

VEILLE TECHNOLOGIQUE

---

**SWAGGER**

# PLAN

- I) Vue d'ensemble sur Swagger
- II) API REST
- III) Outils Swagger
- IV) Formats Swagger
- V) Solutions alternatives
- VI) Conclusion
- VII) Démonstration

## I) VUE D'ENSEMBLE SUR SWAGGER

- Documentations
- Tests
- Générations
- d'API REST
- OpenSource



## II) API REST

- Style d'architecture
- Créé par Roy Fielding en 2000
- Utilisation de HTTP
- Plusieurs langages et plateformes
- JAX-RS en Java et Jersey

## III) OUTILS SWAGGER

- Swagger Core
- Swagger Codegen
- Swagger UI
- Swagger Editor
- Et plus ...

## IV) FORMATS SWAGGER

- JSON

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        { "value": "New", "onclick": "CreateNewDoc()" },
        { "value": "Open", "onclick": "OpenDoc()" },
        { "value": "Close", "onclick": "CloseDoc()" }
      ]
    }
  }
}
```

## IV) FORMATS SWAGGER

- YAML

```
%YAML 1.2
---
YAML: YAML Ain't Markup Language

What It Is: YAML is a human friendly data serialization
            standard for all programming languages.

YAML Resources:
  YAML 1.2 (3rd Edition): http://yaml.org/spec/1.2/spec.html
  YAML 1.1 (2nd Edition): http://yaml.org/spec/1.1/
  YAML 1.0 (1st Edition): http://yaml.org/spec/1.0/
  YAML Issues Page: https://github.com/yaml/yaml/issues
  YAML Mailing List: yaml-core@lists.sourceforge.net
  YAML IRC Channel: #yaml on irc.freenode.net
  YAML Cookbook (Ruby): http://yaml4r.sourceforge.net/cookbook/ (local)
  YAML Reference Parser: http://ben-kiki.org/ypaste/

Projects:
  C/C++ Libraries:
  - libyaml          # "C" Fast YAML 1.1
  - Syck             # (dated) "C" YAML 1.0
  - yaml-cpp        # C++ YAML 1.2 implementation
  Ruby:
  - psych           # libyaml wrapper (in Ruby core for 1.9.2)
  - RbYaml         # YAML 1.1 (PyYAML Port)
  - yaml4r         # YAML 1.0, standard library syck binding
```

## V) SOLUTIONS ALTERNATIVES

- RAML
  - ▶ Les plus : constructions des metadatas avancées
  - ▶ Les moins : manque des outils au niveau du code
- API Blueprint
  - ▶ Les plus : facile à comprendre
  - ▶ Les moins : installation complexe



## VI) CONCLUSION

- Simple
- Support
- Indispensable

# VII) DÉMONSTRATION

**MERCI POUR VOTRE ATTENTION !**

**AVEZ-VOUS DES QUESTIONS ?**

## SOURCES

- [swagger.io](https://swagger.io)
- [github.com/swagger-api/](https://github.com/swagger-api/)
- [raml.org](https://raml.org)
- [apiblueprint.org](https://apiblueprint.org)
- [nordicapis.com/top-specification-formats-for-rest-apis/](https://nordicapis.com/top-specification-formats-for-rest-apis/)