[Overview](http://docs.google.com/index.html)

### Packages

* [CodeIgniter](http://docs.google.com/package-CodeIgniter.html)
  + [Libraries](http://docs.google.com/package-CodeIgniter.Libraries.html)
  + [Rest](http://docs.google.com/package-CodeIgniter.Rest.html)
* [None](http://docs.google.com/package-None.html)

### Classes

* [REST\_Controller](http://docs.google.com/class-REST_Controller.html)
* [Overview](http://docs.google.com/index.html)
* Package
* Class

Class REST\_Controller

CodeIgniter Rest Controller A fully RESTful server implementation for CodeIgniter using one library, one config file and one controller.

CI\_Controller  **REST\_Controller**

Direct known subclasses

[Example](http://docs.google.com/class-Example.html), [Key](http://docs.google.com/class-Key.html)

**Abstract**

**Package:** [CodeIgniter](http://docs.google.com/package-CodeIgniter.html)\[Libraries](http://docs.google.com/package-CodeIgniter.Libraries.html)

**Category:** Libraries

**License:** [MIT](http://docs.google.com/MIT)

**Author:** Phil Sturgeon, Chris Kacerguis

**Version:** 3.0.0

**Link:** <https://github.com/chriskacerguis/codeigniter-restserver>

**Located at** [libraries/REST\_Controller.php](http://docs.google.com/source-class-REST_Controller.html#5-2211)

Methods summary

| protected | [#](#gjdgxs) [early\_checks](http://docs.google.com/source-class-REST_Controller.html#362-370)( )  Extend this function to apply additional checking early on in the process  Extend this function to apply additional checking early on in the process |
| --- | --- |
| public | [#](#30j0zll) [\_\_construct](http://docs.google.com/source-class-REST_Controller.html#372-577)( string $config = 'rest' )  Constructor for the REST API  Constructor for the REST API Parameters $config  Configuration filename minus the file extension e.g: my\_rest.php is passed as 'my\_rest' |
| public | [#](#1fob9te) [\_\_destruct](http://docs.google.com/source-class-REST_Controller.html#579-596)( )  Deconstructor  Deconstructor Author Chris Kacerguis |
| public | [#](#3znysh7) [\_remap](http://docs.google.com/source-class-REST_Controller.html#598-716)( string $object\_called, array $arguments = [] )  Requests are not made to methods directly, the request will be for an "object". This simply maps the object and method to the correct Controller method  Requests are not made to methods directly, the request will be for an "object". This simply maps the object and method to the correct Controller method Parameters $object\_called $arguments The arguments passed to the controller method |
| public | [#](#2et92p0) [response](http://docs.google.com/source-class-REST_Controller.html#718-799)( array|null $data = NULL, integer|null $http\_code = NULL, boolean $continue = FALSE )  Takes mixed data and optionally a status code, then creates the response  Takes mixed data and optionally a status code, then creates the response Parameters $data Data to output to the user $http\_code HTTP status code $continue  TRUE to flush the response to the client and continue running the script; otherwise, exit |
| public | [#](#tyjcwt) [set\_response](http://docs.google.com/source-class-REST_Controller.html#801-814)( array|null $data = NULL, integer|null $http\_code = NULL )  Takes mixed data and optionally a status code, then creates the response within the buffers of the Output class. The response is sent to the client lately by the framework, after the current controller's method termination. All the hooks after the controller's method termination are executable  Takes mixed data and optionally a status code, then creates the response within the buffers of the Output class. The response is sent to the client lately by the framework, after the current controller's method termination. All the hooks after the controller's method termination are executable Parameters $data Data to output to the user $http\_code HTTP status code |
| protected string|null | [#](#3dy6vkm) [\_detect\_input\_format](http://docs.google.com/source-class-REST_Controller.html#816-848)( )  Get the input format e.g. json or xml  Get the input format e.g. json or xml Returns string|null  Supported input format; otherwise, NULL |
| protected string | [#](#1t3h5sf) [\_get\_default\_output\_format](http://docs.google.com/source-class-REST_Controller.html#850-862)( )  Gets the default format from the configuration. Fallbacks to 'json' if the corresponding configuration option $config['rest\_default\_format'] is missing or is empty  Gets the default format from the configuration. Fallbacks to 'json' if the corresponding configuration option $config['rest\_default\_format'] is missing or is empty Returns string  The default supported input format |
| protected mixed|null|string | [#](#4d34og8) [\_detect\_output\_format](http://docs.google.com/source-class-REST_Controller.html#864-933)( )  Detect which format should be used to output the data  Detect which format should be used to output the data Returns mixed|null|string  Output format |
| protected string|null | [#](#2s8eyo1) [\_detect\_method](http://docs.google.com/source-class-REST_Controller.html#935-965)( )  Get the HTTP request string e.g. get or post  Get the HTTP request string e.g. get or post Returns string|null  Supported request method as a lowercase string; otherwise, NULL if not supported |
| protected boolean | [#](#17dp8vu) [\_detect\_api\_key](http://docs.google.com/source-class-REST_Controller.html#967-1039)( )  See if the user has provided an API key  See if the user has provided an API key Returns boolean |
| protected string|null | [#](#3rdcrjn) [\_detect\_lang](http://docs.google.com/source-class-REST_Controller.html#1041-1073)( )  Preferred return language  Preferred return language Returns string|null  The language code |
| protected boolean | [#](#26in1rg) [\_log\_request](http://docs.google.com/source-class-REST_Controller.html#1075-1101)( boolean $authorized = FALSE )  Add the request to the log table  Add the request to the log table Parameters $authorized TRUE the user is authorized; otherwise, FALSE Returns boolean  TRUE the data was inserted; otherwise, FALSE |
| protected boolean | [#](#lnxbz9) [\_check\_limit](http://docs.google.com/source-class-REST_Controller.html#1103-1203)( string $controller\_method )  Check if the requests to a controller method exceed a limit  Check if the requests to a controller method exceed a limit Parameters $controller\_method The method being called Returns boolean  TRUE the call limit is below the threshold; otherwise, FALSE |
| protected boolean | [#](#35nkun2) [\_auth\_override\_check](http://docs.google.com/source-class-REST_Controller.html#1205-1395)( )  Check if there is a specific auth type set for the current class/method/HTTP-method being called  Check if there is a specific auth type set for the current class/method/HTTP-method being called Returns boolean |
| protected | [#](#1ksv4uv) [\_parse\_get](http://docs.google.com/source-class-REST_Controller.html#1397-1407)( )  Parse the GET request arguments  Parse the GET request arguments |
| protected | [#](#44sinio) [\_parse\_post](http://docs.google.com/source-class-REST_Controller.html#1409-1423)( )  Parse the POST request arguments  Parse the POST request arguments |
| protected | [#](#2jxsxqh) [\_parse\_put](http://docs.google.com/source-class-REST_Controller.html#1425-1442)( )  Parse the PUT request arguments  Parse the PUT request arguments |
| protected | [#](#z337ya) [\_parse\_head](http://docs.google.com/source-class-REST_Controller.html#1444-1457)( )  Parse the HEAD request arguments  Parse the HEAD request arguments |
| protected | [#](#3j2qqm3) [\_parse\_options](http://docs.google.com/source-class-REST_Controller.html#1459-1472)( )  Parse the OPTIONS request arguments  Parse the OPTIONS request arguments |
| protected | [#](#1y810tw) [\_parse\_patch](http://docs.google.com/source-class-REST_Controller.html#1474-1492)( )  Parse the PATCH request arguments  Parse the PATCH request arguments |
| protected | [#](#4i7ojhp) [\_parse\_delete](http://docs.google.com/source-class-REST_Controller.html#1494-1507)( )  Parse the DELETE request arguments  Parse the DELETE request arguments |
| protected | [#](#2xcytpi) [\_parse\_query](http://docs.google.com/source-class-REST_Controller.html#1509-1518)( )  Parse the query parameters  Parse the query parameters |
| public array|string|null | [#](#1ci93xb) [get](http://docs.google.com/source-class-REST_Controller.html#1522-1539)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a GET request  Retrieve a value from a GET request Parameters $key  Key to retrieve from the GET request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the GET request; otherwise, NULL |
| public array|string|null | [#](#3whwml4) [options](http://docs.google.com/source-class-REST_Controller.html#1541-1558)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a OPTIONS request  Retrieve a value from a OPTIONS request Parameters $key  Key to retrieve from the OPTIONS request. If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the OPTIONS request; otherwise, NULL |
| public array|string|null | [#](#2bn6wsx) [head](http://docs.google.com/source-class-REST_Controller.html#1560-1577)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a HEAD request  Retrieve a value from a HEAD request Parameters $key  Key to retrieve from the HEAD request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the HEAD request; otherwise, NULL |
| public array|string|null | [#](#qsh70q) [post](http://docs.google.com/source-class-REST_Controller.html#1579-1596)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a POST request  Retrieve a value from a POST request Parameters $key  Key to retrieve from the POST request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the POST request; otherwise, NULL |
| public array|string|null | [#](#3as4poj) [put](http://docs.google.com/source-class-REST_Controller.html#1598-1615)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a PUT request  Retrieve a value from a PUT request Parameters $key  Key to retrieve from the PUT request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the PUT request; otherwise, NULL |
| public array|string|null | [#](#1pxezwc) [delete](http://docs.google.com/source-class-REST_Controller.html#1617-1634)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a DELETE request  Retrieve a value from a DELETE request Parameters $key  Key to retrieve from the DELETE request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the DELETE request; otherwise, NULL |
| public array|string|null | [#](#49x2ik5) [patch](http://docs.google.com/source-class-REST_Controller.html#1636-1653)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from a PATCH request  Retrieve a value from a PATCH request Parameters $key  Key to retrieve from the PATCH request If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the PATCH request; otherwise, NULL |
| public array|string|null | [#](#2p2csry) [query](http://docs.google.com/source-class-REST_Controller.html#1655-1672)( null $key = NULL, null $xss\_clean = NULL )  Retrieve a value from the query parameters  Retrieve a value from the query parameters Parameters $key  Key to retrieve from the query parameters If NULL an array of arguments is returned  $xss\_clean Whether to apply XSS filtering Returns array|string|null  Value from the query parameters; otherwise, NULL |
| protected string | [#](#147n2zr) [\_xss\_clean](http://docs.google.com/source-class-REST_Controller.html#1674-1688)( string $value, boolean $xss\_clean )  Sanitizes data so that Cross Site Scripting Hacks can be prevented  Sanitizes data so that Cross Site Scripting Hacks can be prevented Parameters $value Input data $xss\_clean Whether to apply XSS filtering Returns string |
| public array | [#](#3o7alnk) [validation\_errors](http://docs.google.com/source-class-REST_Controller.html#1690-1701)( )  Retrieve the validation errors  Retrieve the validation errors Returns array |
| protected boolean | [#](#23ckvvd) [\_perform\_ldap\_auth](http://docs.google.com/source-class-REST_Controller.html#1705-1798)( string $username = '', string $password = NULL )  Perform LDAP Authentication  Perform LDAP Authentication Parameters $username The username to validate $password The password to validate Returns boolean |
| protected boolean | [#](#ihv636) [\_perform\_library\_auth](http://docs.google.com/source-class-REST_Controller.html#1800-1837)( string $username = '', string $password = NULL )  Perform Library Authentication - Override this function to change the way the library is called  Perform Library Authentication - Override this function to change the way the library is called Parameters $username The username to validate $password The password to validate Returns boolean |
| protected boolean | [#](#32hioqz) [\_check\_login](http://docs.google.com/source-class-REST_Controller.html#1839-1894)( string $username = NULL, boolean|string $password = FALSE )  Check if the user is logged in  Check if the user is logged in Parameters $username The user's name $password The user's password Returns boolean |
| protected | [#](#1hmsyys) [\_check\_php\_session](http://docs.google.com/source-class-REST_Controller.html#1896-1916)( )  Check to see if the user is logged in with a PHP session key  Check to see if the user is logged in with a PHP session key |
| protected | [#](#41mghml) [\_prepare\_basic\_auth](http://docs.google.com/source-class-REST_Controller.html#1918-1957)( )  Prepares for basic authentication  Prepares for basic authentication |
| protected | [#](#2grqrue) [\_prepare\_digest\_auth](http://docs.google.com/source-class-REST_Controller.html#1959-2014)( )  Prepares for digest authentication  Prepares for digest authentication |
| protected | [#](#vx1227) [\_check\_blacklist\_auth](http://docs.google.com/source-class-REST_Controller.html#2016-2036)( )  Checks if the client's ip is in the 'rest\_ip\_blacklist' config and generates a 401 response  Checks if the client's ip is in the 'rest\_ip\_blacklist' config and generates a 401 response |
| protected | [#](#3fwokq0) [\_check\_whitelist\_auth](http://docs.google.com/source-class-REST_Controller.html#2038-2064)( )  Check if the client's ip is in the 'rest\_ip\_whitelist' config and generates a 401 response  Check if the client's ip is in the 'rest\_ip\_whitelist' config and generates a 401 response |
| protected | [#](#1v1yuxt) [\_force\_login](http://docs.google.com/source-class-REST_Controller.html#2066-2097)( string $nonce = '' )  Force logging in by setting the WWW-Authenticate header  Force logging in by setting the WWW-Authenticate header Parameters $nonce  A server-specified data string which should be uniquely generated each time |
| protected boolean | [#](#4f1mdlm) [\_log\_access\_time](http://docs.google.com/source-class-REST_Controller.html#2099-2114)( )  Updates the log table with the total access time  Updates the log table with the total access time Returns boolean  TRUE log table updated; otherwise, FALSE Author Chris Kacerguis |
| protected boolean | [#](#2u6wntf) [\_log\_response\_code](http://docs.google.com/source-class-REST_Controller.html#2116-2132)( $http\_code )  Updates the log table with HTTP response code  Updates the log table with HTTP response code Parameters $http\_code HTTP status code Returns boolean  TRUE log table updated; otherwise, FALSE Author Justin Chen |
| protected boolean | [#](#19c6y18) [\_check\_access](http://docs.google.com/source-class-REST_Controller.html#2134-2164)( )  Check to see if the API key has access to the controller and methods  Check to see if the API key has access to the controller and methods Returns boolean  TRUE the API key has access; otherwise, FALSE |
| protected | [#](#3tbugp1) [\_check\_cors](http://docs.google.com/source-class-REST_Controller.html#2166-2209)( )  Checks allowed domains, and adds appropriate headers for HTTP access control (CORS)  Checks allowed domains, and adds appropriate headers for HTTP access control (CORS) |

Constants summary

| integer | [**HTTP\_CONTINUE**](http://docs.google.com/source-class-REST_Controller.html#23) | [#](#28h4qwu) 100 |
| --- | --- | --- |
| integer | [**HTTP\_SWITCHING\_PROTOCOLS**](http://docs.google.com/source-class-REST_Controller.html#24) | [#](#nmf14n) 101 |
| integer | [**HTTP\_PROCESSING**](http://docs.google.com/source-class-REST_Controller.html#25) | [#](#37m2jsg) 102 |
| integer | [**HTTP\_OK**](http://docs.google.com/source-class-REST_Controller.html#29-32)  The request has succeeded  The request has succeeded | [#](#1mrcu09) 200 |
| integer | [**HTTP\_CREATED**](http://docs.google.com/source-class-REST_Controller.html#34-37)  The server successfully created a new resource  The server successfully created a new resource | [#](#46r0co2) 201 |
| integer | [**HTTP\_ACCEPTED**](http://docs.google.com/source-class-REST_Controller.html#38) | [#](#2lwamvv) 202 |
| integer | [**HTTP\_NON\_AUTHORITATIVE\_INFORMATION**](http://docs.google.com/source-class-REST_Controller.html#39) | [#](#111kx3o) 203 |
| integer | [**HTTP\_NO\_CONTENT**](http://docs.google.com/source-class-REST_Controller.html#41-44)  The server successfully processed the request, though no content is returned  The server successfully processed the request, though no content is returned | [#](#3l18frh) 204 |
| integer | [**HTTP\_RESET\_CONTENT**](http://docs.google.com/source-class-REST_Controller.html#45) | [#](#206ipza) 205 |
| integer | [**HTTP\_PARTIAL\_CONTENT**](http://docs.google.com/source-class-REST_Controller.html#46) | [#](#4k668n3) 206 |
| integer | [**HTTP\_MULTI\_STATUS**](http://docs.google.com/source-class-REST_Controller.html#47) | [#](#2zbgiuw) 207 |
| integer | [**HTTP\_ALREADY\_REPORTED**](http://docs.google.com/source-class-REST_Controller.html#48) | [#](#1egqt2p) 208 |
| integer | [**HTTP\_IM\_USED**](http://docs.google.com/source-class-REST_Controller.html#49) | [#](#3ygebqi) 226 |
| integer | [**HTTP\_MULTIPLE\_CHOICES**](http://docs.google.com/source-class-REST_Controller.html#53) | [#](#2dlolyb) 300 |
| integer | [**HTTP\_MOVED\_PERMANENTLY**](http://docs.google.com/source-class-REST_Controller.html#54) | [#](#sqyw64) 301 |
| integer | [**HTTP\_FOUND**](http://docs.google.com/source-class-REST_Controller.html#55) | [#](#3cqmetx) 302 |
| integer | [**HTTP\_SEE\_OTHER**](http://docs.google.com/source-class-REST_Controller.html#56) | [#](#1rvwp1q) 303 |
| integer | [**HTTP\_NOT\_MODIFIED**](http://docs.google.com/source-class-REST_Controller.html#58-61)  The resource has not been modified since the last request  The resource has not been modified since the last request | [#](#4bvk7pj) 304 |
| integer | [**HTTP\_USE\_PROXY**](http://docs.google.com/source-class-REST_Controller.html#62) | [#](#2r0uhxc) 305 |
| integer | [**HTTP\_RESERVED**](http://docs.google.com/source-class-REST_Controller.html#63) | [#](#1664s55) 306 |
| integer | [**HTTP\_TEMPORARY\_REDIRECT**](http://docs.google.com/source-class-REST_Controller.html#64) | [#](#3q5sasy) 307 |
| integer | [**HTTP\_PERMANENTLY\_REDIRECT**](http://docs.google.com/source-class-REST_Controller.html#65) | [#](#25b2l0r) 308 |
| integer | [**HTTP\_BAD\_REQUEST**](http://docs.google.com/source-class-REST_Controller.html#69-72)  The request cannot be fulfilled due to multiple errors  The request cannot be fulfilled due to multiple errors | [#](#kgcv8k) 400 |
| integer | [**HTTP\_UNAUTHORIZED**](http://docs.google.com/source-class-REST_Controller.html#74-77)  The user is unauthorized to access the requested resource  The user is unauthorized to access the requested resource | [#](#34g0dwd) 401 |
| integer | [**HTTP\_PAYMENT\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#78) | [#](#1jlao46) 402 |
| integer | [**HTTP\_FORBIDDEN**](http://docs.google.com/source-class-REST_Controller.html#80-83)  The requested resource is unavailable at this present time  The requested resource is unavailable at this present time | [#](#43ky6rz) 403 |
| integer | [**HTTP\_NOT\_FOUND**](http://docs.google.com/source-class-REST_Controller.html#85-91)  The requested resource could not be found  The requested resource could not be found  Note: This is sometimes used to mask if there was an UNAUTHORIZED (401) or FORBIDDEN (403) error, for security reasons | [#](#2iq8gzs) 404 |
| integer | [**HTTP\_METHOD\_NOT\_ALLOWED**](http://docs.google.com/source-class-REST_Controller.html#93-96)  The request method is not supported by the following resource  The request method is not supported by the following resource | [#](#xvir7l) 405 |
| integer | [**HTTP\_NOT\_ACCEPTABLE**](http://docs.google.com/source-class-REST_Controller.html#98-101)  The request was not acceptable  The request was not acceptable | [#](#3hv69ve) 406 |
| integer | [**HTTP\_PROXY\_AUTHENTICATION\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#102) | [#](#1x0gk37) 407 |
| integer | [**HTTP\_REQUEST\_TIMEOUT**](http://docs.google.com/source-class-REST_Controller.html#103) | [#](#4h042r0) 408 |
| integer | [**HTTP\_CONFLICT**](http://docs.google.com/source-class-REST_Controller.html#105-109)  The request could not be completed due to a conflict with the current state of the resource  The request could not be completed due to a conflict with the current state of the resource | [#](#2w5ecyt) 409 |
| integer | [**HTTP\_GONE**](http://docs.google.com/source-class-REST_Controller.html#110) | [#](#1baon6m) 410 |
| integer | [**HTTP\_LENGTH\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#111) | [#](#3vac5uf) 411 |
| integer | [**HTTP\_PRECONDITION\_FAILED**](http://docs.google.com/source-class-REST_Controller.html#112) | [#](#2afmg28) 412 |
| integer | [**HTTP\_REQUEST\_ENTITY\_TOO\_LARGE**](http://docs.google.com/source-class-REST_Controller.html#113) | [#](#pkwqa1) 413 |
| integer | [**HTTP\_REQUEST\_URI\_TOO\_LONG**](http://docs.google.com/source-class-REST_Controller.html#114) | [#](#39kk8xu) 414 |
| integer | [**HTTP\_UNSUPPORTED\_MEDIA\_TYPE**](http://docs.google.com/source-class-REST_Controller.html#115) | [#](#1opuj5n) 415 |
| integer | [**HTTP\_REQUESTED\_RANGE\_NOT\_SATISFIABLE**](http://docs.google.com/source-class-REST_Controller.html#116) | [#](#48pi1tg) 416 |
| integer | [**HTTP\_EXPECTATION\_FAILED**](http://docs.google.com/source-class-REST_Controller.html#117) | [#](#2nusc19) 417 |
| integer | [**HTTP\_I\_AM\_A\_TEAPOT**](http://docs.google.com/source-class-REST_Controller.html#118) | [#](#1302m92) 418 |
| integer | [**HTTP\_UNPROCESSABLE\_ENTITY**](http://docs.google.com/source-class-REST_Controller.html#119) | [#](#3mzq4wv) 422 |
| integer | [**HTTP\_LOCKED**](http://docs.google.com/source-class-REST_Controller.html#120) | [#](#2250f4o) 423 |
| integer | [**HTTP\_FAILED\_DEPENDENCY**](http://docs.google.com/source-class-REST_Controller.html#121) | [#](#haapch) 424 |
| integer | [**HTTP\_RESERVED\_FOR\_WEBDAV\_ADVANCED\_COLLECTIONS\_EXPIRED\_PROPOSAL**](http://docs.google.com/source-class-REST_Controller.html#122) | [#](#319y80a) 425 |
| integer | [**HTTP\_UPGRADE\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#123) | [#](#1gf8i83) 426 |
| integer | [**HTTP\_PRECONDITION\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#124) | [#](#40ew0vw) 428 |
| integer | [**HTTP\_TOO\_MANY\_REQUESTS**](http://docs.google.com/source-class-REST_Controller.html#125) | [#](#2fk6b3p) 429 |
| integer | [**HTTP\_REQUEST\_HEADER\_FIELDS\_TOO\_LARGE**](http://docs.google.com/source-class-REST_Controller.html#126) | [#](#upglbi) 431 |
| integer | [**HTTP\_INTERNAL\_SERVER\_ERROR**](http://docs.google.com/source-class-REST_Controller.html#130-136)  The server encountered an unexpected error  The server encountered an unexpected error  Note: This is a generic error message when no specific message is suitable | [#](#3ep43zb) 500 |
| integer | [**HTTP\_NOT\_IMPLEMENTED**](http://docs.google.com/source-class-REST_Controller.html#138-141)  The server does not recognise the request method  The server does not recognise the request method | [#](#1tuee74) 501 |
| integer | [**HTTP\_BAD\_GATEWAY**](http://docs.google.com/source-class-REST_Controller.html#142) | [#](#4du1wux) 502 |
| integer | [**HTTP\_SERVICE\_UNAVAILABLE**](http://docs.google.com/source-class-REST_Controller.html#143) | [#](#2szc72q) 503 |
| integer | [**HTTP\_GATEWAY\_TIMEOUT**](http://docs.google.com/source-class-REST_Controller.html#144) | [#](#184mhaj) 504 |
| integer | [**HTTP\_VERSION\_NOT\_SUPPORTED**](http://docs.google.com/source-class-REST_Controller.html#145) | [#](#3s49zyc) 505 |
| integer | [**HTTP\_VARIANT\_ALSO\_NEGOTIATES\_EXPERIMENTAL**](http://docs.google.com/source-class-REST_Controller.html#146) | [#](#279ka65) 506 |
| integer | [**HTTP\_INSUFFICIENT\_STORAGE**](http://docs.google.com/source-class-REST_Controller.html#147) | [#](#meukdy) 507 |
| integer | [**HTTP\_LOOP\_DETECTED**](http://docs.google.com/source-class-REST_Controller.html#148) | [#](#36ei31r) 508 |
| integer | [**HTTP\_NOT\_EXTENDED**](http://docs.google.com/source-class-REST_Controller.html#149) | [#](#1ljsd9k) 510 |
| integer | [**HTTP\_NETWORK\_AUTHENTICATION\_REQUIRED**](http://docs.google.com/source-class-REST_Controller.html#150) | [#](#45jfvxd) 511 |

Properties summary

| protected string|null | [$rest\_format](http://docs.google.com/source-class-REST_Controller.html#152-158)  This defines the rest format Must be overridden it in a controller so that it is set  This defines the rest format Must be overridden it in a controller so that it is set | [#](#2koq656) NULL |
| --- | --- | --- |
| protected array | [$methods](http://docs.google.com/source-class-REST_Controller.html#160-165)  Defines the list of method properties such as limit, log and level  Defines the list of method properties such as limit, log and level | [#](#zu0gcz) [] |
| protected array | [$allowed\_http\_methods](http://docs.google.com/source-class-REST_Controller.html#167-172)  List of allowed HTTP methods  List of allowed HTTP methods | [#](#3jtnz0s) ['get', 'delete', 'post', 'put', 'options', 'patch', 'head'] |
| protected object | [$request](http://docs.google.com/source-class-REST_Controller.html#174-181)  Contains details about the request Fields: body, format, method, ssl Note: This is a dynamic object (stdClass)  Contains details about the request Fields: body, format, method, ssl Note: This is a dynamic object (stdClass) | [#](#1yyy98l) NULL |
| protected object | [$response](http://docs.google.com/source-class-REST_Controller.html#183-190)  Contains details about the response Fields: format, lang Note: This is a dynamic object (stdClass)  Contains details about the response Fields: format, lang Note: This is a dynamic object (stdClass) | [#](#4iylrwe) NULL |
| protected object | [$rest](http://docs.google.com/source-class-REST_Controller.html#192-199)  Contains details about the REST API Fields: db, ignore\_limits, key, level, user\_id Note: This is a dynamic object (stdClass)  Contains details about the REST API Fields: db, ignore\_limits, key, level, user\_id Note: This is a dynamic object (stdClass) | [#](#2y3w247) NULL |
| protected array | [$\_get\_args](http://docs.google.com/source-class-REST_Controller.html#201-206)  The arguments for the GET request method  The arguments for the GET request method | [#](#1d96cc0) [] |
| protected array | [$\_post\_args](http://docs.google.com/source-class-REST_Controller.html#208-213)  The arguments for the POST request method  The arguments for the POST request method | [#](#3x8tuzt) [] |
| protected array | [$\_put\_args](http://docs.google.com/source-class-REST_Controller.html#215-220)  The arguments for the PUT request method  The arguments for the PUT request method | [#](#2ce457m) [] |
| protected array | [$\_delete\_args](http://docs.google.com/source-class-REST_Controller.html#222-227)  The arguments for the DELETE request method  The arguments for the DELETE request method | [#](#rjefff) [] |
| protected array | [$\_patch\_args](http://docs.google.com/source-class-REST_Controller.html#229-234)  The arguments for the PATCH request method  The arguments for the PATCH request method | [#](#3bj1y38) [] |
| protected array | [$\_head\_args](http://docs.google.com/source-class-REST_Controller.html#236-241)  The arguments for the HEAD request method  The arguments for the HEAD request method | [#](#1qoc8b1) [] |
| protected array | [$\_options\_args](http://docs.google.com/source-class-REST_Controller.html#243-248)  The arguments for the OPTIONS request method  The arguments for the OPTIONS request method | [#](#4anzqyu) [] |
| protected array | [$\_query\_args](http://docs.google.com/source-class-REST_Controller.html#250-255)  The arguments for the query parameters  The arguments for the query parameters | [#](#2pta16n) [] |
| protected array | [$\_args](http://docs.google.com/source-class-REST_Controller.html#257-262)  The arguments from GET, POST, PUT, DELETE, PATCH, HEAD and OPTIONS request methods combined  The arguments from GET, POST, PUT, DELETE, PATCH, HEAD and OPTIONS request methods combined | [#](#14ykbeg) [] |
| protected string | [$\_insert\_id](http://docs.google.com/source-class-REST_Controller.html#264-269)  The insert\_id of the log entry (if we have one)  The insert\_id of the log entry (if we have one) | [#](#3oy7u29) '' |
| protected boolean | [$\_allow](http://docs.google.com/source-class-REST_Controller.html#271-276)  If the request is allowed based on the API key provided  If the request is allowed based on the API key provided | [#](#243i4a2) TRUE |
| protected string | [$\_user\_ldap\_dn](http://docs.google.com/source-class-REST_Controller.html#278-283)  The LDAP Distinguished Name of the User post authentication  The LDAP Distinguished Name of the User post authentication | [#](#j8sehv) '' |
| protected string | [$\_start\_rtime](http://docs.google.com/source-class-REST_Controller.html#285-290)  The start of the response time from the server  The start of the response time from the server | [#](#338fx5o) '' |
| protected string | [$\_end\_rtime](http://docs.google.com/source-class-REST_Controller.html#292-297)  The end of the response time from the server  The end of the response time from the server | [#](#1idq7dh) '' |
| protected array | [$\_supported\_formats](http://docs.google.com/source-class-REST_Controller.html#299-313)  List all supported methods, the first will be the default format  List all supported methods, the first will be the default format | [#](#42ddq1a) [ 'json' => 'application/json', 'array' => 'application/json', 'csv' => 'application/csv', 'html' => 'text/html', 'jsonp' => 'application/javascript', 'php' => 'text/plain', 'serialized' => 'application/vnd.php.serialized', 'xml' => 'application/xml' ] |
| protected object | [$\_apiuser](http://docs.google.com/source-class-REST_Controller.html#315-320)  Information about the current API user  Information about the current API user | [#](#2hio093) |
| protected boolean | [$check\_cors](http://docs.google.com/source-class-REST_Controller.html#322-327)  Whether or not to perform a CORS check and apply CORS headers to the request  Whether or not to perform a CORS check and apply CORS headers to the request | [#](#wnyagw) NULL |
| protected boolean | [$\_enable\_xss](http://docs.google.com/source-class-REST_Controller.html#329-337)  Enable XSS flag Determines whether the XSS filter is always active when GET, OPTIONS, HEAD, POST, PUT, DELETE and PATCH data is encountered Set automatically based on config setting  Enable XSS flag Determines whether the XSS filter is always active when GET, OPTIONS, HEAD, POST, PUT, DELETE and PATCH data is encountered Set automatically based on config setting | [#](#3gnlt4p) FALSE |
| protected array | [$http\_status\_codes](http://docs.google.com/source-class-REST_Controller.html#339-360)  HTTP status codes and their respective description Note: Only the widely used HTTP status codes are used  HTTP status codes and their respective description Note: Only the widely used HTTP status codes are used Link <http://www.restapitutorial.com/httpstatuscodes.html> | [#](#1vsw3ci) [ self::HTTP\_OK => 'OK', self::HTTP\_CREATED => 'CREATED', self::HTTP\_NO\_CONTENT => 'NO CONTENT', self::HTTP\_NOT\_MODIFIED => 'NOT MODIFIED', self::HTTP\_BAD\_REQUEST => 'BAD REQUEST', self::HTTP\_UNAUTHORIZED => 'UNAUTHORIZED', self::HTTP\_FORBIDDEN => 'FORBIDDEN', self::HTTP\_NOT\_FOUND => 'NOT FOUND', self::HTTP\_METHOD\_NOT\_ALLOWED => 'METHOD NOT ALLOWED', self::HTTP\_NOT\_ACCEPTABLE => 'NOT ACCEPTABLE', self::HTTP\_CONFLICT => 'CONFLICT', self::HTTP\_INTERNAL\_SERVER\_ERROR => 'INTERNAL SERVER ERROR', self::HTTP\_NOT\_IMPLEMENTED => 'NOT IMPLEMENTED' ] |

API documentation generated by [ApiGen](http://apigen.org)