# Formation Webservice RESTful

## Travaux Pratiques

# TP 1 : quizz

## Objectifs

L'objectif de ce TP est d'utiliser les bons principes du protocole HTTP pour développer ses Web Services

## Quizz

A l'aide des tableaux en annexes, répondre aux questions suivantes :

- Quelles entêtes et codes de retour doit-on utiliser pour la récupération d'une liste de réservations?
- Quelles entêtes et codes de retour doit-on utiliser pour la création d'une réservation ?
- Quelles entêtes et codes de retour doit-on utiliser pour la modification d'une réservation ?
- Comment doit-on traiter la suppression d'une ressource ? (2 appels successifs) ?
- Quelles entêtes et codes de retour doit-on utiliser si un paramètre obligatoire est manquant ?
- Comment gérer une erreur ?
    - Que doit renvoyer une requête d'accès à une ressource nécessitant une authentification quand on n'est pas authentifié ?
    - Que doit renvoyer une requête pour une ressource dont on n'a pas les droits d'accès ?
    - Que doit-on renvoyer quand on fait une opération non autorisée sur une ressource ?
    - Que doit renvoyer une recherche dont les paramètres ne correspondent à aucune ressource ?
- Comment modéliser un traitement asynchrone ?
- Quelles entêtes et codes de retour doit-on utiliser lors de la migration d'un Web Service ?

# TP 2 : créer un service REST et le tester

## Objectif

Créer un service REST de type *hello world*.

## Procédure

- Importer le projet Restful dans eclipse, ajuster la variable "JRE System Library" pour la faire pointer sur un JDK 8. Le projet fournit est basé sur la lib jersey implémentant `javax.ws.rs`. Les échanges de données se feront au format JSON.
- Configurer dans Eclipse un serveur tomcat v8 et déployer le projet Restful
- Créer une classe `HelloWorldService` dans le package `com.resanet.ws` exposant un service sur l'URL `/hello`. Ce service répond au verbe `GET`, recoit un paramètre `param` et renvoi une réponse de la forme :

```
"Jersey répond : " + param
```

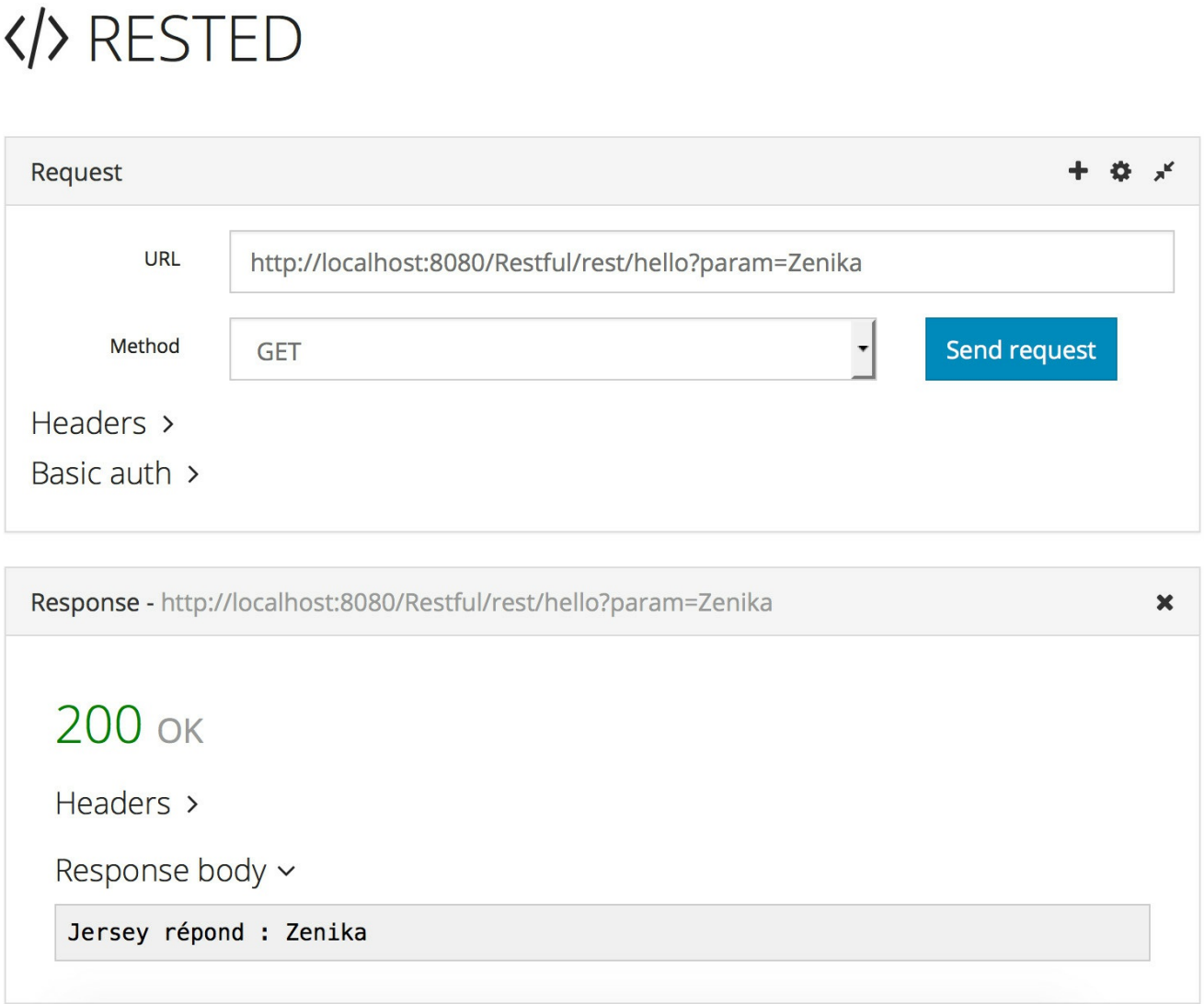
- Ajouter les annotations `jax-rs` pour publier le webservice

La requete d'appel est de la forme : http://localhost:8080/Restful/rest/hello?param=zenika

Pour tester le service, il est possible d'utiliser au choix :

- un navigateur tout simple,
- RESTED sous firefox
- DHC de restlet ou PostMan sous chrome
- un client de test écrit en java

Exemple avec RESTED sous firefox:



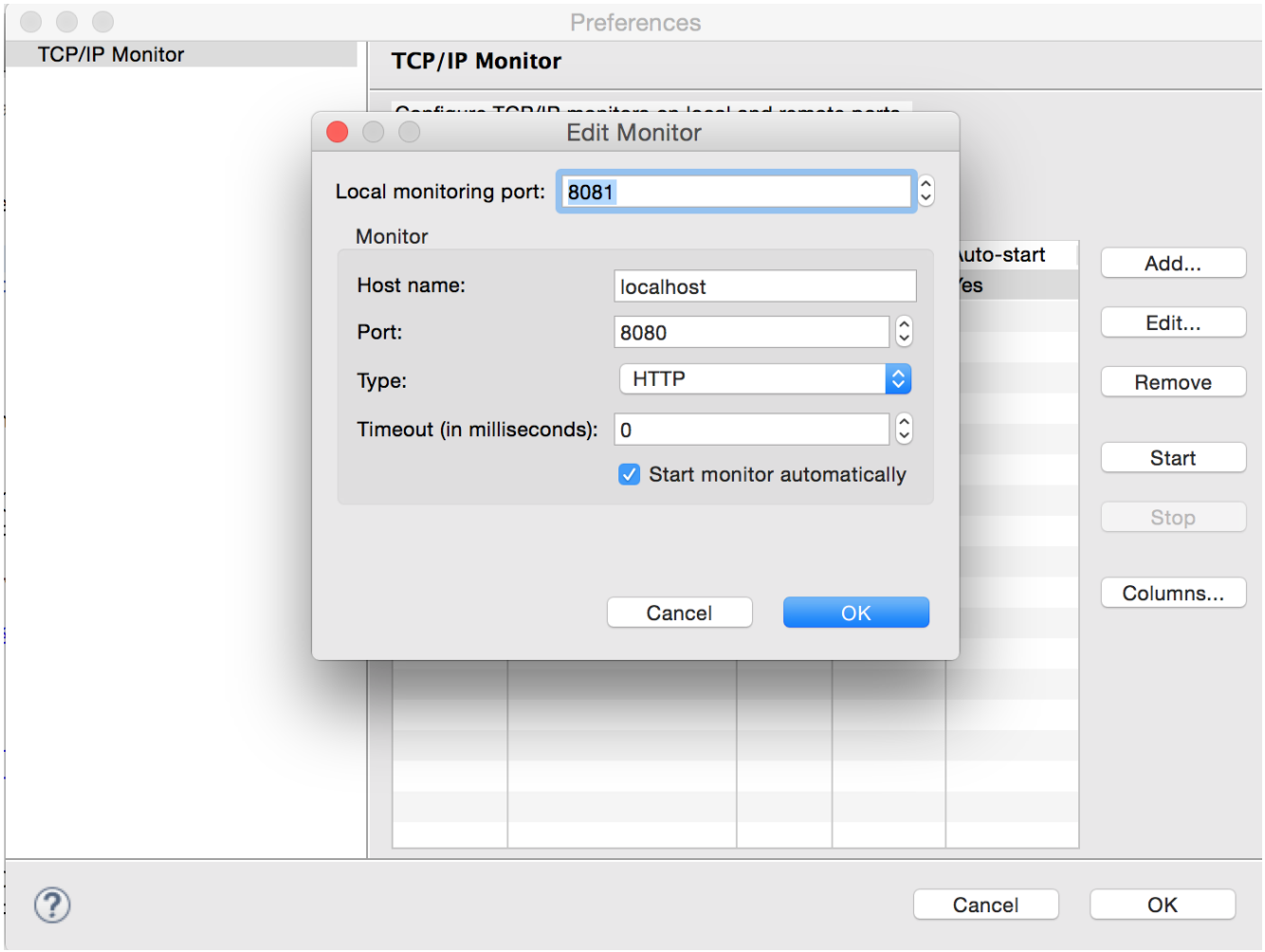

Créer un client REST dans `src/test/java/com/resanet/ws` en java à l'aide de `javax.ws.rs.client.ClientBuilder` qui effectue le même appel que précédement et affiche en console la réponse reçue.

Utiliser le proxy "tcp/ip monitor" d'Eclipse pour voir les requêtes / réponses circuler.

- Window > Show View > Other > TCP/IP Monitor > OK

- Configurer le proxy : Properties > Add ... > Local monitoring port : 8081, Hostname : localhost, port : 8080, cocher la case "start automatically" puis OK > OK

Changer l'url d'appel de http://localhost:8080/Restful/rest/XXX en http://localhost:8081/Restful/rest/XXX




# TP 3 : Développer un Web Service REST

## Contexte

L'agence de voyages Resanet a décidé de mettre en oeuvre une architecture SOA à l'aide d'un Web Services REST au sein de son Système d'Information afin de lui apporter une souplesse et une maintenabilité optimale. L'objectif principal est de fournir l'ensemble des services Resanet à l'ensemble des partenaires (voyagistes, compagnie aérienne...) afin d'enrichir l'offre Resanet rapidement et à moindre coût.

## Installation du service

- Importer les sources java fournis dans le dossier `src/main/java/com/resanet/ws` et `src/test/java` du projet Restful. Le service de gestion des voyages fournit les opérations suivantes
  - créer un voyage
  - récupérer un voyage
  - mettre à jour un voyage
  - supprimer un voyage

- Annoter l'ensemble des méthodes afin de fournir un service REST et les implémenter pour que le client de tests unitaires `VoyageClient` présent dans `src/test/java/com/resanet/ws` passe.
- Enrichir le service addVoyage() pour retourner un objet de type `javax.ws.rs.core.Response` contenant l'url de récupération du nouveau voyage créé.
- Enrichir le service getVoyage() pour retourner une erreur 404 lorsque le voyage demandé n'existe pas
- Est-ce que le service adopte une architecture RESTful ?
- Quelles modifications doit-on apporter pour adopter une architecture RESTful ?

# TP4 : Préparation pour la mise en cache + HATEOAS

## Objectifs

On souhaite garantir les performances de notre service pour un nombre important de clients. L'objectif de ce TP est d'adopter une politique de mise en cache des données coté client pour éviter de surcharger le serveur.

## Modification des services

Pour prendre HATEOAS, que faut il modifier / améliorer sur l'exposition REST ?

Implémenter les modifications et ajouter les tests unitaires associé.

Ajouter les entêtes de cache au niveau des services de récupération de données. Pour assurer une transition progressive des clients, versionner les API publiée.

## Ajout d'opérations de recherche de voyages

1. Ajouter une opération `listVoyages()` prenant en paramètre un libellé et retournant une collection d'objet de type `Voyage` commençant par le libellé donné.
   La réponse est au format XML.
2. Ajouter une opération de recherche paginée prenant en paramètres :

- `libelle` : libelle dont les voyages commencent
- `size` : nombre de résultats à retourner par page
- `numeroPage` : numéro de la page à retourner
  Ce service doit prendre en compte l'approche HATEAOS (c'est à dire, l'ajout d'url dans la réponse, comme l'élement suivant ou précédent de la liste).

3. Créer les tests unitaires associés
4. Ajouter les entêtes de cache sur toutes les méthodes de récupération de ressources

# TP5 : Concurrence & Sécurité

On souhaite fournir la possiblité aux utilisateurs d'acheter des réservations de voyages à la manière d'un panier d'achat.

## Objectifs

L'objectif de ce TP est de sécuriser les accès et d'implémenter un mecanisme de verrou pour fournir un scénario fonctionnel complet impliquant des transactions.

## Sécurisation

- Ajouter une authentification de type "Basic" aux différentes opérations.
- Modifier les tests unitaires pour prendre en compte l'authentification

## Gestion de la concurrence d'accès

- Exposer les services de gestion suivants en prenant en compte les principes HATEOAS et d'accès concurrent
  - récupération d'une réservation
  - réserver un voyage
  - suppression voyage
- creer les tests unitaires associés

# TP6 : Implémentation CORS

On souhaite ouvrir le webservice à tous les utilisateurs

## Objectifs

L'objectif de ce TP est d'ajouter l'implémentation coté serveur du mécanisme CORS et de documenter l'API

## CORS

- Implémenter le mécanisme CORS
- Tester la bonne valorisation des entêtes

## Swagger

- Importer le projet Swagger-UI dans eclipse. Il s'agit d'une webapp qui est capable de parcourir la documentation swagger du webservice pour la mettre en forme. Une fois déployée sur Tomcat, elle est accessible à l'adresse http://localhost:8080/Swagger-UI/
- Ajouter la documentation swagger aux API exposées

# Annexes Code de retour HTTP

- Codes commençant par 1

| Code | Message | Signification |
|---|---|---|
| 100 | Continue | Attente de la suite de la requête |
| 101 | Switching Protocols | Acceptation du changement de protocole |
| 102 | Processing | WebDAV: Traitement en cours (évite que le client dépasse le temps d'attente limite). |
| 118 | Connection timed out | Délai imparti à l'opération dépassé |

- Codes commençant par 2

| Code | Message | Signification |
|---|---|---|
| 200 | OK | Requête traitée avec succès |
| 201 | Created | Requête traitée avec succès avec création d'un document |
| 202 | Accepted | Requête traitée mais sans garantie de résultat |
| 203 | Non-Authoritative Information | Information retournée mais générée par une source non certifiée |
| 204 | No Content | Requête traitée avec succès mais pas d'information à renvoyer |
| 205 | Reset Content | Requête traitée avec succès, la page courante peut être effacée |
| 206 | Partial Content | Une partie seulement de la requête a été transmise |
| 207 | Multi-Status | WebDAV: Réponse multiple. |
| 210 | Content Different | WebDAV: La copie de la ressource côté client diffère de celle du serveur (contenu ou propriétés). |

- Codes commençant par 3

| Code | Message | Signification |
|---|---|---|
| 300 | Multiple Choices | L'URIdemandée se rapporte à plusieurs ressources |
| 301 | Moved Permanently | Document déplacé de façon permanente |
| 302 | Moved Temporarily | Document déplacé de façon temporaire |
| 303 | See Other | La réponse à cette requête est ailleurs |
| 304 | Not Modified | Document non modifié depuis la dernière requête |
| 305 | Use Proxy | La requête doit être ré-adressée auproxy |

| Code | Message | Signification |
|---|---|---|
| 307 | Temporary Redirect | La requête doit être redirigée temporairement vers l'URI spécifiée |
| 310 | Too many Redirects | La requête doit être redirigée de trop nombreuses fois, ou est victime d'une boucle de redirection. |

- Codes commençant par 4

| Code | Message | Signification |
|---|---|---|
| 400 | Bad Request | La syntaxe de la requête est erronée |
| 401 | Unauthorized | Une authentification est nécessaire pour accéder à la ressource |
| 402 | Payment Required | Paiement requis pour accéder à la ressource (non utilisé) |
| 403 | Forbidden | L'authentification est refusée. Contrairement à l'erreur 401, aucune demande d'authentification ne sera faite |
| 404 | Not Found | Ressource non trouvée |
| 405 | Method Not Allowed | Méthode de requête non autorisée |
| 406 | Not Acceptable | Toutes les réponses possibles seront refusées. |
| 407 | Proxy Authentication Required | Accès à la ressource autorisé par identification avec le proxy |
| 408 | Request Time-out | Temps d'attente d'une réponse du serveur écoulé |
| 409 | Conflict | La requête ne peut être traitée à l'état actuel |
| 410 | Gone | La ressource est indisponible et aucune adresse de redirection n'est connue |
| 411 | Length Required | La longueur de la requête n'a pas été précisée |
| 412 | Precondition Failed | Préconditions envoyées par la requête non-vérifiées |
| 413 | Request Entity Too Large | Traitement abandonné dû à une requête trop importante |
| 414 | Request-URI Too Long | URItrop longue |
| 415 | Unsupported Media Type | Format de requête non-supportée pour une méthode et une ressource données |

| Code | Message | Signification |
|---|---|---|
| 416 | Requested range unsatisfiable | Champs d'en-tête de requête «range» incorrect. |
| 417 | Expectation failed | Comportement attendu et défini dans l'en-tête de la requête insatisfaisable |
| 418 | I'm a teapot | "Je suis une théière". Ce code est défini dans la RFC 2324 datée du premier avril, Hyper Text Coffee Pot Control Protocol. Il n'y a pas d'implémentation de ce code. |
| 422 | Unprocessable entity | WebDAV: L'entité fournie avec la requête est incompréhensible ou incomplète. |
| 423 | Locked | WebDAV: L'opération ne peut avoir lieu car la ressource est verrouillée. |
| 424 | Method failure | WebDAV: Une méthode de la transaction a échoué. |
| 425 | Unordered Collection | WebDAV(RFC 3648). Ce code est défini dans le brouillon WebDAV Advanced Collections Protocol, mais est absent deWeb Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol |
| 426 | Upgrade Required | (RFC 2817) Le client devrait changer de protocole, par exemple au profit deTLS/1.0 |
| 449 | Retry With | Code défini par Microsoft. La requête devrait être renvoyée après avoir effectué une action. |
| 450 | Blocked by Windows Parental Controls | Code défini par Microsoft. Cette erreur est produite lorsque les outils de contrôle parental de Windows sont activés et bloquent l'accès à la page. |

- Codes commençant par 5

| Code | Message | Signification |
|---|---|---|
| 500 | Internal Server Error | Erreur interne du serveur |
| 501 | Not Implemented | Fonctionnalité réclamée non supportée par le serveur |
| 502 | Bad GatewayouProxy Error | Mauvaise réponse envoyée à un serveur intermédiaire par un autre serveur. |
| 503 | Service Unavailable | Service temporairement indisponible ou en maintenance |
| 504 | Gateway Time-out | Temps d'attente d'une réponse d'un serveur à un serveur intermédiaire écoulé |
| 505 | HTTP Version not supported | Version HTTP non gérée par le serveur |
| 507 | Insufficient storage | WebDAV: Espace insuffisant pour modifier les propriétés ou construire la collection |

| Code | Message | Signification |
| --- | --- | --- |
| 509 | Bandwidth Limit Exceeded | Utilisé par de nombreux serveurs pour indiquer un dépassement de quota. |

- Entêtes des requêtes HTTP

| Entête | Description | Exemple |
| --- | --- | --- |
| Accept | Content-Types that are acceptable | Accept: text/plain |
| Accept-Charset | Character sets that are acceptable | Accept-Charset: utf-8 |
| Accept-Encoding | Acceptable encodings. SeeHTTP compression. | Accept-Encoding: gzip, deflate |
| Accept-Language | Acceptable languages for response | Accept-Language: en-US |
| Accept-Datetime | Acceptable version in time | Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT |
| Authorization | Authentication credentials for HTTP authentication | Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ== |
| Cache-Control | Used to specify directives that MUST be obeyed by all caching mechanisms along the request/response chain | Cache-Control: no-cache |
| Connection | What type of connection the user-agent would prefer | Connection: keep-alive |
| Cookie | an HTTP cookie previously sent by the server withSet-Cookie(below) | Cookie: $Version=1; Skin=new; |
| Content-Length | The length of the request body inoctets(8-bit bytes) | Content-Length: 348 |
| Content-MD5 | ABase64-encoded binaryMD5sum of the content of the request body | Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ== |
| Content-Type | TheMIME typeof the body of the request (usedwith POST and PUT requests) | Content-Type:application/x-www-form-urlencoded |
| Date | The date and time that the message was sent | Date: Tue, 15 Nov 1994 08:12:31 GMT |
| Expect | Indicates that particular server behaviors are required by the client | Expect: 100-continue |
| From | The email address of the user making the request | From: user@example.com |

| Entête | Description | Exemple |
| --- | --- | --- |
| Host | The domain name of the server (forvirtual hosting), mandatory since HTTP/1.1. Although domain name are specified as case-insensitive[5][6], it is not specified whether the contents of the Host field should be interpreted in a case-insensitive manner[7]and in practice some implementations of virtual hosting interpret the contents of the Host field in a case-sensitive manner.[citation needed] | Host: en.wikipedia.org |
| If-Match | Only perform the action if the client supplied entity matches the same entity on the server. This is mainly for methods like PUT to only update a resource if it has not been modified since the user last updated it. | If-Match: "737060cd8c284d8af7ad3082f209582d" |
| If-Modified-Since | Allows a 304 Not Modified to be returned if content is unchanged | If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT |
| If-None-Match | Allows a 304 Not Modified to be returned if content is unchanged, seeHTTP ETag | If-None-Match: "737060cd8c284d8af7ad3082f209582d" |
| If-Range | If the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity | If-Range: "737060cd8c284d8af7ad3082f209582d" |
| If-Unmodified-Since | Only send the response if the entity has not been modified since a specific time. | If-Unmodified-Since: Sat, 29 Oct 1994 19:43:31 GMT |
| Max-Forwards | Limit the number of times the message can be forwarded through proxies or gateways. | Max-Forwards: 10 |
| Pragma | Implementation-specific headers that may have various effects anywhere along the request-response chain. | Pragma: no-cache |
| Proxy-Authorization | Authorization credentials for connecting to a proxy. | Proxy-Authorization: BasicQWxhZGRpbjpvcGVuIHNlc2FtZQ== |
| Range | Request only part of an entity. Bytes are numbered from 0. | Range: bytes=500-999 |
| Referer[sic] | This is the address of the previous web page from which a link to the currently requested page was followed. (The word “referrer” is misspelled in the RFC as well as in most implementations.) | Referer: http://en.wikipedia.org/wiki/Main_Page |

| Entête | Description | Exemple |
| --- | --- | --- |
| TE | The transfer encodings the user agent is willing to accept: the same values as for the response header Transfer-Encoding can be used, plus the "trailers" value (related to the "chunked" transfer method) to notify the server it expects to receive additional headers (the trailers) after the last, zero-sized, chunk. | TE: trailers,deflate |
| Upgrade | Ask the server to upgrade to another protocol. | Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11 |
| User-Agent | Theuser agent stringof the user agent | User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101 Firefox/12.0 |
| Via | Informs the server of proxies through which the request was sent. | Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1) |
| Warning | A general warning about possible problems with the entity body. | Warning: 199 Miscellaneous warning |

- Entête de réponse HTTP

| Entête | Description | Exemple |
| --- | --- | --- |
| Accept-Ranges | What partial content range types this server supports | Accept-Ranges: bytes |
| Age | The age the object has been in aproxy cachein seconds | Age: 12 |
| Allow | Valid actions for a specified resource. To be used for a 405 Method not allowed | Allow: GET, HEAD |
| Cache-Control | Tells all caching mechanisms from server to client whether they may cache this object. It is measured in seconds | Cache-Control: max-age=3600 |
| Connection | Options that are desired for the connection[17] | Connection: close |
| Content-Encoding | The type of encoding used on the data. SeeHTTP compression. | Content-Encoding: gzip |
| Content-Language | The language the content is in | Content-Language: da |
| Content-Length | The length of the response body inoctets(8-bit bytes) | Content-Length: 348 |
| Content-Location | An alternate location for the returned data | Content-Location: /index.htm |

| Entête | Description | Exemple |
|---|---|---|
| Content-MD5 | ABase64-encoded binaryMD5sum of the content of the response | Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ== |
| Content-Disposition[18][19][20] | An opportunity to raise a "File Download" dialogue box for a known MIME type with binary format or suggest a filename for dynamic content. Quotes are necessary with special characters. | Content-Disposition: attachment; filename="fname.ext" |
| Content-Range | Where in a full body message this partial message belongs | Content-Range: bytes 21010-47021/47022 |
| Content-Type | TheMIME typeof this content | Content-Type: text/html; charset=utf-8 |
| Date | The date and time that the message was sent | Date: Tue, 15 Nov 1994 08:12:31 GMT |
| ETag | An identifier for a specific version of a resource, often amessage digest | ETag: "737060cd8c284d8af7ad3082f209582d" |
| Expires | Gives the date/time after which the response is considered stale | Expires: Thu, 01 Dec 1994 16:00:00 GMT |
| Last-Modified | The last modified date for the requested object, inRFC 2822format | Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT |
| Link | Used to express a typed relationship with another resource, where the relation type is defined byRFC 5988 | Link: ; rel="alternate"[21] |
| Location | Used in redirection, or when a new resource has been created. | Location: http://www.w3.org/pub/WWW/People.html |
| P3P | This header is supposed to setP3Ppolicy, in the form ofP3P:CP="your_compact_policy". However, P3P did not take off, [22]most browsers have never fully implemented it, a lot of websites set this header with fake policy text, that was enough to fool browsers the existence of P3P policy and grant permissions forthird party cookies. | P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=151657 for more info." |
| Pragma | Implementation-specific headers that may have various effects anywhere along the request-response chain. | Pragma: no-cache |

| Entête | Description | Exemple |
| --- | --- | --- |
| Proxy-Authenticate | Request authentication to access the proxy. | Proxy-Authenticate: Basic |
| Refresh | Used in redirection, or when a new resource has been created. This refresh redirects after 5 seconds. This is a proprietary, non-standard header extension introduced by Netscape and supported by most web browsers. | Refresh: 5; url=http://www.w3.org/pub/WWW/People.html |
| Retry-After | If an entity is temporarily unavailable, this instructs the client to try again after a specified period of time (seconds). | Retry-After: 120 |
| Server | A name for the server | Server: Apache/2.4.1 (Unix) |
| Set-Cookie | anHTTP cookie | Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1 |
| Strict-Transport-Security | A HSTS Policy informing the HTTP client how long to cache the HTTPS only policy and whether this applies to subdomains. | Strict-Transport-Security: max-age=16070400; includeSubDomains |
| Trailer | The Trailer general field value indicates that the given set of header fields is present in the trailer of a message encoded withchunked transfer-coding. | Trailer: Max-Forwards |
| Transfer-Encoding | The form of encoding used to safely transfer the entity to the user.Currently defined methodsare:chunked, compress, deflate, gzip, identity. | Transfer-Encoding: chunked |
| Vary | Tells downstream proxies how to matchfuture request headers to decide whether the cached response can be used rather than requesting a fresh one from the origin server. | Vary: * |
| Via | Informs the client of proxies through which the response was sent. | Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1) |
| Warning | A general warning about possible problems with the entity body. | Warning: 199 Miscellaneous warning |

| Entête | Description | Exemple |
| --- | --- | --- |
| WWW-Authenticate | Indicates the authentication scheme that should be used to access the requested entity. | WWW-Authenticate: Basic |