

Architecture REST

Guy Lapalme
IFT3225

Défini originalement par Roy Thomas Fielding (2000)

Thèse Ph.D. - Architectural Styles and the Design of Network-based Software Architectures

REpresentational State Transfer

contraintes sur WWW

1. Client-serveur
2. Interaction sans état
3. Cache
4. Interface uniforme entre composantes
5. Système en couches
6. Code sur demande

I. - Client-serveur

- Sépare l'interface usager du stockage des données
- Conséquences
 - portabilité sur plusieurs plateformes
 - améliore la mise à l'échelle en simplifiant les composants du serveur
 - évolution indépendante des clients et serveurs

2. Interaction sans état (a)

- chaque requête au serveur contient **toute** l'information pour la comprendre
- ne peut se fier à un contexte sur le serveur
- information de session gardée sur le client

2. Interaction sans état (b)

- avantages
 - Meilleure visibilité
 - monitoring n'a pas besoin d'autres infos que la requête pour la comprendre
 - Meilleure fiabilité
 - facilité de récupération d'erreur
 - Extensibilité (scalability)
 - pas besoin de gardes des ressources entre les requêtes (libération plus rapide)
- désavantages
 - Répétition d'information

3. Cache

- réponse du serveur doit être
 - implicitement ou explicitement étiquetée comme *cachable*
 - éviter de redemander de l'information

4. Interface uniforme entre composantes

- HTTP
 - idéal pour transferts à *gros grain*
 - moins efficace pour d'autres

5. Systèmes en couches

- Système ne connaît qu'un seul niveau d'interaction
- Serveur de l'un est client de l'autre
- Isole l'architecture
- Risque d'*overhead* et de *latence*

6. Code sur demande *optionnel*

- extension de la fonctionnalité d'un client
 - téléchargement de code (applets/scripts)
- traitement des cas où ce n'est pas toujours possible

Éléments de données REST

ressource

- types
 - document/image
 - service (e.g. météo d'aujourd'hui)
 - collection de ressources
 - personne
- généralité en abstrayant l'implantation
- liaison tardive de la référence
- réfère au concept plutôt qu'à une représentation

Qualité d'un identificateur dépend des efforts pour en garder la validité (e.g. liens brisés)

Éléments de données REST

représentation

- capture l'état actuel ou désiré d'une ressource
- séquence de bytes ou de métadonnées qui la décrit
- données de contrôle qui décrivent l'état de la transmission (e.g. progression, cache)

Éléments de données REST

connecteurs

- Encapsulent les accès aux ressources et les transferts de représentations de ressources
- Exemples
 - client: libwww
 - serveur: libwww, Apache API
 - cache: cache du browser
 - resolver: bind (DNS lookup)
 - tunnel : SSL après connexion HTTP

Exemple de *RESTful* design WWW

- DB
 - CREATE, READ, UPDATE, DELETE (CRUD)
- HTTP
 - POST, GET, PUT, DELETE
 - Contrainte du *sans état* violée par les cookies et le *back* du browser
- HTML
 - Javascript pour le code sur demande
 - Support pour les hyperliens (GET)
 - FORM : GET, POST