YANG Catalog

Ops Area

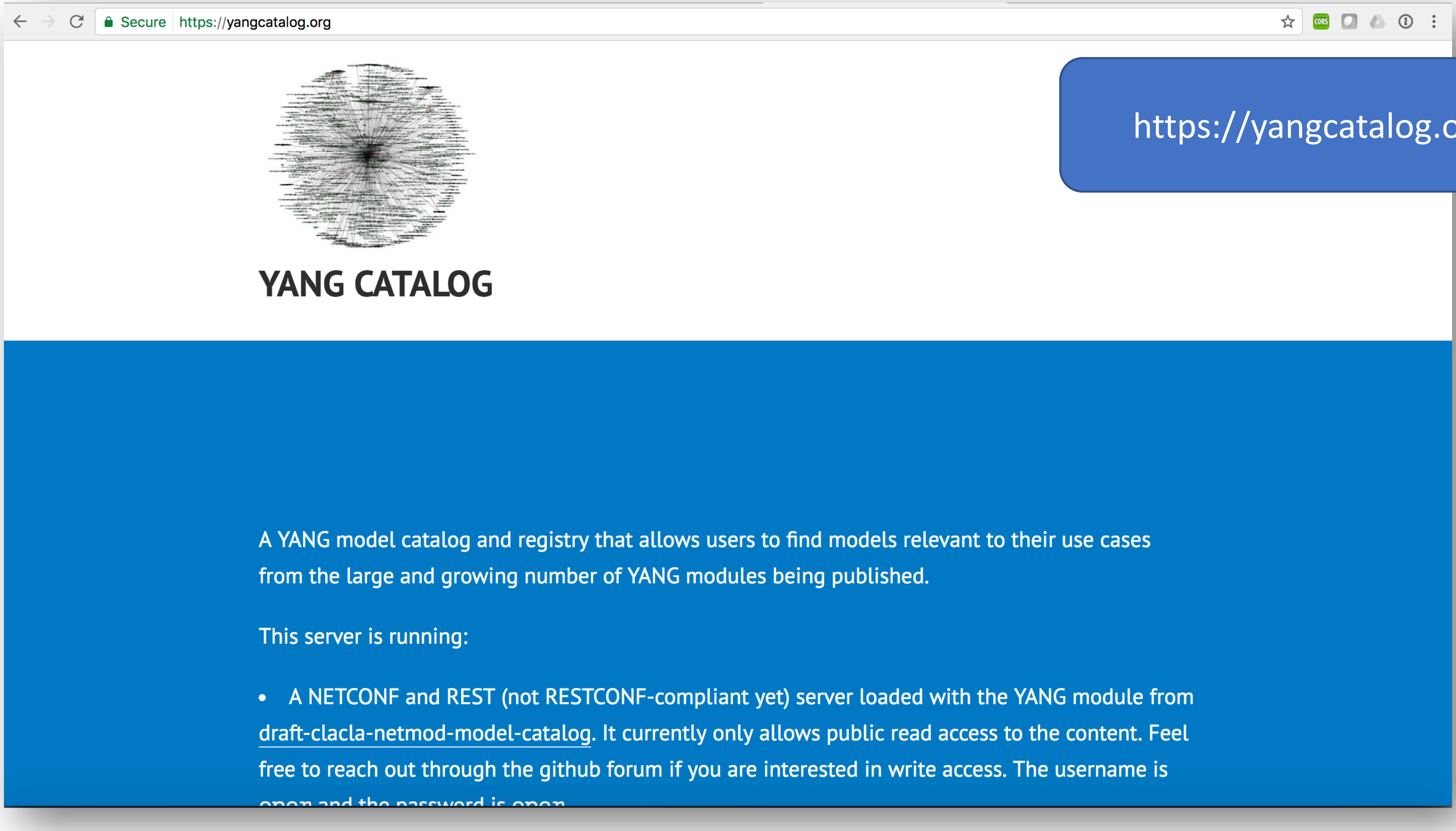
July 18, 2017

Joe Clarke, Cisco

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



YANG CATALOG

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:

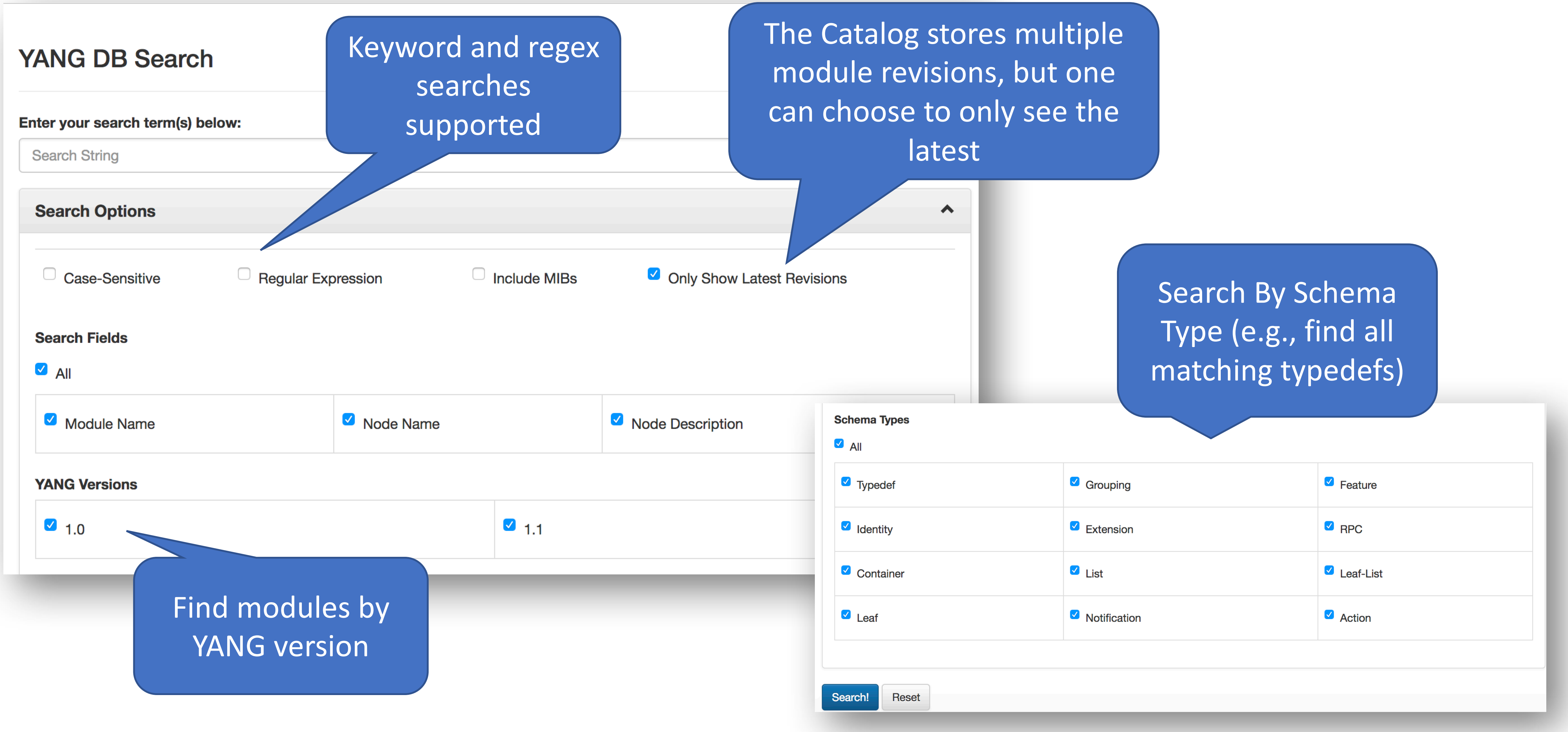
- A NETCONF and REST (not RESTCONF-compliant yet) server loaded with the YANG module from draft-clacla-netmod-model-catalog. It currently only allows public read access to the content. Feel free to reach out through the github forum if you are interested in write access. The username is open and the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

https://yangcatalog.org

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

☑ Module Name	☑ Node Name	☑ Node Description

YANG Versions

☑ 1.0	☑ 1.1

Schema Types

☑ All

☑ Typedef	☑ Grouping	☑ Feature
☑ Identity	☑ Extension	☑ RPC
☑ Container	☑ List	☑ Leaf-List
☑ Leaf	☑ Notification	☑ Action

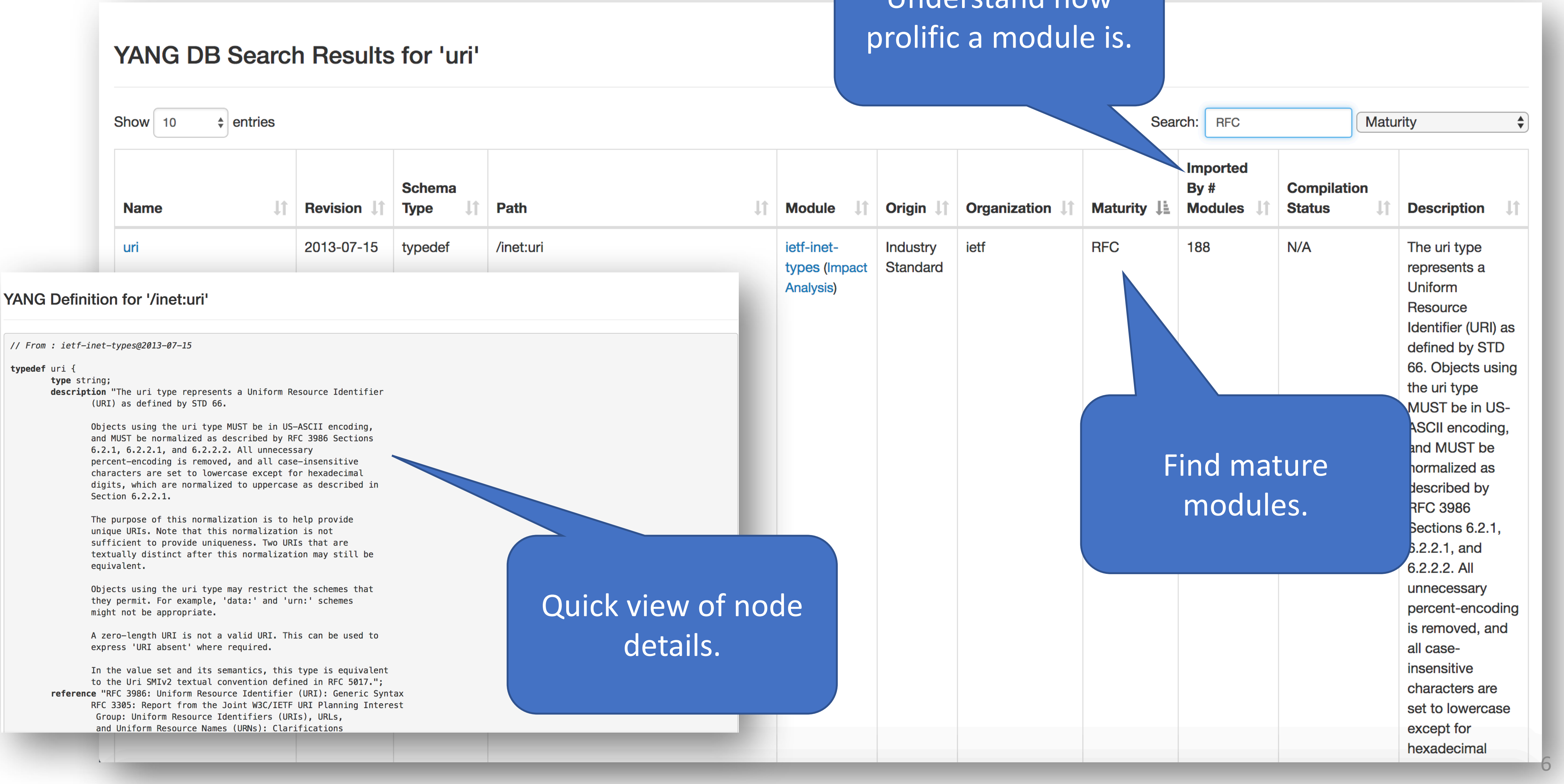
Keyword and regex searches supported

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

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

Find modules by YANG version

Search Results



YANG DB Search Results for 'uri'

Show 10 entries

Search: RFC Maturity

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

YANG Definition for '/inet:uri'

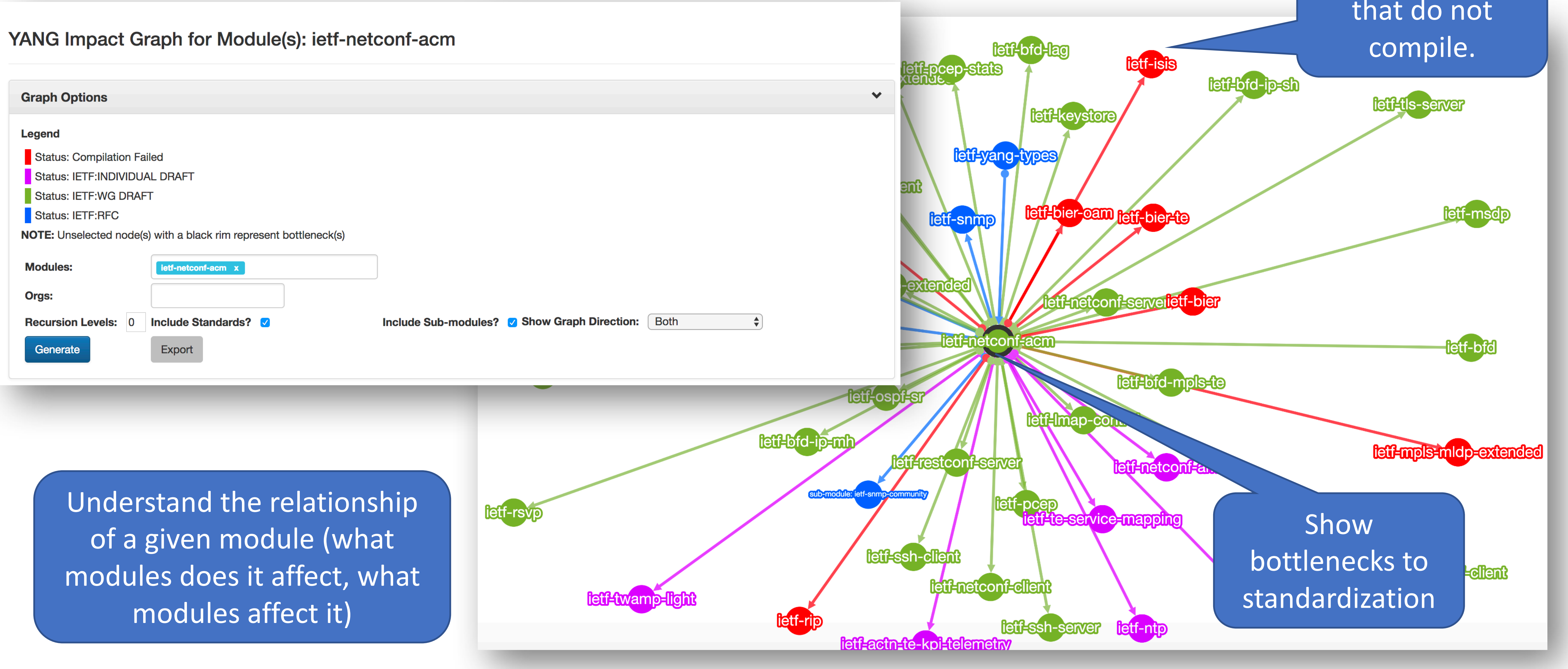
```
// From : ietf-inet-types@2013-07-15

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 URI 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

Graph Options

Legend

- 🟥 Status: Compilation Failed
- 🟪 Status: IETF:INDIVIDUAL DRAFT
- 🟩 Status: IETF:WG DRAFT
- 🟦 Status: IETF:RFC

NOTE: Unselected node(s) with a black rim represent bottleneck(s)

Modules: ietf-netconf-acm x

Orgs:

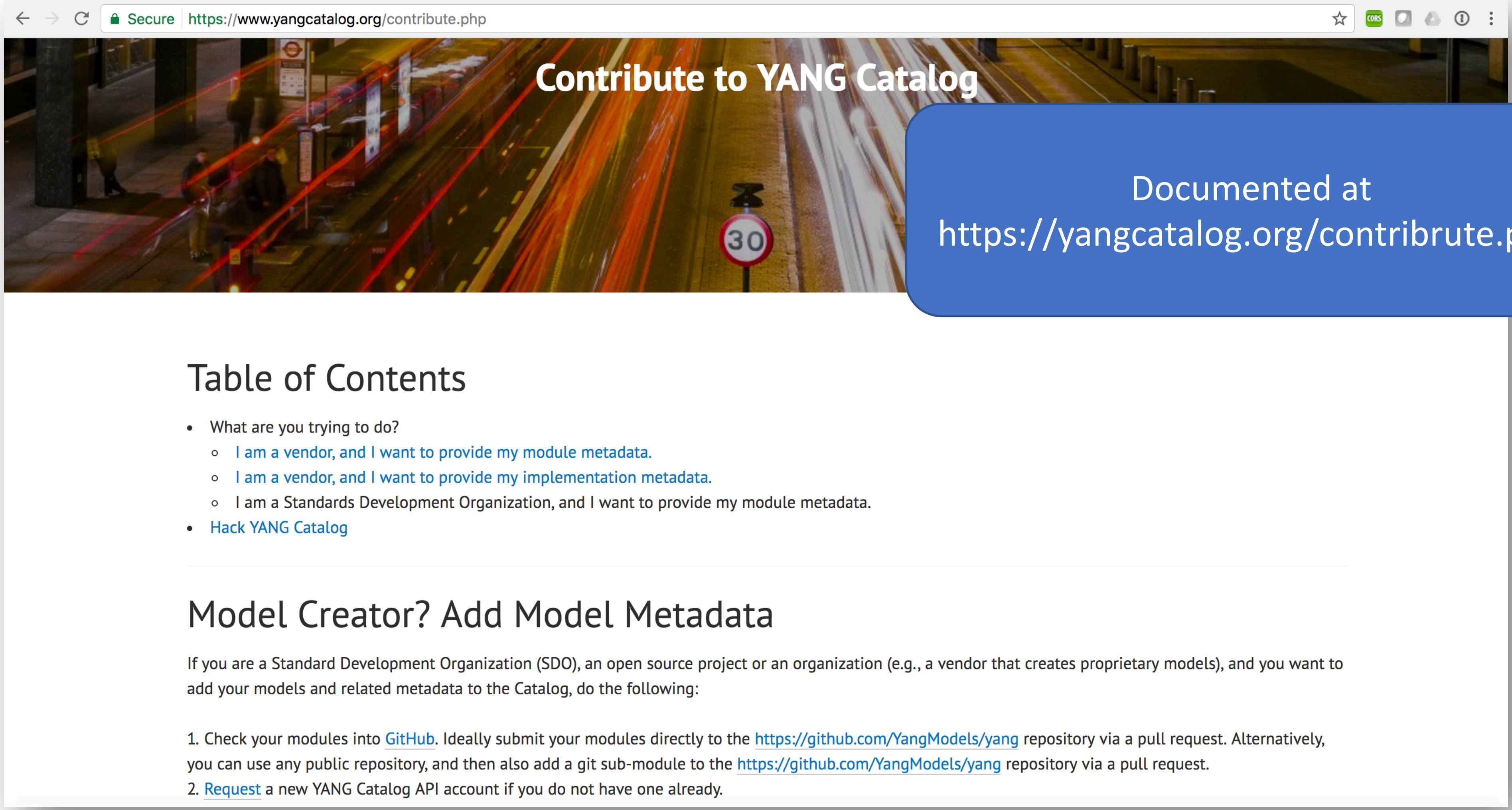
Recursion Levels: 0 **Include Standards?** ☑ **Include Sub-modules?** ☑ **Show Graph Direction:** Both

Identify modules that do not compile.

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

Show bottlenecks to standardization

The YANG Catalog API



Contribute to YANG Catalog

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

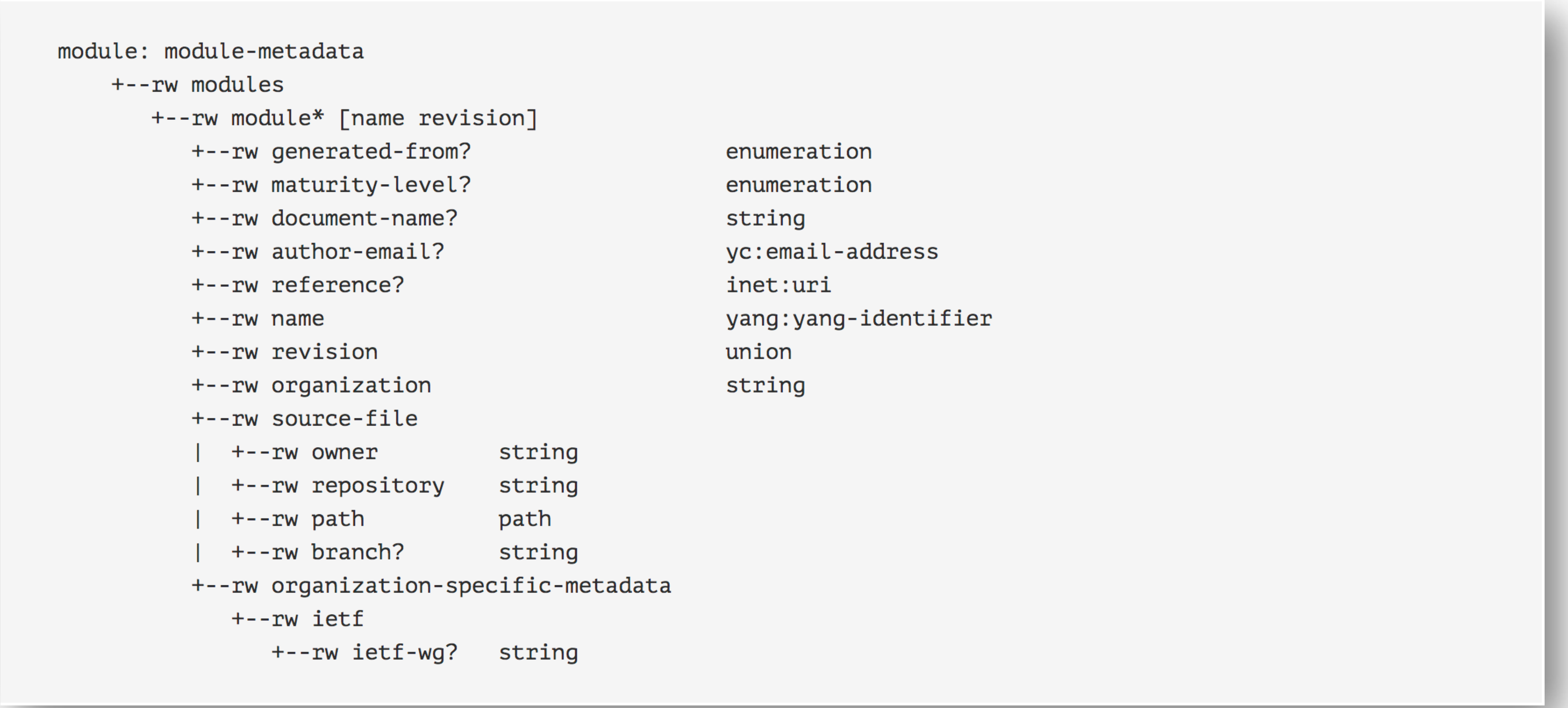
Model Creator? Add Model Metadata

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.

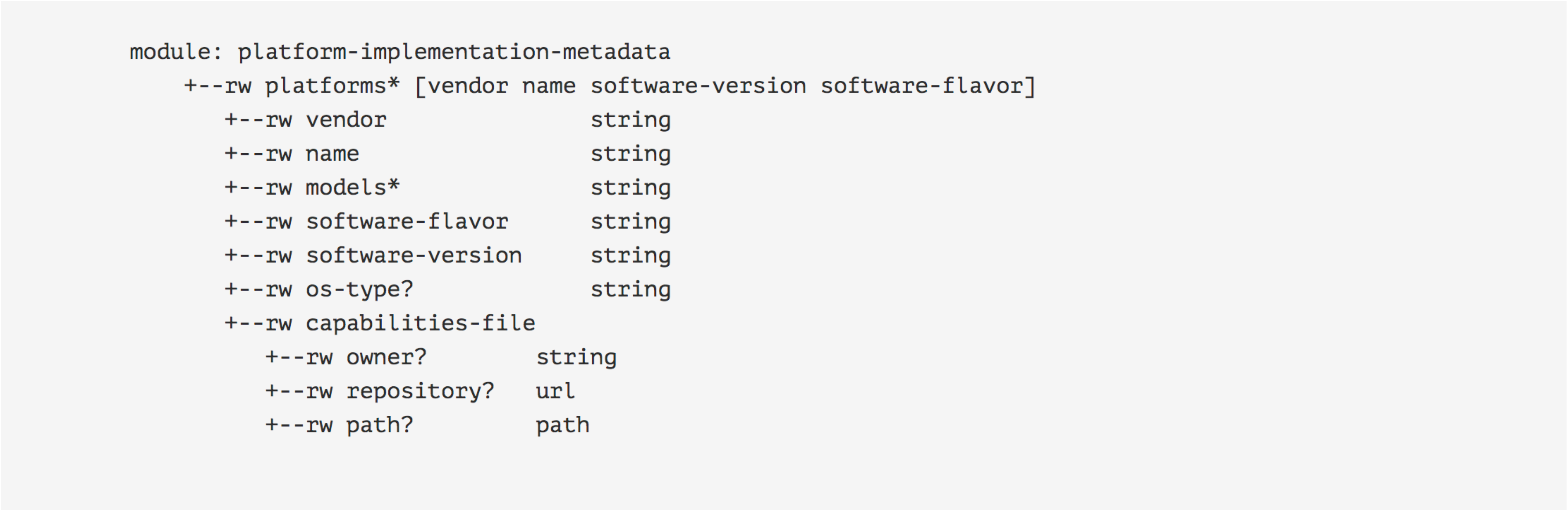
Documented at
https://yangcatalog.org/contribrute.php

Module Metadata API Structure



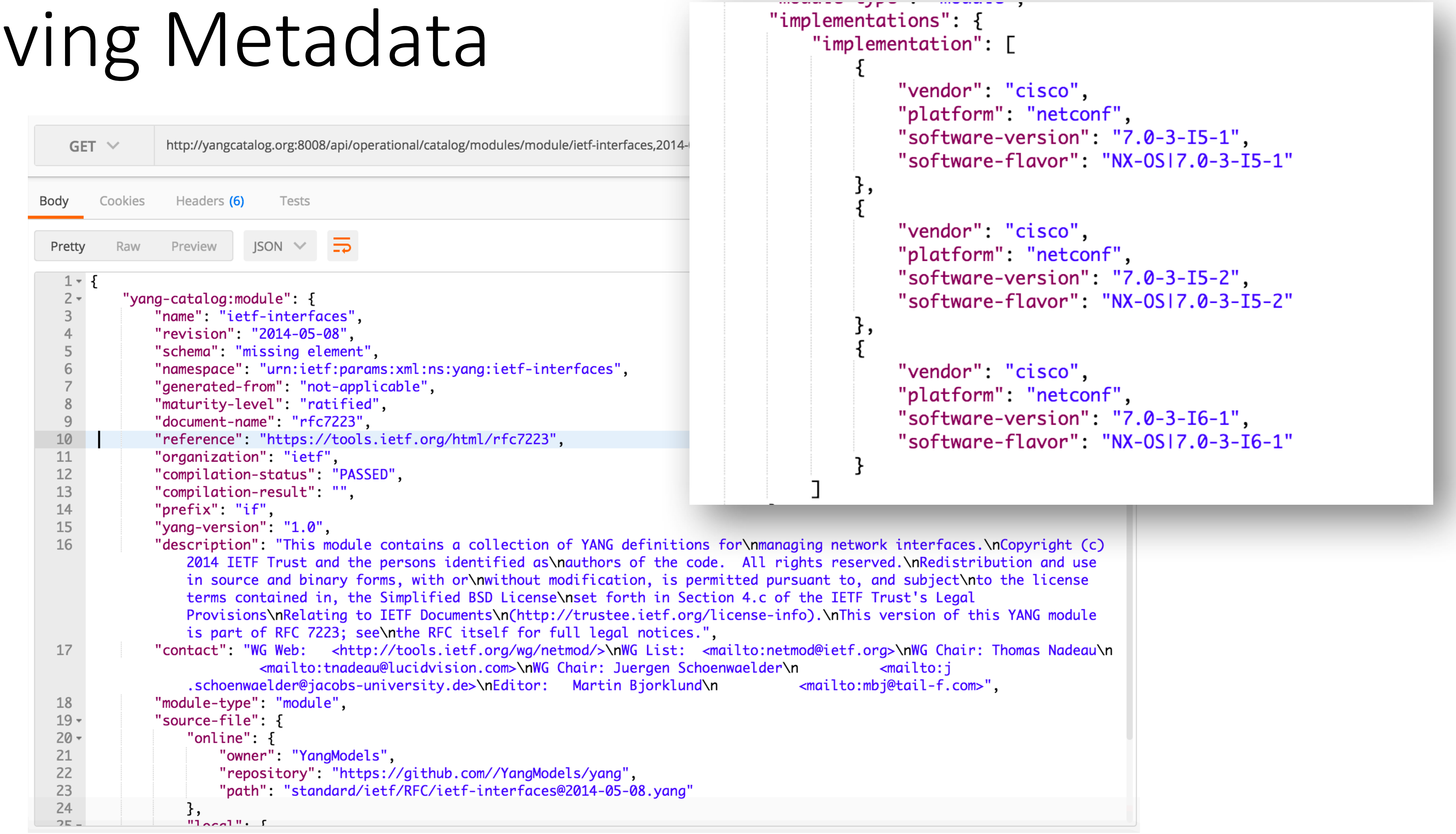
```
module: module-metadata
    +--rw modules
       +--rw module* [name revision]
          +--rw generated-from?                   enumeration
          +--rw maturity-level?                   enumeration
          +--rw document-name?                    string
          +--rw author-email?                     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

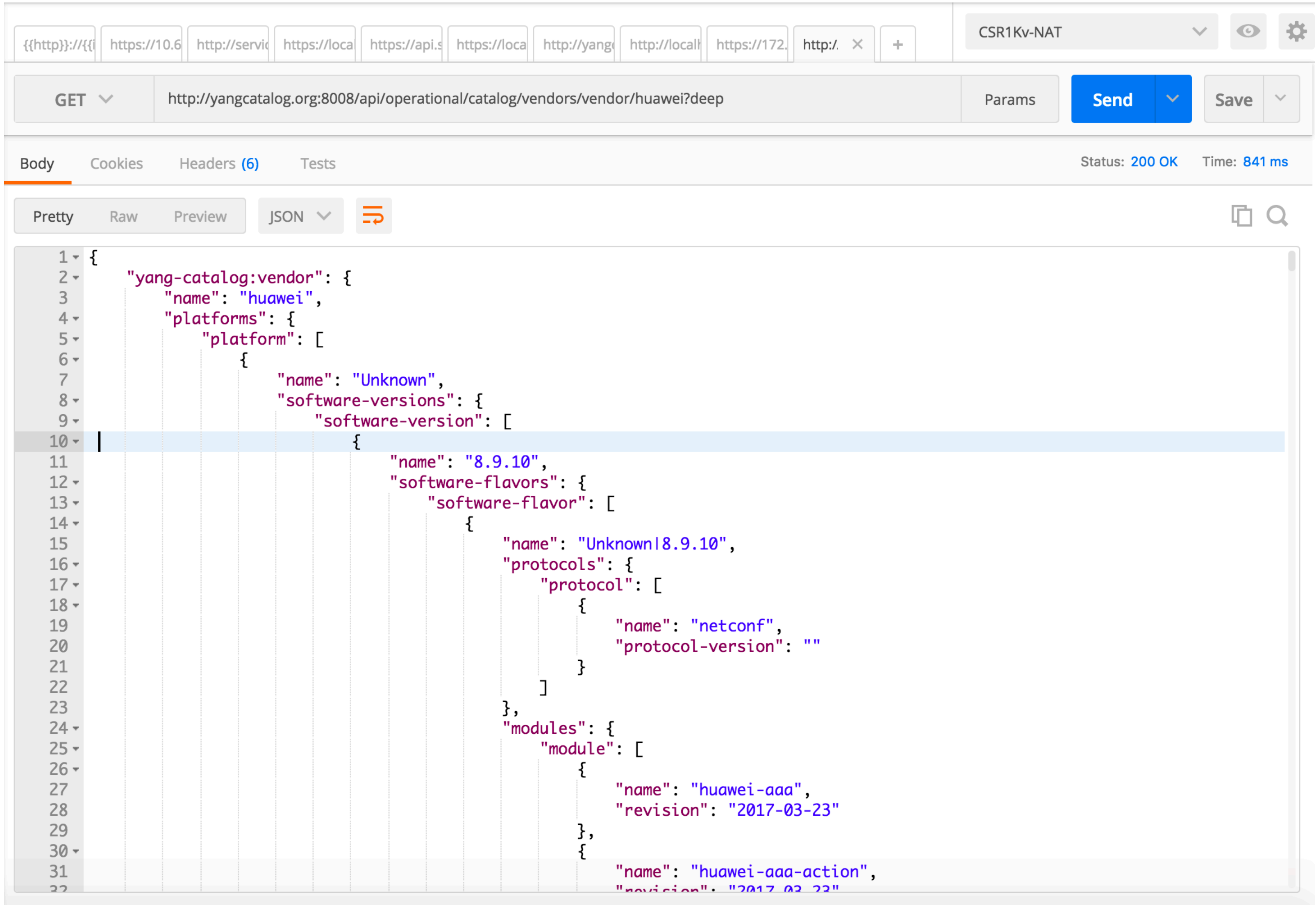


GET
http://yangcatalog.org:8008/api/operational/catalog/modules/module/ietf-interfaces,2014-
Body
Cookies
Headers (6)
Tests
Pretty
Raw
Preview
JSON

```
{
    "yang-catalog:module": {
        "name": "ietf-interfaces",
        "revision": "2014-05-08",
        "schema": "missing element",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
        "generated-from": "not-applicable",
        "maturity-level": "ratified",
        "document-name": "rfc7223",
        "reference": "https://tools.ietf.org/html/rfc7223",
        "organization": "ietf",
        "compilation-status": "PASSED",
        "compilation-result": "",
        "prefix": "if",
        "yang-version": "1.0",
        "description": "This module contains a collection of YANG definitions for\nmanaging network interfaces.\nCopyright (c)
            2014 IETF Trust and the persons identified as\nauthors of the code.  All rights reserved.\nRedistribution and use
            in source and binary forms, with or\nwithout modification, is permitted pursuant to, and subject\nto the license
            terms contained in, the Simplified BSD License\nset forth in Section 4.c of the IETF Trust's Legal
            Provisions\nRelating to IETF Documents\n(http://trustee.ietf.org/license-info).\nThis version of this YANG module
            is part of RFC 7223; see\nthe RFC itself for full legal notices.",
        "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>",
        "module-type": "module",
        "source-file": {
            "online": {
                "owner": "YangModels",
                "repository": "https://github.com//YangModels/yang",
                "path": "standard/ietf/RFC/ietf-interfaces@2014-05-08.yang"
            },
            "local": {
```

```
        "module-type": "module",
        "implementations": {
            "implementation": [
                {
                    "vendor": "cisco",
                    "platform": "netconf",
                    "software-version": "7.0-3-I5-1",
                    "software-flavor": "NX-OS|7.0-3-I5-1"
                },
                {
                    "vendor": "cisco",
                    "platform": "netconf",
                    "software-version": "7.0-3-I5-2",
                    "software-flavor": "NX-OS|7.0-3-I5-2"
                },
                {
                    "vendor": "cisco",
                    "platform": "netconf",
                    "software-version": "7.0-3-I6-1",
                    "software-flavor": "NX-OS|7.0-3-I6-1"
                }
            ]
```

Retrieving Metadata



CSR1Kv-NAT

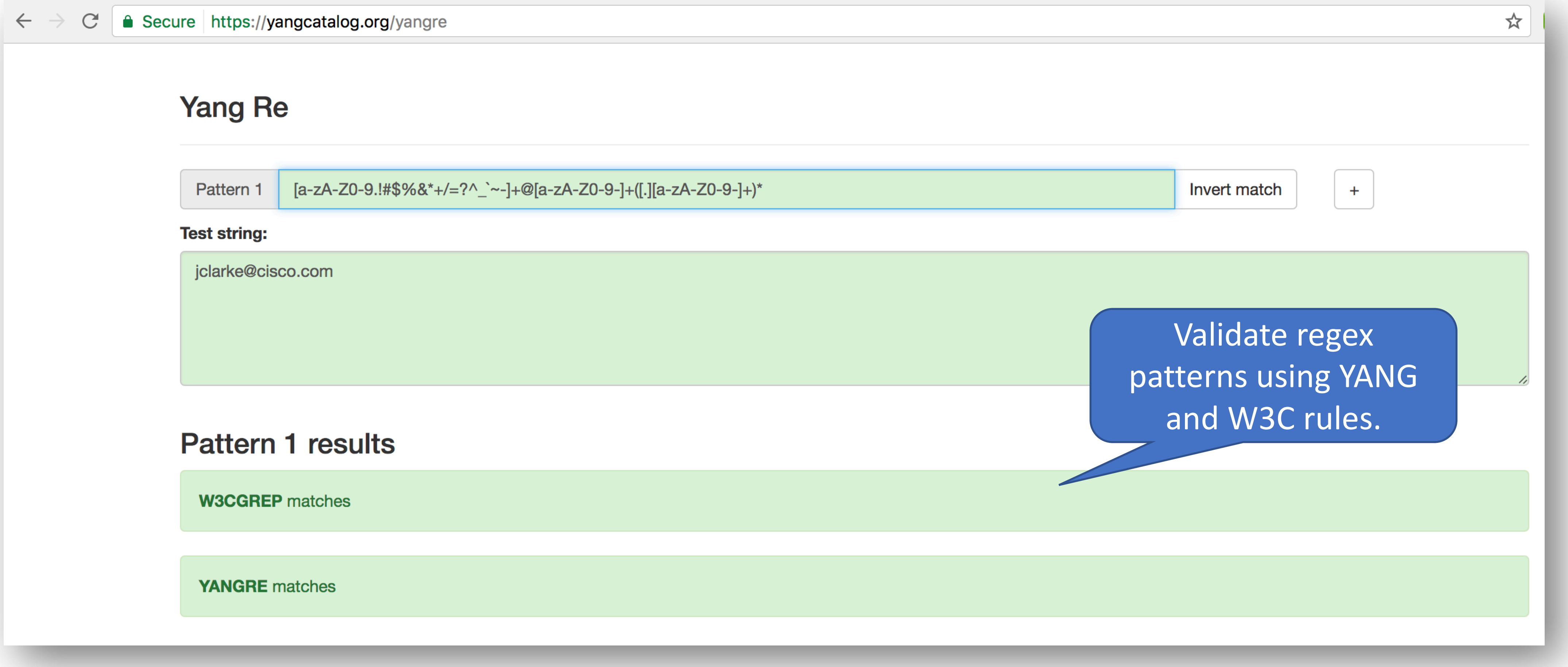
GET http://yangcatalog.org:8008/api/operational/catalog/vendors/vendor/huawei?deep

Status: 200 OK Time: 841 ms

```
{
    "yang-catalog:vendor": {
        "name": "huawei",
        "platforms": {
            "platform": [
                {
                    "name": "Unknown",
                    "software-versions": {
                        "software-version": [
                            {
                                "name": "8.9.10",
                                "software-flavors": {
                                    "software-flavor": [
                                        {
                                            "name": "Unknown|8.9.10",
                                            "protocols": {
                                                "protocol": [
                                                    {
                                                        "name": "netconf",
                                                        "protocol-version": ""
                                                    }
                                                ]
                                            },
                                            "modules": {
                                                "module": [
                                                    {
                                                        "name": "huawei-aaa",
                                                        "revision": "2017-03-23"
                                                    },
                                                    {
                                                        "name": "huawei-aaa-action",
                                                        "revision": "2017-03-23"
```

Hackathon 99

YANG Regular Expression Validator



Secure https://yangcatalog.org/yangre

Yang Re

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

Test string:

jclarke@cisco.com

Pattern 1 results

W3CGREP matches

YANGRE matches

Validate regex patterns using YANG and W3C rules.

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