

Memento REST

- Services Web
 - Protocoles standardisés / ouverts (en général HTTP)
 - Deux familles : SOAP (orienté services) et REST (orienté ressources)
 - REpresentational State Transfer, chapitre 5 thèse de Roy Fielding (2000), systèmes distribués, <http://opikanoba.org/tr/fielding/rest>
 - Architecture, ensemble de contraintes...
 - Client / serveur ; Stateless ; interface uniforme (identification, représentation, auto-descriptif, HATEOAS / Hypermedia) ; couches ; code à la demande (optionnel)
- Client / serveur : sépare IHM et stockage (évolution séparée, meilleure portabilité et possibilités montée en charge...)
- Stateless : requête auto-portante ; pas de contexte sur le serveur (côté client) ; améliore visibilité
- Données réponses marquées pour la mise en cache
- Interfaces uniformes (standardisé) : cf. SQL / SGBDR (schema.table.colonne) : SELECT, INSERT, UPDATE et DELETE
- Contraintes d'interface : identification uniforme (URI : Uniform Resource Identifier)...
- Représentations (seule la représentation de l'état transite)
- Hypermédia comme moteur de l'état de l'application (HATEOAS : Hypertext As The Engine Of Application State) : facilité, guidage utilisateur, agent utilisateur stocke l'information, pas de contraintes pré-définies...
- ...
- HTTP / URI = Uniform Resource Identifier (RFC 3986) <https://tools.ietf.org/html/rfc3986>
 - URL : Uniform Resource Locator (identifie une ressource sur un réseau)
 - URN : Uniform Resource Name (identifie un nom de ressource)
 - Schema + Chemin d'accès
- Requête : Structure textuelle normalisée
 - Ligne de requête : <méthode> <URI> <version-HTTP> CRLF
 - Lignes d'en-tête : *(<champs> : <valeur> CRLF)
 - Ligne vide : CRLF
 - Corps (optionnel) : [*OCTET]
- Réponse : Structure assez similaire à la requête
 - Ligne de statut : <version-HTTP> <statut> <message> CRLF
 - Lignes d'en-tête : *(<champs> : <valeur> CRLF)
 - Ligne vide : CRLF
 - orps (optionnel) : [*OCTET]
- De nombreux champs d'en-tête sont prédéfinis
 - General header : les champs communs (Cache-Control, Date, ...)
 - Request header : les champs spécifiques de requête (Host, Accept, ...)
 - Response header : les champs spécifiques de réponse (Age, ETag, ...)
 - Entity header : les champs spécifiques des entités (Content-Encoding, ...)
- Spécifications : <http://www.w3.org/Protocols/>
 - HTTP 1.1 : <https://tools.ietf.org/html/rfc2616>
 - HTTP 1.1bis : reformulation dans plusieurs documents (RFC7230, ..., RFC7237)
 - HTTP 2 : RFC7540 publiée en mai 2015
- 8 méthodes de base
 - GET : obtenir une représentation d'une ressource
 - HEAD : demander des informations sur une ressource
 - POST : transmettre des données à une ressource pour traitement
 - PUT : remplacer ou créer une ressource (dont on connaît l'URI)
 - DELETE : supprimer une ressource
 - CONNECT : pour utiliser un Proxy comme un tunnel de communication
 - OPTIONS : demander les options de communications (ressource ou serveur)
 - TRACE : demander au serveur de retourner ce qu'il a reçu (echo pour diagnostic)
- Codes de réponse HTTP
- ...

Code		Signification
1xx	Information	messages d'information (non utilisé)
2xx	Succès	la requête s'est déroulée correctement
3xx	Redirection	indique une redirection
4xx	Erreur client	à l'origine du client
5xx	Erreur serveur	à l'origine du serveur

TABLE 1 – Table résumant les codes HTTP