

Vue + Django

I am using babel as my transpiler and web pack as my bundler. Although I am using web pack it is possible to use Rollup.js or Browserify to help with bundling. I have not tried it, but please take a look and let me know if you do!

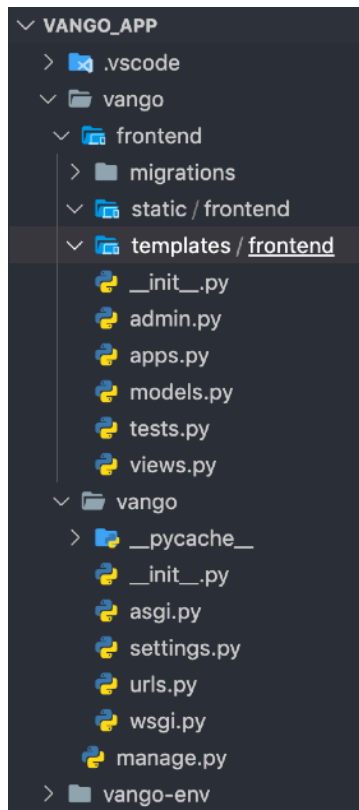
P.S. This is a quick rough draft doc and is my mind thinking way passed my energy lol. I will do my best to update soon.

Setting up Django

- mkdir vango_app && cd vango_app
- python -m venv vango-env
- pip install django djangorestframework
- django-admin startproject vango
- cd vango
- python manage.py startapp frontend
- You can now open the top level 'vango_app' folder
- Go to vango > vango > settings.py
- Under INSTALLED_APPS add the 'frontend' app at the end

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'frontend'  
]
```

- Under vango > frontend create two directories (we will be going back to these later):
 - Option 1 (command line):
 - `mkdir -p {static, templates}/frontend`
 - Option 2 (manually)
 - under frontend create two folders: 'static' and 'templates'
 - Under both of these two new folders, create a 'frontend' folder for each of them

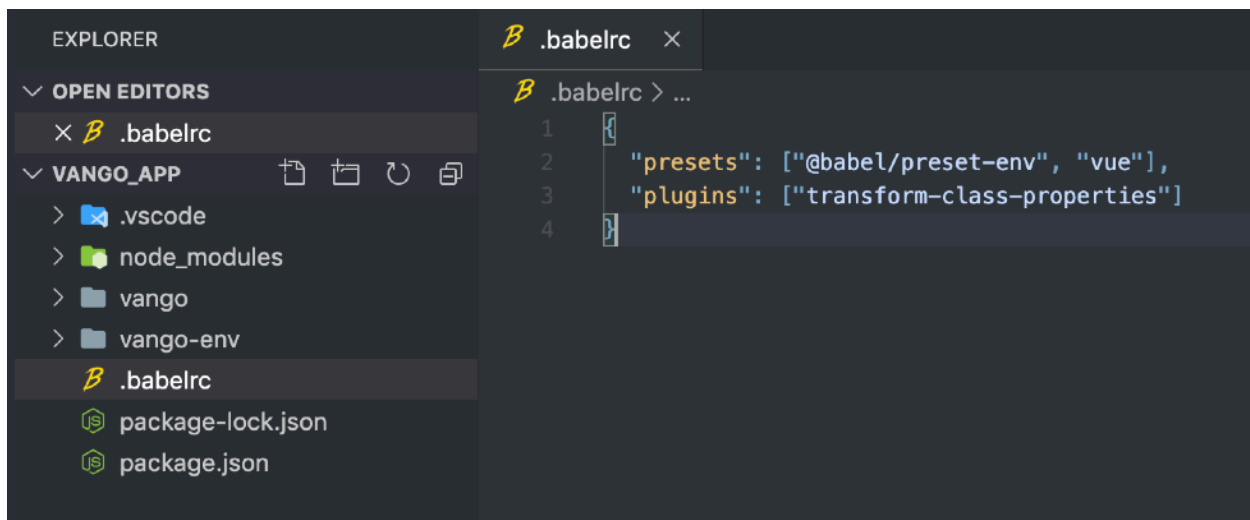


Setting up Vue

- Go to the first directory right under 'vango_app'
- Type 'npm init -y' (if you do not have the npm command you will need to install NodeJS and npm). You should now have a package.json in your main repo
- Go to the command line and cd <base path>/vango_app

Babel

- In the same directory, type "npm install -D @babel/core babel-loader @babel/preset-env babel-preset-vue babel-plugin-transform-class-properties"



- While you are here, create a babel config file by typing touch .babelrc
- Add presets and plugins for babel as shown below

Webpack

- While you are in 'django_app' on the command line, type "npm i -D webpack webpack-cli vue vue-loader css-loader vue-style-loader vue-template-compiler"
- Create a file titled "webpack.config.js" right under "vango_app". This should be at the same level as your "package.json" and ".babelrc" files
- In your webpack.config.js insert the following code (used to allow Vue's template/script/style structure to be bundled correctly; reference: <https://vue-loader.vuejs.org/>)

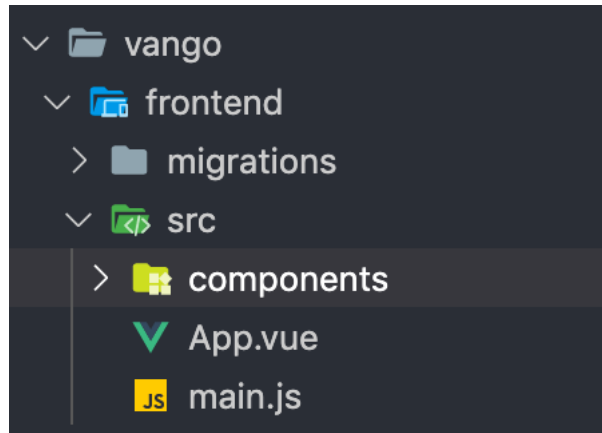
[guide/
#manual-
setup\)](#)

```
webpack.config.js ×
webpack.config.js > <unknown>
1  const VueLoaderPlugin = require('vue-loader/lib/plugin')
2
3  module.exports = {
4    mode: 'development',
5    module: {
6      rules: [
7        {
8          test: /\.vue$/,
9          loader: 'vue-loader'
10         },
11         // this will apply to both plain `.js` files
12         // AND `
```

Monday, March 2, 2020

- Inside src create 'App.vue' file, 'components'

these items:
'main.js' file, and a folder



- We will make this app very simple, so in the main.js file create a Vue instance that will be mounted to our html with the code below.

```
JS main.js ×
vango > frontend > src > JS main.js > ...
1  import Vue from 'vue'
2  import App from './App.vue'
3
4  new Vue({
5    render: h => h(App)
6  }).$mount('#app')
```

- Add simple to the file

some
template
App.vue

```
✓ App.vue ×
vango > frontend > src > ✓ App.vue > {} "App.vue"
1  <template>
2    <div id="app">
3      <p class="hi-world">{{ greeting }}</p>
4    </div>
5  </template>
6
7  <script>
8    export default {
9      data() {
10       greeting: "Look at the vue."
11     }
12   }
13 </script>
14
15 <style>
16   body {
17     font-family: -apple-system, 'Open Sans', sans-serif;
18   }
19   .hi-world {
20     display: flex;
21     background-color: darkslateblue;
22     color: white;
23     border-radius: 5px;
24     padding: 10px;
25   }
26 </style>
```

Putting it all Together / Final Touches

- Now we bundle our app's javascript and output this to a new 'bundle.js' file. To use web pack seamlessly, we will add some scripts to handle the bundling.

Go to your "package.json" file under the vango_app repo and update the "scripts" values as shown below.

```
"scripts": {
  "dev": "webpack --watch --mode development ./vango/frontend/src/main.js --output ./vango/frontend/static/frontend/bundle.js",
  "build": "webpack --watch --mode production ./vango/frontend/src/main.js --output ./vango/frontend/static/frontend/bundle.js"
},
```

```
"dev": "webpack --watch --mode development ./vango/frontend/src/main.js --output ./vango/frontend/static/frontend/bundle.js",
```

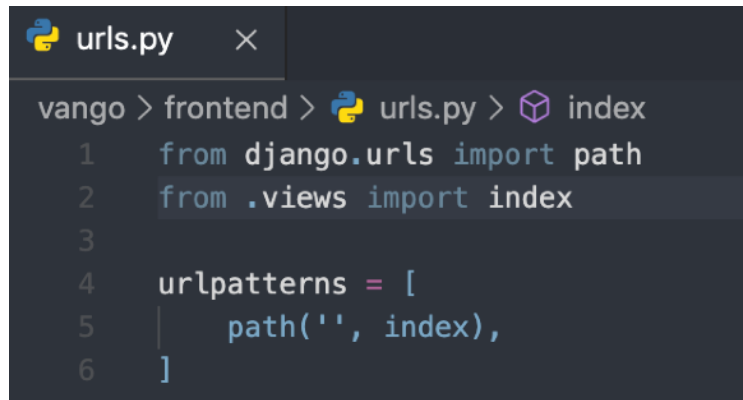
```
"build": "webpack --watch --mode production ./vango/frontend/src/main.js --output ./vango/frontend/static/frontend/bundle.js"
```

- Now in your terminal, go to your vango_app directory. Type in 'npm run dev'. You should see a 'bundle.js' file in your frontend/static/frontend folder.
- Under frontend/templates/frontend create a new 'index.html' file. This simple code allows the bundled javascript to be imported as a static file. In turn, it also imports all

```
index.html ×
vango > frontend > templates > frontend > index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  |   <title>Vango</title>
7  </head>
8  <body>
9  |   <div id="app">
10 |     <!-- Vue code injected here-->
11 |   </div>
12 |
13 |   {% load static %}
14 |   <script src="{% static 'frontend/bundle.js' %}"></script>
15 </body>
16 </html>
```

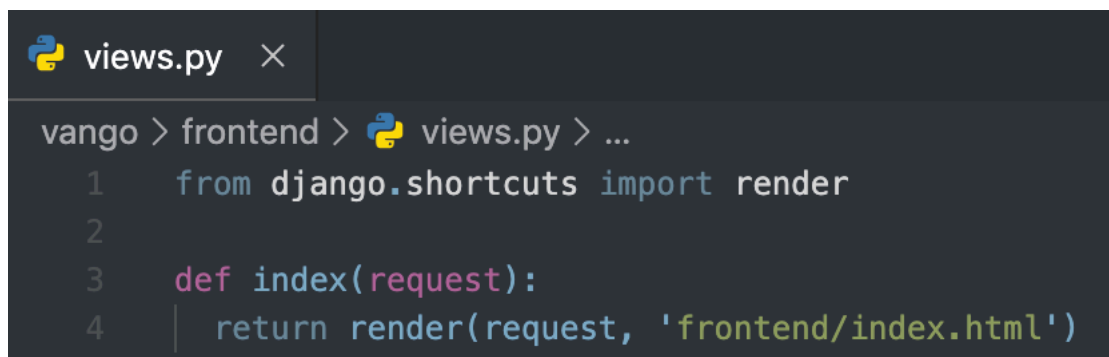
of the app through the '#app' id inside the div due to our App.vue mounting to the given id.

- We're almost done! Now we need to register our index.html to a url in Django so our server can recognize it. In the vango_app > mango > frontend folder create a new file named 'urls.py'. Add the code below.




```
urls.py ×  
vango > frontend > urls.py > index  
1 from django.urls import path  
2 from .views import index  
3  
4 urlpatterns = [  
5     path('', index),  
6 ]
```

- Now we update the views.py file within the same 'frontend' folder.



```
views.py ×  
vango > frontend > views.py > ...  
1 from django.shortcuts import render  
2  
3 def index(request):  
4     return render(request, 'frontend/index.html')
```

- We updated the url paths for specifically for the 'frontend' app. Now we lastly have to update the urls for the parent global urls.py located at vango_app > vango > tango > urls.py


```
urls.py ×  
vango > vango >  urls.py > ...  
1 from django.contrib import admin  
2 from django.urls import path, include  
3  
4 urlpatterns = [  
5     path('admin/', admin.site.urls),  
6     path('', include('frontend.urls'))  
7 ]
```

- Now we run the server by going to vango_app > vango and typing 'python manage.py runserver'. This should show your website at the url localhost:8000. Navigate to this URL in your browser and you should see your app!

Congratulations we made a Django app with a Vue frontend!

