

Get Django Form Data

- Validate Data/ Field Validation
- Get Cleaned Data

How to send GET Request

- Open browser and write url hit enter this is by default get request
- Anchor Tag
- Form tag contains method='GET'
- Form tag with specifying method is by default GET

How to send POST Request

- form tag contains method='POST'

Django Form and Field Validation

`is_valid()` - This method is used to run validation and return a Boolean designating whether the data was valid as True or not as False. This returns True or False.

Syntax:- `Form.is_valid()`

`cleaned_data` – This attribute is used to access clean data. Each field in a Form class is responsible not only for validating data, but also for “cleaning” it normalizing it to a consistent format. This is a nice feature, because it allows data for a particular field to be input in a variety of ways, always resulting in consistent output. Once you’ve created a Form instance with a set of data and validated it, you can access the clean data via its `cleaned_data` attribute.

Any text based field such as `CharField` or `EmailField` always cleans the input into a string.

If your data does not validate, the `cleaned_data` dictionary contains only the valid fields.

`cleaned_data` will always only contain a key for fields defined in the Form, even if you pass extra data when you define the Form.

When the Form is valid, `cleaned_data` will include a key and value for all its fields, even if the data didn’t include a value for some optional fields.

Get Django Form Data in Terminal

1. First of all Create form inside forms.py file

```
forms.py
from django import forms
class StudentRegistration(forms.Form):
    name=forms.CharField()
    email=forms.EmailField()
```

2. Get submitted Data in views.py

```
views.py
from .forms import StudentRegistration
def showformdata(request):
    if request.method=='POST':
        fm = StudentRegistration(request.POST)
        if fm.is_valid():
            print('Form Validated')
            print('Name:', fm.cleaned_data['name'])
            print('email:', fm.cleaned_data['email'])
        else:
            fm = StudentRegistration()
    return render(request, 'enroll/userregistration.html', {'form':fm})
```

3. Get object from views.py in template file

```
templates/enroll / userregistration.html
<!DOCTYPE html>
<html>
    <body>
        <form method="POST"> {% csrf_token%}
            {{form.as_p}}
            <input type="submit" value="Submit">
        </form>
    </body>
</html>
```