**RELAZIONI TRA ENTITA:**

**But how do you deal with relations?** If a relation can only exist within another resource, RESTful principles provide useful guidance. Let's look at this with an example. A ticket in [Enchant](http://www.enchant.com/)consists of a number of messages. These messages can be logically mapped to the /ticketsendpoint as follows:

* GET /tickets/12/messages - Retrieves list of messages for ticket #12
* GET /tickets/12/messages/5 - Retrieves message #5 for ticket #12
* POST /tickets/12/messages - Creates a new message in ticket #12
* PUT /tickets/12/messages/5 - Updates message #5 for ticket #12
* PATCH /tickets/12/messages/5 - Partially updates message #5 for ticket #12
* DELETE /tickets/12/messages/5 - Deletes message #5 for ticket #12

Alternatively, if a relation can exist independently of the resource, it makes sense to just include an identifier for it within the output representation of the resource. The API consumer would then have to hit the relation's endpoint. However, if the relation is commonly requested alongside the resource, the API could offer functionality to automatically embed the relation's representation and avoid the second hit to the API.