

YANG Catalog

Ops Area

July 18, 2017

Joe Clarke, Cisco

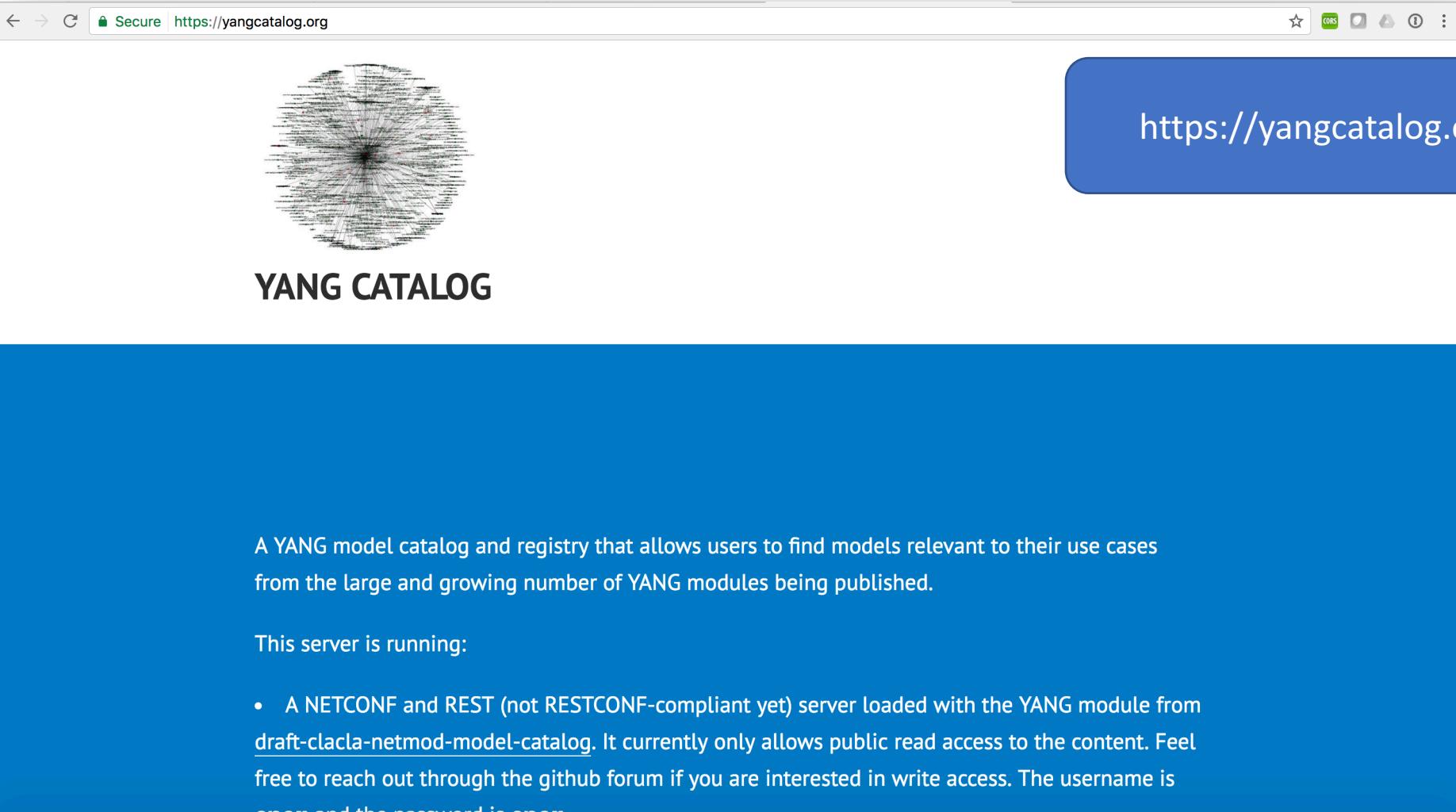


I E T F

Introduction

- We're beginning to have a “good” problem with YANG
 - Lots of YANG modules
 - Lots of bodies working on YANG modules
 - Lots of vendors implementing YANG modules
- Is it sufficient just to have a module developed and implemented?
- We need an industry-wide, single stop, open catalog and toolchain to help YANG consumers, YANG implementors, and YANG authors find and explore available modules

Yangcatalog.org



The screenshot shows a web browser displaying the Yangcatalog.org homepage. The address bar at the top left shows the URL <https://yangcatalog.org>. The page itself features a large, intricate network graph icon above the text "YANG CATALOG". To the right of the graph, a blue callout box contains the URL <https://yangcatalog.org>. The main content area is a large blue box containing descriptive text and a bulleted list.

A YANG model catalog and registry that allows users to find models relevant to their use cases from the large and growing number of YANG modules being published.

This server is running:

- A NETCONF and REST (not RESTCONF-compliant yet) server loaded with the YANG module from [draft-clacla-netmod-model-catalog](#). It currently only allows public read access to the content. Feel free to reach out through the github forum if you are interested in write access. The username is `open` and the password is `open`.

YANG Catalog Overview

- Set of open source tools
 - YANG Keyword Search
 - YANG Module Impact Analysis
 - YANG Validators
 - YANG-compliant Regular Expression Validator
 - *YANG Modules Exploration* (changing soon)
 - *YANG Implementation Discovery* (coming soon)
- An REST-like API to fetch per-module metadata and per-vendor implementation data
- An authenticated API to upload per-module metadata and per-vendor implementation data

Yang Search

YANG DB Search

Enter your search term(s) below:

Search String

Search Options

Case-Sensitive Regular Expression Include MIBs Only Show Latest Revisions

Search Fields

All

| | | |
|---|---|--|
| <input checked="" type="checkbox"/> Module Name | <input checked="" type="checkbox"/> Node Name | <input checked="" type="checkbox"/> Node Description |
|---|---|--|

YANG Versions

1.0 1.1

Schema Types

All

| | | |
|---|--|---|
| <input checked="" type="checkbox"/> Typedef | <input checked="" type="checkbox"/> Grouping | <input checked="" type="checkbox"/> Feature |
| <input checked="" type="checkbox"/> Identity | <input checked="" type="checkbox"/> Extension | <input checked="" type="checkbox"/> RPC |
| <input checked="" type="checkbox"/> Container | <input checked="" type="checkbox"/> List | <input checked="" type="checkbox"/> Leaf-List |
| <input checked="" type="checkbox"/> Leaf | <input checked="" type="checkbox"/> Notification | <input checked="" type="checkbox"/> Action |

Keyword and regex searches supported

The Catalog stores multiple module revisions, but one can choose to only see the latest

Find modules by YANG version

Search By Schema Type (e.g., find all matching typedefs)

Search! Reset

Search Results

YANG DB Search Results for 'uri'

Show 10 entries

| Name | Revision | Schema Type | Path | Module | Origin | Organization | Maturity | Imported By # Modules | Compilation Status | Description |
|------|------------|-------------|-----------|---|-------------------|--------------|----------|-----------------------|--------------------|---|
| uri | 2013-07-15 | typedef | /inet:uri | ietf-inet-types (Impact Analysis) | Industry Standard | ietf | RFC | 188 | N/A | The uri type represents a Uniform Resource Identifier (URI) as defined by STD 66. Objects using the uri type MUST be in US-ASCII encoding, and MUST be normalized as described by RFC 3986 Sections 6.2.1, 6.2.2.1, and 6.2.2.2. All unnecessary percent-encoding is removed, and all case-insensitive characters are set to lowercase except for hexadecimal digits, which are normalized to uppercase as described in Section 6.2.2.1. The purpose of this normalization is to help provide unique URIs. Note that this normalization is not sufficient to provide uniqueness. Two URIs that are textually distinct after this normalization may still be equivalent. Objects using the uri type may restrict the schemes that they permit. For example, 'data:' and 'urn:' schemes might not be appropriate. A zero-length URI is not a valid URI. This can be used to express 'URI absent' where required. In the value set and its semantics, this type is equivalent to the Uri SMIv2 textual convention defined in RFC 5017."; |

YANG Definition for '/inet:uri'

```
// From : ietf-inet-types@2013-07-15
typedef uri {
    type string;
    description "The uri type represents a Uniform Resource Identifier (URI) as defined by STD 66.

    Objects using the uri type MUST be in US-ASCII encoding, and MUST be normalized as described by RFC 3986 Sections 6.2.1, 6.2.2.1, and 6.2.2.2. All unnecessary percent-encoding is removed, and all case-insensitive characters are set to lowercase except for hexadecimal digits, which are normalized to uppercase as described in Section 6.2.2.1.

    The purpose of this normalization is to help provide unique URIs. Note that this normalization is not sufficient to provide uniqueness. Two URIs that are textually distinct after this normalization may still be equivalent.

    Objects using the uri type may restrict the schemes that they permit. For example, 'data:' and 'urn:' schemes might not be appropriate.

    A zero-length URI is not a valid URI. This can be used to express 'URI absent' where required.

    In the value set and its semantics, this type is equivalent to the Uri SMIv2 textual convention defined in RFC 5017.";
```

reference "RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
RFC 3305: Report from the Joint W3C/IETF URN Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications

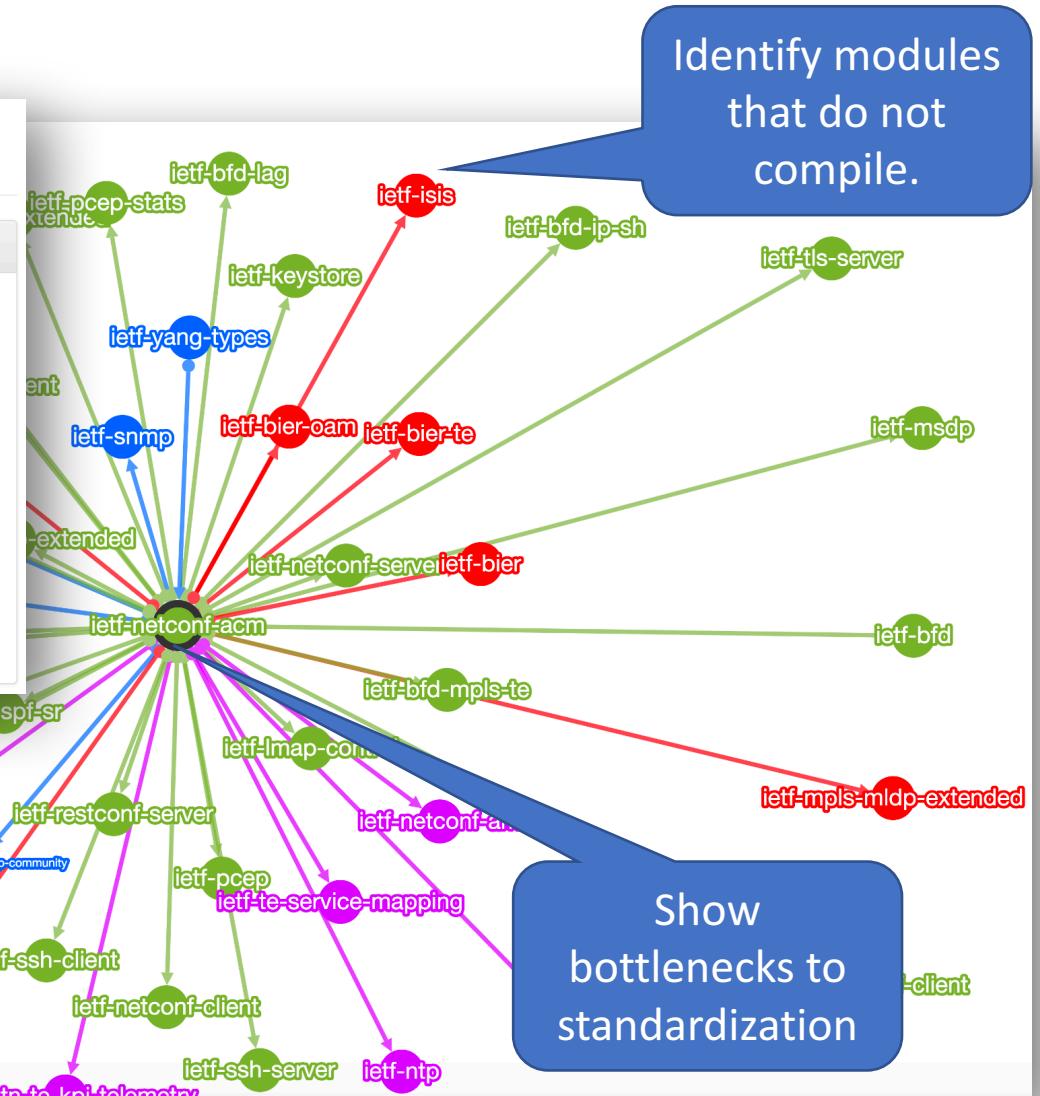
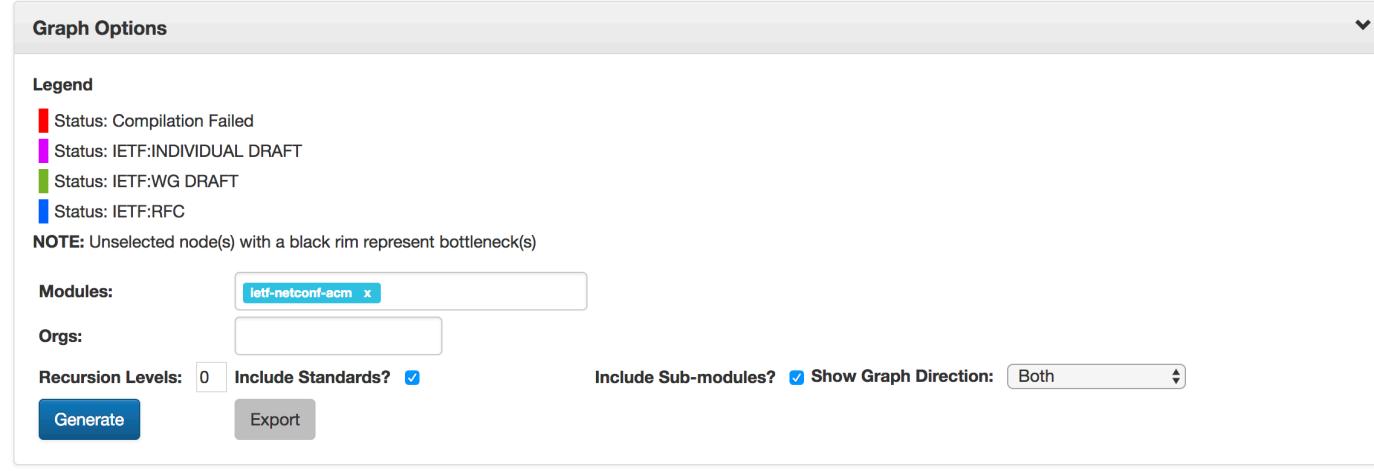
Understand how prolific a module is.

Find mature modules.

Quick view of node details.

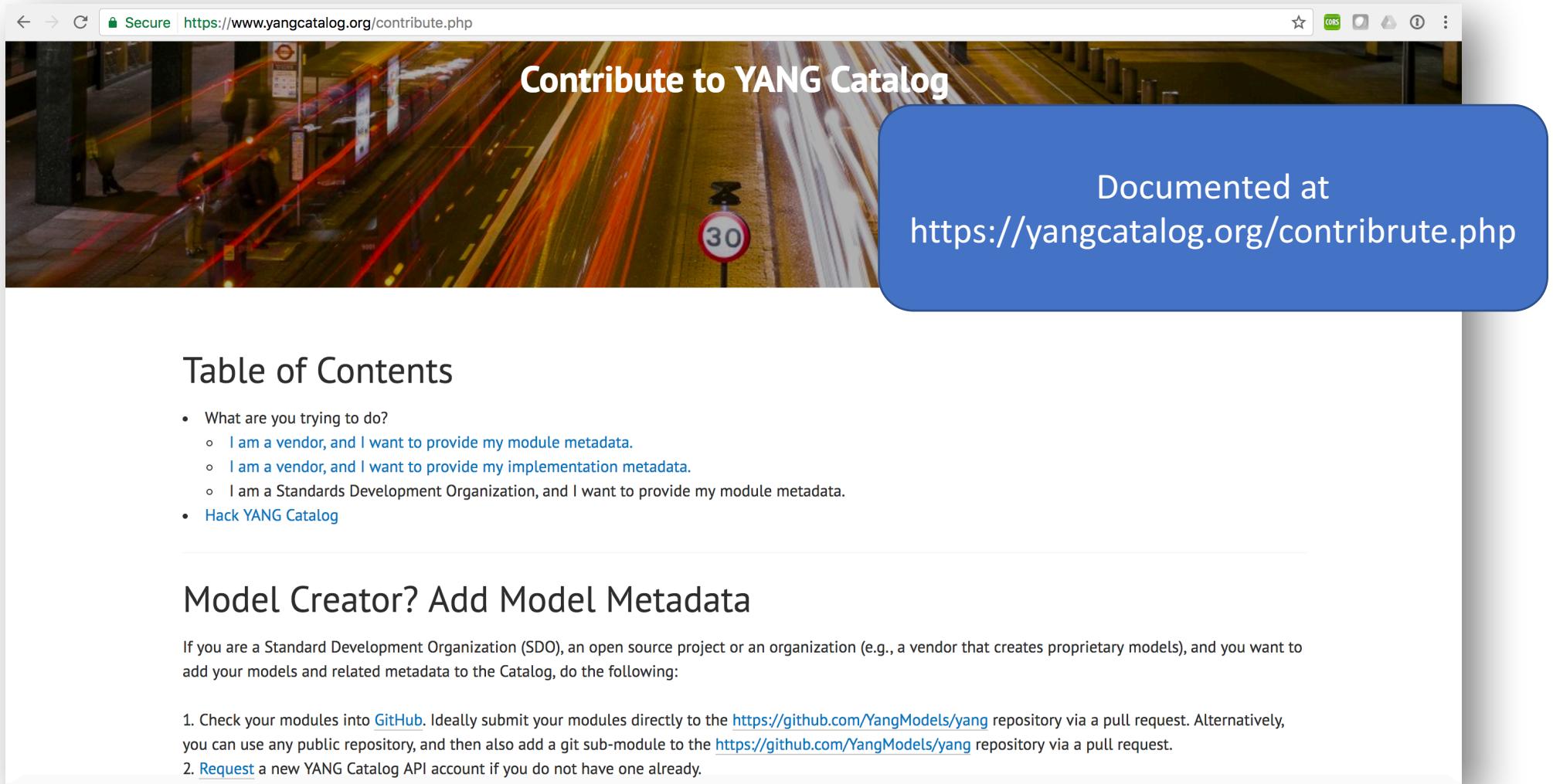
Impact Analysis

YANG Impact Graph for Module(s): ietf-netconf-acm



Understand the relationship of a given module (what modules does it affect, what modules affect it)

The YANG Catalog API



A screenshot of a web browser displaying the 'Contribute to YANG Catalog' page at <https://www.yangcatalog.org/contribute.php>. The page features a blurred background image of a city street at night with motion blur from lights. The title 'Contribute to YANG Catalog' is centered above a blue callout box. The callout box contains the text 'Documented at <https://yangcatalog.org/contribrute.php>'.

Table of Contents

- What are you trying to do?
 - [I am a vendor, and I want to provide my module metadata.](#)
 - [I am a vendor, and I want to provide my implementation metadata.](#)
 - I am a Standards Development Organization, and I want to provide my module metadata.
- [Hack YANG Catalog](#)

Model Creator? Add Model Metadata

If you are a Standard Development Organization (SDO), an open source project or an organization (e.g., a vendor that creates proprietary models), and you want to add your models and related metadata to the Catalog, do the following:

1. Check your modules into [GitHub](#). Ideally submit your modules directly to the <https://github.com/YangModels/yang> repository via a pull request. Alternatively, you can use any public repository, and then also add a git sub-module to the <https://github.com/YangModels/yang> repository via a pull request.
2. [Request](#) a new YANG Catalog API account if you do not have one already.

Module Metadata API Structure

```
module: module-metadata
  +-rw modules
    +-rw module* [name revision]
      +-rw generated-from?          enumeration
      +-rw maturity-level?         enumeration
      +-rw document-name?          string
      +-rw author-email?           yc:email-address
      +-rw reference?              inet:uri
      +-rw name                    yang:yang-identifier
      +-rw revision                union
      +-rw organization             string
      +-rw source-file
        | +-rw owner                 string
        | +-rw repository            string
        | +-rw path                  path
        | +-rw branch?               string
      +-rw organization-specific-metadata
        +-rw ietf
          +-rw ietf-wg?             string
```

Vendor Implementation Metadata API Structure

```
module: platform-implementation-metadata
  +-rw platforms* [vendor name software-version software-flavor]
    +-rw vendor          string
    +-rw name            string
    +-rw models*         string
    +-rw software-flavor string
    +-rw software-version string
    +-rw os-type?        string
    +-rw capabilities-file
      +-rw owner?       string
      +-rw repository?  url
      +-rw path?        path
```

Retrieving Metadata

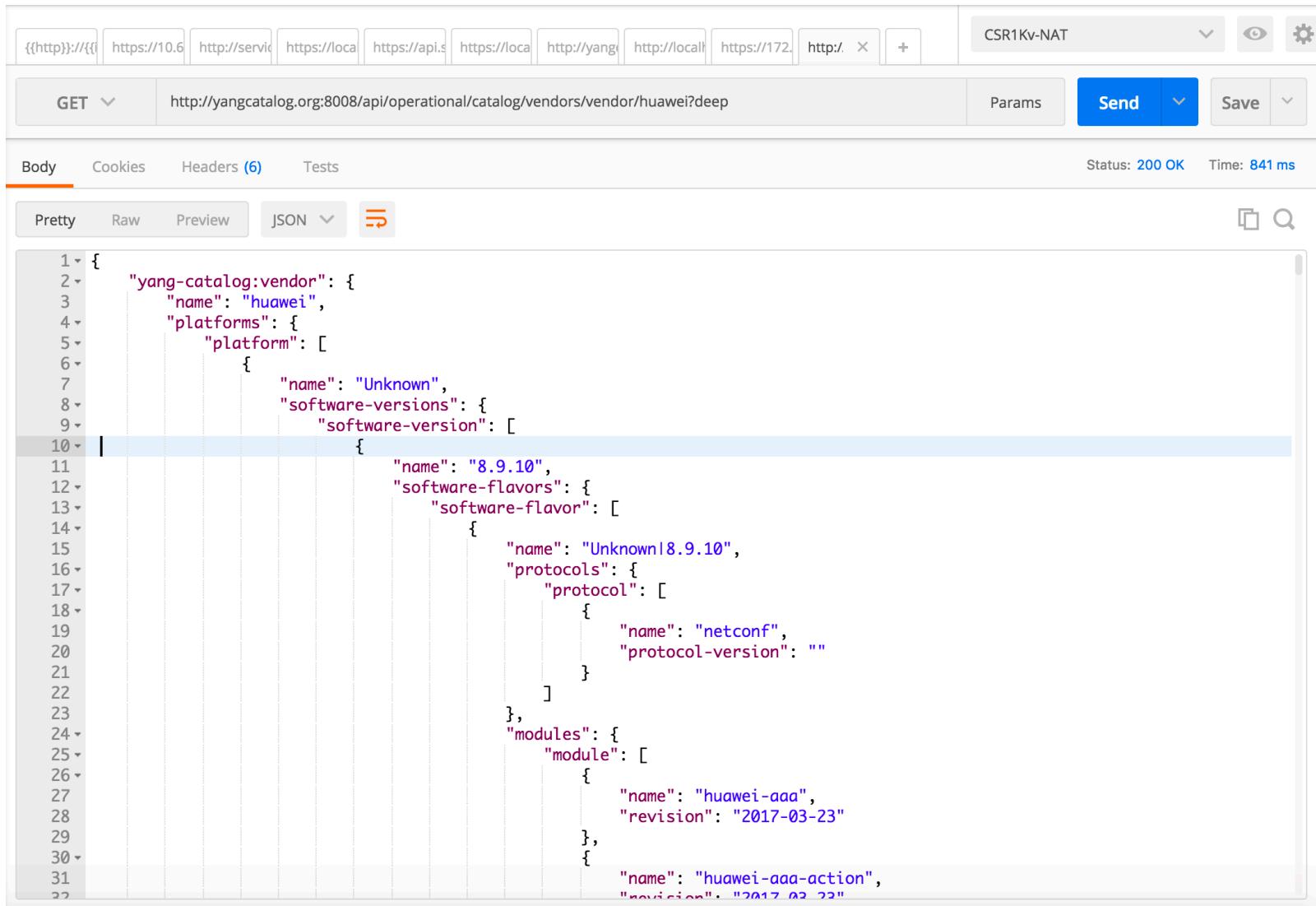
GET <http://yangcatalog.org:8008/api/operational/catalog/modules/module/ietf-interfaces,2014-05-08>

Body Cookies Headers (6) Tests

Pretty Raw Preview JSON

```
1 {  
2   "yang-catalog:module": {  
3     "name": "ietf-interfaces",  
4     "revision": "2014-05-08",  
5     "schema": "missing element",  
6     "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",  
7     "generated-from": "not-applicable",  
8     "maturity-level": "ratified",  
9     "document-name": "rfc7223",  
10    "reference": "https://tools.ietf.org/html/rfc7223",  
11    "organization": "ietf",  
12    "compilation-status": "PASSED",  
13    "compilation-result": "",  
14    "prefix": "if",  
15    "yang-version": "1.0",  
16    "description": "This module contains a collection of YANG definitions for managing network interfaces.\nCopyright (c)  
2014 IETF Trust and the persons identified as authors of the code. All rights reserved.  
Redistribution and use  
in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license  
terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal  
Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info). This version of this YANG module  
is part of RFC 7223; see the RFC itself for full legal notices.",  
17    "contact": "WG Web: <http://tools.ietf.org/wg/netmod/>\nWG List: <mailto:netmod@ietf.org>\nWG Chair: Thomas Nadeau\n<mailto:tnadeau@lucidvision.com>\nWG Chair: Juergen Schoenwaelder\n<mailto:j.schoenwaelder@jacobs-university.de>\nEditor: Martin Bjorklund\n<mailto:mbj@tail-f.com>",  
18    "module-type": "module",  
19    "source-file": {  
20      "online": {  
21        "owner": "YangModels",  
22        "repository": "https://github.com/YangModels/yang",  
23        "path": "standard/ietf/RFC/ietf-interfaces@2014-05-08.yang"  
24      },  
25      "local": {}  
26    }  
27  }  
28 }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 }  
35 }  
36 }  
37 }  
38 }  
39 }  
40 }  
41 }  
42 }  
43 }  
44 }  
45 }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

Retrieving Metadata



The screenshot shows a REST client interface with the following details:

- URL:** `http://yangcatalog.org:8008/api/operational/catalog/vendors/vendor/huawei?deep`
- Status:** 200 OK
- Time:** 841 ms
- Body (Pretty JSON):**

```
1 {  
2   "yang-catalog:vendor": {  
3     "name": "huawei",  
4     "platforms": {  
5       "platform": [  
6         {  
7           "name": "Unknown",  
8           "software-versions": {  
9             "software-version": [  
10               {  
11                 "name": "8.9.10",  
12                 "software-flavors": {  
13                   "software-flavor": [  
14                     {  
15                       "name": "Unknown|8.9.10",  
16                       "protocols": {  
17                         "protocol": [  
18                           {  
19                             "name": "netconf",  
20                             "protocol-version": ""  
21                           }  
22                         ]  
23                       },  
24                       "modules": {  
25                         "module": [  
26                           {  
27                             "name": "huawei-aaa",  
28                             "revision": "2017-03-23"  
29                           },  
30                           {  
31                             "name": "huawei-aaa-action",  
32                             "revision": "2017-03-23"  
33                           }  
34                         ]  
35                       }  
36                     ]  
37                   ]  
38                 ]  
39               ]  
40             ]  
41           ]  
42         ]  
43       ]  
44     ]  
45   ]  
46 }
```

YANG Regular Expression Validator

The screenshot shows a web browser window with the URL <https://yangcatalog.org/yangre>. The page title is "Yang Re". A search bar contains the pattern `[a-zA-Z0-9.!#$%&*+=?^_`~-]+@[a-zA-Z0-9-]+([.][a-zA-Z0-9-]+)*`. Below the pattern is a "Test string" input field containing `jclarke@cisco.com`. To the right of the input fields is a blue speech bubble with the text: "Validate regex patterns using YANG and W3C rules." Below the test string is a section titled "Pattern 1 results" with two buttons: "W3CGREP matches" and "YANGRE matches".

Pattern 1 [a-zA-Z0-9.!#\$%&*+=?^_`~-]+@[a-zA-Z0-9-]+([.][a-zA-Z0-9-]+)*

Invert match +

Test string:

jclarke@cisco.com

Validate regex patterns using YANG and W3C rules.

Pattern 1 results

W3CGREP matches

YANGRE matches

Results And Next Steps

- We have fully integrated IETF and BBF models
- Working to incorporate MEF models
- Working with Cisco, Huawei and Juniper to add metadata for their models
- Submitted **draft-clacla-netmod-model-catalog-00** to describe the backing store for the Catalog
- Evangelizing YANG Catalog with vendors, SDOs, and customers
- Continue to develop the yang-catalog.yang model
- Take feedback to enhance the Catalog tools
- Add a report for YANG implementation discovery and revamp the YANG exploration module