Link : [http://framework.zend.com/manual/2.0/en/modules/zend.mvc.controllers.html#the-abstractrestfulcontroller](http://framework.zend.com/manual/2.0/en/modules/zend.mvc.controllers.html%23the-abstractrestfulcontroller)

**The AbstractRestfulController**[**¶**](http://framework.zend.com/manual/2.0/en/modules/zend.mvc.controllers.html#the-abstractrestfulcontroller)

The second abstract controller ZF2 provides is Zend\Mvc\Controller\AbstractRestfulController. This controller provides a naive RESTful implementation that simply maps HTTP request methods to controller methods, using the following matrix:

* **GET** maps to either get() or getList(), depending on whether or not an “id” parameter is found in the route matches. If one is, it is passed as an argument to get(); if not, getList() is invoked. In the former case, you should provide a representation of the given entity with that identification; in the latter, you should provide a list of entities.
* **POST** maps to create(). That method expects a $data argument, usually the $\_POST superglobal array. The data should be used to create a new entitiy, and the response should typically be an HTTP 201 response with the Location header indicating the URI of the newly created entity and the response body providing the representation.
* **PUT** maps to update(), and requires that an “id” parameter exists in the route matches; that value is passed as an argument to the method. It should attempt to update the given entity, and, if successful, return either a 200 or 202 response status, as well as the representation of the entity.
* **DELETE** maps to delete(), and requires that an “id” parameter exists in the route matches; that value is passed as an argument to the method. It should attempt to delete the given entity, and, if successful, return either a 200 or 204 response status.

Additionally, you can map “action” methods to the AbstractRestfulController, just as you would in the AbstractActionController; these methods will be suffixed with “Action”, differentiating them from the RESTful methods listed above. This allows you to perform such actions as providing forms used to submit to the various RESTful methods, or to add RPC methods to your RESTful API.