

# Question - Create & Update View

---

In the earlier section we created a question from admin site, now let's create functionality so that a user can create and update a question from frontend.

## Forms

We will create forms to handle our create and update views, the same forms will be used to pass to our views and get data from POST requests.

```
# Create Question
class QuestionCreationForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['title', 'body', 'status']

# Update Question
class QuestionUpdateForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['title', 'body', 'status']
```

## Create View

Import the forms into the views file.

```
# Question Create
@login_required
def question_create(request):
    if request.method == 'POST':
        # Pass data to form
        form = QuestionCreationForm(request.POST)
        if form.is_valid():
            # Create an instance but do not save yet!
            instance = form.save(commit=False)
            # Assign question user to the current user
            instance.author = request.user
            # Save and commit to database
            instance.save()
            # Send a success message
            messages.success(request, 'Question created successfully!')
            # Redirect to the question detail view
            return redirect(reverse('question_detail', args=
(instance.id,)))
        else:
            # Send a blank form
```

```
        form = QuestionCreationForm()
    context = {
        'form': form
    }
    return render(request, 'djora/question_create.html', context)
```

## URL's & Templates

Update all the URL's and base.html template.

## Update View

Update view is similar to create view.

```
# Question Update
@login_required
def question_update(request, pk):
    # Check for valid question
    question = get_object_or_404(Question, pk=pk)
    # Check for valid user
    if question.author == request.user:
        if request.method == 'POST':
            # Pass data to form with instance as the requested question
            form = QuestionUpdateForm(request.POST, instance=question)
            if form.is_valid():
                # Save and commit to database
                instance = form.save()
                # Send success message
                messages.success(request, 'Question updated successfully!')
                # Redirect to the detail view of updated question
                return redirect(reverse('question_detail', args=(instance.id,)))
            else:
                # Pre-fill the form with existing data
                form = QuestionUpdateForm(instance=question)
        else:
            # If current user is not author then raise an error
            raise PermissionDenied
    context = {
        'form': form
    }
    return render(request, 'djora/question_update.html', context)
```

## URL's & Templates

Update all the URL's and question\_detail & profile templates.