

CO565  
Web Services

# Chain of Web Services

Draft 0b6649b  
5th March 2018 12:51pm

Changes since this draft  
<https://github.com/wopian/hibari-api/compare/0b6649b...master>

James Harris  
21606555

**5th March 2018**

*Computing & Web Development*  
Buckinghamshire New University

# Summary

<b>A</b>	<b>Research</b>	<b>4</b>
<b>1</b>	<b>Technology - Wider Context</b>	<b>5</b>
1.1	Technological Solutions . . . . .	5
1.2	Architectural Solutions . . . . .	5
1.3	Protocol Solutions . . . . .	5
1.4	Merits and Limitations . . . . .	5
1.5	Security, Reliability and Performance Issues . . . . .	5
<b>2</b>	<b>Web Services Technology</b>	<b>6</b>
2.1	Definition of Web Services . . . . .	6
2.2	Underlying Technology of Web Services . . . . .	6
2.3	Underlying Architecture of Web Services . . . . .	6
2.4	Merits and Limitations of Web Services . . . . .	6
2.5	Security Concerns with Web Services . . . . .	6
2.6	Reliability Concerns with Web Services . . . . .	6
<b>3</b>	<b>REST as an Alternative to Web Services</b>	<b>7</b>
3.1	Definition of REST . . . . .	7
3.2	What Is REST . . . . .	7
3.3	Comparison to Web Services . . . . .	7
<b>4</b>	<b>Examples of Web Services</b>	<b>8</b>
4.1	How and Why Large Organisations Publish Web Services . . . . .	8
4.2	How and Why Large Organisations Consume Web Services . . . . .	8
4.3	Useful Web Service Resources for Developers . . . . .	8
<b>B</b>	<b>Implementation</b>	<b>9</b>
<b>5</b>	<b>Web Service Proposal</b>	<b>10</b>
5.1	Project Name . . . . .	10
5.2	Overview of the Project . . . . .	10
5.3	External Web Services to Be Consumed . . . . .	10
5.4	What Will Be Exposed in My Web Service . . . . .	11
5.5	Initial Database Schema . . . . .	11
5.6	Initial Wireframes / UI Ideas . . . . .	11

<b>6</b>	<b>Engineering Approach</b>	<b>12</b>
6.1	Testing Life-Cycle . . . . .	12
6.2	Implementation Life-Cycle . . . . .	12
6.3	Designing/Modelling Life-Cycle . . . . .	12
6.4	Requirements/Specification/Confirmation Life-Cycle . . . . .	12
<b>7</b>	<b>Project Management</b>	<b>13</b>
7.1	Processes . . . . .	13
7.2	Roles . . . . .	13
7.3	Responsibilities of Group Members . . . . .	13
7.4	Evidence of Collaborative Activity . . . . .	13
7.5	Usage of Source Control Management . . . . .	13
7.6	How the Project Will Be Managed . . . . .	13
<b>8</b>	<b>Development</b>	<b>14</b>
8.1	Requirements Elicitation . . . . .	14
8.2	Design . . . . .	14
8.3	Implementation . . . . .	14
8.4	Testing . . . . .	14

## List of Figures

8.1	src/hibari/headers.cr . . . . .	14
-----	---------------------------------	----

## List of Tables

# **Part A**

# **Research**

## **Section 1**

# **Technology - Wider Context**

Not started

**1.1 Technological Solutions**

**1.2 Architectural Solutions**

**1.3 Protocol Solutions**

**1.4 Merits and Limitations**

**1.5 Security, Reliability and Performance Issues**

## **Section 2**

# **Web Services Technology**

Not started

- 2.1 Definition of Web Services**
- 2.2 Underlying Technology of Web Services**
- 2.3 Underlying Architecture of Web Services**
- 2.4 Merits and Limitations of Web Services**
- 2.5 Security Concerns with Web Services**
- 2.6 Reliability Concerns with Web Services**

## **Section 3**

### **REST as an Alternative to Web Services**

Not started

#### **3.1 Definition of REST**

#### **3.2 What Is REST**

#### **3.3 Comparison to Web Services**



## Section 4

### Examples of Web Services

Not started

- 4.1 How and Why Large Organisations Publish Web Services**
- 4.2 How and Why Large Organisations Consume Web Services**
- 4.3 Useful Web Service Resources for Developers**

# **Part B**

# **Implementation**

## **Section 5**

# **Web Service Proposal**

### **5.1 Project Name**

Hibari API

### **5.2 Overview of the Project**

The project will provide extra statistics related to users and media on Kitsu, which would require hundreds of requests to generate client-side. The provided statistics from this project can then be used to provide users insights into what they are watching/reading and how they compare to other users on the service.

### **5.3 External Web Services to Be Consumed**

The Web Service I will be consuming in this project is Kitsu<sup>1</sup>, an anime & manga discovery and tracking service (similar to IMDb). Its Web Service published as a RESTful API<sup>2</sup> that uses the JSON:API<sup>3</sup> specification.

## 5.4 What Will Be Exposed in My Web Service

In progress

**Note:** *media* refers to *anime* and *manga* exposed as separate API methods that share the same functionality and output.

1. Computed statistics of media ratings
  - (a) Count
  - (b) Mean
  - (c) Median
  - (d) Mode
  - (e) Variance
  - (f) Standard deviation
  - (g) Raw rating frequency - key-value pair of { rating: occurrence }
2. User libraries
  - (a) Statistics of media in the user's library (see 1.)
  - (b) Statistics of media in the user's library grouped by airing year (see 1.)
  - (c) Statistics of media in the user's library grouped by category (see 1.)
  - (d) Top 10 media per category
  - (e) Total episodes/chapters seen
  - (f) Total episodes/chapters seen per status type (Currently Watching, Plan to Watch, Completed, On Hold and Dropped)
  - (g) Total episodes/chapters not seen
  - (h) Total episodes/chapters not seen per status type
3. Categories
  - (a) Statistics of media in a category (see 1)
  - (b) Statistics of media in a category grouped by airing year (see 1.)
4. Media
  - (a) Rank in category (position and percentile)

Maybe also  
Chebyshev's  
Theorem and  
Coefficient of  
Variation

## 5.5 Initial Database Schema

Not started

## 5.6 Initial Wireframes / UI Ideas

Not started

- 
- 1 <https://kitsu.io>
  - 2 <https://kitsu.docs.apiary.io>
  - 3 <http://jsonapi.org>

## **Section 6**

# **Engineering Approach**

Not started

**6.1 Testing Life-Cycle**

**6.2 Implementation Life-Cycle**

**6.3 Designing/Modelling Life-Cycle**

**6.4 Requirements/Specification/Confirmation Life-Cycle**

## **Section 7**

# **Project Management**

Not started

### **7.1 Processes**

### **7.2 Roles**

### **7.3 Responsibilities of Group Members**

### **7.4 Evidence of Collaborative Activity**

### **7.5 Usage of Source Control Management**

### **7.6 How the Project Will Be Managed**

## Section 8

# Development

Not started

### 8.1 Requirements Elicitation

### 8.2 Design

### 8.3 Implementation

#### 8.3.1 Example syntax highlighting

Figure 8.1: src/hibari/headers.cr

```
1 module Hibari
2   # Response headers for the Hibari API
3   HEADERS = {
4     "Content-Type" => JsonAPI::CONTENT_TYPE,
5     "X-Powered-By" => "", # TODO: Remove this header entirely
6     "X-Frame-Options" => "SAMEORIGIN",
7     "X-Xss-Protection" => "1; mode=block",
8     "X-Content-Type-Options" => "nosniff",
9     "Referrer-Policy" => "strict-origin-when-cross-origin"
10  }
11 end
```

### 8.4 Testing

# List of Corrections

Not started	5
Not started	6
Not started	7
Not started	8
In progress	11
Maybe also Chebyshev's Theorem and Coefficient of Variation	11
Not started	11
Not started	11
Not started	12
Not started	13
Not started	14