

Perl Dancer 1.X cheat sheet v1.0

		ROUTES			ACCESSORS/SETTERS
any '/route' => sub {}	Defines a rou	ute for multiple HTTP methods at once		ret type	
get '/route' => sub {}	Defines a rou	ute for HTTP GET requests to the given path	captures	hashref	Returns a reference to a copy of %+, if there are named captures in the route Regexp
patch '/route' => sub {}	Defines a rou	ute for HTTP PATCH requests to the given URL	cookie 'name' => 'value'	scalar/none	Get/set cookie value
post '/route' => sub {}	Defines a rou	ute for HTTP POST requests to the given URL	cookies	hashref	Accesses to cookies keys/values
del '/route' => sub {}	Defines a rou	ute for HTTP DELETE requests to the given URL	config	hashref	Accesses the configuration of the application
options '/route' => sub {}	Defines a rou	ute for HTTP OPTIONS requests to the given URL	content-type 'content/type'	none	Sets the content-type rendered, for the current route handler
put '/route' => sub {}	Defines a rou	ute for HTTP PUT requests to the given URL	engine 'name'	objectref	Given a namespace, returns the current engine object
•		HOOKS/FILTERS	headers 'name' => 'value',	none	Adds custom headers to response
after	Add a hook a	at the after position	header 'name => 'value'	none	Adds custom header to response
before sub {}	Defines a bet	fore filter	push_header 'name' => 'value'	none	Do the same as header, but allow for multiple headers with the same name
before_template sub {}	Defines a bet	fore_template filter	mime	objectref	Shortcut to access the instance object of Dancer::MIME
hook 'name' => sub {}	Adds a hook	at some position	params	hashref	Returns an hash reference to all defined parameters
before_deserializer	This hook red	ceives no arguments	param 'name'	scalar	This method is an accessor to the parameters hash table
before_file_render	This hook red	ceives as argument the path of the file to render	prefix '/route"	none	Defines a prefix for each route handler
before error init	This hook red	ceives as argument a Dancer::Error object	request	objectref	Returns a Dancer::Request object representing the current request
before_error_render		ceives as argument a Dancer::Error object	set 'name' => 'value',	none	Defines a configuration setting(s)
before	This is an alia	as to before . This hook receives no arguments	setting 'name'	any	Returns the value of a given setting
before_template_render	This is an alia	as to before_template .This hook receives as argument a HashRef, containing lat will be passed to the template	set_cookie 'name' => 'value', %opts	none	Creates or updates cookie values
before_layout_render	This hook red	ceives two arguments. The first one is a HashRef containing the tokens. The ScalarRef representing the content of the template	session 'name' => 'value'	hashref/none	Provides access to all data stored in the user's session (if any). It can also be used as a setter to store data in the session
before_serializer		ceives as argument a Dancer::Response object	splat	array	Returns the list of captures made from a route handler with a route pattern which includes wildcards
after deserializer	This hook red	ceives no arguments	upload 'input'	objectref	Provides access to file uploads
after file render		ceives as argument a Dancer::Response object	uri for '/route'	scalar	Returns a fully-qualified URI for the given path
after_template_render	This hook red template	ceives as argument a ScalarRef representing the content generated by the	var 'name' => 'value'	any/none	Provides an accessor for variables shared between filters and route handlers. Given a key/value pair, it sets a variable
after_layout_render	This hook receives as argument a ScalarRef representing the content generated by the layout		vars	hashref	Returns the HashRef of all shared variables set during the filter/route chain with the var keyword
after	This is an alias for after. This hook runs after a request has been processed, but before the response is sent. It receives a Dancer::Response object				ACTIONS
after error render	This hook receives as argument a Dancer::Response object		dance	Alias for the s	start keyword
on_handler_exception	This hook is called when an exception has been caught, at the handler level, just before creating and rendering Dancer::Error. This hook receives as argument a Dancer::Exception object		debug 'message'	Logs a message of debug level	
on_route_exception	This hook is called when an exception has been caught, at the route level, just before rethrowing it higher. This hook receives the exception as argument. It can be a Dancer::Exception, or a string, or whatever was used to die		error 'message'	Logs a message of error level	
	UTILS		forward '/route' => {%params} => {method => 'METHOD'}	Runs an internal redirect of the current request to another request	
	ret type		halt 'message'	Sets a response object with the content given	
dirname 'path'	scalar	scalar Returns the dirname of the path given load 'script		Loads one or more perl scripts in the current application's namespace. Syntactic sugar around Perl's require	
path 'path', 'path',	scalar	Concatenates multiple paths together, without worrying about the underlying operating system	load_app 'app_name'	Loads a Dancer package. This method sets the libdir to the current ./lib directory	
false	scalar	Constant that returns a false value (0)	pass	Tells Dancer	to pass the processing of the request to the next matching route
true	scalar	Constant that returns a true value (1)	redirect '/route' => CODE	Generates a l	HTTP redirect (302)
from_dumper 'struct'	hashref	Deserializes a Data::Dumper structure	send_error 'message' => CODE	Returns a HT	TP error. By default the HTTP code returned is 500
from_json 'struct' => {%opts}	hashref	Deserializes a JSON structure	send_file '/path' => %opts	Lets the curre	ent route handler send a file to the client
from_yaml 'struct'	hashref	Deserializes a YAML structure	start	Starts the app	olication or the standalone server (depending on the deployment choices)
from_xml 'struct'	hashref	Deserializes a XML structure	status 'status'	Changes the status code provided by an action. By default, an action will produce an HTTP 200 OK status code, meaning everything is OK	
to_dumper \$struct	scalar	Serializes a structure with Data::Dumper	template 'name' {%tokens}, {%opts}	Returns the response of processing the given template with the given parameters (and optional settings), wrapping it in the default or specified layout too, if layouts are in use	
to_json \$struct => {%opts}	scalar	Serializes a structure to JSON	warning 'message'	Logs a warning message through the current logger engine	
to_yaml \$struct	scalar	Serializes a structure to YAML			
to_xml \$struct => {%opts}	scalar	Serializes a structure to XML			