

Services API

Important note: All the services should be created by using JSON as the response format.

General Response Format

All the requests will follow a simple basic format for the errors and another one for the success.

In the case where there was an error from the server side, the return will be:

```
{"success":false,"error_code": XXX, "error_msg":"ERROR_MESSAGE"}
```

Then, if everything is working fine the return will be one of two possible results, the following is presented in the case where the service will return just one element (find by id).

```
{"success":true,"MODEL_NAME":{"attr_name1":"attr_value1", ...}}
```

Finally, the following element will present the object that will be returned in the case where the request is for getting a list of elements

```
{"success":true,"PLURAL_MODEL_NAME":[{"attr_name1":"attr_value1", ...}, {...}, {...}, ...]}
```

Service Error Responses

All the responses for the errors have the following format.

```
{"success":false,"error_code": XXX, "error_msg":"ERROR_MESSAGE"}
```

Also, the list of all the error codes and messages are:

400 - Bad request

401 – Not authorized

404 - Record not found

500 - Server Error

General User/Password validation

As we may want to use this API live, we may need to create a simple “HTTP Basic Authentication” (http://en.wikipedia.org/wiki/Basic_access_authentication) in order to prevent unwanted visitors to our API.

So, to reduce the effort required in this topic (security), we may only validate that the username is “my_user” and the password is “my_password” on each API Call (see the details of the implementation of HTTP_BASIC_AUTHENTICATION in the Wikipedia URL or on Grape documentation), and you should “hard code” this information to avoid the implementation of a table with the credentials for all the users (do it as simple as possible).

Services description

List all stores

Request Method	GET
Path	/services/stores
Parameters	None
Sample request	Load all the stores that are stored in the Database.
Response	<pre>{ "stores": [{ "id": 1, "address": "Somewhere over the rainbow", "name": "Super Store" }, { ... }], "success": true, "total_elements": 2 }</pre>
Errors	None

List all the articles

Request Method	GET
Path	/services/articles
Parameters	None
Sample request	Load all the articles that are in the Database.
Response	<pre>{ "articles": [{ "id": 1, "description": "The best quality of shoes in a green color", "name": "green shoes", "price": 20.15, "total_in_shelf": 25, "total_in_vault": 40, "store_name": "Super Store" }, { ... }], "success": true, "total_elements": 2 }</pre>
Errors	None

List all the articles that are in a specific store

Request Method	GET
Path	/services/articles/stores/:id
Parameters	id: a numeric value of the ID of the store
Sample request	Load all the articles from a specific store that are in the Database.
Response	<pre>{ "articles": [{ "id": 1, "description": "The best quality of shoes in a green color", "name": "green shoes", "price": 20.15, "total_in_shelf": 25, "total_in_vault": 40, "store_name": "Super Store" }, { ... }], "success": true, "total_elements": 2 }</pre>
Errors	<p>Wrong parameters (id is not a number) {"error_msg": "Bad Request", "error_code": 400, "success": false}</p> <p>No store with that ID {"error_msg": "Record not Found", "error_code": 404, "success": false}</p>