

WEB API: WHY THEY MATTER

ECOL 453/553 2012

Nirav Merchant

nirav@email.arizona.edu

WHAT IS A WEB API

- ▶ API: Application Programming Interface
- ▶ Calling API over the web (http)
- ▶ What is this http thing you keep talking about ?
- ▶ So why do you want to call API over the web ?
 - ▶ Data size
 - ▶ Data Complexity
 - ▶ Linked data
 - ▶ Specialized databases and data structure
- ▶ It allows for automation (mashup, workflows)
- ▶ How is this different then a perl module ?

WHERE HAVE YOU SEEN WEB API

- ▶ Internet apps
 - ▶ Science apps
 - ▶ <http://www.programmableweb.com/apis/directory/1?apicategory=Medical&protocol=REST>
 - ▶ We will use EBI and TogoWS for our class today

 - ▶ Getting started ...
- 

Nucleic Acids Research

Nucleic Acids Res. 2010 July 1; 38(Web Server issue): W706–W711.

PMCID: PMC2896079

Published online 2010 May 14. doi: [10.1093/nar/gkq386](https://doi.org/10.1093/nar/gkq386)

TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web services

Toshiaki Katayama,^{1,*} Mitsuteru Nakao,^{2,3,*} and Toshihisa Takagi^{2,4}

[Author information](#) ▶ [Article notes](#) ▶ [Copyright and License information](#) ▶

This article has been [cited by](#) other articles in PMC.

ABSTRACT

[Go to:](#)

Web services have become widely used in bioinformatics analysis, but there exist incompatibilities in interfaces and data types, which prevent users from making full use of a combination of these services. Therefore, we have developed the TogoWS service to provide an integrated interface with advanced features. In the TogoWS REST (REpresentative State Transfer) API (application programming interface), we introduce a unified access method for major database resources through intuitive URIs that can be used to search, retrieve, parse and convert the database entries. The TogoWS SOAP API resolves compatibility issues found on the server and client-side SOAP implementations. The TogoWS service is freely available at: <http://togows.dbcls.jp/>.

INTRODUCTION

[Go to:](#)

In recent years, major bioinformatics centers have begun providing SOAP-based (<http://www.w3.org/2002/ws/>) Web services that enable users to use these database resources with client programs in an automated manner. These include the E-Utilities service (1) provided by the National Center for Biotechnology Information (NCBI), Web services provided by the European Bioinformatics Institute (EBI) (2,3), the Web API for Bioinformatics (WABI) from the DNA Data Bank of Japan (DDBJ) (4–7), the Protein Data Bank Japan's (PDBj) Web services (8) and the KEGG API service from the Kyoto Encyclopedia of Genes and Genomes (KEGG) (9). Thanks to these services, users can easily perform various bioinformatics tasks through their choice of client software and can reproduce each procedure as a workflow.

SOME TERMS

- ▶ RESTful (Representational State Transfer)
- ▶ Architectural Styles and the Design of Network-based Software Architectures
Fielding, Roy Thomas (2000)
Doctoral dissertation, University of California, Irvine
- ▶ Some concepts
 - ▶ GET
 - ▶ POST
 - ▶ PUT
 - ▶ DELETE

JSON (AND SOME XML)

- ▶ JSON (JavaScript Object Notation)
lightweight data-interchange format
- ▶ It is easy for humans to read and write
- ▶ It is easy for machines to parse and generate
- ▶ It is based on a subset of the JavaScript Programming Language
- ▶ Built on two structures:
 - ▶ A collection of name/value pairs.
(object, record, struct, dictionary, hash table, keyed list, or associative array)
 - ▶ An ordered list of values
(array, vector, list, or sequence)

JSON EXAMPLE

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```

EXERCISE (MORE TERMS)

- ▶ Go to <http://togows.dbcls.jp/site/en/rest.html>
- ▶ Let's follow some URL
- ▶ Now can you do this for 100 id's ?
- ▶ Log on to your HPC account
- ▶ Utilities you need
 - ▶ wget
 - ▶ curl
 - ▶ JSON pretty printing: `python -mjson.tool`
 - ▶ Unix pipe | to connect things together

MORE EXERCISE

<http://beta.rest.ensembl.org>

beta.rest.ensembl.org

e!Ensembl Endpoints User Guide Change Log About the Ensembl Project Contact Ensembl

Ensembl REST API Endpoints

Comparative Genomics

Resource	Description
GET <code>genetree/id/:id</code>	Retrieves Gene Tree dumps for a given Gene Tree stable identifier
GET <code>homology/id/:id</code>	Retrieves homology information by ensembl gene id
GET <code>homology/symbol/:species/:symbol</code>	Retrieves homology information by symbol

Cross References

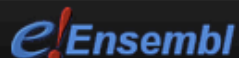
Resource	Description
GET <code>xrefs/id/:id</code>	Perform lookups of Ensembl Identifiers and retrieve their external cross referenc
GET <code>xrefs/name/:species/:name</code>	Performs a lookup based upon the primary accession or display label of an exte about the entry
GET <code>xrefs/symbol/:species/:symbol</code>	Looks up an external symbol and returns all Ensembl objects linked to it. This ca synonym or an externally linked reference. If a Gene's transcript is linked to the Transcript (it supports transient links).

Features

http://beta.rest.ensembl.org/documentation/info/feature_region

beta.rest.ensembl.org/documentation/info/feature_region

beta.rest.ensembl.org/documentation/info/feature_region



Endpoints

User Guide

Change Log

About the Ensembl Project

Contact Ensembl

by. Will search the given and child terms

species	Yes	String	Registry name/aliases used to restrict searches by	-	homo_sapiens
species_set	No	String	The species set name to retrieve constrained elements for	mammals	

Example Requests

[/feature/region/human/7:140424943-140624564?feature=gene;feature=transcript;feature=cds;feature=exon;content-type=application/json](#)

Example output

Perl

Python

Ruby

Curl

Wget

```
1. wget -q --header='Content-type:application/json' 'http://beta.rest.ensembl.org/feature/region/human/7:140424943-140624564?feature=gene;feature=transcript;feature=cds;feature=exon' -O -
```

[/feature/region/human/7:140424943-140624564?feature=gene;feature=transcript;feature=cds;feature=exon;content-type=text/x-gff3](#)

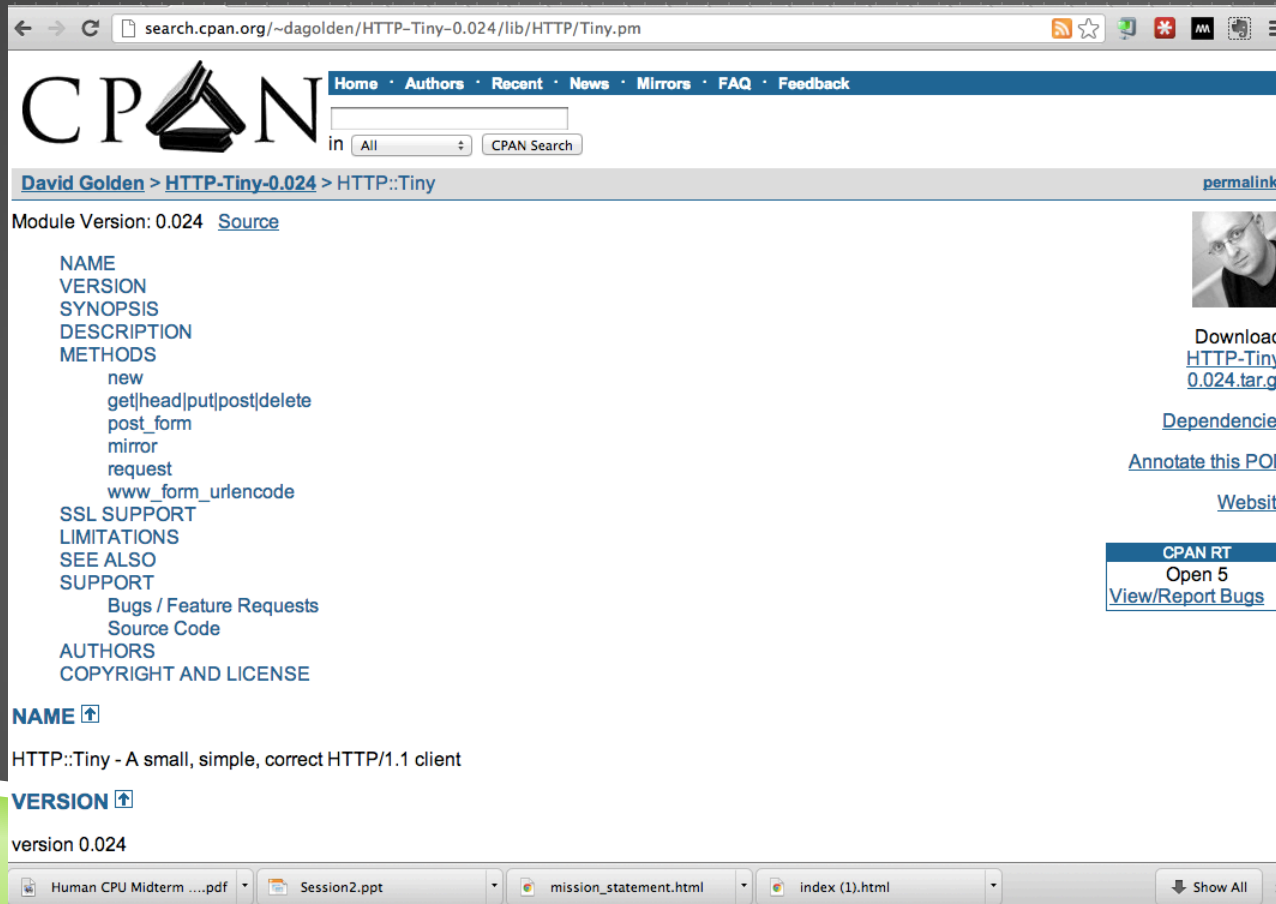
```
"ID" : "ENSG00000207040",  
"logic name" : "ncrna".
```

LETS LOOK AT PERL TAB ON THE PAGE

- ▶ What modules do we need
- ▶ use HTTP::Tiny;
- ▶ use JSON;
- ▶ use strict;
- ▶ use warnings;
- ▶ use Data::Dumper;

LETS GET THE ONES WE ARE MISSING

► <http://search.cpan.org>



The screenshot shows a web browser window with the address bar containing `search.cpan.org/~dagolden/HTTP-Tiny-0.024/lib/HTTP/Tiny.pm`. The page header features the CPAN logo and navigation links: Home, Authors, Recent, News, Mirrors, FAQ, and Feedback. Below the header is a search bar with a dropdown menu set to 'All' and a 'CPAN Search' button. The main content area displays the module page for 'HTTP::Tiny' by 'David Golden'. The breadcrumb trail is 'David Golden > HTTP-Tiny-0.024 > HTTP::Tiny', with a 'permalink' link on the right. The module version is listed as '0.024' with a 'Source' link. A list of module features is shown on the left, including NAME, VERSION, SYNOPSIS, DESCRIPTION, METHODS (new, get|head|put|post|delete, post_form, mirror, request, www_form_urlencode), SSL SUPPORT, LIMITATIONS, SEE ALSO, SUPPORT (Bugs / Feature Requests, Source Code), AUTHORS, and COPYRIGHT AND LICENSE. On the right side, there is a profile picture of David Golden, a 'Download: HTTP-Tiny-0.024.tar.gz' link, and links for 'Dependencies', 'Annotate this POD', and 'Website'. A 'CPAN RT' box indicates 'Open 5' bugs, with a 'View/Report Bugs' link. The browser's taskbar at the bottom shows several open tabs: 'Human CPU Midtermpdf', 'Session2.ppt', 'mission_statement.html', and 'index (1).html', along with a 'Show All' button.

HOW DO WE GET THE MODULE

- ▶ wget to rescue
- ▶ Copy and paste url !
- ▶ tar -xvzf <downloaded file>
- ▶ Make a directory in your home
perl-module
- ▶ Go to downloaded dir (HTTP-Tiny-0.024)
- ▶ perl Makefile.PL PREFIX=~/.perl-module
- ▶ make
- ▶ make test
- ▶ make install
- ▶ check perl-module for installed files !

HOW DO YOU USE IT

- ▶ `setenv PERL5LIB ~/perl-module/share/perl5/`
- ▶ Now you can run the program
- ▶ To make your life easy I have already built the modules that you can use
- ▶ Its in `/genome/perl-lib/share/perl5/`

SO:

- ▶ `set PERL5LIB /genome/perl-lib/share/perl5/`

Now you can run the sample perl scripts from ensembl rest services !