

Flask Notes Application – Bug Analysis and Resolution

This document explains the bugs identified in a simple Flask Notes application, their causes, exact error-prone code lines, and the correct approach used to resolve them. Each issue is explained in a clear and structured manner for easy understanding.

Bug 1: Route Only Allowed POST Method

Error Code Line:

```
@app.route('/', methods=['POST'])
```

Problem Explanation: The root URL (/) was configured to accept only POST requests. When a browser loads a page, it always sends a GET request first. Because GET was not allowed, the application could not display the page properly.

Resolved Code Line:

```
@app.route('/', methods=['GET', 'POST'])
```

Resolution Approach: The route was updated to accept both GET and POST methods so the page can load normally and also handle form submissions.

Bug 2: Using request.args for POST Data

Error Code Line:

```
note = request.args.get('note')
```

Problem Explanation: `request.args` is used only for query parameters sent through GET requests. Since the form submits data using POST, the submitted note could not be retrieved correctly.

Resolved Code Line:

```
note = request.form.get('note')
```

Resolution Approach: `request.form` was used to correctly capture form data sent via POST requests.

Bug 3: Form Action Missing

Error Code Line:

```
<form action="">
```

Problem Explanation: The form action attribute was empty, which can cause unpredictable behavior. The application did not clearly know where to send the form data.

Resolved Code Line:

```
<form action="/",method="POST">
```

Resolution Approach: The form action was explicitly set to the root route and the method was defined as POST.

Bug 4: Empty Notes Being Added

Error Code Line:

```
notes.append(note)
```

Problem Explanation: The application allowed empty strings to be added to the notes list, resulting in blank entries being displayed.

Resolved Code Lines:

```
if note:  
    notes.append(note)
```

Resolution Approach: A validation check was added to ensure that only non-empty notes are stored.

Conclusion

By correcting routing methods, handling form data properly, defining form actions clearly, and validating input, the Flask Notes application now functions reliably. These fixes follow standard Flask best practices and improve both stability and user experience.