API Wrapper

Api wrapper for magento2 REST api written with Express.js

This project assumes you had already installed these tools:

1. docker
2. docker-compose
3. nvm
4. pm2
5. postman

Used Ports

1. **3000** for express.js
2. **9200** for elasticsearch
3. **5601** for kibana

Installation

- Installation of node, npm, pm2, elasticsearch and kibana by using the commands below:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.2/install.sh | bash
```

- Now add these lines to your ~/.bashrc, ~/.profile, or ~/.zshrc file to have it automatically sourced upon login: (you may have to add to more than one of the above files)

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads
nvm bash_completion
```

- After *nvm* successfully installed, use the commands below:

```
nvm install 10.15.3
nvm use 10.15.3
npm install pm2@4.2.3 -g
cd {path-to-api-wrapper}
docker-compose up -d
npm install
```

- You should wait **approximately 10** seconds for elasticsearch and kibana to be alive. After you are certain, you can use `update-elasticsearch.js` for populating the elasticsearch.

```
cd {path-to-api-wrapper}
node update-elasticsearch.js
```

Sample Usage of API

- You can start the API by using the command below:

```
cd {path-to-api-wrapper}
pm2 start ecosystem.config.js
```

- You can watch your cluster by using the commands below:

```
pm2 monit
pm2 list
```

- You can find the example `POSTMAN` collection as *api-wrapper.postman_collection.json* on the directory. You can import the file and try each request on `POSTMAN`.

Read and Bulk Insert

- You can use `bulk-customer-insert.js` for reading csv and bulk inserting customers. Sample usage:

```
cd {path-to-api-wrapper}
chmod +x bulk-customer-insert.js
./bulk-customer-insert.js --file=sample.csv --url=http://localhost:3000
```

Localization

- You can add *en*, *tr*, *ar* language files as a **JSON** file under the *lang* directory, api will support the localization for these languages.
- Sample *tr* file:

```
{
  "United Arab Emirates": "Birlesik Arap Emirlikleri",
  "Afghanistan": "Afganistan",
  "Albania": "Arnavutluk",
  "Andorra": "Andora",
  "Default Category": "Standart Kategori",
  "What's New": "Yeni Ne Var",
  "Jackets": "Ceketler",
  "Email is not valid.": "Email gecerli degil.",
  "Firstname is required.": "Firstname gerekli.",
  "Lastname is required.": "Lastname gerekli.",
  "Password min length is 5 and should be alphanumeric.": "Password en az 5
karakter ve alfanumerik olmali.",
  "Internal Server Error": "Icsel Sunucu Hatasi.",
  "A customer with the same email address already exists in an associated
website.": "Ayni emaili olan kayitli bir kullanici zaten var.",
  "Bad Request": "Gecersiz istek."
}
```

Customer Password Constraints

- Customer password should be greater than **5** and the characters in the password should be **Alphanumeric**.
- Sample Customer Data:

```
{
  "email": "johndoe@gmail.com",
  "firstname": "John",
  "lastname": "Doe",
  "password": "asd123ASD"
}
```