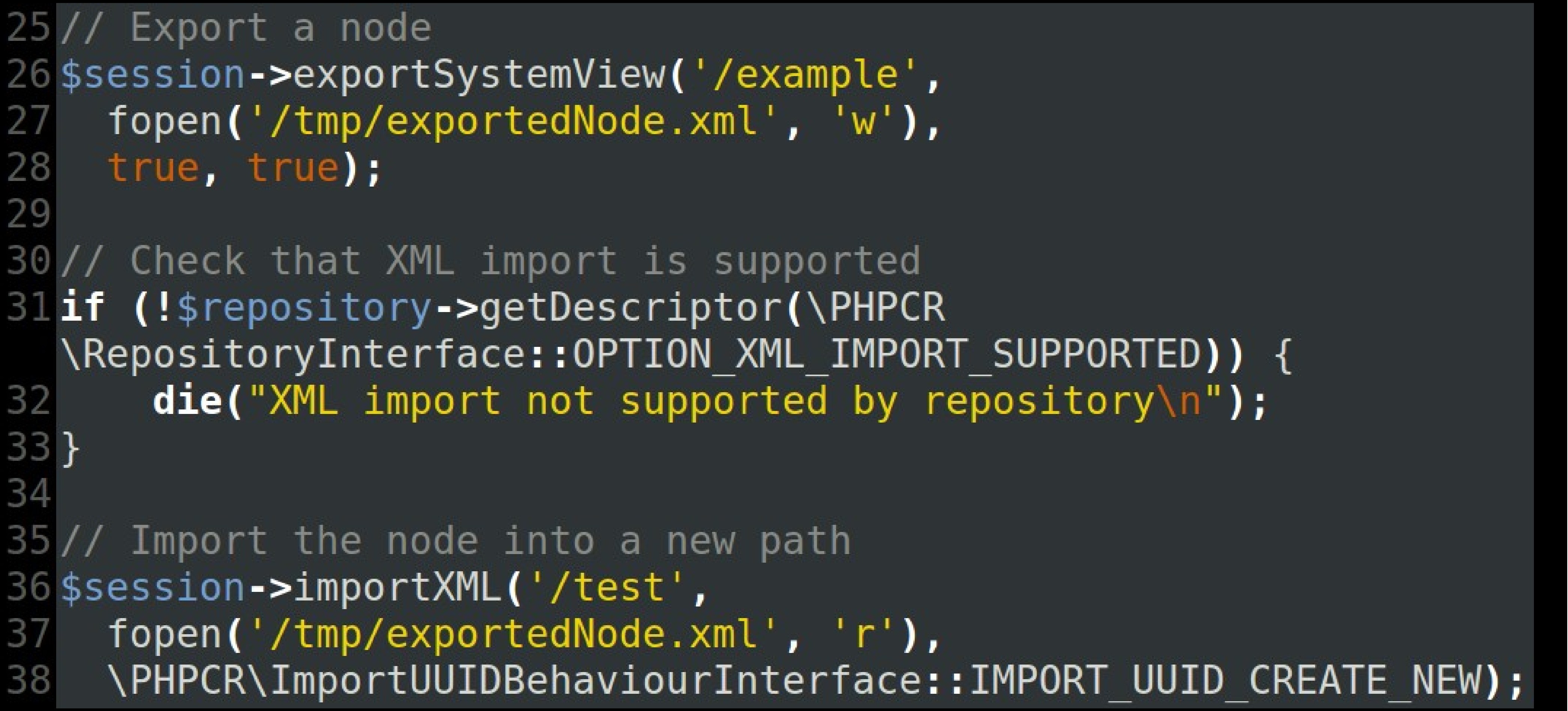
```php
$node->setProperty('property', 'value');
$session->save();
// Commit current version of the node
$vm->checkpoint('/example');
// Change property value
$node->setProperty('property', 'other value');
$session->save();
// Commit again
$vm->checkpoint('/example');
```

Versioning: restore



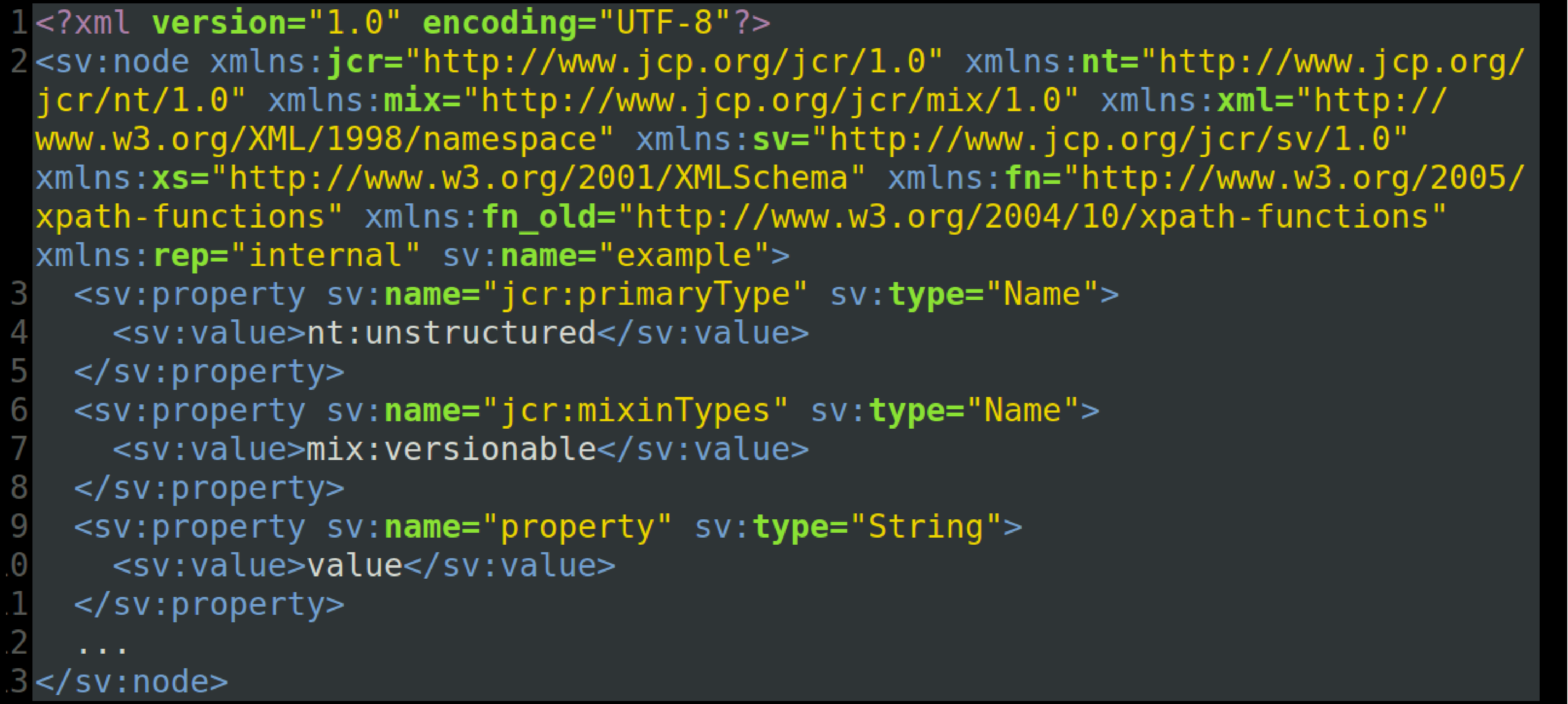
```
$node = $session->getNode('/example');
var_dump($node->getPropertyValue('property')); // "other value"

// Restore previous version
$base = $vm->getBaseVersion('/example');
$previous = array_pop($base->getPredecessors());
$vm->restore(true, $previous->getName(), $node->getPath());

$node = $session->getNode('/example');
var_dump($node->getPropertyValue('property')); // "value"
```

Export and import



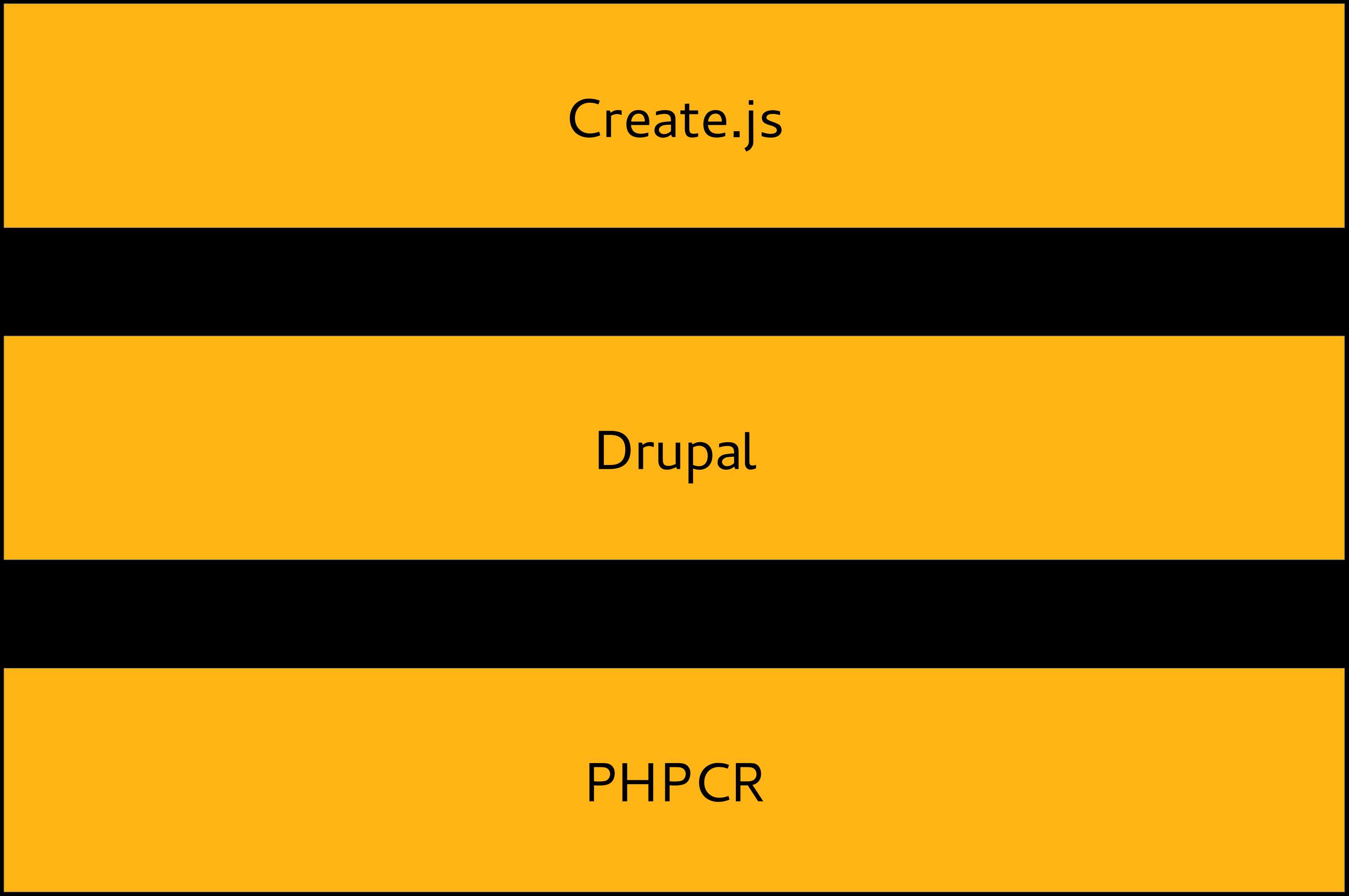
```php
// Export a node
$session->exportSystemView('/example',
  fopen('/tmp/exportedNode.xml', 'w'),
  true, true);

// Check that XML import is supported
if (!$repository->getDescriptor(\PHPCR\RepositoryInterface::OPTION_XML_IMPORT_SUPPORTED)) {
    die("XML import not supported by repository\n");
}

// Import the node into a new path
$session->importXML('/test',
  fopen('/tmp/exportedNode.xml', 'r'),
  \PHPCR\ImportUUIDBehaviourInterface::IMPORT_UUID_CREATE_NEW);
```

Export and import



```xml
<?xml version="1.0" encoding="UTF-8"?>
<sv:node xmlns:jcr="http://www.jcp.org/jcr/1.0" xmlns:nt="http://www.jcp.org/jcr/nt/1.0" xmlns:mix="http://www.jcp.org/jcr/mix/1.0" xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:sv="http://www.jcp.org/jcr/sv/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fn="http://www.w3.org/2005/xpath-functions" xmlns:fn_old="http://www.w3.org/2004/10/xpath-functions" xmlns:rep="internal" sv:name="example">
  <sv:property sv:name="jcr:primaryType" sv:type="Name">
    <sv:value>nt:unstructured</sv:value>
  </sv:property>
  <sv:property sv:name="jcr:mixinTypes" sv:type="Name">
    <sv:value>mix:versionable</sv:value>
  </sv:property>
  <sv:property sv:name="property" sv:type="String">
    <sv:value>value</sv:value>
  </sv:property>
  ...
</sv:node>
```

Your decoupled CMS



Create.js

Drupal

PHPCR

Decoupling means collaboration

m
OpenCms
Symfony
sf
TYPO3

Questions?

Fork me on GitHub

@bergie

createjs.org
hallojs.org
viejs.org

phpcr.github.com
iks-project.eu