

Aula 04: Configuração do Django Admin

Nesta aula, vamos explorar as diversas possibilidades de configuração do Django Admin para o modelo Aluno. O Django Admin é uma ferramenta poderosa que permite gerenciar os dados da aplicação de forma eficiente através de uma interface web.

Objetivos da Aula

- Entender como personalizar a interface do Django Admin.
- Aprender a configurar diferentes opções para exibir e editar dados no Django Admin.
- Conhecer as funcionalidades de busca e filtro no Django Admin.

Configuração Básica

A configuração básica do Django Admin envolve a criação de uma classe que herda de `admin.ModelAdmin` e a definição de várias opções para personalizar a interface.

```
from django.contrib import admin
from apps.aluno.models import Aluno

class AlunoAdmin(admin.ModelAdmin):
    list_display = ('email', 'nome', 'cpf', 'data_de_cadastro', 'is_active',
                    'is_superuser')
    list_display_links = ('email', 'nome')
    search_fields = ('nome', 'email', 'cpf')
    list_filter = ('is_superuser', 'is_active', 'data_de_cadastro')
    list_editable = ('is_active',)

admin.site.register(Aluno, AlunoAdmin)
```

Opções de Configuração

`list_display`

Define os campos que serão exibidos na lista de registros do modelo.

`list_display_links`

Define quais campos na lista de registros serão clicáveis, levando à página de edição do registro.

`search_fields`

Define os campos que serão pesquisáveis na interface do admin.

`list_filter`

Adiciona opções de filtro na barra lateral da interface do admin.

`list_editable`

Permite que certos campos sejam editáveis diretamente na lista de registros.

Conclusão

Com essas configurações, você pode personalizar a interface do Django Admin para melhor atender às necessidades da sua aplicação. Experimente ajustar essas opções e veja como elas afetam a usabilidade do admin.

Exercícios

1. Adicione novos campos ao `list_display` e veja como eles aparecem na interface do admin.
2. Experimente adicionar novos filtros usando `list_filter`.
3. Teste a funcionalidade de busca adicionando diferentes campos em `search_fields`.

Próximos Passos

Na próxima aula, vamos explorar como adicionar ações personalizadas e como estender ainda mais a funcionalidade do Django Admin.