# Handles Sufia Integration

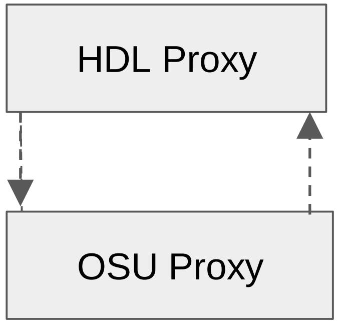
By: Chris Bartos

# What is a Handle?

# How does it work?

http://hdl.handle.net/<handleX>
i.e. http://hdl.handle.net/1811.1/OSU124

Therefore:
http://hdl.handle.net/<prefix>/<handle>
(Essentially a "PERMALINK")



Do you own this handle prefix?

I sure do! Redirect to me!

- The HDL Proxy will redirect to the URL of our Proxy Server (Database holds our handles)
- There is makes a 1-1 mapping to URL that this handle points to.
- (Latency of microseconds to find Handle)
- How about a quick example of it working?

# This seems a bit over-simplified...

Sorry -- I only have 5 minutes

https://www.ietf.org/rfc/rfc3652.txt

RFC 3652 for less enjoyable reading

# What does this get me?

The URL to hdl.handle.net will **ALWAYS** stay the same...maybe

If you are citing this URL in a publication, the URL will ALWAYS (maybe...) point to the same URL.

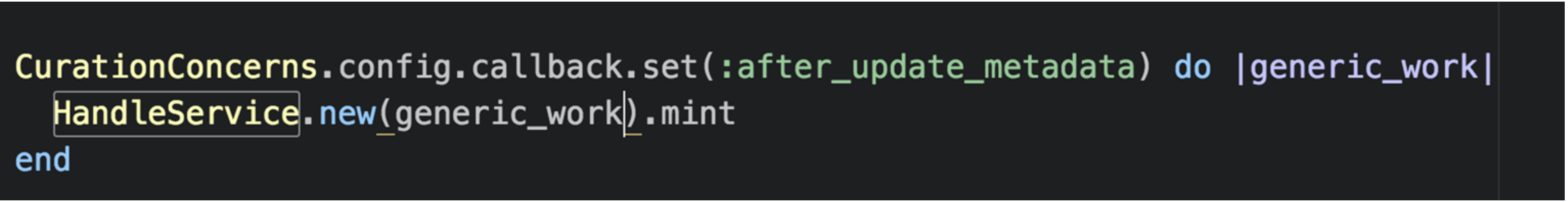
You don’t have to update links everywhere on the interwebs.
(just where the handle is pointing to)

A handle that doesn’t point to the URL it’s always pointed to = Angry Librarians

# How about some code?

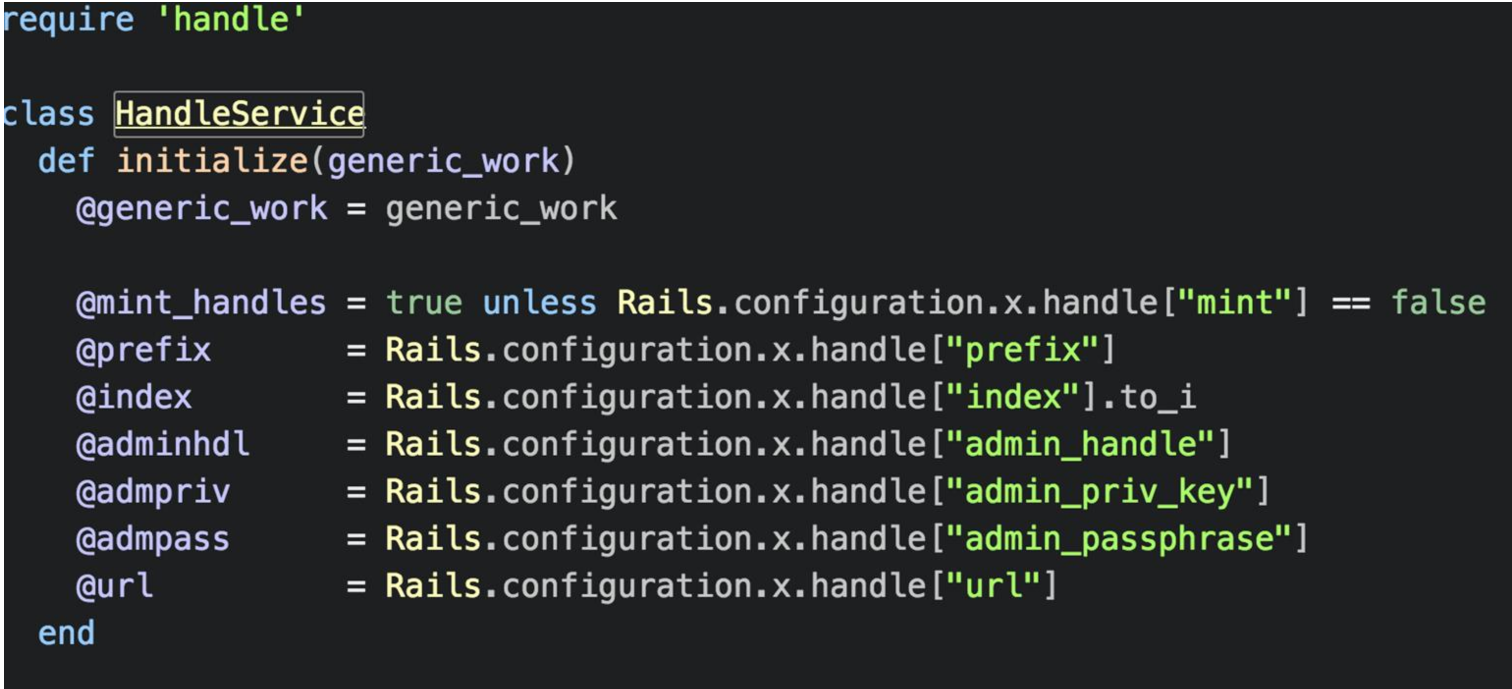
Hook into the update metadata callback. (config/initializers/sufia.rb)

When updating a GenericWork’s metadata, we mint a handle on our Handle Server.



```ruby
CurationConcerns.config.callback.set(:after_update_metadata) do |generic_work|
  HandleService.new(generic_work).mint
end
```

# HandleService (mints handles)

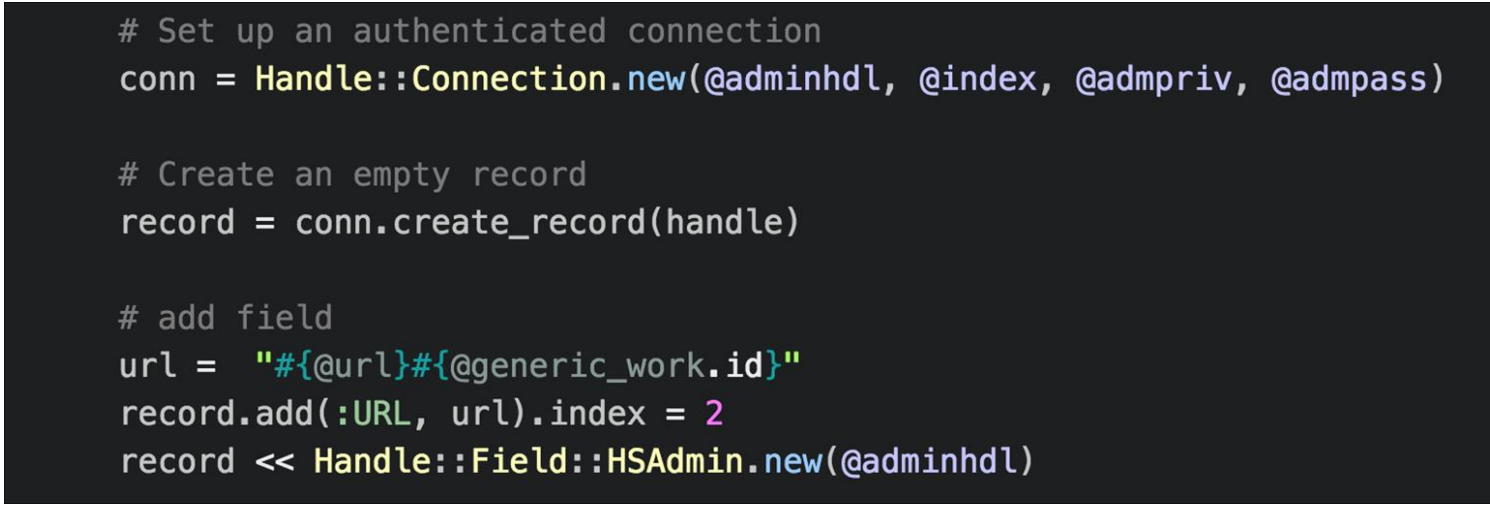


```
require 'handle'

class HandleService
  def initialize(generic_work)
    @generic_work = generic_work

    @mint_handles = true unless Rails.configuration.x.handle["mint"] == false
    @prefix       = Rails.configuration.x.handle["prefix"]
    @index        = Rails.configuration.x.handle["index"].to_i
    @adminhdl     = Rails.configuration.x.handle["admin_handle"]
    @admpriv      = Rails.configuration.x.handle["admin_priv_key"]
    @admpass      = Rails.configuration.x.handle["admin_passphrase"]
    @url          = Rails.configuration.x.handle["url"]
  end
```

# Mint Handles

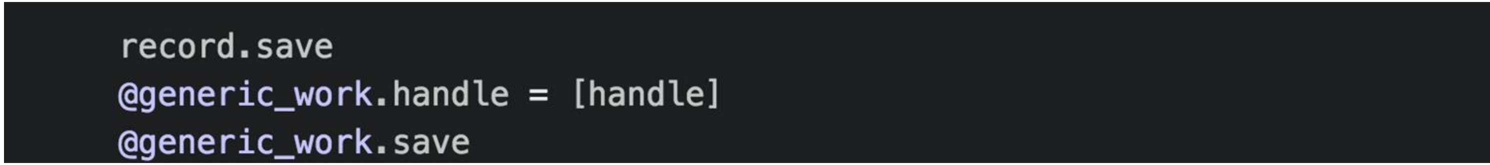


```
# Set up an authenticated connection
conn = Handle::Connection.new(@adminhdl, @index, @admpriv, @admpass)

# Create an empty record
record = conn.create_record(handle)

# add field
url =  "#{@url}#{@generic_work.id}"
record.add(:URL, url).index = 2
record << Handle::Field::HSAdmin.new(@adminhdl)
```

- This has all been taken from the documentation of the ‘handle-system’ gem.
- You can update and delete handles as well. (Update = delete + create)



```
record.save
@generic_work.handle = [handle]
@generic_work.save
```

# Other Stuff...

You need to download the Handle Server tools from handle.net

You need your handle server to be external facing with open UDP, HTTP, and TCP ports in order for you Handle Proxy to talk to HDL Proxy as quickly as possible.

Our code will only create a handle if the GenericWork is publicly available. It doesn’t make sense to create a Handle for a GenericWork that only YOU can see...

# Thank You!

**Sufia 6 Implementation: https://github.com/osulibraries/ims**