



DIGITAL SERVICE AT CMS

CMS Code.json: The journey behind creating a metadata standard

Natalia Luzuriaga + OSPO // September 24, 2024

The Journey

May - Today!

1. The Idea 
2. repometrics 
3. code.json 
4. publiccode.yml 
5. Implementation + Demo 



The Idea



→ CMS Source Code Stewardship Long-term Vision



We know where our repos are.

Inventory all the components using GitHub



We know who is working on them.

Understand who is maintaining and contributing to projects, and how often.



We know what is inside them.

Identifying FISMA labels, repository topics, maturity models, security vulnerabilities, and library composition

4 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>



→ CMS Source Code Stewardship Long-term Vision



We know where our repos are.

Inventory all the components using GitHub



We know who is working on them.

Understand who is maintaining and contributing to projects, and how often.



We know what is inside them.

Identifying FISMA labels, repository topics, maturity models, security vulnerabilities, and library composition

5 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>

OSAB Q2 Meeting

Source Code Stewardship Taskforce: Metrics Website

CMS.gov Open Source Repository Metrics

The Centers for Medicare and Medicaid Services is comprised of many GitHub Organizations.

 **CMS-Enterprise**
56 30

 **CMSgov**
230 30

 **DSACMS**
Digital Service at CMS
22 12

 **Enterprise-CMCS**
Center for Medicaid & CHIP Services
88 30

CMSgov

AB2D-Libs

1 0 0 373 8

CMCS-DSG-DSS-Certification

This is the PROD repo. Commits made to the main branch of the staging repo (<https://github.com/CMSgov/CMCS-DSG-DSS-Certification-Staging>) will be automatically merged in and deployed here. Please open Issues and Pull Requests in the Staging repo instead.

8 2 0 10 13

CMCS-DSG-DSS-Certification-Staging

(This is the STAGING repo.) Welcome to the MES Certification Repository, a collaborative community for CMS, states, and vendors. For more information about the repository, and how to use it, take a look at the ReadMe section.

10 7 63 130 12

Report for metrics

project type midsize

Repo Stats

Stars	Forks	Issues	Watchers	Pull Requests
5	2	17	3	113

Summary Table

Metric	Latest	Previous	Diff	% Diff
Commits	568	538	30	5.4%
Issues	17	15	2	12%
Open Issues	7	6	1	15%
Closed Issues	10	9	1	11%
Open Pull Requests	11	9	2	20%
Merged Pull Requests	95	87	8	8.8%
Closed Pull Requests	7	7	0	0%
Forks	2	2	0	0%
Stars	5	5	0	0%
Watchers	3	3	0	0%

dsacms.github.io/metrics/

6 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>



OSAB Q2 Meeting

Source Code Stewardship Taskforce

- CXO Ariele Faber:
 - What types of projects are you seeing across repos? Are there buckets certain projects fall under?
 - Project “Bucket”: operational, policy, software
 - Customer / User Type: patients, providers, government
 - Subset in Healthcare:
 - Project Type: Utility, B2B, etc.
- We realized that we don't have this type of information/data on our repos at the moment

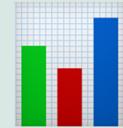
7 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>



That got us thinking...



We should collect
metadata on our repos!



→ Brainstorming Sesh 🧠

What information about a project should we collect from project teams?

This is information we can't get from an API or is not on the repo themselves.

- FISMA Level
- User Type
- Home Department / Org / Group that owns project



Brainstorming Sesh 🧠

How should we go about doing this data collection?

1. Collect facts about a repo directly from the project team
2. Store this information in a file in the repository
3. Use this file to display this data on our metrics website reports

¹¹ <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>



Brainstorming Sesh 🧠

How is this metadata helpful? 🌄

- Metadata allows us to:
 1. Identify & understand the purpose of software
 2. Classify it based on a variety of ways
- As a OSPO, how we can best serve the needs of our development teams and community?
- By getting data on all of our repos, we can provide high level insights on all our repos at CMS
 - How many projects have a medium FISMA level?
 - How many projects are websites?
 - How many projects are written in Python?
 - Contractors vs internal



repometrics.json

Research stage + our initial attempt

→ **repometrics.json**

- We decided to store this metadata in a file called **repometrics.json**
- Deciding on the fields
 - What information is unavailable through an API or isn't included in the repository itself?
 - What key information should we ask project teams about the repository that would be useful to know?



repometrics.json fields

Field	Type	Options	Notes
repo_type	string Multiple Choice (MC)	Package, Website, Standards, Libraries, Data, Applications, Tools, API	Type of "Thing", considering renaming to type -> format
user_input: does it accept user input? (e.g. Allows user to query a database, allows login by users, etc.)	boolean MC	Yes, No	OpenSSF Scorecard Requirement
repository_host: where is the source code hosted?	string MC	github.com: CMSgov, CMS-Enterprise github.cms.gov, CCSQ GitHub	
fisma_level	string MC	Low, Moderate, High	
group: Home Department / Org / Group	string Free Response (FR)	CMS/OA/DSAC	Could make MC
maturity_model_tier	number MC	1, 2, 3, 4	
project/initiative***	string FR	Bluebutton, PECOS	Based on Meg's findings, a project usually has multiple repositories
status***	string MC	Active, Archived, Deprecated, Empty	Can automate
user_type***	string Multiple Select (MS)	Providers, Patients, Government	Customers of software
subset_in_healthcare***	string MS	Policy, Operational, Medicare, Medicaid	Still figuring out options!



repometrics.json Initial Schema

```
{  
    "projectType": "Tools",  
    "userInput": "Yes",  
    "fismaLevel": "Moderate",  
    "group": "CMS/OA/DSAC",  
    "subsetInHealthcare": "Operational", ← repometrics.json  
    "userType": "Providers",  
    "repositoryHost": "Github.com",  
    "maturityModelTier": "3"  
}
```

of DedupliFHIR

Options:

```
2     "project_type" : ["Package", "Website", "Standards", "Libraries", "Data", "Apps", "Tools", "APIs"],  
3     "user_input": ["Yes", "No"],  
4     "project_fisma_level": ["Low", "Moderate", "High"],  
5     "group": "CMS/OA/DSAC",  
6     "subset_in_healthcare": "Policy, Operational",  
7     "user_type": "Providers, Patients, Government",  
8     "repository_host": ["Github.com", "GitHub ENT", "GitHub Cloud", "GitLab.com", "GitLab ENT", "GitLab ENT CCSQ"],  
9     "maturity_model_tier": ["1", "2", "3", "4"],
```



Research

Meg did a deep dive into github.com/CMSgov, home to 210 repos

Github categories for code.json metadata

7/3/2024

★★ Repo shows in more than one category below

Implementation guides



Design standards

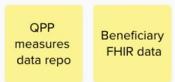


Code compliance



???

Data repositories



APIs



Contracts data



Findings

- There are common project formats / types
 - Technical: APIs, Web apps, Data Repositories
 - Design: Design systems, Style guides
 - Docs: Implementation guides, API docs, Policy
- Multiple repos for a single project/initiative
 - Bluebutton has 10+ repos for APIs, web server, docs, and sample projects



Incorporating other metadata standards

code.json



Our federal-wide metadata standard

● M-16-21: Mandated Agency Software Inventory

- As part of the [Federal Source Code Policy](#) released in 2016, agencies were required to publish an inventory of their custom-developed code to make source code easier to find and support reuse of software across the federal government

²¹ <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>



Code.gov

- A website made by GSA that serves as a centralized catalog where federal agencies can list and share their custom-developed software

code_



code.json

→ How do agencies go about this?

1. Agency makes a code.json for each of their repos. This is a project level code.json

```
"name": "Animal Disease Spread Model",
"description": "A simulation of disease spread in livestock populations.
Includes detection and containment simulation.",
"repositoryURL": "https://github.com/NAVADMC/ADSM",
"permissions": {
  "licenses": [
    {
      "URL": "https://github.com/NAVADMC/ADSM/blob/master/LICENSE",
      "name": "BSD-2-Clause"
    }
  ],
  "usageType": "openSource"
},
"laborHours": 2838,
"tags": [
  "USDA",
  "source code",
  "animal",
  "disease",
  "herds",
  "livestock",
  "simulation",
  "model"
],
"contact": {
  "email": "Melissa.Schoenbaum@aphis.usda.gov",
  "name": "Missy Schoenbaum"
},
"status": "Beta",
"vcs": "git",
"languages": [
  "python"
],
"homepageURL": "https://github.com/NAVADMC/ADSM",
"date": {
  "created": "",
  "lastModified": "2018-01-11",
  "metadataLastUpdated": "2018-03-27"
}
```

code.json:
USDA's Animal Disease
Spread Model

Optional Object Fields

→ How do agencies go about this?

2. Use a code.json generator tool to collect info from files and curate the agency-wide code.json inventory

```
{  
  "agency": "USDA",  
  "version": "2.0.0",  
  "measurementType": {  
    "method": "projects"  
  },  
  "releases": [  
    {  
      "organization": "OC",  
      "name": "DigitalGov Analytics",  
      "description": "A Drupal Module that implements all features defined within the DAP Implementation Guide by exposing an admin configuration page to edit each resource reference customization.",  
      "repositoryURL": "https://www.drupal.org/project/gov_analytics",  
      "permissions": {  
        "licenses": [  
          {  
            "URL": "http://www.gnu.org/licenses/old-licenses/gpl-2.0.html",  
            "name": "GPL-2.0"  
          }  
        ],  
        "usageType": "openSource"  
      },  
      "laborHours": 16,  
      "tags": [  
        "Drupal",  
        "DigitalGov",  
        "analytics",  
        "DAP"  
      ],  
      "contact": {  
        "email": "Source.Code@ocio.usda.gov",  
        "name": "USDA Source Code"  
      },  
      "status": "Production",  
      "vcs": "git",  
      "languages": [  
        "Drupal",  
        "php"  
      ],  
      "homepageURL": "https://www.drupal.org/project/gov_analytics",  
      "date": {  
        "created": "2015-05-29",  
        "lastModified": "2015-06-01",  
        "metadataLastUpdated": "2019-06-06"  
      },  
    }  
  ]  
}
```

USDA's code.json



→ How do agencies go about this?

3. Agency submits their code.json to code.gov

The screenshot shows the 'AGENCIES' section of the code.gov website. At the top, there are navigation links for 'AGENCIES' and 'GUIDANCE'. Below this, a large blue header box contains the word 'AGENCIES' in white. Underneath, there's a table with two columns. The left column, titled 'Federal Agency', lists several agency names with checkboxes next to them. The right column displays the 'Department of Agriculture' entry, which includes its name, a link to its Code.JSON file, and a list of its organizations.

Federal Agency	
<input type="checkbox"/> Agency for International Development	i For inquiries into agency source code, please refer to the agency liaison list .
<input type="checkbox"/> Department of Agriculture	Department of Agriculture
<input type="checkbox"/> Department of Commerce	Code.JSON: https://www.usda.gov/code.json
<input type="checkbox"/> Department of Defense	Organizations: GrainGenes , usda , usda-ars-agil , usda-ars-gbru , usda-ars-ltar , usda-ars-nwrc , usda-ars-wmsru , USDA-ERS , USDA-FSA , USDA-NIFA , usda-vs , usdaForestService , WFMRDA



→ How do agencies go about this?

4. GSA checks for compliance



Department of Agriculture

Yellow square icon: Agency policy is being updated for consistency with the Federal Source Code Policy.

Red square icon: Agency has not updated acquisition language.

Red square icon: Agency has not completed inventory.



— This is our report card 😭

Only 6 out of 24 agencies have fully completed and submitted their inventory



Department of Health and Human Services

■ Agency policy is consistent with the Federal Source Code Policy.

■ Agency has not updated acquisition language.

■ Agency has not completed inventory.



**code.json / agency inventory
is no longer actively
maintained 😔**

But that got us thinking...



**Let's bring it back to CMS
bay-bee!!!**



publiccode.yml



Mr. Worldwide of software metadata standards

Metadata at the International Level?!

In OSPOs for Good Symposium hosted by the United Nations, Remy learned about **publiccode.yml**, a worldwide metadata description standard for public software.

Goal: Making public software easily discoverable and reusable by other entities

```
1 publiccodeYmlVersion: "0.4"
2
3 name: Medusa
4 url: "https://example.com/italia/medusa.git"
5 platforms:
6   - web
7
8 categories:
9   - financial-reporting
10
11 developmentStatus: development
12
13 softwareType: "standalone/desktop"
14
15 description:
16   en:
17     shortDescription: >
18       A rather short description which
19       is probably useless
20
21     longDescription: >
22       Very long description of this software, also split
23       on multiple rows. You should note what the software
24       is and why one should need it. We can potentially
25       have many pages of text here.
26
27   features:
28     - Just one feature
29
30 legal:
31   license: AGPL-3.0-or-later
32
33 maintenance:
34   type: "community"
35
36 contacts:
37   - name: Francesco Rossi
38
39 localisation:
40   localisationReady: true
41   availableLanguages:
42     - en
```

[publiccode.yml schema](#)



→ Supports Country-Specific Fields

- Designed to be interoperable internationally, thus the country-specific keys are separated by the core part and are defined in specific sections that each government can rule.
- Italy 
 - Mandatory for all public software developed in Italy
 - Building a national public catalog
- Germany 
 - German registry of open source

2.1. Italy

All the keys listed below are specific for Italy and, as such, they must be inserted in a section named with the `it` code. Every Country is specified using a two letters *country code* following the ISO 3166-1 alpha-2 standard.

2.1.2. Key `conforme`

This section contains the keys for auto-declaring the compliance with the current legislation, with respect to the following sections. Not including these keys implies that the compliance is not known or not declared.

2.1.2.1. Key `conforme/lineeGuidaDesign`

- Type: boolean
- Presence: optional

If present and set to `true`, the software is compliant with the Italian accessibility laws (L. 4/2004), as further explained in the [linee guida di design](#) (Italian language).

2.1.2.2. Key `conforme/modelloInteroperabilita`

- Type: boolean
- Presence: optional

If present and set to `true`, the software is compliant with the [linee guida sull'interoperabilità](#).

Regulatory reference: [Art. 73 del CAD](#) (Italian language).

[Docs \(yml.publiccode.tools\)](#)



That got us thinking...

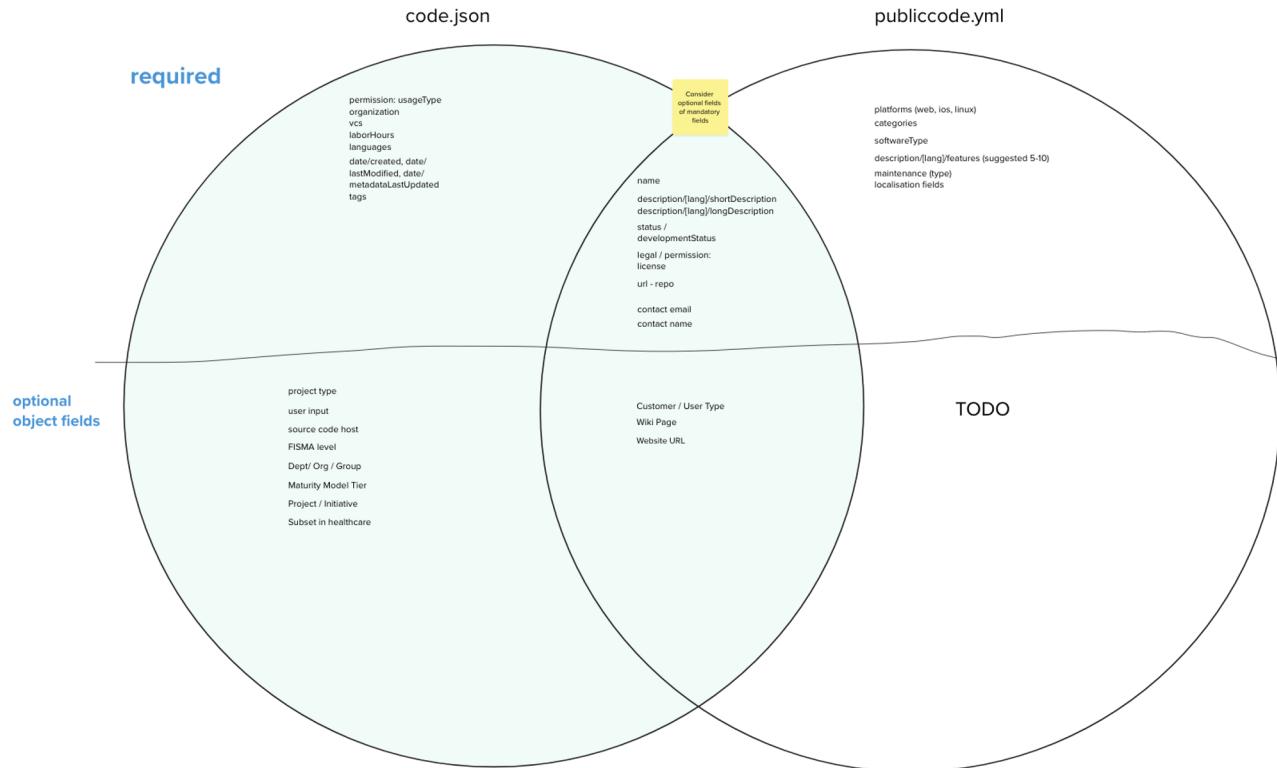


We can have our repos
comply on both a national
and international level! 😮

Putting it all together 🤝

We are cooking 🥚

Compare & Contrast



Common Fields

```
{  
    "organization": "OC",  
    "name": "DigitalGov Analytics",  
    "description": "A Drupal Module that implements all features defined within the  
DAP Implementation Guide by exposing an admin configuration page to edit each  
resource reference customization.",  
    "repositoryURL": "https://www.drupal.org/project/gov_analytics",  
    "permissions": {  
        "licenses": [  
            {  
                "URL": "http://www.gnu.org/licenses/old-licenses/gpl-2.0.html",  
                "name": "GPL-2.0"  
            }  
        ],  
        "usageType": "openSource"  
    },  
    "laborHours": 16,  
    "tags": [  
        "Drupal",  
        "DigitalGov",  
        "analytics",  
        "DAP"  
    ],  
    "contact": {  
        "email": "Source.Code@ocio.usda.gov",  
        "name": "USDA Source Code"  
    },  
    "status": "Production",  
    "vcs": "git",  
    "languages": [  
        "Drupal",  
        "php"  
    ],  
    "homepageURL": "https://www.drupal.org/project/gov_analytics",  
    "date": {  
        "created": "2015-05-29",  
        "lastModified": "2015-06-01",  
        "metadataLastUpdated": "2019-06-06"  
    },  
}
```

39 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>

```
1 publiccodeYmlVersion: "0.4"  
2  
3 name: Medusa  
4 url: "https://example.com/italia/medusa.git"  
5 platforms:  
6   - web  
7  
8 categories:  
9   - financial-reporting  
10  
11 developmentStatus: development  
12  
13 softwareType: "standalone/desktop"  
14  
15 description:  
16   en:  
17     shortDescription: >  
18       A rather short description which  
19       is probably useless  
20  
21     longDescription: >  
22       Very long description of this software, also split  
23       on multiple rows. You should note what the software  
24       is and why one should need it. We can potentially  
25       have many pages of text here.  
26  
27     features:  
28       - Just one feature  
29  
30     legal:  
31       license: AGPL-3.0-or-later  
32  
33 maintenance:  
34   type: "community"  
35  
36 contacts:  
37   - name: Francesco Rossi  
38  
39 localisation:  
40   localisationReady: true  
41   availableLanguages:  
42     - en
```



publiccode.yml fields



```
{  
    "organization": "OC",  
    "name": "DigitalGov Analytics",  
    "description": "A Drupal Module that implements all features defined within the  
DAP Implementation Guide by exposing an admin configuration page to edit each  
resource reference customization.",  
    "repositoryURL": "https://www.drupal.org/project/gov_analytics",  
    "permissions": {  
        "licenses": [  
            {  
                "URL": "http://www.gnu.org/licenses/old-licenses/gpl-2.0.html",  
                "name": "GPL-2.0"  
            }  
        ],  
        "usageType": "openSource"  
    },  
    "laborHours": 16,  
    "tags": [  
        "Drupal",  
        "DigitalGov",  
        "analytics",  
        "DAP"  
    ],  
    "contact": {  
        "email": "Source.Code@ocio.usda.gov",  
        "name": "USDA Source Code"  
    },  
    "status": "Production",  
    "vcs": "git",  
    "languages": [  
        "Drupal",  
        "php"  
    ],  
    "homepageURL": "https://www.drupal.org/project/gov_analytics",  
    "date": {  
        "created": "2015-05-29",  
        "lastModified": "2015-06-01",  
        "metadataLastUpdated": "2019-06-06"  
    }  
},  
40 https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf
```

```
1 publiccodeYmlVersion: "0.4"  
2  
3 name: Medusa  
4 url: "https://example.com/italia/medusa.git"  
5 platforms:  
6   - web  
7  
8 categories:  
9   - financial-reporting  
10  
11 developmentStatus: development  
12  
13 softwareType: "standalone/desktop"  
14  
15 description:  
16   en:  
17     shortDescription: >  
18       A rather short description which  
19       is probably useless  
20  
21     longDescription: >  
22       Very long description of this software, also split  
23       on multiple rows. You should note what the software  
24       is and why one should need it. We can potentially  
25       have many pages of text here.  
26  
27   features:  
28     - Just one feature  
29  
30 legal:  
31   license: AGPL-3.0-or-later  
32  
33 maintenance:  
34   type: "community"  
35  
36 contacts:  
37   - name: Francesco Rossi  
38  
39 localisation:  
40   localisationReady: true  
41   availableLanguages:  
42     - en
```



code.json fields



```
{  
  "organization": "OC",  
  "name": "DigitalGov Analytics",  
  "description": "A Drupal Module that implements all features defined within the  
DAP Implementation Guide by exposing an admin configuration page to edit each  
resource reference customization.",  
  "repositoryURL": "https://www.drupal.org/project/gov_analytics",  
  "permissions": {  
    "licenses": [  
      {  
        "URL": "http://www.gnu.org/licenses/old-licenses/gpl-2.0.html",  
        "name": "GPL-2.0"  
      }  
    ],  
    "usageType": "openSource"  
  },  
  "laborHours": 16,  
  "tags": [  
    "Drupal",  
    "DigitalGov",  
    "analytics",  
    "DAP"  
  ],  
  "contact": {  
    "email": "Source.Code@ocio.usda.gov",  
    "name": "USDA Source Code"  
  },  
  "status": "Production",  
  "vcs": "git",  
  "languages": [  
    "Drupal",  
    "php"  
  ],  
  "homepageURL": "https://www.drupal.org/project/gov_analytics",  
  "date": {  
    "created": "2015-05-29",  
    "lastModified": "2015-06-01",  
    "metadataLastUpdated": "2019-06-06"  
  }  
}
```

41 <https://github.com/DSACMS/decks/blob/main/cms-code-json2024.pdf>

```
1 publiccodeYmlVersion: "0.4"  
2  
3 name: Medusa  
4 url: "https://example.com/italia/medusa.git"  
5 platforms:  
6   - web  
7  
8 categories:  
9   - financial-reporting  
10  
11 developmentStatus: development  
12  
13 softwareType: "standalone/desktop"  
14  
15 description:  
16   en:  
17     shortDescription: >  
18       A rather short description which  
19       is probably useless  
20  
21     longDescription: >  
22       Very long description of this software, also split  
23       on multiple rows. You should note what the software  
24       is and why one should need it. We can potentially  
25       have many pages of text here.  
26  
27     features:  
28       - Just one feature  
29  
30 legal:  
31   license: AGPL-3.0-or-later  
32  
33 maintenance:  
34   type: "community"  
35  
36 contacts:  
37   - name: Francesco Rossi  
38  
39 localisation:  
40   localisationReady: true  
41   availableLanguages:  
42     - en
```



Implementation

Workflow

1

Create the repo

Project team uses repo-scaffolder to create a new repo that complies with our repo hygiene standards.

2

Generate code.json

When project team is ready to outbound project, they can use cookiecutter to generate code.json file.

3

Ready for use

Our metrics websites can grab this info and display on our reports.



Introducing CMS' code.json V0 (minimum version)

CMS' code.json

Field	Type	Where from?	Notes
name	string Free Response (FR)	🇺🇸 🌎	we retrieve this from previous cookiecutter variables
description/[lang]/shortDescription	string FR	🌐	Max 150 characters
description/[lang]/longDescription	string FR	🌐	Between 150 to 1,000 characters
status	string Multiple Choice (MC)	🇺🇸	Development status of project: "Ideation", "Development", "Alpha", "Beta", "Release Candidate", "Production", "Archival"
permissions/license/url permissions/license/name	array MC	🇺🇸 🌎	"CC0-1.0", "Apache-2.0", "MIT", "MPL-2.0", "GPL-2.0-only", "GPL-3.0-only", "GPL-3.0-or-later", "LGPL-2.1-only", "LGPL-3.0-only", "BSD-2-Clause", "BSD-3-Clause", "EPL-2.0", "Other"
organization	string 🌐	🇺🇸	Automated, value is "Centers for Medicare & Medicaid Services"
vcs (version control system)	string MC	🇺🇸	"git", "hg", "svn", "rcs", "bzr"
laborHours	number 🌐	🇺🇸	Automated through use of COCOMO & SCC tool
platforms	string Multiple Select (MS)	🌐	"web", "windows", "mac", "linux", "ios", "android", "other"
categories	string MS	🌐	List of options found here: https://yml.publiccode.tools/categories-list.html



CMS' code.json cont.

Field	Type	Where from?	Notes
softwareType	string MC		"standalone/mobile", "standalone/iot", "standalone/desktop", "standalone/web", "standalone/backend", "standalone/other", "addon", "library", "configurationFiles"
languages	array FR		programming languages that make up the codebase
maintenance	string MC		"internal", "contract", "community", "none"
date/created date/lastModified date/metaDataLastUpdated	timestamp		Automated! We don't ask
tags	array FR		self-reported tags
contact/email contact/name	string FR		Save info on point of contact
localisation	boolean MC		whether the project supports multiple languages



CMS' code.json: agency-specific fields

Field	Type	Options / Examples	Notes
name	string Multiple Choice (MC)	Package, Website, Standards, Libraries, Data, Applications, Tools, API	Type of "Thing", considering renaming to type -> format
user_input: does it accept user input? (e.g. Allows user to query a database, allows login by users, etc.)	boolean MC	Yes, No	OpenSSF Scorecard Requirement
repository_host: where is the source code hosted?	string MC	github.com: CMSgov, CMS-Enterprise github.cms.gov, CCSQ GitHub	
fisma_level	string MC	Low, Moderate, High	
group: Home Department / Org / Group	string Free Response (FR)	CMS/OA/DSAC	Could make MC
maturity_model_tier	number MC	1, 2, 3, 4	
project/initiative***	string FR	Bluebutton, PECOS	Based on Meg's findings, a project usually has multiple repositories
Status***	string MC	Active, Archived, Deprecated, Empty	Can automate
user_type***	string Multiple Select (MS)	Providers, Patients, Government	Customers of software
subset_in_healthcare***	string MS	Policy, Operational, Medicare, Medicaid	Still figuring out options!



Code.json V0 Minimum Version Schema

```
{  
  "name": "metrics",  
  "description": {  
    "en": [  
      {"shortDescription": "A website with reports on CMS' repositories",  
       "longDescription": "The CMS Repository Metrics Website shows an overview of software development activity"}  
    ],  
    "status": "production",  
    "permissions": {  
      "licenses": [  
        {"URL": "LICENSE",  
         "name": "CC0-1.0"}  
      ]  
    },  
    "organization": "Centers for Medicare & Medicaid Services",  
    "repositoryURL": "https://github.com/DSACMS/metrics",  
    "vcs": "git",  
    "laborHours": "250",  
    "platforms": [ "web" ],  
    "categories": [ "healthcare" ],  
    "softwareType": "standalone/web",  
    "languages": [ "Python, HTML, CSS, Javascript" ],  
    "maintenance": "internal",  
    "date": {  
      "lastModified": "2024-09-24T06:56:53+0000",  
      "metadataLastUpdated": "2024-09-24T06:56:53+0000"  
    },  
    "tags": [ "metrics, oss, visualizations, data, activity" ],  
    "contact": {  
      "email": "opensource@cms.hhs.gov",  
      "name": "CMS Open Source Program Office"  
    },  
    "localisation": "false",  
    "repoType": "Website",  
    "userInput": "No",  
    "fismaLevel": "Low",  
    "group": "CMS/OA/DSAC",  
    "subsetInHealthcare": "Operational",  
    "userType": "Government",  
    "repositoryHost": "Github.com",  
    "maturityModelTier": "1"  
}
```



→ Intern Rotation completed by Ricardo!

Backend:

- Ask questions and populate code.json using cookiecutter

Frontend:

- Tags in reports on the metrics website



DEMO DEMO DEMO



```
l9a1@JXD9TH5QCW repo-scaffolder % cookiecutter . --directory=tier1
[1/8] project_name (My Project): party-planning-website
[2/8] project_slug (party-planning-website):
[3/8] project_org (DSACMS):
[4/8] project_repo_name (party-planning-website):
[5/8] project_description (This is the project description, could match github.com repo description.): Website for planning parties
[6/8] Select project_visibility
  1 - public
  2 - internal
  3 - private
Choose from [1/2/3] (1): 3
[7/8] Would you like to create a repo on github.com?
  1 - True
  2 - False
Choose from [1/2] (1): 
```

Created party-planning-website
using repo-scaffolder Tier 1 repository



```
④ l9a1@JXD9TH5QCW .github % cookiecutter . --directory=repometrics
[1/24] Project name: (party-planning-website):
[2/24] GitHub organization: (DSACMS):
[3/24] Provide a short description of the software. It should be a single line containing a single sentence. Maximum 150 characters a
description of the project.):
[4/24] Provide longer description of the software, between 150 and 10000 chars. It is meant to provide an overview of the capabilitie
potential user. (A longer description of the project.):
[5/24] Which subset of healthcare does the project belong to? (Policy, Operational): ^C
Aborted!
○ l9a1@JXD9TH5QCW .github % cookiecutter . --directory=repometrics
[1/24] Project name: (party-planning-website):
[2/24] GitHub organization: (DSACMS):
[3/24] Provide a short description of the software. It should be a single line containing a single sentence. Maximum 150 characters a
description of the project.: Website for planning parties
[4/24] Provide longer description of the software, between 150 and 10000 chars. It is meant to provide an overview of the capabilitie
potential user. (A longer description of the project.): Doing a demonstration of repo-scaffolder and code.json work for the DSAC onsite
[5/24] Which subset of healthcare does the project belong to? (Policy, Operational): Operational
[6/24] Who are the intended users? (Providers, Patients, Government): Government
[7/24] Does the project accept user input? (e.g. allows user to query a database, allows login by users, upload files, etc.)
  1 - Yes
  2 - No
Choose from [1/2] (1): 2
```

Created code.json on party-planning-website repo using repo-scaffolder



DSACMS / party-planning-website

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

party-planning-website Public

main · 1 Branch · 0 Tags

Go to file Add file Code About

Website for planning parties

Readme CCO-1.0 license Code of conduct Security policy Activity Custom properties 0 stars 0 watching 0 forks Report repository

Releases
No releases published Create a new release

Packages
No packages published Publish your first package

party-planning-website

Website for planning parties

About the Project

This is a demo project to demonstrate the code.json functionality integration with cookiecutter and the DSACMS metrics website.

Core Team

A full list of contributors can be found on <https://github.cms.gov/DSACMS/party-planning-website/graphs/contributors>.

party-planning-website / code.json

natalialuzuriaga Added code.json and updated README.md ad1c48a · 4 days ago 2 Commits

Code Blame 45 lines (45 loc) · 1.2 KB

```

1  {
2   "name": "party-planning-website",
3   "description": {
4     "en": {
5       "shortDescription": "Website for planning parties!",
6       "longDescription": "This website was created for demoing at the DSAC onsite."
7     }
8   },
9   "status": "development",
10  "permissions": {
11    "licenses": [
12      {
13        "URL": "LICENSE",
14        "name": "CC0-1.0"
15      }
16    ],
17  },
18  "organization": "Centers for Medicare & Medicaid Services",
19  "repositoryURL": "https://github.com/DSACMS/party-planning-website",
20  "vcs": "git",
21  "laborHours": "180",
22  "platforms": [ "web" ],
23  "categories": [ "web-collaboration" ],
24  "softwareType": "standalone/web",
25  "languages": [ "HTML", "CSS", "JS" ],
26  "maintenance": "internal",
27  "date": {
28    "lastModified": "2024-09-23T20:36:43+0000",
29    "metadataLastUpdated": "2024-09-23T20:36:43+0000"
30  },
31  "tags": [ "party", "planning", "dsac" ],
32  "contact": {
33    "email": "opensource@cms.hhs.gov",
34    "name": "CMS Open Source Program Office"
35  },
36  "localisation": "false",
37  "projectType": "Website",
38  "userInput": "Yes",
39  "fismaLevel": "Low",
40  "group": "OMS/OA/DSAC",
41  "subsetInHealthcare": "Operational",
42  "userType": "Government",
43  "repositoryHost": "Github.com",
44  "maturityModelTier": "1"
45 }

```

Created party-planning-website repo and code.json using repo-scaffolder





CMS.gov Open Source Repository Metrics

Home Organizations Projects

The Centers for Medicare and Medicaid Services develops and supports many open-source projects found in our various GitHub Organizations.

Search Sort by

Organization
<input type="checkbox"/> DSACMS
<input type="checkbox"/> CMS-Enterprise
<input type="checkbox"/> CMSgov
<input type="checkbox"/> Enterprise-CMCS

Maturity Model Tier

DSACMS

party-planning-website
Website for planning parties

0 0 0 0 0

CMS.gov Open Source Repository Metrics

Home Organizations Projects

Report for party-planning-website

GITHUB.COM CMS/OA/DSAC

WEBSITE TIER 1 LOW ^



Repo Stats

★ Stars	🍴 Forks	⌚ Issues	⌚ Watchers	↗ Pull Requests
0	0	0	0	0

Summary Table

Metric	Latest	Previous	Diff	% Diff
Commits	2	2	0	0%
Issues	0	0	0	0%
Open Issues	0	0	0	0%
Closed Issues	0	0	0	0%

party-planning-website repo is on metrics website with report!



The dream .



Next steps

1. Create a minimum version of CMS' project-level code.json. Incorporate into reposcaffolder for project teams to create code.json file on their own ✓
2. Add optional fields from code.json & publiccode.yml to create an extended version of CMS' project-level code.json
3. Finalize our agency-specific fields
4. Add code.json to all repos in CMS. Integrate into our public metrics site & internal metrics site.
5. OSPO generates an updated version of agency-level code.json inventory & submits to GSA



→ Code.json fields: laborHours & languages

- Automatic Calculation of Labor Hours using Sloc Cloc and Code (SCC)
- Automatic Calculation of Language using Sloc Cloc and Code (SCC)
- **What is COCOMO?**
 - COnstructive COst MOdeling
 - # of Lines of Code * average salary
- **How do we Calculate it per project?**
 - Sloc, Cloc and Code: scc is a very fast accurate code counter with complexity calculations and COCOMO estimates



→ Inventory Enables Key Metrics

- **Maturity Metrics:**

- Tiering of public repositories quantify *security* and *continuity* risk.

- **Burden Metrics:**

- Quantify Time/Effort needed to contribute or improve repositories
 - e.g. Time-to-first-response/Time-to-resolution of Issues/Pull Requests

- **Cost Metrics:**

- Quantify Time/Effort invested into repositories
 - e.g. COCOMO (Constructive Cost Modeling) Metrics for Repositories



The Dream: Planetary Public Sector Software Inventory



SUSTAINABLE DEVELOPMENT GOALS

United Nations Sustainable Development Goals

- “Which SDG does this project address?”

Digital Public Goods Alliance

- Designation Form



DPGA Standard

Indicator	Requirement
1. Relevance to Sustainable Development Goals	Digital public goods must demonstrate relevance to advancing the Sustainable Development Goals (SDGs)
2. Use of Approved Open Licenses	Digital public goods must demonstrate the use of an approved open license. For open-source software, only OSI approved licenses are accepted. For open content collections the use of a Creative Commons license is required. DPGs are encouraged to use a license that allows for both derivatives and commercial reuse (CC-BY and CC-BY-SA), or dedicate content to the public domain (CC0); licenses that do not allow for commercial reuse (CC-BY-NC and CC-BY-NC-SA) are also accepted. For open data, an Open Data Commons approved license is required. See The full license list for reference.
3. Clear Ownership	Ownership of assets that the digital public good produces must be clearly defined and documented. For example, through copyright, trademark or other publicly available information.
4. Platform Independence	When the digital public good has mandatory dependencies that create more restrictions than the original license, proving independence from the closed component(s) and/or indicating the existence of functional, open alternatives that can be used without significant changes to the core product is required.
5. Documentation	Digital public goods require documentation of the source code, use cases, and/or functional requirements. For content collections, this should include all relevant/compatible apps, software, or hardware required to access the content collection, and instructions regarding how to use it. For software solutions, this should be technical documentation that would allow a technical person unfamiliar with the project to launch and run the software. For data projects, this should be documentation that describes all the fields in the set, and provides context on how the dataset was collected, and how it should be interpreted.
6. Mechanism for Extracting Data	Digital public goods with non-personally identifiable information (PII) design for possibility of extracting or importing non-PII data and content from the system in a non-proprietary format.
7. Adherence to Privacy and Applicable Laws	Digital public goods must be designed and developed to comply with privacy and other applicable laws.
8. Adherence to Standards & Best Practices	Digital public goods must be designed and developed to align with relevant standards, best practices, and/or principles. For example, the Principles for Digital Development .
9. Do No Harm by Design	Digital public goods must be designed to anticipate, prevent, and do no harm by design.
9.a) Data Privacy & Security	Digital public goods that collect, store and distribute personally identifiable data , must demonstrate how they ensure the privacy, security and integrity of this data in addition to the steps taken to prevent adverse impacts resulting from its collection, storage and distribution.
9.b) Inappropriate & Illegal Content	Digital public goods that collect, store or distribute content must have policies identifying inappropriate and illegal content such as child sexual abuse materials in addition to processes for detecting, moderating, reporting and removing inappropriate/ illegal content.
9.c) Protection from Harassment	If the digital public good facilitates interactions with or between users or contributors there must be a process for users and contributors to protect themselves against grief, abuse, and harassment. The project must have system(s) to address the safety and security of underage users.



→ The Dream: Planetary Public Sector Software Inventory



First we start with an inventory of CMS, then HHS, then all of USA, then...

We unite Hackers on Planet Earth to create a Grand Unified Inventory of all open source software, categorized by:

- What SDGs they address,
- What languages they support,
- What purposes they serve,
- Which nations are working together to maintain them,
- What is the common dependencies graph,
- And whatever other metadata will help!



Feedback from team.  

**What fields would you like to
see on code.json?**



CMS specific fields: what we have so far

Field	Type	Options	Notes
repo_type	string Multiple Choice (MC)	Package, Website, Standards, Libraries, Data, Applications, Tools, API	Type of "Thing", considering renaming to type -> format
user_input: does it accept user input? (e.g. Allows user to query a database, allows login by users, etc.)	boolean MC	Yes, No	OpenSSF Scorecard Requirement
repository_host: where is the source code hosted?	string MC	github.com: CMSgov, CMS-Enterprise github.cms.gov, CCSQ GitHub	
fisma_level	string MC	Low, Moderate, High	
group: Home Department / Org / Group	string Free Response (FR)	CMS/OA/DSAC	Could make MC
maturity_model_tier	number MC	1, 2, 3, 4	
Project/initiative***	string FR	Bluebutton, PECOS	Based on Meg's findings, a project usually has multiple repositories
Status***	string MC	Active, Archived, Deprecated, Empty	Can automate
user_type***	string Multiple Select (MS)	Providers, Patients, Government	Customers of software
subset_in_healthcare***	string MS	Policy, Operational, Medicare, Medicaid	Still figuring out options!



Thank you!