Introduction aux Services Web REST

Pierrick Charron, Janvier 2011

pierrick@php.net

REST ce n'est pas

- Une technologie
- Un protocole
- Un standard

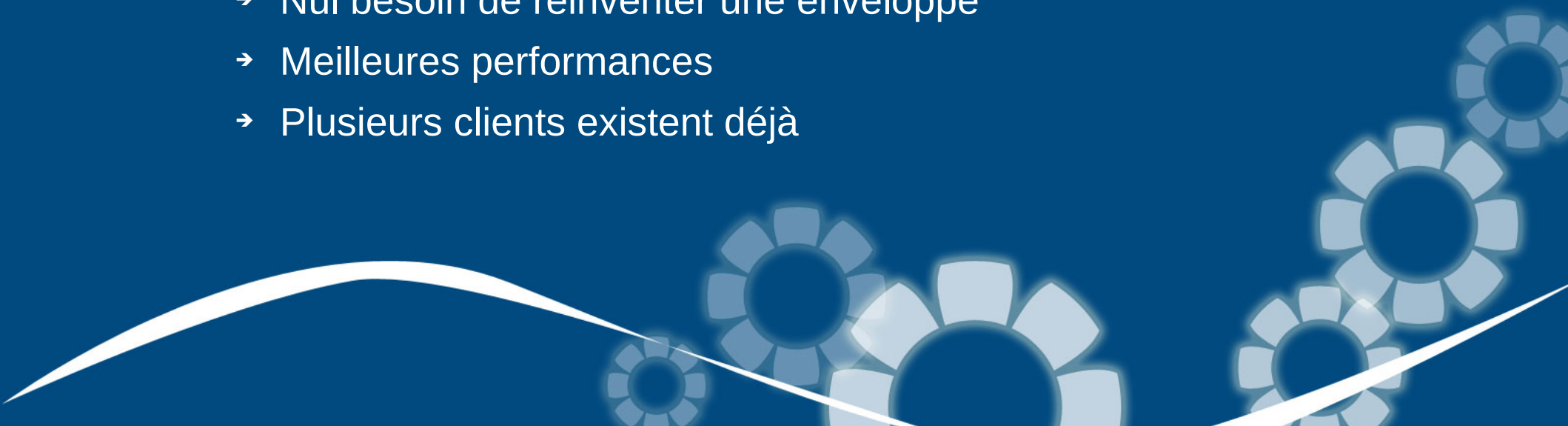
REST c'est

- Un acronyme : REpresentational State Transfer

- Un style architectural défini en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP dans sa thèse “*Chapter 5 – Representational State Transfer*”

Avantages

- Chaque requête comprend toutes les informations nécessaires au serveur pour y répondre
  - Facilite l'utilisation de cache
  - Possibilité de distribuer les requêtes sur plusieurs serveurs
- Utilisation du protocole HTTP
  - Nul besoin de réinventer une enveloppe
  - Meilleures performances
  - Plusieurs clients existent déjà



Contraintes

- Le client doit conserver toutes les données nécessaires à la continuité de ses opérations

- Plus grosse consommation en bande passante

Une ressource

- Une chose, une entité, un événement, une liste d'autres ressources, ou toute autre information pouvant être nommée

- Identifiée par une URI (qui ne doit pas contenir de verbe, uniquement des noms)

Exemples de Ressource

- Un document :

  http://localhost/document/12
  http://localhost/document/12/chapitre/1
  http://localhost/document/dernier

- Une liste de documents :

  http://localhost/documents/

Représentation

- Capture de l'état courant d'une ressource
- Méta-données de la ressource
- La représentation est ce qui est envoyé au client
- Une même ressource peut avoir plusieurs représentations :
  - XML
  - JSON
  - YAML
  - ...

Exemple de représentation

- GET /document/12
  Host: localhost
  Accept: application/xml

- <document>
    <titre>Titre du document</title>
    <contenu>Bonjour le monde !!!</contenu>
  </document>

Exemple de représentation

- GET /document/12
  Host: localhost
  Accept: application/x-javascript

- {titre: “Titre du document”, body: “Bonjour le monde!!!”}

Interface uniforme

- Ensemble limité d'opérations bien définis afin d'accéder et de manipuler les ressources (basé sur les spécifications HTTP – RFC2616)
  - GET : Récupération de la représentation d'une ressource
  - POST : Création d'une nouvelle ressource
  - PUT : Modification d'une ressource existante (ou création)
  - DELETE : Suppression d'une ressource

Réponses HTTP

- 2XX
  - Succès de l'opération
- 3XX
  - Redirection
- 4XX
  - Erreur côté client
- 5XX
  - Erreur côté serveur

Réponses HTTP 2XX

- 200 : OK
- 201 : Created
- 202 : Accepted
- 204 : No Content
- 205 : Reset content
- 206 : Partial Content

Réponses HTTP 3XX

- 301 : Moved Permanently
- 304 : Not Modified

Réponses HTTP 4XX

- 400 : Bad Request
- 401 : Unauthorized
- 404 : Not Found
- 405 : Method Not Allowed
- 408 : Request Timeout
- 409 : Conflict
- 415 : Unsupported Media Type

Réponses HTTP 5XX

- 500 : Internal Server Error
- 501 : Not Implemented
- 503 : Service Unavailable

Entêtes HTTP

- Accept
- Accept-Charset
- Accept-Encoding
- Accept
- ...
- ...

- Content-Type
- Location
- Range
- WWW-Authenticate
- ...
- ...

Exemple d'appel REST

- GET /documents/ HTTP1.1
  Host: localhost
  Range: items=0-10
  Accept: application/xml

- HTTP/1.0 200 OK
  Content-Type: application/xml
  Content-Length : 1245

  <documents>...</documents>

Autre Exemple

- POST /documents/ HTTP1.1
  Host: localhost
  Content-Type: application/json
  Content-Lenght: ...

  {titre: “Titre de mon document”, body: “Bonjour le monde !!!”}

- HTTP/1.0 201 Created
  Location: http://localhost/document/20
  Content-Type: application/json
  Content-Length : 1245

  {titre: “Titre de mon document”, body: “Bonjour le monde !!!”}

Quelques Ressources :)

- Sites internet :
  - http://www.rfc-editor.org/rfc/rfc2616.txt
  - http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
  - http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

- Livre :
  - RESTful Web Services - O'Reilly Media