

# Web Crawler Project - Task Documentation

Author: **Bahy Helmi Hartoyo Putra**  
(bahyhelmi97@gmail.com)

## Generate Training Data

1. Prepare .csv that contain training data (websites), format the .csv as shown in this [example](#) dataset:

	merchant_name	website	label
0	CV.ASIATRIPINDONESIA	http://asiatrip.id/	APPROVED
1	Pt.BloomingLotusYoga	https://www.blooming-lotus-yoga.com	APPROVED
2	YayasanGreenSejahtera	https://greensejahterafoundation.com/	APPROVED
3	PTMatairTerraSolusi	http://www.matair.co.id	APPROVED
4	SimplyMii	https://www.simplymii.com/	APPROVED
5	Nyetak.ID	https://www.nyetak.id	APPROVED
6	nonandnik	https://nonandnik.com	APPROVED
7	YoYoMats	https://yoyomatsindonesia.myshopify.com	REJECTED
8	hiendguitar.com	http://hiendguitar.com/	APPROVED
9	Fipper	http://www.fippersandal.co.id	APPROVED

2. Run the *batch processing* API, change the *input\_file* into your file:

► Crawl - Batch

GET http://127.0.0.1:5000/api/v1/crawler-batch?input\_file=datasets/merchants\_cleaned.csv&start\_index=9&end\_...

Params Authorization Headers Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	input_file	datasets/merchants_cleaned.csv	
<input checked="" type="checkbox"/>	start_index	9	
<input checked="" type="checkbox"/>	end_index	12	
<input checked="" type="checkbox"/>	output_file	df_test	
<input checked="" type="checkbox"/>	train	true	

3. Your train data will be available on *datasets/* directory under the *output\_file* name.

## Update Training Data

1. Open [Modelling](#) notebook.
2. Change this line (*df\_cleaned.csv*) to your new generated training data.

```
In [69]: ## Change this to your new training data  
df = pd.read_csv("df_cleaned.csv").iloc[:,1:]  
df = df.drop_duplicates(subset='website')
```

3. Run all the cells. It will automatically dump the new model with your newest training dataset.

## Changing Model

1. Open [Modelling](#) notebook.
2. Change this line to your desired classifier model.

```
In [88]: ## Gaussian Naive Bayes has a good ability to predict REJECTED websites, but it is hard to get a Low FP Rate with this model.  
## While XGBoost provides more reliable model with Low FP rate and enough TP rate  
## Though, it goes back to the business decision which rate is more important  
  
model_choice = GaussianNB()  
model_choice = BernoulliNB()  
# model_choice = MultinomialNB()  
# model_choice = XGBClassifier(**params)
```

3. Run all the cells. It will automatically dump the new model with your newest training dataset.

## Set Timeout

1. Open [base\\_functions.py](#).
2. Ctrl + F “timeout”, set your desired timeout. Save. Reload the API.

## Reference

- [Source Code](#)
- [Beautifulsoup](#)
- [Selenium with Python](#)
- [Model Validation Notebook](#)
- [Hyperparameter Tuning Notebook](#)