

Scripting DSpace in JRuby with the jrdspace gem



Monika Mevenkamp

monikam@princeton.edu

DSpace

Hierarchy

Communities

Collections

Items

Bitstreams

Web UI

APIs oai, rest, ...

Accounts & Rights

Stats/Reporting

Administration

Web UI & CLI

DSpace @ Princeton

DataSpace

60,000 Senior Thesis in ~50 departmental collections

~ 70 library collections with close to 5000 articles

various midsize and smaller collections

some research data - but mostly pdf

Soon new instance with faculty articles

At Princeton

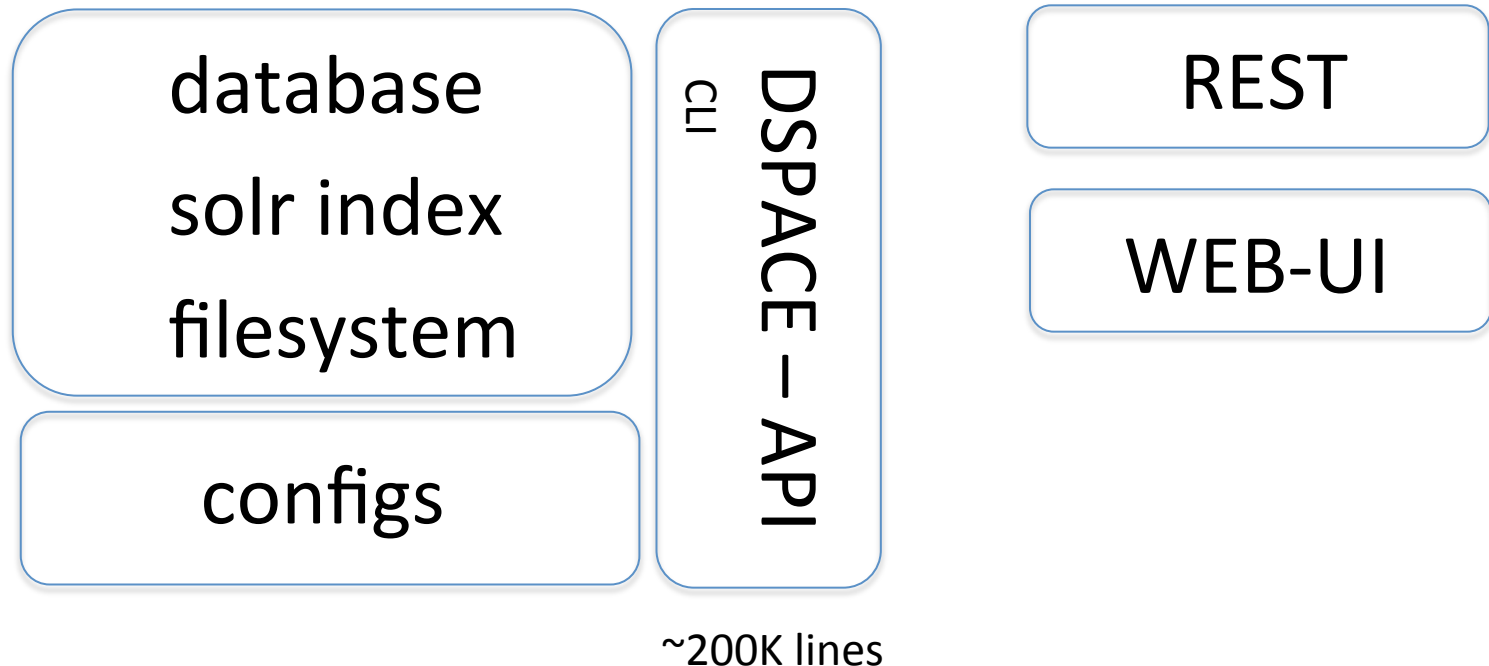
Princeton Senior Theses

- who are submitters for department
- new default value for class year
- set metadata this way iff access permissions are these
- bitstream downloads over time

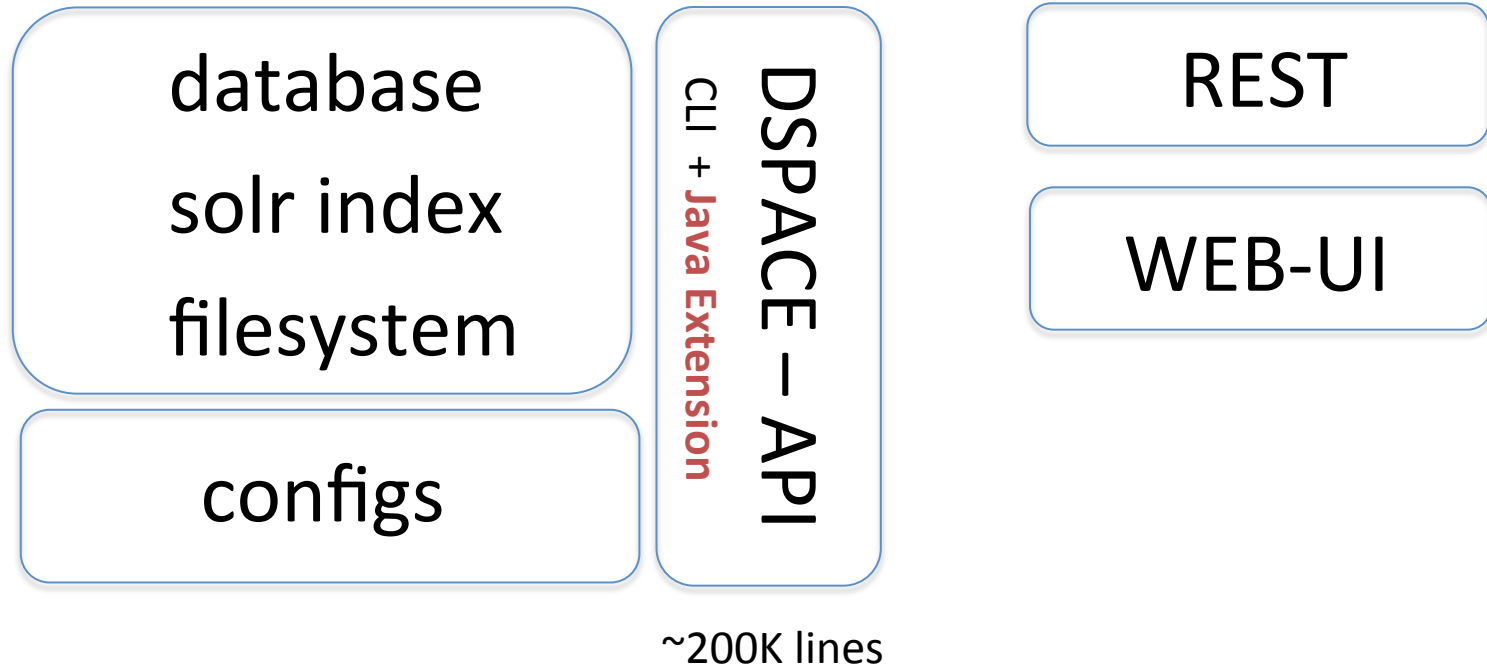
Library Content

- create a new collection such that

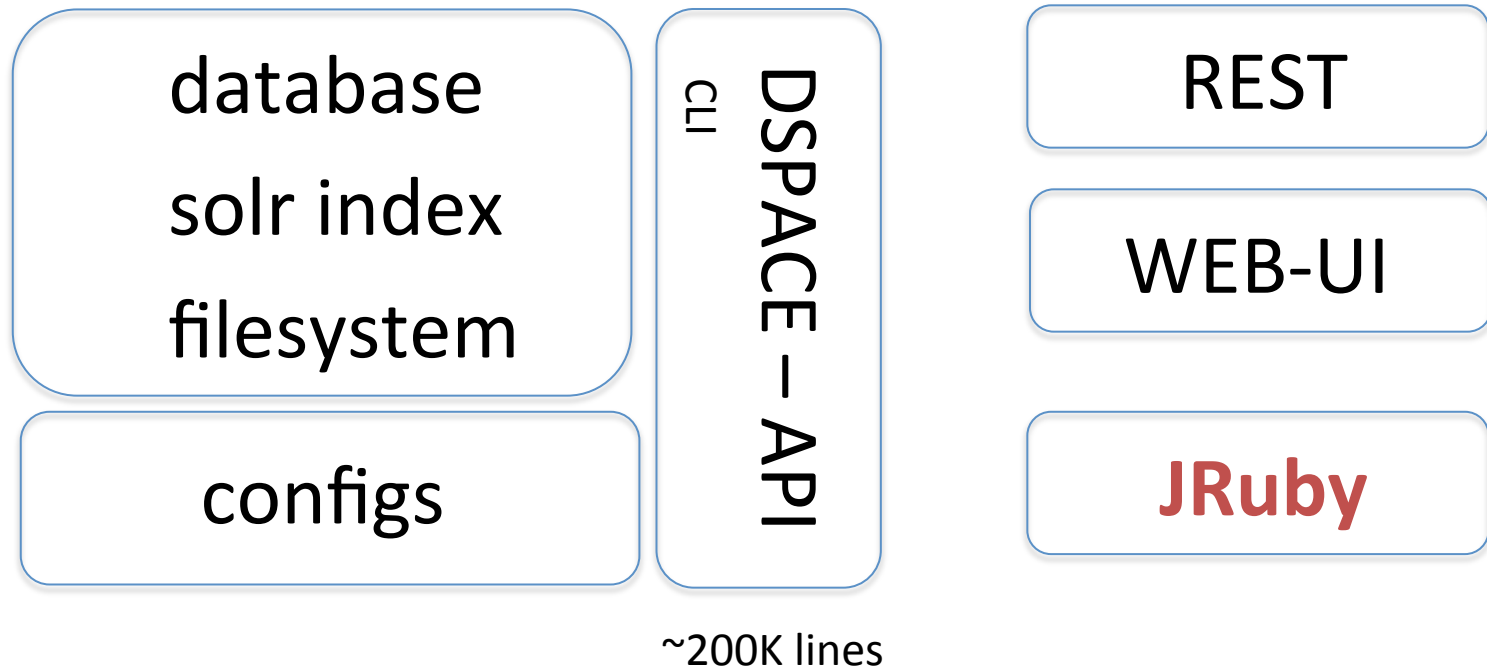
Extending DSpace



Extending DSpace



Extending DSpace



Extending DSpace

Java Development Cycle

- code
- mvn package **90 sec**
- ant update
- test/run

JRuby Development Cycle

- start interactive interpreter
- load dspace-api jars **12 sec**
- load Ruby gems
- code, test/run code

Extending with JRuby

Ruby interpreter written in Java

Ruby

load compiled Java

call Java methods

Java

execute Ruby from Java

JRuby – not Jython

interfaces with Java

interpreted not compiled

widely used

many packages

I know Ruby better

Time For Examples

Set Password

Stick it in a file – make it executable

```
#!/usr/bin/env jruby  
require 'dspace'
```

```
netid, pwd = ARGV    # or from somewhere else
```

```
DSpace.load  
DSpace.login(ENV['USER'])  
p = DEPerson.find(netid)  
raise "no such eperson" if p.nil?  
p.setPassword(pwd)  
p.update  
DSpace.commit
```

Collection Submitters

```
handle = '123456789/5858'
```

```
DSpace.load
```

```
DSpace.fromString(handle).getCollections.each do |col|  
  subs = col.getSubmitters.getMembers  
  members = subs.collect {|p| p.getFullName }  
  puts col.getName + "\t\t" + members.join "\t"  
end
```

Collection Submitters

```
handle = '123456789/5858'
```

```
DSpace.load
```

```
DSpace.fromString(handle).getCollections.each do |col|  
  subs = col.getSubmitters.getMembers  
  members = subs.collect {|p| p.getFullName }  
  puts col.getName + "\t\t" + members.join "\t"  
end
```

Output

```
Anthropology           Bergstrom McLaughlin   Kuhlman Ni  
Architecture School   Schamberger Cremin  
Art and Archaeology   Koepp Dach  
... .
```

Set Default Class Year

```
year = '2016'  
schema, element, qualifier = ['pu', 'date', 'classyear']  
handle = '123456789/5858'
```

```
DSpace.load  
DSpace.login(ENV['USER'])
```

```
DSpace.fromString(handle).getCollections.each do |col|  
  template = col.getTemplateItem  
  if (template) then  
    template.setMetadataSingleValue(schema, element,  
                                     qualifier, nil, year)  
  
    template.updateMetadata  
    template.update  
  end  
end
```

```
DSpace.commit
```

Submissions Over Time

```
date_field = 'dc.date.accessioned'
iter, submissions = DItem.iter, {}
while (i = iter.next())
  begin
    date = i.getMetadata(date_field)
    year, month = date.split('-')
    submissions[year] ||= []
    submissions[year][month.to_i] ||= 0
    submissions[year][month.to_i] += 1
  rescue
    $stderr.puts "Item #{i} can't parse #{date}"
  end
end
end
puts "Year\t" + (1..12).collect{|m| m.to_s}.join("\t")
submissions.keys.sort.each { |y|
  counts = (1..12).collect { |m| submissions[y][m] || 0 }
  puts y + "\t" + counts.join("\t")
}
```


Submissions Over Time

```
date_field = 'dc.date.issued'
iter, submissions DItem.iter, {}
while (i = iter.next()) do
  date = i.getMetadata(date_field)
  begin
    year, month = date.split('-')
    submissions[year] ||= []
    submissions[year][month.to_i] ||= 0
    submissions[year][month.to_i] += 1
  rescue
    $stderr.puts "Item #{i} can't parse #{date}"
  end
end
end
```

Year	1	2	3	4	5	6	7	8	9	10	11	12
2010	12	11	5	5	9	11	7	5	11	6	10	7
2011	8	6	4	8	5	10	1	9	8	8	8	6
2012	11	3	5	8	5	4	12	7	7	14	4	6

Strict Policies -> MetaData

```
year , group_name = 2016, "SrTheses_Bitstream_Read_Mudd"
set_value = 'Walk-in Access. ...';

items = DSpace.findByMetadataValue('pu.date.classyear', year, nil)
items.each do |i|
  DSpace.create(i).bitstreams.each do |bit|
    DSpace.create(bit).policies.each do |hsh|
      if (0 ==hsh[:action] and group_name == hsh[:group]) then
        i.setMetadataSingleValue('dc', 'right', 'accessRights',
                                nil, set_value)
        puts "ITEM #{i.getHandle}setting dc.rights.accessRights"
        i.update
      end
    end
  end
end
end
end
```

Create Collection

```
require 'dspace'
```

```
class DCollection  
  def copy(name, parent) ... end  
end
```

```
DSpace.Load()
```

```
DSpace.login "monikam"  
template_coll = DSpace.fromString('123456789/6625')  
parent, name = prompt_and_collect_answers
```

```
dcol = DSpace.create(template_coll)  
dcol.copy(name, parent)
```

```
DSpace.commit
```

Create Collection

```
class DCollection
  def copy(name, parent)
    new_col = DCollection.create(name, parent)
    group = @obj.getSubmitters();
    if (group) then
      new_group = new_col.createSubmitters
      copy_group(group, new_group)
    end
    [1, 2, 3].each do |i|
      group = @obj.getWorkflowGroup(i);
      if (group) then
        new_group = new_col.createWorkflowGroup(i)
        copy_group(group, new_group)
      end
    end
    new_col.update()
    return new_col
  end
end
```

```
private
  def copy_group(from, to)
    from.getMemberGroups().each do |g|
      to.addMember(g);
    end
    from.getMembers().each do |g|
      to.addMember(g);
    end
    to.update
  end
end
```

@ GitHub

<https://github.com/akinom/dspace-jruby>

<https://rubygems.org/gems/jrdspace>

[@rubydoc.info](http://rubydoc.info)

@RubyGems.org