

Suivi de projet dev - Ynov Campus Rennes - Évaluation 2/3 : API RESTful

Date: 10/07/23

Version: 1.0

Durée estimée : 1h max (sur place)

À rendre

Au choix :

- Sur papier
- Au format **PDF** à l'adresse mail suivante: paul.schuhmacher@ynov.com, avec le sujet avec le sujet suivant **rendu - api restful - x_abc** où **x** est la première lettre de votre nom et **abc** votre prénom (j'aurai donc pour sujet de mon e-mail **rendu - api restful - s_paul**).

Exercices

Exercice 1 (2pt)

Relier chaque terme à gauche avec la définition correspondante

- | | |
|----------------------------|---|
| (1) <i>Adressabilité</i> • | • (a) Document échangeable sur le réseau |
| (2) Représentation • | • (b) Identifiant unique permettant d'identifier une ressource |
| (3) Ressource • | • (c) Méthode HTTP idempotente utilisée pour envoyer des données au serveur |
| (4) URI • | • (d) Ensemble de propriétés ou contraintes sur lesquelles est fondé le web |
| (5) REST • | • (e) Permettent à un client et à un serveur de se transmettre des informations supplémentaires lors d'une requête ou d'une réponse |
| (6) En-tête HTTP • | • (f) Méthode HTTP utilisée pour envoyer des données au serveur |
| (7) POST • | • (g) Permettre de donner à chaque ressource une URI ou nom unique |
| (8) PUT • | • (h) Peut représenter <i>n'importe quoi</i> : une oeuvre d'art, un ensemble de références vers d'autres ressources, etc. |

Exercice 2 (3pt)

Barrer *la ou les* mauvaises réponses.

1. Sur le web, un client manipule des ressources

- Vrai
- Faux

2. La contrainte d'*absence d'état* dit que

- Le client ne connaît pas le serveur
- Le serveur ne maintient aucun *état*
- Le serveur ne garde aucune information sur ses clients

3. Les *hypermedia* doivent diriger l'état de l'application

- Vrai
- Faux

4. Toutes les APIs sur le web sont par définition RESTful

- Vrai
- Faux

5. Le protocole HTTP

- Impose des contraintes sur le contenu (corps) de l'*enveloppe*
- Impose des contraintes sur le formatage des *enveloppes*

6. Lorsque le serveur répond au client, il doit lui renvoyer

- La représentation de la ressource demandée
- Toutes les représentations possibles de la même ressource
- Les URLs vers les ressources connexes exposées par le serveur

7. Le web est supposé être navigable de la même manière pour tous les types d'agent (humains ou non)

- Vrai
- Faux

8. Une ressource est

- Un fichier sur un serveur web (page html, image, etc.)
- Conceptuelle et abstraite
- Définie par une URI (ou URL)
- Peut posséder plusieurs URI

9. Un site web statique (de simples fichiers html et assets) est *RESTful* par définition

- Vrai
- Faux

Exercice 3 (5pt)

Vous êtes en charge du développement d'un service web de carnet d'adresses. Au sein de cette application, les clients doivent pouvoir lire des cartes de visites (coordonnées d'une personne), en ajouter, les modifier ou les supprimer. Il faut également pouvoir créer des *groupes* pour regrouper des fiches (professionnel, famille, amis, etc.). Le service doit permettre d'accéder à une fiche individuelle ou à un groupe de fiches et d'effectuer une recherche par *nom de famille*.

Les *coordonnées* d'une personne sont définies par :

- le prénom
- le nom de famille
- le genre
- le numéro de téléphone
- l'adresse (rue, numéro de rue, ville, code postal)

Ce service doit pouvoir être consommé par des clients *REST*. Par client *REST*, nous entendons un programme, écrit dans un langage quelconque, qui interrogera des URL via le protocole HTTP pour accéder aux données du carnet d'adresses, dans un format à définir (HTML, XML, JSON, etc.).

1. **Décomposer** les données en ressources.
2. **Nommer** chacune des ressources avec des URI et un libellé.
3. **Définir** un sous-ensemble de l'interface uniforme (GET, POST, PUT, DELETE) pour chaque ressource identifiée.
4. **Donner** un exemple de données retournée par le serveur au format JSON pour la ressource "Groupes de fiches" ainsi que le code de retour.
5. **Définir** les représentations acceptées *par le serveur* pour modifier les ressources. Le client REST envoie sa représentation au format `application/x-www-form-urlencoded` (format des données soumises via un formulaire via une balise `<form>`), soit de simples `clef=valeur` dans le corps de la requête HTTP. Pour chaque représentation, fournir une pseudo requête HTTP en utilisant le template suivant :

MÉTHODE URL HTTP/1.1

clef=valeur