

REST

- Representational State Transfer, thèse R. Fielding 2000, co-fondateur F. Apache
- style d'architecture, pas un protocole ni un format
- Des contraintes : client-serveur, sans état, cache, interface uniforme (hypermédia), couche
- Les contraintes, telles que définies par Roy Fielding, peuvent s'appliquer à d'autres protocoles d'application que HTTP

REST

- HTTP va fournir les opérations nécessaires à la manipulation de la ressource :
- GET pour récupérer la ressource
- POST pour créer la ressource
- PUT pour modifier la ressource
- DELETE pour supprimer la ressource.
- CRUD : Create, Read, Update, Delete

REST

- Exemples
- Utiliser 2 URLs par ressource (des noms, pas des verbes)

Resource	POST create	GET read	PUT update	DELETE delete
/dogs	create a new dog	list dogs	bulk update dogs	delete all dogs
/dogs/1234	error	show Bo	if exists update Bo if not error	delete Bo

REST

- Cacher la complexité des relations dans l'application
- GET owners/5678/dogs
- POST owners/5678/dogs
- resource name/identifiant/resource name
- GET /dogs?color=red&state=running&location=park

REST

- HAL <http://api.archives-ouvertes.fr/docs>
- Mathdoc
- Agenda des Maths
- Freenas
- Proxmox
- Request Tracker
- Piwik

REST

- Les codes d'erreur
- http://en.wikipedia.org/wiki/Http_error_codes
- https://blog.apigee.com/detail/restful_api_design_what_about_errors/

Google GData

200 201 304 400 401 403 404 409 410 500

Netflix

200 201 304 400 401 403 404 412 500

Digg

200 400 401 403 404 410 500 503

API Freenas

- <http://api.freenas.org/>
- Nécessite au moins Freenas 9.x
- Exemples dans <https://github.com/freenas/freenas/tree/master/examples/api>

API Proxmox

- http://pve.proxmox.com/wiki/Proxmox_VE_API

REST

- Representational State Transfer, thèse R. Fielding 2000, co-fondateur F. Apache
- style d'architecture, pas un protocole ni un format
- Des contraintes : client-serveur, sans état, cache, interface uniforme (hypermédia), couche
- Les contraintes, telles que définies par Roy Fielding, peuvent s'appliquer à d'autres protocoles d'application que HTTP

- Hypermédia : chaque accès aux états suivants de l'application est décrit dans le message courant
- chaque requête d'un client vers un serveur doit contenir toute l'information nécessaire pour permettre au serveur de comprendre la requête, sans avoir à dépendre d'un contexte conservé sur le serveur. Cela libère de nombreuses interactions entre le client et le serveur.
- Toute l'information n'est pas envoyée dans une seule ressource unique

L'absence de gestion d'état du client sur le serveur conduit à une consommation de mémoire inférieure, une plus grande simplicité et donc à une capacité plus grande de répondre à un grand nombre de requêtes simultanées.

REST

- HTTP va fournir les opérations nécessaires à la manipulation de la ressource :
- GET pour récupérer la ressource
- POST pour créer la ressource
- PUT pour modifier la ressource
- DELETE pour supprimer la ressource.
- CRUD : Create, Read, Update, Delete

REST

- Exemples
- Utiliser 2 URLs par ressource (des noms, pas des verbes)

Resource	POST create	GET read	PUT update	DELETE delete
/dogs	create a new dog	list dogs	bulk update dogs	delete all dogs
/dogs/1234	error	show Bo	if exists update Bo if not error	delete Bo

REST

- Cacher la complexité des relations dans l'application
- GET owners/5678/dogs
- POST owners/5678/dogs
- resource name/identifiant/resource name
- GET /dogs?color=red&state=running&location=park

REST

- HAL <http://api.archives-ouvertes.fr/docs>
- Mathdoc
- Agenda des Maths
- Freenas
- Proxmox
- Request Tracker
- Piwik

Et aussi freebox qui a une api rest :

<http://dev.freebox.fr/sdk/os/>

Most API uses the REST architecture, pay attention to the http methods used for each request.

Openstack rest api

REST

- Les codes d'erreur
- http://en.wikipedia.org/wiki/Http_error_codes
- https://blog.apigee.com/detail/restful_api_design_what_about_errors/

Google GData

200 201 304 400 401 403 404 409 410 500

Netflix

200 201 304 400 401 403 404 412 500

Digg

200 400 401 403 404 410 500 503

API Freenas

- <http://api.freenas.org/>
- Nécessite au moins Freenas 9.x
- Exemples dans <https://github.com/freenas/freenas/tree/master/examples/api>

API Proxmox

- http://pve.proxmox.com/wiki/Proxmox_VE_API