Formset & Form Improvements

Parth Patil

ABSTRACT

- ### OVERVIEW OF PROBLEM

Django currently has a formset class which is used as a collection instance for the forms. But the main problem with it is that passing the request object to individual form in the formset is difficult(see this), due to the nature for the BaseFormSet & ModelFormSet classes. Also, I'm planning to add some new features like pagination to inline formset and also solving some issues like #13060 (making Management Form exceptions more descriptive) and #10403

- ### GOALS

To add request field to the BaseFormsetClass, and then add few functions to handle the request field everywhere through the formsets. Also, revamp the architecture of the formset such that it will be easy for the user to have more control over the saving process and which arguments are passed to the individual forms in the formset.

After that, the next goal will add some new features to formset like the one mentioned above. And also documenting everything along the way.

- **BENEFITS**

  Apart from obvious benefits like closing the tickets, improving the formset architecture will ease the users in handling the many forms at once (e.g. Blog sites). The main limiting factor now for people not using the formset for the blog site is that linking a user with form in formset is difficult. This project will clearly help those.

The Important Milestones

- **Drafting a full specification of the Formsets**

  The first important task according to me is getting the full specifications of the new formset class. Along with any changes that, would be required in any other places like the ModelForm or BaseForm class. Brainstorming and getting this verified from the community will be the first task according to me.

  Since the complexity of the project is low according to the Django GSoC page. Addressing a few more issues that fall under formsets will be a wise thing to do. Devising a plan of action for this issue will be done in the beginning so that adding them afterward in the new framework won't be much difficult. Some issue that can be addressed 1, 2(#28405), 3(#26607). This will be the secondary issues I'm planning to address

- **Testing and documenting the already available solutions**

  As suggest by Carlton Gibson (a contributor at Django) in this post going through the existing workaround and testing them would be a good thing. This will also help to document different approach and test them for any flaws.

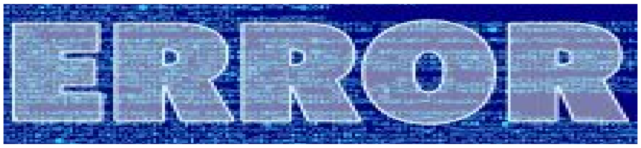
- **Adding extra Features**

There are many features that can be added to the formset

1. **Pagination for inline forms**

I'm thinking of adding an option for the user to toggle between the paginated forms or normal tabular forms. Seems trivial but will require some effort.

2. **Making management form errors more helpful**



Nothing is more frustrating than non-descriptive error. This is no simple task!!! :)

I would try to make some sense out some errors that occur in management formset. (refer #13060)

3. **Declarative syntax support**

Supporting this will be a little difficult compared to the previous tasks. For this, I will need to change the basic way of how BaseFormSet works. I would be thinking whether to try fixing this or not in initial phase of the project (since more information needs to be collected on this). Still, I’m mentioning this as it is present at the back of my mind, so if i find a good way this could be implemented.

- **Testing**

  “Only the best programmers write tests for their code”

  Writing the tests for the code is crucial, even though it won’t be a separate milestone rather a default milestone attached with everything else.

Timeline

My summer vacation start from May 6, 2019, during which I plan to spend around 35 - 40 hours a week. My third academic year starts from July 29, after which I might have to cut it down to around 3-4 hours on weekdays and the regular 5-7 hours on weekends. This period is a little shorter, however, I can also work during the semester until the official end of GSoC. Hence I intend to do most of GSoC during the period 6th May to the 30th July instead of the usual periodof 6th May to 19th August. I have tried to structure the timeline in a way that there are fewer weeks purely for coding or only for learning, to ensure a good mix of both. I intend to write code daily in a week. 1-2 days of every week will be designated for writing documentation and compiling all resources ,adding significant changes I’ve made. I also plan to write a blog post every week to track my GSoC progress, containing links to my contributions, a brief overview, and the plan for next week. This will ensure that these are not lost in the future.

I may be taking a week off in June, therefore I’m keeping a buffer week in June to ensure it won’t affect my GSoC.

- **Pre-Summer Most of April (Application Review Period):**
  - Solve a few bugs related forms (easy pickings may be) to get a little more involved in forms.
  - Getting to know people from community.
  - Getting myself familiarize with the conventions.
- **May 6 - May 10**
  - Discuss the Implementation of the project in detail with mentor and senior contributors from the community
  - Understand whether my approach is correct and figure out a good way to approach the problem.
- **May 11 - May 19**
  - Testing already available solution to the problem and documenting them (2 - 3 days)
  - Formalizing on the new architecture for the BaseFormSet class
- **May 20 - May 27**
  - Actually implementing in code
  - Debugging !! Testing!!
- **May 28 - May 30**
  - Documenting everything completed everything until now.
  - Also some buffer time if anything goes wrong
- **May 31 - June 7**
  - Closing the above mentioned issues (1, 2(#28405), 3(#26607))
  - Writing any extra patches that may be required
- **June 7 - June 12**
  - Starting with planning for (#10403)
  - Understanding the problem first
  - Coming up with a plan of action for this
- **June 13 - June 20**
  - **BUFFER WEEK**
  - I’m going for a vacation during this period :)

- **June 21 - June 25**
    - Implementation of the plan for (#10403)
    - Debugging!! Testing!!
- **June 26 - July 1**
    - Implementation of the pagination feature
    - Keeping it simple!1
- **July 2 - July 9**
    - Looking into the management form exception issue
    - May be adding some features to management form
- **July 9 - July 16**
    - Completing my documentation and testing of code
- **July 17 - July 30**
    - Addressing few tickets related to the forms and formsets

ABOUT ME

I'm **Parth Patil** pursuing my B.Tech. in Electrical Engineering at the Indian Institute of Technology,Bombay. I am in my second year. I was introduced to programming right in my childhood, and I developed a passion for it since then. I have always enjoyed Mathematics solving real world problems. I started coding from a very young age, I built my first app in 8th grade and my first custom ROM for my android app in 9th grade. I'm currently a part of software sub-division Team AUV-IITB. Our team is runner up winner in 2016's Robosub Competition which sees participation from team all across the world. I'm also active in many Tech club in institute and mostly fit in well with the community.