Unit 6 - Django and Vue.js

Classwork: The classwork for this unit should be saved in a new folder: **your_repo/unit_6/**
Homework: Homework should be saved here: **your_repo/homework/unit_6/**

*** Day 01 ***

Django and Vue.js

We learn how to develop a backend application with Django. We learn how to do the frontend like a pro with vuejs 🙂

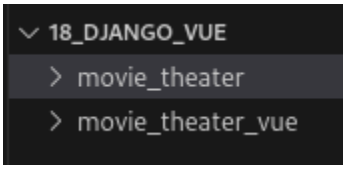
There are more features on the front-end that you can still learn, but you have learned enough to make a beautiful front-end already. If you want, you can learn more on your own, like using a router to have multiple pages in one vue application, using Pinia for state management, …

Well, now let's make them work together. And then you can tell that you are a full-stack developer!!!

Let's see how to make them work together. You know almost everything, we just need to link them together, which is just a matter of configuration (which can be a pain with JS because for me, nodejs is a mess) and then a matter of requests between them.

Configure Django and Vue.js

1. Copy the last django application version you have (authentication lesson). Remember, you need to have communication with the database. If you need to set up part, please set it up.
2. Create a new python env for this lesson. Make sure to have **nodeenv in the dev requirements** (now, we are going to need it as you are expert on front-end).
3. Include a new package on your **main requirements: django_vite**
4. Create a new env for nodejs.
5. Let's create a new vuejs project. So what I like to do is to have a Django folder, and at the same level, the vuejs folder.



6. Check that your vue js app is working.

Insert Vue.js pages into Django templates

1. Let's create a folder apps in the src folder in the vuejs folder. In this apps folder, let's create a movie_edit folder.
2. Let's create a js file movie_edit.js inside the movie_edit folder, that is going to have the vuejs app and mount it in the html (but we still do not know which html, but that does not matter for now, it is going to be mounted somewhere).

```
import 'vite/modulepreload-polyfill';

import { createApp } from 'vue';
import App from './MovieEdit.vue'


createApp(App).mount("#app")
```

3. Let's create the file MovieEdit.vue in the same folder just to have something in the template.

```
<template>
      <div>
      This is the page where we are going to edit movie in vuejs
      </div>
</template>

<script>

export default {
  name: 'App',
  components: {
  },
  data: function() {
      return {

      }},
      methods: {

      },
}
</script>
```

4. Let's rewrite the vite.config.js. This file will define the multiple vuejs app that we can write, it will also define where we want to build our .js file for production.

```
import { defineConfig } from "vite";
import vue from "@vitejs/plugin-vue";

const backendPath = '../movie_theater';

// https://vitejs.dev/config/
export default defineConfig({
 plugins: [vue()],
 base: '/static/vite/',
 server: {
   watch: {
     ignored: [],
   },
 },
 build: {
   manifest: true,
   emptyOutDir: true,
   outDir: backendPath + '/core/static/vite/',
   rollupOptions: {
     input: {
       vue_movie_edit: "./src/apps/movie_edit/movie_edit.js",
     },
   },
 },
});
```

5. What we are going to use right now, it is the rollupOptions, input: (https://vitejs.dev/config/build-options#build-rollupoptions, https://rollupjs.org/configuration-options/#input, https://github.com/MrBin99/django-vite)

   Let's add **'django_vite'** to the **INSTALLED_APPS** list in Django app setting.py

   Also, add the following lines at the end of settings.py:

   ```
   DJANGO_VITE_ASSETS_PATH = os.path.join(BASE_DIR, "core", "static", "vite")
   DJANGO_VITE_DEV_SERVER_PORT = get_secret("vite_dev_server_port")
   DJANGO_VITE_STATIC_URL_PREFIX = "vite/"
   DJANGO_VITE_DEV_MODE = True # This line has to be removed in production
   ```

   In the #Static file section in the same settings.py file add STATIC_ROOT. Otherwise, django_vite is going to complain. This is a parameter only used in production to serve the static files, through a web server like nginx, so that's why you set a directory just for that, and when you ask django to build the static files with the command "python manage.py collectstatic", it will put all the static files in there.

   ```
   STATIC_ROOT = os.path.join(os.path.dirname(BASE_DIR), "movie_theater_static")
   ```

6. Do not forget to set 'vite_dev_server_port' in the secrets.json to 5173 (could be different in your computer, double check). We add this port here in case we want to have the flexibility later to change the vite server port.

   ```
   "vite_dev_server_port": "5176"
   ```

7. Good, let's continue. We are going to work on a Django template now, the one to edit a movie (movie_form.html)

   Right now, we have something like this:

   ```
   {% extends "base.html" %}

   {% block content %}
   <form method="post">{% csrf_token %}
      {{ form.as_p }}
      <input type="submit" value="Save">
   </form>

   {% endblock content %}
   ```

   On top of the template, we are going to add:

   {% load django_vite %}

   And at the end of the template:

   ```
   {% block js %}
      {{ block.super }}

      {% vite_hmr_client %}
      {% vite_asset 'src/apps/movie_edit/movie_edit.js' %}
   {% endblock js %}
   ```

   We should have something like the following code. We are going to keep the original form just in case for now.

```
{% extends "base.html" %}
{% load django_vite %}

{% block content %}
<form method="post">{% csrf_token %}
   {{ form.as_p }}
   <input type="submit" value="Save">
</form>

<br><br>

<div id="app">
   <app></app>
</div>

{% endblock content %}

{% block js %}
  {{ block.super }}

  {% vite_hmr_client %}
  {% vite_asset 'src/apps/movie_edit/movie_edit.js' %}
{% endblock js %}
```

8. At this point, if you refresh the website page on a new movie or update a movie, you should see the part coming from vue js.

   If it is working, you can even change something on the MovieEdit.vue and see the change almost immediately on the html page. (this is the magic of the vite server).

If you get it until there, that is great. We did a lot of configuration and now we are able to load a vuejs app in a Django template. That is great!!!!!

*** Day 02 ***

Pass Django Data into Vue component

First we need to give the csrf token, as we will have to submit information to Django.

Q: Do you remember what a csrf is? When do you need to use it? In a GET request? In a POST request?

So here, as we need to submit a post request with the movie form, the vue js application will need to have this csrf token. We can do the following:

In our **movie_form.html**, in the block js, we can insert a script tag, and then just define our **csrf_token** there. We will configure our vuejs to grab it.

```
{% block js %}
  {{ block.super }}
  <script>
       var ext_csrf_token = '{{ csrf_token }}'
  </script>
  {% vite_hmr_client %}
  {% vite_asset 'src/apps/movie_edit/movie_edit.js' %}
{% endblock js %}
```

Remember, that if you load some JS into your code that is not trusted, people will be able to hack your website, because they will have access to csrf_token from your html page.

In case just check your base.html (in core/templates/) an make sure you have:

```
…
</body>

   {% block js %}
   {% endblock js %}
…
```

Previously, I messed it up by including a <script> before the block js, and the js console was complaining about it. Please move `{% block js %} {% endblock js %}` outside the script tag.

****

Exercise 1:

Search: how you can retrieve this **`ext_csrf_token`** from your vue js app.
Web development is always a quest. You have to search for the information you need and try to find the best information quickly.

Hint: There is something you need to add in the MovieEdit.vue

****

Once you are done with Exercise 1, please verify that vue js app has access to the csrf token by using the vuejs plugin (in firefox or chrome), and checking that the csrf_token has a value.

****

Exercise 2:

Now, find a way to display the same form from django, but from your vue js application.

Hint: You must modify the movie_form.html and the MovieEdit.vue

****

****

Exercise 3:

At some point, it would be great if we recreate the form field by field in Vue.js. But for the moment, let's define the movie object as a dictionary in Vue, as well as the actor list (for the multiple choice).

Let's think about a way to do it and implement it.

Hint: Three files should be modified:

- movies/views.py => class MovieUpdateView(UpdateView) => Add method:
  def get_context_data(self, **kwargs) => In this method define a dictionary for movies and a list for movie actors.
- Movies_form.html => Pass the movies dictionary and actors list
- MovieEdit.vue => Receive the movies dictionary and actors list

****

*** Day 03 ***

Customize your form in Vue

Let's copy the form we have in django to vue, so we can customize it.

```
<template>
 <div>
                This is the page where we are going to edit movie in vuejs, this is cool
                This is the form coming from django, displayed in vue
 </div>
 <div>
                <form method="post" class="form">
                        <input type="hidden" name="csrfmiddlewaretoken" v-bind:value="csrf_token">
                        <p>
                                <label for="id_name">Name:</label>
                                 <input type="text" name="name" value="Matrix" maxlength="100"
required="" id="id_name">
                        </p>
                        <p>
                                <label for="id_running_time">Running time:</label>
                                <input type="text" name="running_time" value="02:18:00" required=""
id="id_running_time">
                        </p>
                        <p>
                                <label for="id_actors">Actors:</label>
                                <select name="actors" required="" id="id_actors" multiple="">
                                        <option value="2">Carrie-Anne Moss</option>
                                        <option value="1" selected="">Keanu Reeves</option>
                                </select>
                        </p>
                        <p>
                                <label for="id_director">Director:</label>
                                <input type="text" name="director" value="The Wachowskis"
maxlength="200" required="" id="id_director">
                        </p>
                        <p>
                                <label for="id_release_date">Release date:</label>
                                <input type="text" name="release_date" value="1999-03-30" required=""
id="id_release_date">
                        </p>
                                <button type="submit" class="btn btn-primary">
                                        Submit
                                </button>
                </form>
  </div>
  <br><br>

</template>
```

Let's use a vue js library for the date field: vue datepicker (https://vue3datepicker.com/).
In package.json, add a dependency: "@vuepic/vue-datepicker": "8.x"

```
"dependencies": {
   "vue": "^3.4.21",
   "@vuepic/vue-datepicker": "8.x"
 },
```

Then, install this new package: **`npm install`**

Now, you should have the new vue js package.

****

Exercise 4:

Read the documentation, and insert the component VueDatePicker to select the date, and when you click on submit, it should work.

Hints:

- In the template, you must insert the new component. Just by adding the component, it does not trigger a new entry in the post request, so you must change the text input where we had the date in string before to a hidden input and get the date in the format expected by Django.
- You must initialize the date variable in data (date: null).
- You might need a computed property to convert date to string.
- You will need to make some changes on the date object, because JS is really a mess concerning the Date object. It is not as easy to manipulate as in Python. There are some libraries to manipulate Date objects better, but for now with 2 lines you can have the result we want. If you need to use Date objects more often on your vue app, it is better to install libraries to format or manipulate the date object way better than the js standard library.

****

Solution:
Also, we put some styling to have the field go next to the label.:

```
<p>
                <label for="id_release_date">Release date:</label>
                <input type="hidden" name="release_date" :value="get_date_string" required=""
id="id_release_date">
```

```
            <VueDatePicker style="display:inline-block;width:
300px;padding-bottom:10px;padding-left:10px" v-model="date"
:enable-time-picker="false"></VueDatePicker>
        </p>
```

We need to initialize the date variable in data:

```
data: function() {
        return {
                ….
                date: null,
                ……
```

And we are going to create the computed property get_date_string:

```
methods: {
        convert_date_to_string(dato){
        const offset = dato.getTimezoneOffset()
        dato = new Date(dato.getTime() - (offset*60*1000))
        console.log('date', dato, dato.toISOString())
        return dato.toISOString().split('T')[0]
        }

        },
        computed: {
        get_date_string() {
        if (this.date == null) {
                return ""
        } else {
                return this.convert_date_to_string(this.date)
        }
        }
        }
```

If that is working, just show the date into the field when we load the page.
Go ahead dear student, show me that you can display the date on the field when we load the page.

Solution:

```
methods: {
        …..
        init_date(date_string){
```

```
        let dato = new Date(date_string)
        const offset = dato.getTimezoneOffset()
        dato = new Date(dato.getTime() + (offset*60*1000))
        return dato
        }
        ….
```

Yes, javascript date is a mess, so annoying.

```
data: function() {
        …..
        date: this.init_date(window.ext_movie_dict.release_date),
        ….
}
```

And that it.

So, now thank to vuejs, we have a beautiful datepicker input.
Let's do the same for the time.
But this is for another day, it is time for dodis now, good night.