



Jsonnet



Alexandra CHATON - INFO5



POLYTECH[®]
GRENOBLE



Un peu d'histoire

- Le nom Jsonnet vient de **JSON** et *sonnet*
- Jsonnet a été lancé le 6 août 2014

Aujourd'hui

- Version : v0.16.0
- GitHub :

 [google / jsonnet](#)

 Watch

110

 Star

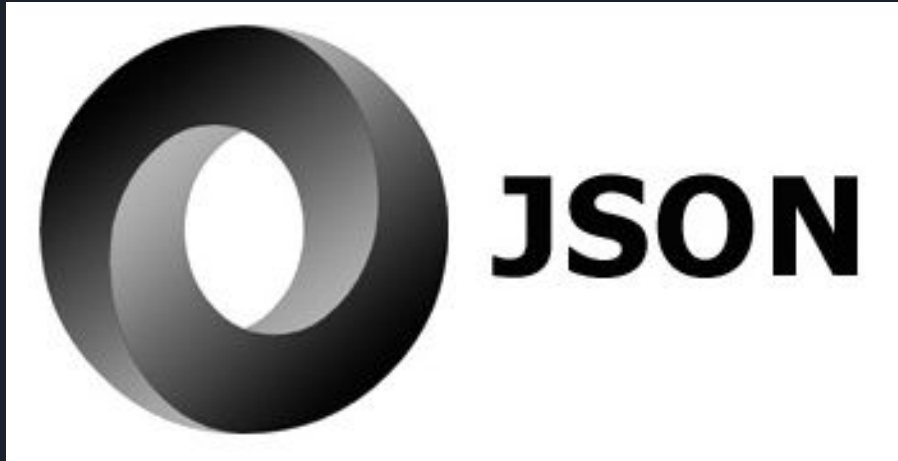
4.5k

 Fork

319

Qu'est ce que Jsonnet ?

- Langage de création de modèle de données (data templating language)
- Une extension de JSON
- Génère des fichiers au format JSON



```
{  
  "employees": [  
    {  
      "firstName": "John",  
      "lastName": "Doe"  
    },  
    {  
      "firstName": "Anna",  
      "lastName": "Smith"  
    },  
    {  
      "firstName": "Peter",  
      "lastName": "Jones"  
    }  
  ]  
}
```

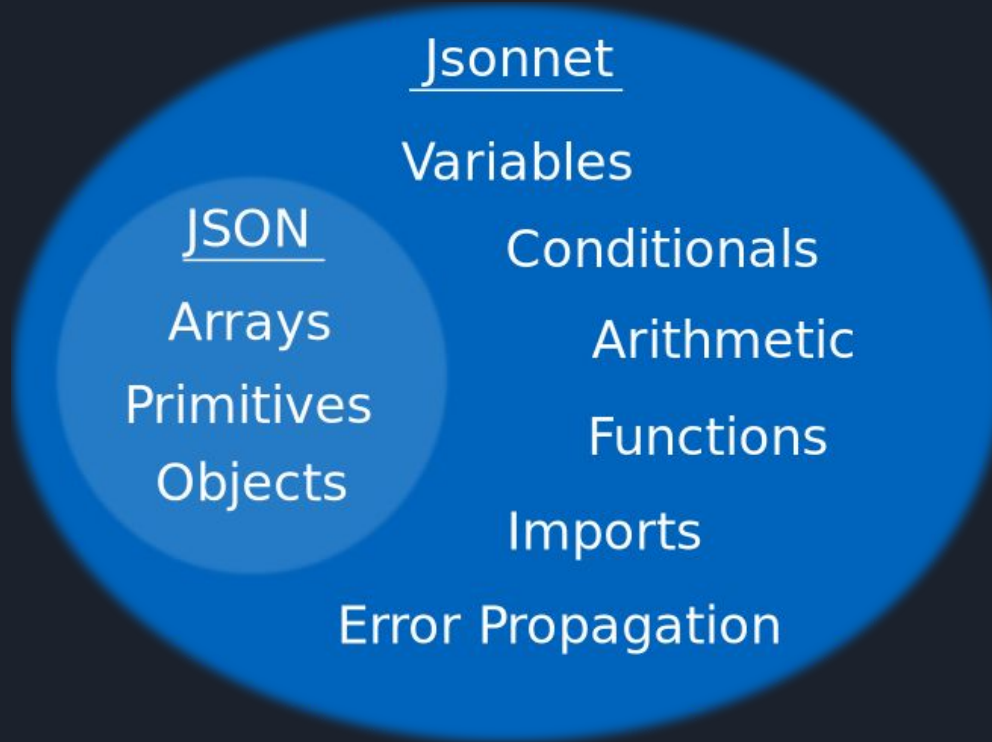
Comparaison JSON vs Jsonnet

- Syntaxe
- Variables
- Fonctions
- Opérations
- Conditions
- Références
- Imports
- Et bien plus ...

```
{
  Martini: {
    local drink = self,
    ingredients: [
      { kind: "Farmer's Gin", qty: 1 },
      {
        kind: 'Dry White Vermouth',
        qty: drink.ingredients[0].qty,
      },
    ],
    garnish: 'Olive',
    served: 'Straight Up',
  },
}
```

```
# Edit me!
local b = import "b.libsonnet"; # comment
local a = import "a.libsonnet";
```

Comparaison JSON vs Jsonnet



→ JSON = décrit les données littéralement

→ Jsonnet plus court que JSON
→ Utile pour les systèmes complexes

→ fichier JSON ⇒ fichier Jsonnet

fichier JSON ⇏ fichier Jsonnet

A quels problèmes répond Jsonnet ?

JSON très populaire MAIS difficile à gérer ⇒ création de scripts qui génèrent du JSON

- Jsonnet veut faciliter cette génération et a donc été conçu selon les critères suivants :
Modularité, Familiarité, Puissant mais simple, Rigueur formelle

Pour quelle utilisation ?

- Pour gérer les configurations des ressources de **Kubernetes**



Autres caractéristiques



example3.jsonnet

```
1 local application = 'my-app';
2 local module = 'uwsgi_module';
3 local dir = '/var/www';
4 local permission = 644;
5
6
7 {
8   'uwsgi.ini': std.manifestIni({
9     sections: {
10      uwsgi: {
11        module: module,
12        pythonpath: dir,
13        socket: dir + '/uwsgi.sock',
14        'chmod-socket': permission,
15        callable: application,
16        logto: '/var/log/uwsgi/uwsgi.log',
17      },
18    },
19  }),
20 }
```

cassandra.conf

init.sh

uwsgi.ini

```
[uwsgi]
callable = my-app
chmod-socket = 644
logto = /var/log/uwsgi/uwsgi.log
module = uwsgi_module
pythonpath = /var/www
socket = /var/www/uwsgi.sock
```

Comparaison avec CUE



Configure Unify Execute

- CUE plus jeune
- CUE gère l'intégration de nombreux formats de données
- CUE fait pour l'automatisation donc va aussi gérer la validation de schémas
- Toutefois la notation est très strict et peu proche du JSON

Create

The builtin `encoding/json.Marshal` generates JSON from within CUE.

```
import "encoding/json"
configMap: data: "point.json":
  json.Marshal({
    x: 4.5
    y: 2.34
  })
```

```
{
  "configMap": {
    "data": {
      "point.json": "{\"x\":4.5,\"y\":2.34}"
    }
  }
}
```




Conclusion



Avantages:

- Syntaxe claire, proche du JSON mais utilisant les codes de la programmation objet
- Jsonnet fait pour du JSON + le meilleur pour JSON
- Communauté active : Kubernetes



Désavantages

- Jsonnet fait pour du JSON
- Petite communauté



Démonstration



Questions ?



Sources

<https://jsonnet.org/>

<https://github.com/google/jsonnet>

<https://collectiveidea.com/blog/archives/2018/03/06/what-s-in-a-templating-language-part-1>

<https://www.json.org/json-fr.html>

<https://kubernetes.io/fr/docs/concepts/overview/what-is-kubernetes/>

<https://databricks.com/fr/blog/2017/06/26/declarative-infrastructure-jsonnet-templating-language.html>

<https://cuelang.org/docs/integrations/json/>