

## Routes

```
GET /items.json      - index of all items
    ?sort_by=:id&order=ASC - sorted index by value name belonging to some property id
    ?order=ASC        - sorted index of all items by ID
    ?offset=:datetime - index of all items after a certain time
GET /items/:id.json  - show a single item by :id
GET /items/new.json  - new (don't add to db yet)
POST /items.json     - create
PUT /items/:id.json  - update
DELETE /items/:id.json - destroy an item

GET /items/search.json - key:value pair and keyword search
    sort_by=:id&order=ASC - sort results by value name belonging to some property id
    order=ASC            - sort results by item ID
    value[0]=...&value[1]=... - search with an array of keywords
    property_id[0]=...&property_id[1]=... - search within an array of properties by property id
    JSON Request:        - Key:value pair search
{"pair"=>["0" => {"property_id"=>"3", "value"=>"Manuscript"},
"1" => {"property_id"=>"1", "value"=>"Thomas Edison"}]
}

GET /properties.json - index of all property names with ID
GET /properties/:id.json - show all values for a specific property in DB

GET /values/filters.json - List of all potential key:value pair filters in the database
GET /values/filters.json (has the same set of search parameters as /items/search.json)
GET /values.json         - index of all value names with ID
```

## Example call to edit an item

```
{
  "item": {
    "values_attributes": {
      "0": {
        "name": "Some book or something",
        "property_attributes": {
          "name": "Type",
          "id": "3"
        }
      }
    }
  }
}
```