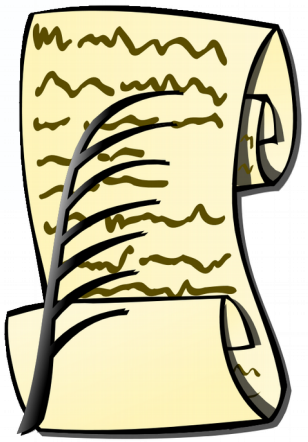
Formation Produire et consommer des web services de type REST

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

Présentation du programme



Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

Présentation du programme

I) Présentation d'une architecture REST

1) Définition
2) Les éléments d'une architecture REST

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

Présentation du programme

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

Présentation du programme

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les bonnes pratiques

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

1) Définition

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

1) Définition

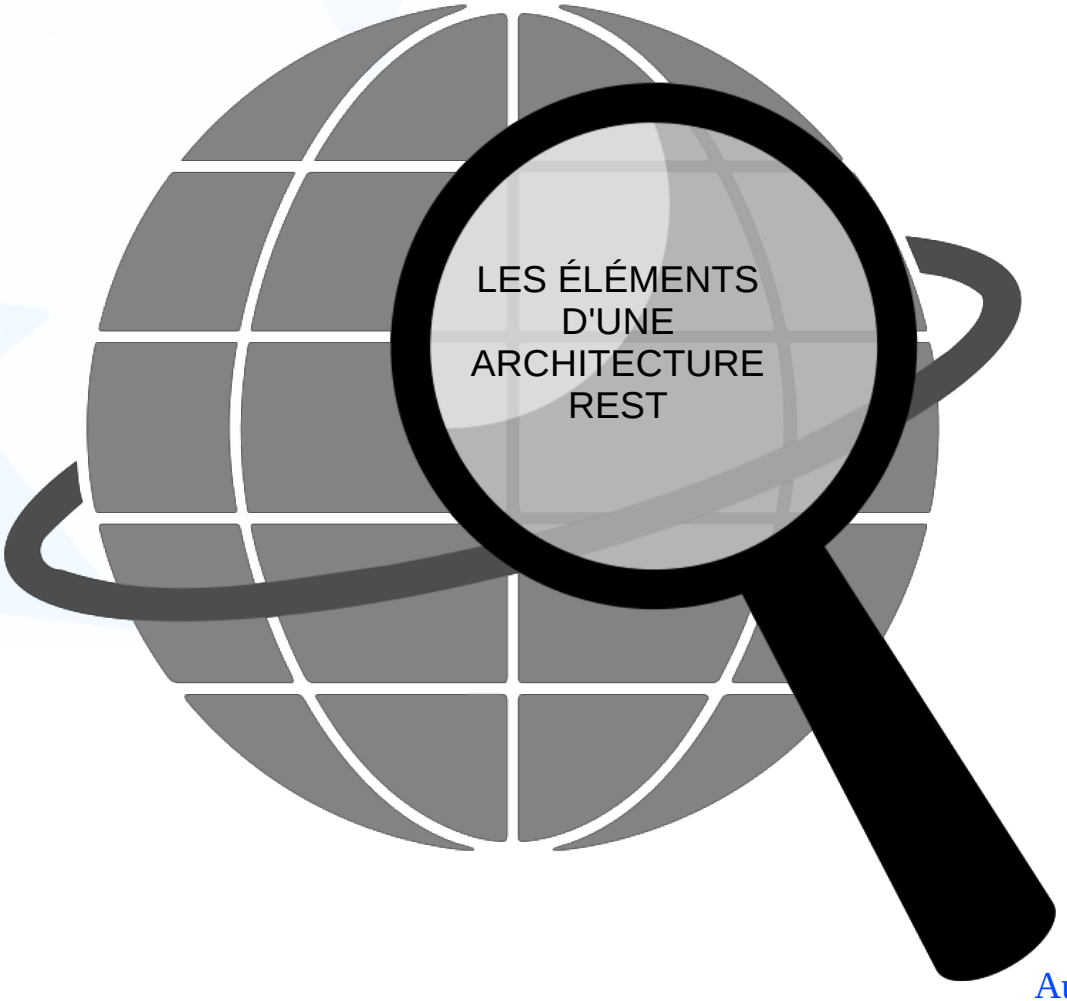
REST (REpresentational State Transfer) est un style d’architecture pour les systèmes hypermédia distribués, créé par Roy Fielding en 2000 dans le chapitre 5 de sa thèse de doctorat.

Source : http://fr.wikipedia.org/wiki/Representational_State_Transfer

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

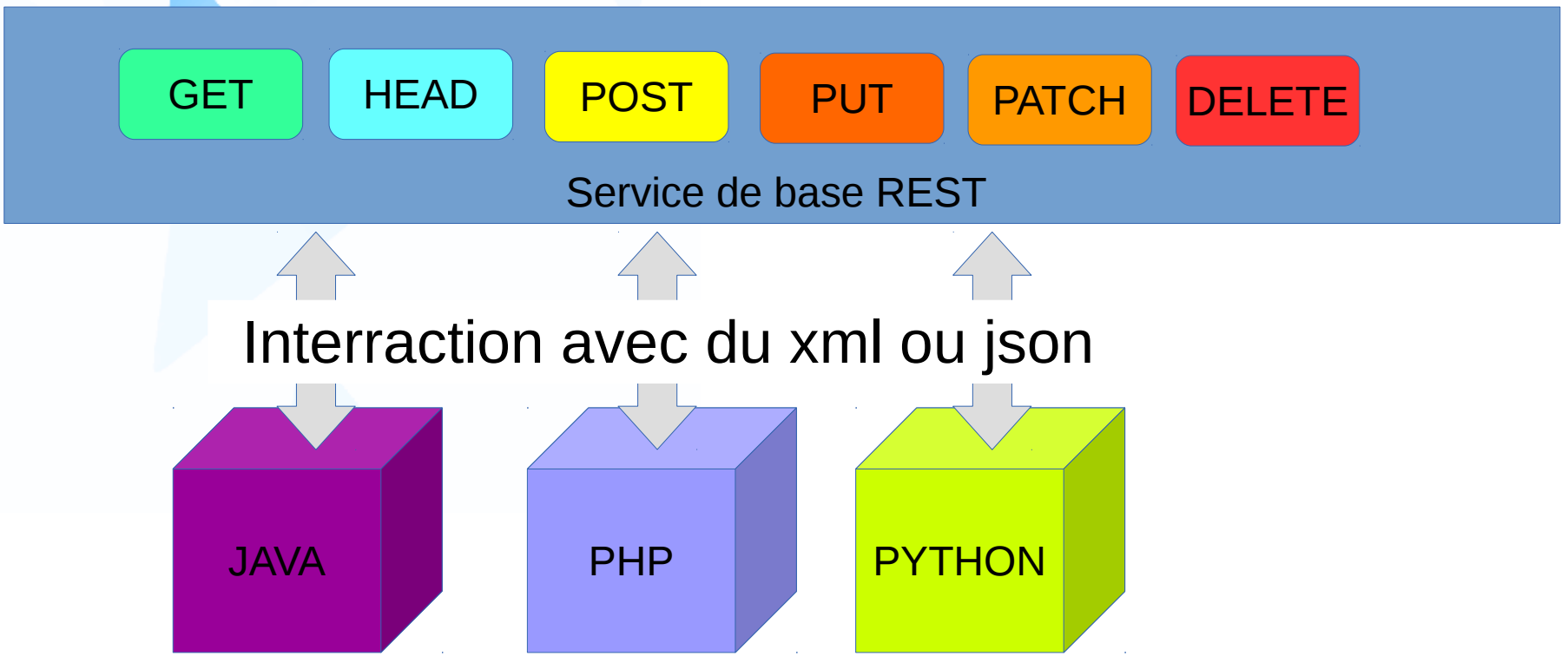


Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

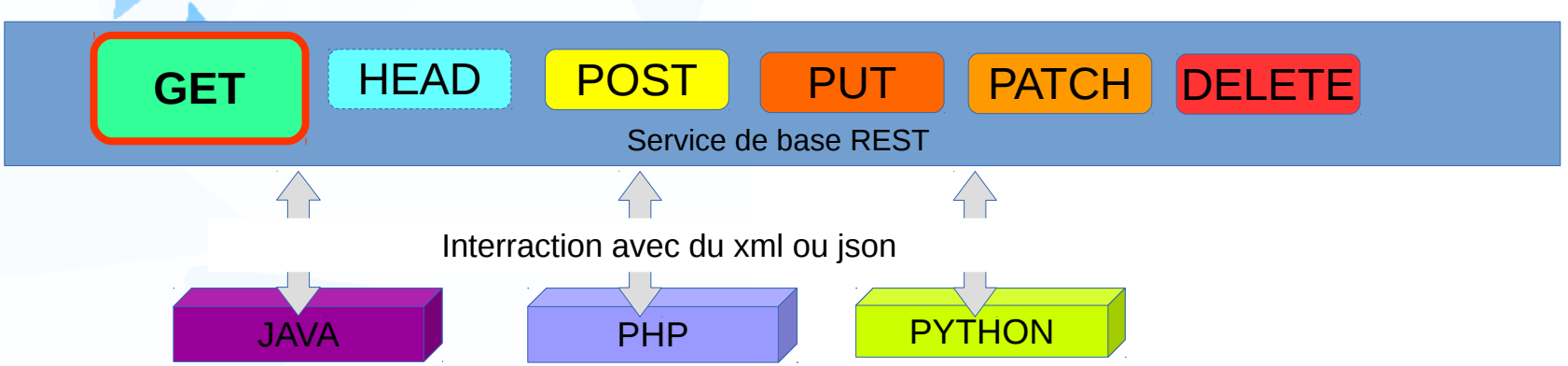


Pour plus d'information : https://roy.gbiv.com/pubs/dissertation/rest_arch_style.htm

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

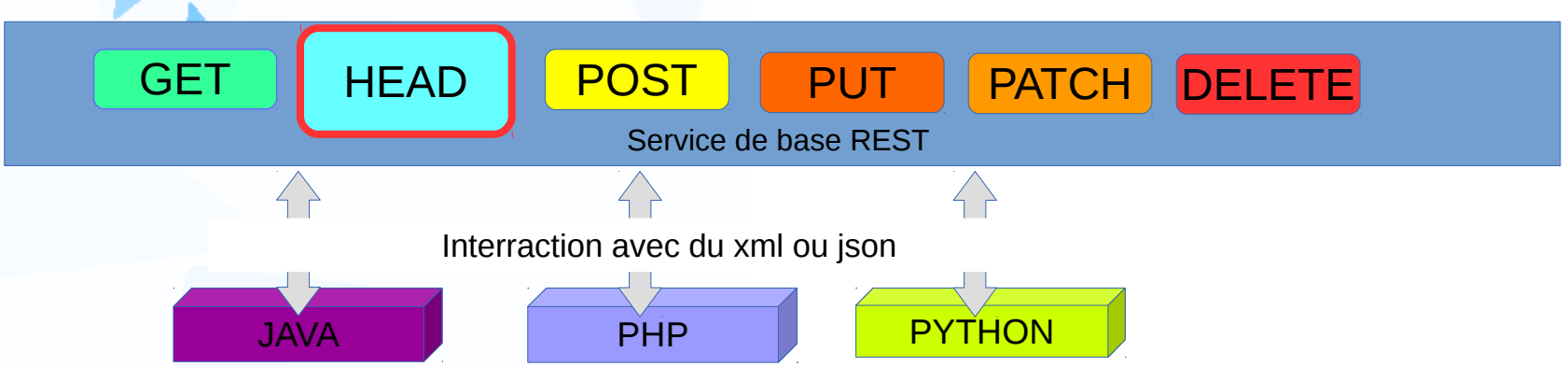


- L'élément GET accède à une ressource.

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

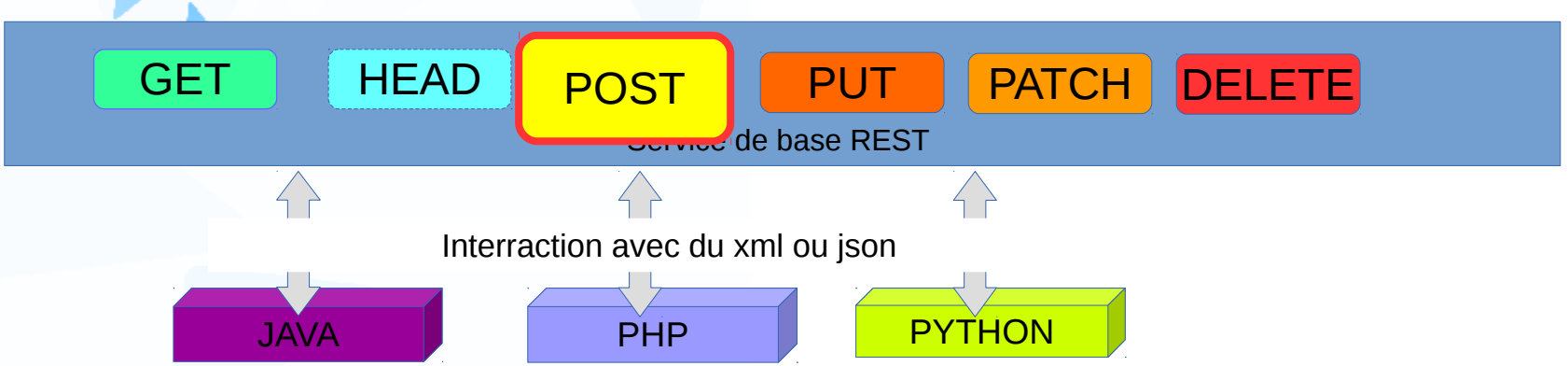


- L'élément HEAD accède à une méta-donnée c'est-à-dire à une donnée servant à définir ou décrire la ressource que l'on obtient par la méthode get.

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

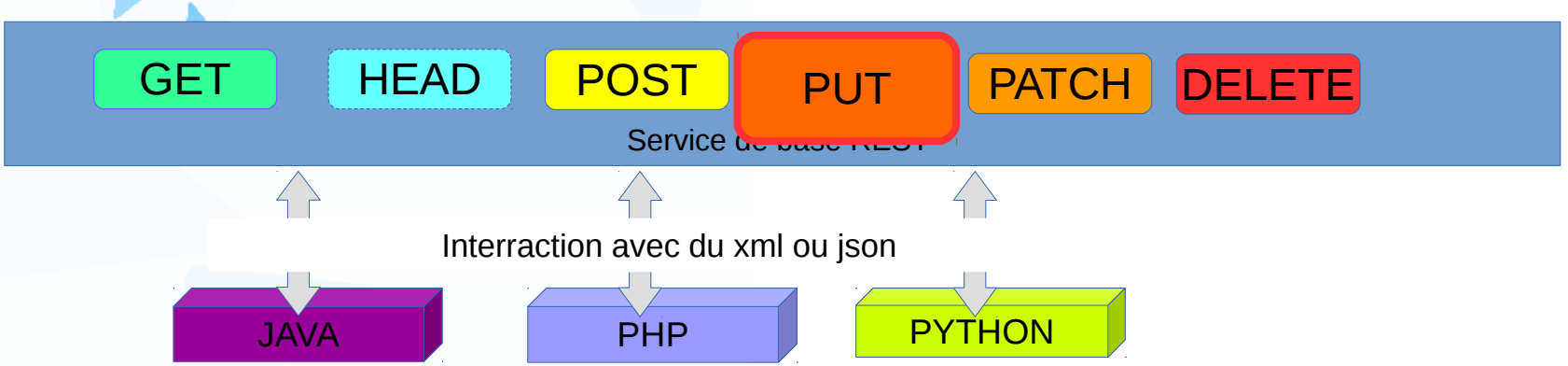


- L'élément POST ajoute une ressource.

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

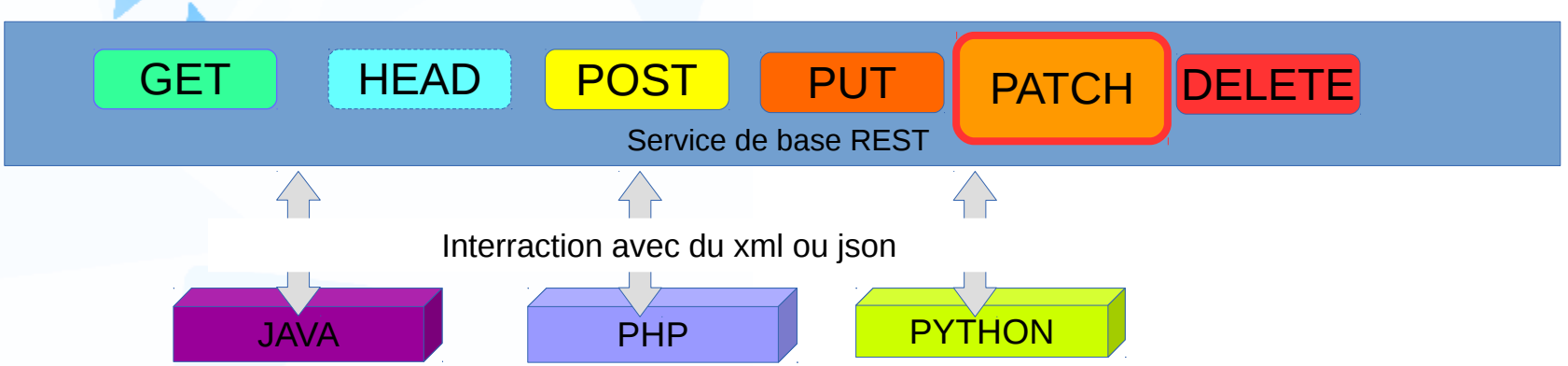


- L'élément PUT met à jour entièrement une ressource.

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

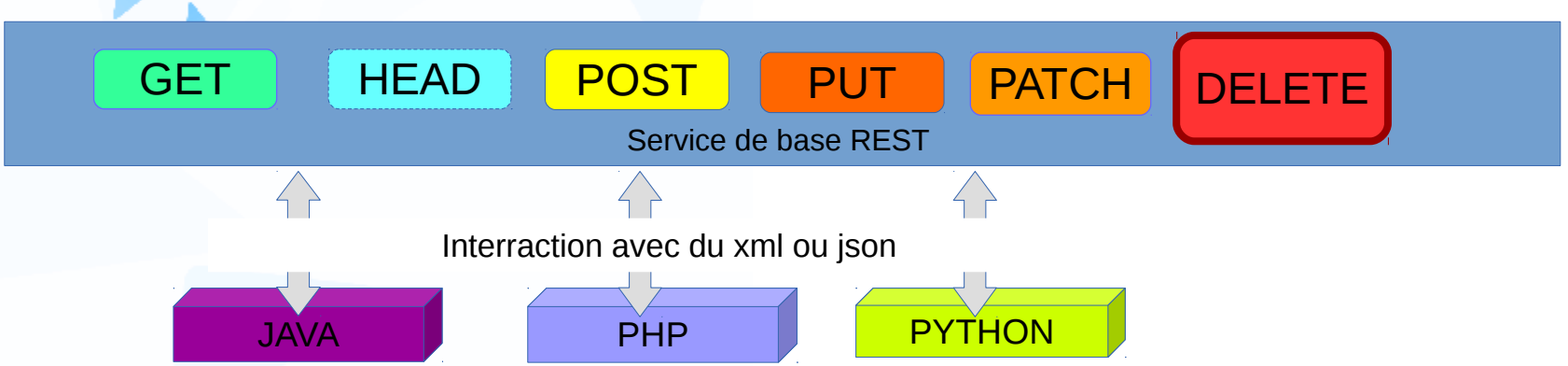


- L'élément PATCH met à jour partiellement une ressource.

Formation Produire et consommer des web services de type REST

I) Présentation d'une architecture REST

2) Les éléments d'une architecture REST

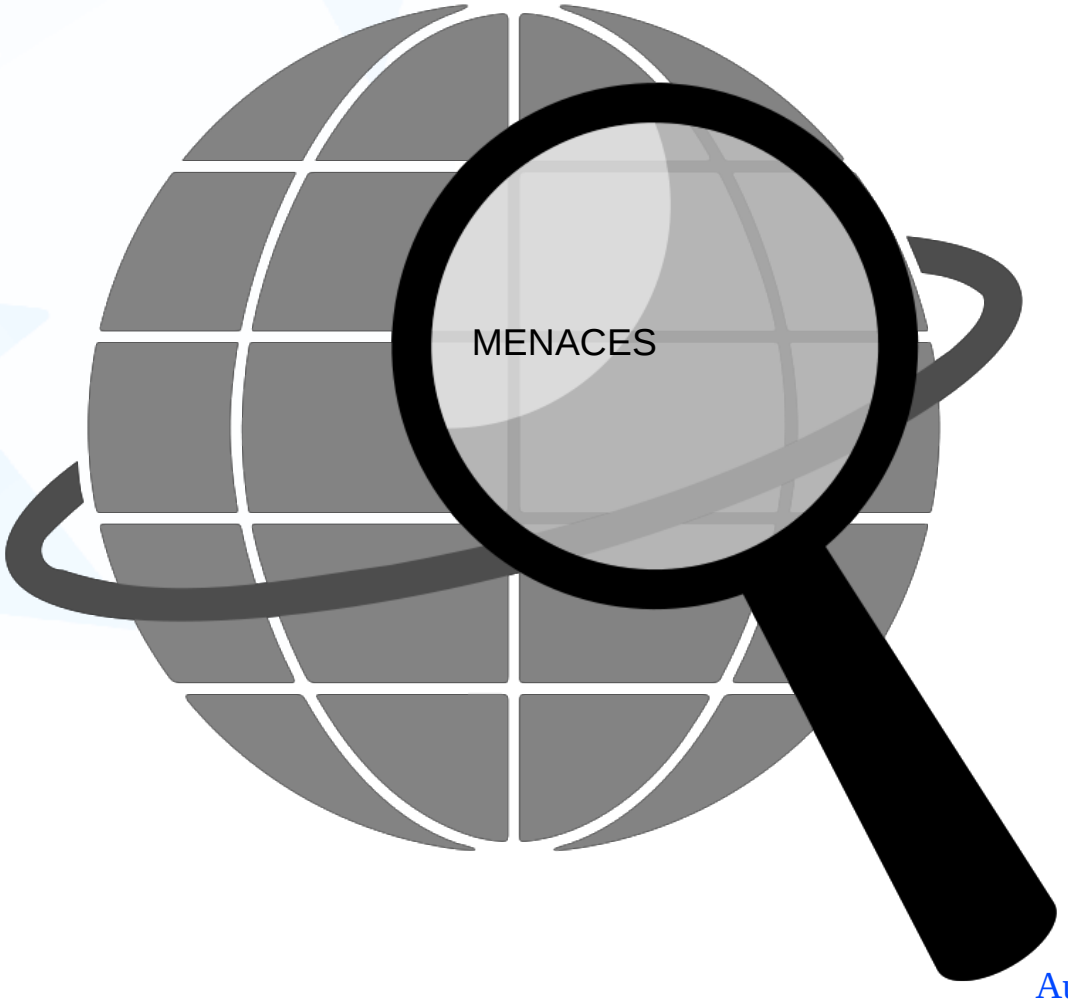


- L'élément DELETE supprime une ressource.

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités



Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Open Web Application Security Project (OWASP) est une communauté en ligne travaillant sur la sécurité des applications Web. Sa philosophie est d'être à la fois libre et ouverte à tous.

Source : https://fr.wikipedia.org/wiki/Open_Web_Application_Security_Project

Régulièrement, il sorte une liste concernant les failles de sécurité les plus rencontrées sur le Web : Top Ten OWASP

Site internet : www.owasp.org/

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Parmi les failles recensées par l'OWASP, voici celles que l'on peut rencontrées :

- ✗ Faille d'injection
- ✗ Broken Authentication
- ✗ Les failles XSS
- ✗ Une attaque CSRF

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Faille d'injection :

Une faille d'injection, telle l'injection SQL, OS et LDAP, se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent duper l'interpréteur afin de l'amener à exécuter des commandes fortuites ou accéder à des données non autorisées.

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Broken Authentication

Les fonctions applicatives relatives à l'authentification et la gestion de session ne sont souvent pas mises en œuvre correctement, permettant aux attaquants de compromettre les mots de passe, clés, jetons de session, ou d'exploiter d'autres failles d'implémentation pour s'approprier les identités d'autres utilisateurs.

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Les failles XSS

Les failles XSS se produisent chaque fois qu'une application accepte des données non fiables et les envoie à un navigateur web sans validation appropriée. XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants.

Formation Produire et consommer des web services de type REST

II) Menaces et vulnérabilité d'une architecture REST

1) Menaces et vulnérabilités

Une attaque CSRF

Une attaque CSRF (Cross Site Request Forgery) force le navigateur d'une victime authentifiée à envoyer une requête HTTP forgée, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement inclue, à une application web vulnérable. Ceci permet à l'attaquant de forcer le navigateur de la victime à générer des requêtes dont l'application vulnérable pense qu'elles émanent légitimement de la victime.

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Auteur : Aurélien MARTINEAU

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Méthodes safes (sûres) et idempotents (idempotentes)

La RFC7231 (remplaçante de la RFC2616) a ajouté deux notions aux méthodes HTTP :

- Les méthodes dites safes
- Les méthodes idempotents

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Méthodes safes (sûres)

Les méthodes dites safes ne modifient pas les données sur le serveur. Peu importe le nombre de fois qu’elles sont appelées.

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Les méthodes idempotentes

Les méthodes idempotentes quant à elles peuvent modifier les données lors du premier appel. Lors des appels suivants, la réponse sera tout le temps identique.

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Méthodes	GET	HEAD	POST	PUT	PATCH	DELETE
SAFE	oui	oui	non	non	non	non
Idempotente	oui	oui	non	oui	non	oui

Remarque : DELETE est idempotent car DELETE doit retourner un code 200 (Success) en cas de réussite et non un 204 (No Content) en cas d'echec.

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Bonnes/mauvaises pratiques

Il y a malheureusement, trop d'API conçue comme ceci :

POST api.mon-application.org/utilisateurs/rechercher

POST api.mon-application.org/utilisateurs/42/supprimer

POST api.mon-application.org/utilisateurs/42/modifier

GET api.mon-application.org/utilisateurs/supprimer?id=42

GET api.mon-application.org/utilisateurs/42?action=supprimer

GET api.mon-application.org/utilisateurs/42?action=envoyerMailConfirmation

Ce sont de mauvaises pratiques liées à une mauvaise compréhension des verbes HTTP. Quand il y a un nom d'action CRUD dans l'URI c'est qu'il y a un problème.

Formation Produire et consommer des web services de type REST

III) Découvrir les bonnes pratiques d'une architecture REST

1) Les Bonnes pratiques

Bonnes/mauvaises pratiques

GET api.mon-application.org/utilisateurs

GET api.mon-application.org/utilisateurs/42

POST api.mon-application.org/utilisateurs/

PUT api.mon-application.org/utilisateurs/42

DELETE api.mon-application.org/utilisateurs/42

GET api.mon-application.org/utilisateurs/42/messages/8

Ce sont les bonnes pratiques à utiliser dans le cadre d'une API REST.