**Innovation time January 2011**

After having read the o’Reilly book “REST in Practice” (Grahams book if anyones interested), I set myself the challenge of using OpenRasta (<http://trac.caffeine-it.com/openrasta>) to create a basic RESTful web service.

I decided for the first day to just concentrate on getting a basic CRUD app as outlined in chapter 4 working. This involved the ability to create, read, update and delete physical file xml representations of Artists. It is described in the book as a Level 2 application on Richardsons maturity model (<http://martinfowler.com/articles/richardsonMaturityModel.html>), as it doesn’t make use of Hypermedia yet

[Stress that OpenRasta and REST focuses on resources and their representations <http://www.zephyros-systems.co.uk/blog/?p=45>]

For this I created an ArtistHandler in the normal OpenRasta way, creating methods for each of these four HTTP verbs:

GET for reading   
POST for creating  
PUT for updating  
DELETE for removing.

The main aim of this exercise was to discover exactly what http response statuses and headers I should be returning, and whether it was possible to adhere strictly to the guidelines using OpenRasta.

My 4 CRUD methods were structured like so:

The Responses I wanted to give were structured [pictures]

[Issues with responses and OpenRasta - not being able to send xml as a response when created to save round trip]

[Using Curl to test]

[Not being able to POST Xml ]

[Adding Location headers for 201 Created/204 No Content]

[Creating a link for a Delete – very basic homage to Atom:link]

[Things I wanted to do – Atom feeds, OAuth (no OpenRasta support as yet), talk about track download endpoint]

[Not being able to create a meaningful response to a Bad Request (e.g. telling you which params are missing)]

[Link to project]

REST in practice

**Day 1**  
Build a RESTful service using OpenRasta  
Use album, artist, track model  
Use xsd to validate schema  
Has to be fully RESTful as outlined in the book, without using Atom

**Day 2**

Build another service that can only be accessed with oAuth

Get Service A talking to service B (e.g., as with the RestBucks voucher system, could be a track purchase/download?)

If there is time, look at implementing Atom links.