Kpax Plataforma d'aprenentatge en Xarxa:
Migració de kPaxServer a NodeJs amb MongoDB

Annexes

Miquel A. Muntaner Morey
Administració web i comerç electrònic
Màster en Programari Lliure

Daniel Riera Terrén
Francisco Javier Noguera Otero

19 de juny de 2016

Índex dels Annexes

Taula HTTP response codes

Font: https://developer.mozilla.org/en-US/docs/Web/HTTP/Response_codes

Status code	Status text	Description	HTTP version
Informational responses			
100	Continue	This interim response indicates that everything so far is OK and that the client should continue with the request or ignore it if it is already finished.	HTTP/1.1 only
101	Switching Protocol	This code is sent in response to an `Upgrade:` request header by the client, and indicates that the protocol the server is switching too. It was introduced to allow migration to an incompatible protocol version, and is not in common use.	HTTP/1.1 only
Successful responses			
200	OK	The request has succeeded. The meaning of a success varies depending on the HTTP method: • GET: The resource has been fetched and is transmitted in the message body. • HEAD: The entity headers are in the message body. • POST: The resource describing the result of the action is transmitted in the message body. • TRACE: The message body contains the request message as received by the server	HTTP/0.9 and later
201	Created	The request has succeeded and a new resource has been created as a result of it. This is typically the response sent after a PUT request.	HTTP/0.9 and later
202	Accepted	The request has been received but not yet acted upon. It is non-committal, meaning that there is no way in HTTP to later send an asynchronous response indicating the outcome of processing the request. It is intended for cases where another process or server handles the request, or for batch processing.	HTTP/0.9 and later
203	Non-Authoritative Information	This response code means returned meta-information set is not exact set as available from the origin server, but collected from a local or a third party copy. Except this condition, 200 OK response should be preferred instead of this response.	HTTP/0.9 and 1.1
204	No Content	There is no content to send for this request, but the headers may be useful. The user-agent may update its cached headers for this resource with the new ones.	HTTP/0.9 and later
205	Reset Content	This response code is sent after accomplishing request to tell user agent reset document view which sent this request.	HTTP/1.1 only
206	Partial Content	This response code is used because of range header sent by the client to separate download into multiple streams.	HTTP/1.1 only
Redirection messages			
300	Multiple Choice	The request has more than one possible responses. User-agent or user should choose one of them. There is no standardized way to choose one of the responses.	HTTP/1.0 and later
301	Moved	This response code means that URI of requested	HTTP/0.9 and later

Status code	Status text	Description	HTTP version
	Permanently	resource has been changed. Probably, new URI would be given in the response.	
302	Found	This response code means that URI of requested resource has been changed *temporarily*. New changes in the URI might be made in the future. Therefore, this same URI should be used by the client in future requests.	HTTP/0.9 and later
303	See Other	Server sent this response to directing client to get requested resource to another URI with an GET request.	HTTP/0.9 and 1.1
304	Not Modified	This is used for caching purposes. It is telling to client that response has not been modified. So, client can continue to use same cached version of response.	HTTP/0.9 and later
305	Use Proxy	This means requested response must be accessed by a proxy. This response code is not largely supported because security reasons.	HTTP/1.1 only
306	*unused*	This response code is no longer used, it is just reserved currently. It was used in a previous version of the HTTP 1.1 specification.	HTTP/1.1 only
307	Temporary Redirect	Server sent this response to directing client to get requested resource to another URI with same method that used prior request. This has the same semantic than the `302 Found` HTTP response code, with the exception that the user agent *must not* change the HTTP method used: if a `POST` was used in the first request, a `POST` must be used in the second request.	HTTP/1.1 only
308	Permanent Redirect	This means that the resource is now permanently located at another URI, specified by the `Location:` HTTP Response header. This has the same semantics as the `301 Moved Permanently` HTTP response code, with the exception that the user agent *must not* change the HTTP method used: if a `POST` was used in the first request, a`POST` must be used in the second request. **Note:** This is an experimental response code whose specification is currently in draft form.	draft-reschke-http-status-308
Client error responses			
400	Bad Request	This response means that server could not understand the request due to invalid syntax.	HTTP/0.9 and later
401	Unauthorized	Authentication is needed to get requested response. This is similar to 403, but in this case, authentication is possible.	HTTP/0.9 and later
402	Payment Required	This response code is reserved for future use. Initial aim for creating this code was using it for digital payment systems however this is not used currently.	HTTP/0.9 and 1.1
403	Forbidden	Client does not have access rights to the content so server is rejecting to give proper response.	HTTP/0.9 and later
404	Not Found	Server can not find requested resource. This response code probably is most famous one due to its frequency to occur in web.	HTTP/0.9 and later
405	Method Not Allowed	The request method is known by the server but has been disabled and cannot be used. The two	HTTP/1.1 only

Status code	Status text	Description	HTTP version
		mandatory methods, `GET` and `HEAD`, must never be disabled and should not return this error code.	
406	Not Acceptable	This response is sent when the web server, after performing server-driven content negotiation, doesn't find any content following the criteria given by the user agent.	HTTP/1.1 only
407	Proxy Authentication Required	This is similar to 401 but authentication is needed to be done by a proxy.	HTTP/1.1 only
408	Request Timeout	This response is sent on an idle connection by some servers, even without any previous request by the client. It means that the server would like to shut down this unused connection. This response is used much more since some browsers, like Chrome or IE9, use HTTP preconnection mechanisms to speed up surfing (see bug 881804, which tracks the future implementation of such a mechanism in Firefox). Also note that some servers merely shut down the connection without sending this message.	HTTP/1.1 only
409	Conflict	This response would be sent when a request conflict with current state of server.	HTTP/1.1 only
410	Gone	This response would be sent when requested content has been deleted from server.	HTTP/1.1 only
411	Length Required	Server rejected the request because the `Content-Length` header field is not defined and the server requires it.	HTTP/1.1 only
412	Precondition Failed	The client has indicated preconditions in its headers which the server does not meet.	HTTP/1.1 only
413	Payload Too Large	Request entity is larger than limits defined by server; the server might close the connection or return an `Retry-After` header field.	HTTP/1.1 only
414	URI Too Long	The URI requested by the client is longer than the server is willing to interpret.	HTTP/1.1 only
415	Unsupported Media Type	The media format of the requested data is not supported by the server, so the server is rejecting the request.	HTTP/1.1 only
416	Requested Range Not Satisfiable	The range specified by the `Range` header field in the request can't be fulfilled; it's possible that the range is outside the size of the target URI's data.	HTTP/1.1 only
417	Expectation Failed	This response code means the expectation indicated by the `Expect` request header field can't be met by the server.	HTTP/1.1 only
418	I'm a teapot	Any attempt to brew coffee with a teapot should result in the error code "418 I'm a teapot". The resulting entity body MAY be short and stout.	HTCPCP/1.0
421	Misdirected Request	The request was directed at a server that is not able to produce a response. This can be sent by a server that is not configured to produce responses for the combination of scheme and authority that are included in the request URI.	HTTP/2.0

Status code	Status text	Description	HTTP version
426	Upgrade Required	The server refuses to perform the request using the current protocol but might be willing to do so after the client upgrades to a different protocol. The server MUST send an Upgrade header field in a 426 response to indicate the required protocol(s) (Section 6.7 of [RFC7230]).	HTTP/1.1 and newer
428	Precondition Required	The origin server requires the request to be conditional. Intended to prevent "the 'lost update' problem, where a client GETs a resource's state, modifies it, and PUTs it back to the server, when meanwhile a third party has modified the state on the server, leading to a conflict."	HTTP/1.1 and newer
429	Too Many Requests	The user has sent too many requests in a given amount of time ("rate limiting").	HTTP/1.1 and newer
431	Request Header Fields Too Large	The server is unwilling to process the request because its header fields are too large. The request MAY be resubmitted after reducing the size of the request header fields.	HTTP/1.1 and newer
Server error responses			
500	Internal Server Error	The server has encountered a situation it doesn't know how to handle.	HTTP/0.9 and later
501	Not Implemented	The request method is not supported by the server and cannot be handled. The only methods that servers are required to support (and therefore that must not return this code) are `GET` and `HEAD`.	HTTP/0.9 and later
502	Bad Gateway	This error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response.	HTTP/0.9 and later
503	Service Unavailable	The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded. Note that together with this response, a user-friendly page explaining the problem should be sent. This responses should be used for temporary conditions and the `Retry-After:` HTTP header should, if possible, contain the estimated time before the recovery of the service. The webmaster must also take care about the caching-related headers that are sent along with this response, as these temporary condition responses should usually not be cached.	HTTP/0.9 and later
504	Gateway Timeout	This error response is given when the server is acting as a gateway and cannot get a response in time.	HTTP/1.1 only
505	HTTP Version Not Supported	The HTTP version used in the request is not supported by the server.	HTTP/1.1 only
506	Variant Also Negotiates	The server has an internal configuration error: transparent content negotiation for the request results in a circular reference.	HTTP/1.1
507	Variant Also Negotiates	The server has an internal configuration error: the chosen variant resource is configured to engage in transparent content negotiation itself, and is therefore not a proper end point in the negotiation process.	HTTP/1.1
511	Network Authentication	The 511 status code indicates that the client needs to authenticate to gain network access.	HTTP/1.1

Status code	Status text	Description	HTTP version
	Required		

Taula de serveis Web[1] definits al servidor.

Serveis Web d'usuari	User/ /user/list?q={} /user/listall /user/:user_id /user/del

Serveis Web de jocs	/game /game/list?q={} /game/listall /game/:game /game/:game/like /game/:game/unlike /games/del

Detall

1 Endpoints (serveis web)

user/

URL: /user
METHOD: POST
PARAMS: name required
login required

Object: Adds a **new** user in the system

Info stru

Info Out: None.

Info Saved:
```
var user = {
        login: req.body.login,
        name: req.body.name,
        created_at: now,
        updated_at: now,
        status: 1
}
```

/user/list?q={}

URL: /user/list?q={ *JSON query* }
METHOD: GET
PARAMS: { JSON MongoDB query } optional

Object: List users under a free condition
* if no parameter passed, all users are listed
* the 'q' query must be a valid JSON query condition in MongoBD format

Llista de usuaris sota una condició parametritzable
* Si no hi ha cap paràmetre, es retornen tots els usuaris del sistema
* La consulta 'q' ha de ser una cadena JSON per filtratges segons la sintaxi de MongoDB

Info stru:

info Out: A MongoDB cursor containing list of users.
Un cursor de MongoDB que conté la col·lecció de registres demanda.

/user/listall

URL: /user/listall
METHOD: GET
PARAMS: No parameters

Object: List users ALL the users in the system
* No parameters needed

Llista tots els usuaris del sistema
* No necessita cap paràmetre

Info stru:

info Out: A MongoDB cursor containing list of users.
Un cursor de MongoDB que conté la col·lecció de registres demanda.

/user/:user

URL: /user/user
METHOD: GET
PARAMS: user

Object: List info of ONE user (by Id of the user)
* user_id : id of the user in the system

Llista UN usuari del sistema
* user: la Id del usuari requerit

Info stru:

info Out: A MongoDB cursor containing info of the required user
Un cursor de MongoDB que conté la informació del registre demanda.

/user/del

URL: /user/del
METHOD: POST
PARAMS: user_id The Id of the user

Object: Deletes the user from the system
IN FACT , just changes the status value into '3' , deleted!

* Sets status = 3 , (meaning DELETED)

Esborra un usuari del sistema. De fet, simplement estableix el camp ststus a '3' per indicar que el usuari esta 'esborrat'

* user_id: identificador del usuari

Info stru:

info Out: No Out

/game

URL: /game
METHOD: POST
PARAMS: name required
owner_id required

Object: Adds a **new** game in the system

Info stru

Info Out: None.

Info Saved:
```
var game = {
        name: req.body.name,
        owner: req.body.owner_id,
        status: 1,
        nlikes: 0,
        created_at: now,
        updated_at: now
}
```

/game/list?q={}

URL: /games/list?q={ *JSON query* }
METHOD: GET
PARAMS: { JSON MongoDB query } optional

Object: List games under a free condition
* if no parameter passed, all games are listed
* the 'q' query must be a valid JSON query condition in MongoBD format

Llista de jocs sota una condició parametritzable
* Si no hi ha cap paràmetre, es retornen tots els jocs del sistema
* La consulta 'q' ha de ser una cadena JSON per filtratges segons la sintaxi de MongoDB

Info stru:

info Out: A MongoDB cursor containing list of games.
Un cursor de MongoDB que conté la col·lecció de registres demanda.

/game/listall

URL: /games/listall
METHOD: GET
PARAMS: No parameters

Object: List games ALL the games in the system
* No parameters needed

Llista tots els jocs del sistema
* No necessita cap paràmetre

Info stru:

info Out: A MongoDB cursor containing list of games.
Un cursor de MongoDB que conté la col·lecció de registres demanda.

/game/:game

URL: /games/:game
METHOD: GET
PARAMS: game (id)

Object: List info of ONE game (by Id of the Game)
* game (id) : id of the Game in the system

Llista UN joc del sistema
* game (id): la Id del Joc requerit

Info stru:

info Out: A MongoDB cursor containing info of the required game
Un cursor de MongoDB que conté la informació del registre demanda.

/game/:game/like

URL: /games/like
METHOD: POST
PARAMS: name The name of the game

Object: Stores additional info of user_id and date/time

* adds 1 to the nlike counter of the game , identified by game_id
* additinal info of user and date/time when the 'nlike' is added

Incrementa el comptador nlike del joc identificat pel seu nom
* name: nom del joc en el sistema
* grava informació del l'usuari i la data i hora en que s'ha produït el like, sempre que no repeteixi el like sobre el mateix joc; en aquest darrer cas no actua.

Info stru:

info Out: No Out

/game/:game/unlike

URL: /games/unlike
METHOD: POST
PARAMS:

game	The Id of the game
user	The user who un-likes the program

Object:

decreases by one the nlikes countes
Stores additional info of user_id and date/time

* adds 1 to the nlike counter of the game , identified by game_id
* additinal info of user and date/time when the 'nlike' is added

Decrementa el comptador nlike del joc identificat pel Id
* game (id): identificador del joc
* user (id) : identificador de l'usuari que marca el 'nlike'
* esborra del sistema la informació del like de l'usuari, si previamenta havia marca el joc amb like, si no, no realitza cap acció.

Info stru:

info Out: No Out

/games/del

URL: /games/del
METHOD: POST
PARAMS: game_id The Id of the game

Object: Deletes the game from the system
IN FACT , just changes the status value into '3' , deleted!

* Sets status = 3 , (meaning DELETED)

Esborra un joc del sistema. De fet, simplement estableix el camp ststus a '3' per indicar que el joc esta 'esborrat'

* game_id: identificador del joc

Info stru:

info Out: No Out