Disheng Lin (contact: 0406423125)

This APP has been developed by Symfony 6.1 & PHP8.1.4 in the docker environment which mean you can run it by using docker-compose. It needs ports 8000, 8001 and 1025 to open for running the containers. To access the Mailcatcher console, please type <http://localhost:8001> after the containers are all up and running.

Graphical user interface, application, Word

Description automatically generated

A screenshot of a computer

Description automatically generated

1, Please go to its root folder and type the following command:

**docker-compose up -d**

A picture containing text, indoor, screenshot

Description automatically generated

2, After the images has been built and the container is up and running, please access the container by typing the following command:

**docker-compose exec output\_file bash**

Shape

Description automatically generated with medium confidence

3, Install the vendor by typing the following command:

**php -d memory\_limit=-1 /usr/local/bin/composer install -vvv**

Text

Description automatically generated

4, Once the installation complete and then can use the command line for testing. Here, we can use the MailCatcher for viewing email and its attachments. Type the following command and it will generate one big csv file in file folder and send it as the attachment to [dixon@abc.com](mailto:dixon@abc.com). The first parameter is the type of the output as csv, xml etc. The second parameter is the recipient email address. The third parameter is the threshold number to split the one big csv or xml file into multiple files according to the number records it read from the json.

**php bin/console app:output-file csv** [**dixon@abc.com**](mailto:dixon@abc.com)

A picture containing graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application, Word

Description automatically generated

5, Type the following command and it will generate multiple csv files and send it as the attachments to [dixon@abc.com](mailto:dixon@abc.com) (the attachment size in email might be limited)

**php bin/console app:output-file csv** [**dixon@abc.com**](mailto:dixon@abc.com) **20**

Text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

6, Type the following command and it will generate one big XML file in file folder and send it as the attachment to [dixon@abc.com](mailto:dixon@abc.com).

**php bin/console app:output-file xml** [**dixon@abc.com**](mailto:dixon@abc.com)

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

7, Type the following command and it will generate multiple XML files in file folder and send it as the attachments to [dixon@abc.com](mailto:dixon@abc.com).

**php bin/console app:output-file xml** [**dixon@abc.com**](mailto:dixon@abc.com) **20**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

8, The App apply for modified strategy design patterns and follow the OC principle for the output files and validators for each line of data read from the Json file. That is to add a new output format or validator is extremely simple. For adding a new output format, just need to implement OutputMethodInterface. For adding a new validator, just need to implement ValidatorInterface without the any need to change the other code.

9, Type the following command and it will execute the phpunit test example of this app.

**php bin/** **phpunit**

A picture containing text, screenshot, close

Description automatically generated

Notice: If needed, I can also share the source code in Git hub.