

# 1 Add a panel to admin panel / Database

## Introduction

To enable database manipulation / editing without using tools like DB browser and their like. There's an admin panel, that allows editing of DB entries. The following is a guide on how to edit the codebase s.t. it can display a new table

1. Add the database schema by creating a model class in *main\_page/models.py* file.
2. Update the database by calling the django commands `make migration` and `migrate`.
3. Create the class **Edit<model name>Form** corresponding to the edit form and the class **Add<Model name>Form** corresponding to the adding form in the *main\_page/forms.py* file
4. Add an Endpoint by adding the class **<model name>EndPoint** in the *main\_page/views/api/api.py* file.
5. Update *main\_page/url.py* with the endpoint created in step 4
6. In the classes **AdminPanelEditView** and **AdminPanelAddview** found in the *main\_page/views/admin\_panel.py* file, add an entry with to the dictionary **MODEL\_NAME\_MAPPINGS** with the model from step one in both classes. Add an Entry to the dictionary in the: **EDIT\_FORM\_MAPPINGS** / **ADD\_FORM\_MAPPING** in the respective classes with the form you created in step 3.
7. Update the javascript in the files *main\_page/static/main\_page/js/admin\_panel.js*, *main\_page/static/main\_page/js/admin\_panel\_add.js* and *main\_page/static/main\_page/js/admin\_panel\_edit.js*
8. Add the option to the button in the admin

Total list of files that require to be edited

```
main_page/
- forms.py
- models.py
- urls.py
views/:
  * admin_panel.py
  api/
    · api.py
static/main_page/js/
```

- \* admin\_panel.js
- \* admin\_panel\_add.js
- \* admin\_panel\_edit.js

**Author note:** Legends speak of a programmer who claim it's simple to add extra tables. Some even speak of the famous 'GOOD design'. HETETIC I say.