Content Negotiation for REST APIs
## SOAP vs. REST

<table>
<thead>
<tr>
<th>SOAP</th>
<th>POST /api.xml</th>
</tr>
</thead>
</table>

**SOAP**

**POST /api.xml**
| SOAP | POST /api.xml |
| SOAP | POST /api.xml |
## SOAP vs. REST

<table>
<thead>
<tr>
<th></th>
<th>SOAP</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>POST /api.xml</td>
<td>GET /users/sferik.json</td>
</tr>
</tbody>
</table>
## SOAP vs. REST

<table>
<thead>
<tr>
<th></th>
<th>Operation</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP</td>
<td>POST</td>
<td>/api.xml</td>
</tr>
<tr>
<td>REST</td>
<td>GET</td>
<td>/users/sferik.json</td>
</tr>
</tbody>
</table>
## SOAP vs. REST

<table>
<thead>
<tr>
<th></th>
<th>SOAP</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>POST</td>
<td>GET</td>
</tr>
<tr>
<td>/api.xml</td>
<td>GET</td>
<td>/users/sferik.json</td>
</tr>
</tbody>
</table>
## SOAP vs. REST

<table>
<thead>
<tr>
<th></th>
<th>SOAP</th>
<th>POST /api.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST</td>
<td>GET</td>
<td>/users/sferik.json</td>
</tr>
</tbody>
</table>
FILE NAME
EXTENSIONS
Welcome to FreeDOS

CuteMouse v1.9.1 alpha 1 [FreeDOS]
Installed at PS/2 port
C:\>ver

FreeCom version 0.82 pl 3 XMS_Swap [Dec 10 2003 06:49:21]

C:\>dir
Volume in drive C is FREEDOS_C95
Volume Serial Number is 0E4F-19EB
Directory of C:\

FDOS        <DIR>  08-26-04  6:23p
AUTOEXEC  BAT    435  08-26-04  6:24p
BOOTSECT  BIN    512  08-26-04  6:23p
COMMAND  COM   93,963  08-26-04  6:24p
CONFIG   SYS    801  08-26-04  6:24p
FDOSBOOT  BIN    512  08-26-04  6:24p
KERNEL    SYS   45,815  04-17-04  9:19p

6 file(s)    142,038 bytes
1 dir(s)    1,064,517,632 bytes free

C:\>_
RESOURCE VS. REPRESENTATION
XML REPRESENTATION

GET /users/sferik.xml

<user>
  <id>sferik</id>
  <name>Erik</name>
</user>
JSON REPRESENTATION

GET /users/sferik.json

{
   "id": "sferik",
   "name": "Erik"
}
**RESOURCE VS. REPRESENTATION**

<table>
<thead>
<tr>
<th>GET /users/sferik.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /users/sferik.json</td>
</tr>
</tbody>
</table>
RESOURCE VS. REPRESENTATION

GET /users/sferik
ACCEPT HEADER

The Accept request-header field can be used to specify certain media types which are acceptable for the response.
GET /avatars/sferik.png

Accept: image/png
ACCEPT HEADER

GET /avatars/sferik

Accept: image/png,
    image/jpeg; q=0.8,
image/gif; q=0.8,
image/*; q=0.5,
application/json; q=0.1
GET /avatars/sferik

Accept: video/*,
    image/gif; q=0.8,
    image/*; q=0.5,
application/json; q=0.1
The Accept-Language request-header field is similar to Accept, but restricts the set of natural languages that are preferred as a response to the request.
GET /index.se.html

Accept-Language: se
GET /index

Accept-Language: se,
en-us;q=0.8,
en;q=0.7
Build Status Badges now Support SVG

Mathias Meyer, 20 Mar 2014

Build status badges for readme files have a long history at Travis CI. Originally conceived by Jeff Kreeftmeijer, they were initially added on 20 February 2011. That's more than three years ago!

Here's what the first ever build status badge looked like: build status stable

Now they've become part of the trust chain in a project. The green badge instills confidence that the project maintainer is not only actively writing tests, but that they're also caring for having the automated build pass.

Since then, the badges have become omnipresent, adopted by a lot of services like Code Climate and Gemnasium.

They've gotten so popular that a group of fine people huddled together to build Shields, a common endpoint and provider of unified status badges.

Shields was the driving force behind the unified PNG badges that have been in use for the...
Wednesday, May 19, 2010

Introducing WebM, an open web media project

A key factor in the web's success is that its core technologies such as HTML, HTTP, TCP/IP, etc. are open and freely implementable. Though video is also now core to the web experience, there is unfortunately no open and free video format that is on par with the leading commercial choices. To that end, we are excited to introduce WebM, a broadly-backed community effort to develop a world-class media format for the open web.

WebM includes:
- VP8, a high-quality video codec we are releasing today under a BSD-style, royalty-free license
- **Vorbis**[^1], an already open source and broadly implemented audio codec
- a container format based on a subset of the **Matroska**[^2] media container

The team that created VP8 have been pioneers in video codec development for over a decade. VP8 delivers high quality video while efficiently adapting to the varying processing and bandwidth conditions found on today's broad range of web-connected devices. VP8's efficient bandwidth usage will mean lower serving costs for content publishers and high quality video for end-users. The codec's relative simplicity makes it easy to integrate into existing environments and requires less manual tuning to produce high quality results. These existing attributes and the rapid innovation we expect through the open-development process make VP8 well suited for the unique requirements of video on the web.

A developer preview of WebM and VP8, including source code, specs, and encoding tools is available today at [www.webmproject.org](http://www.webmproject.org).

We want to thank the many industry leaders and web community members who are collaborating on the development of WebM and integrating it into their products. Check out what [Mozilla](http://mozilla.org), [Opera](http://opera.com), [Google Chrome](http://chrome.com), [Adobe](http://adobe.com), and many others below have to say about the importance of WebM to the future of web video.

[^1]: [Vorbis](http://www.xiph.org/vorbis)
[^2]: [Matroska](http://www.matroska.org)
WebP, a new image format for the Web
Thursday, September 30, 2010

As part of Google's initiative to make the web faster, over the past few months we have released a number of tools to help site owners speed up their websites. We launched the Page Speed Firefox extension to evaluate the performance of web pages and to get suggestions on how to improve them, we introduced the Speed Tracer Chrome extension to help identify and fix performance problems in web applications, and we released a set of closure tools to help build rich web applications with fully optimized JavaScript code. While these tools have been incredibly successful in helping developers optimize their sites, as we've evaluated our progress, we continue to notice a single component of web pages is consistently responsible for the majority of the latency on pages across the web: images.

Most of the common image formats on the web today were established over a decade ago and are based on technology from around that time. Some engineers at Google decided to figure out if there was a way to further compress lossy images like JPEG to make them load faster, while still preserving quality and resolution. As part of this effort, we are releasing a developer preview of a new image format, WebP, that promises to significantly reduce the byte size of photos on the web.
Starting today, you can share and view animated GIFs on Twitter.com, Android and iPhone. pic.twitter.com/XBrAbOm4Ya
Introducing GIFV

GIFs are no longer about .GIFs—the culture of the GIF now trumps the file format. With Project GIFV, Imgur is reimagining the looping GIF video with all the richness it deserves as a key piece of Internet culture.

OH, WHAT AN OCCASION WE HAVE HERE BEFORE US.
FORMATS CHANGE
FIELDING ON REST

REST is software design on the scale of decades: every detail is intended to promote software longevity and independent evolution. Many of the constraints are directly opposed to short-term efficiency. Unfortunately, people are fairly good at short-term design, and usually awful at long-term design. Most don’t think they need to design past the current release.
EXPOSE RESOURCES,
NOT REPRESENTATIONS
TACK