# Title: DevOps: Setting up customer repeated asks using freshservice (FSBOT scripts)

1) All the scripts should be **approved** by **respective leads** and uploaded to S3 bucket only in case of overriding otherwise it will be available in <https://github.com/freshdesk/freshsales/tree/prestaging/fsbot> . So for every changes we must have to update here also.URL of S3 Bucket: <https://s3.console.aws.amazon.com/s3/buckets/fsbot-staging/?region=us-east-1>(Staging), [https://s3.console.aws.amazon.com/s3/buckets/fsales-supreme/fsbot/?region=us-east-1&tab=overview#](https://s3.console.aws.amazon.com/s3/buckets/fsales-supreme/fsbot/?region=us-east-1&tab=overview)(Prod)

**Sample Script Template**

class Fsbot<ClassName>  
 def self.run\_script(args = {})  
 ..... # Code Part Here  
 { type: '<Type>', message: "<Message>", attachments: ["/path/to/attachment/1", "/path/to/attachment/2"] }  
 end  
end

**Note:**

All the attachments will be sent out to user and will be deleted in the memory. No uploading to S3 or uploading to Freshservice as a note.

Be sure about the **class name** the script will provide while approving. If it mistakenly have the same name as defined classes in the freshsales application, it might override the class definition itself. So class name must start with **Fsbot**. So the final class name will be like **Fsbot<class name>**.

If script throws an **exception**, the **stack trace** will be sent to the requester.

**Return Parameters Explained:**

* **type:** can be "note" or "mail"
* **message:**can be anything in String format to sent out to user as a message header
* **attachments:** should be absolute paths of files and should be as a list of string paths of files.

2) Create a Service Item for every script in <https://freshsales.freshservice.com/catalog/items> with meaningful name.

**Note:**

For passing arguments to script, you need to specify all the values in **Custom Fields** section of the item. Both **Account Id** and **Domain** will be used for validation.

* **Account Id:** Should be with same name for all the scripts (required field)
* **Domain:**Should be with same name for all the scripts (required field)
* **Custom Field 1:** Can be anything that's gonna be required by the respective script.
* **Custom Field 2:** Can be anything that's gonna be required by the respective script.
* **Custom Field n:** Can be anything that's gonna be required by the respective script.

3) Edit the workflow automator in freshservice. URL: <https://freshsales.freshservice.com/admin/ticket_automators/10000317510>

**Note:**

Trigger Webhook action has been already configured. So no need configure for every script. Configured as:

**Event** → **Condition** → **Webhook**

**Event:**

* **Service Request** is **Raised**
* These events can be performed by **Anyone**

Up to **Event** is common for all. So you have to create a new **Condition → Webhook** for every item you want to include.

**Condition:**

* Match **any**of these conditions
* **Requested Items** includes any **<Choose all configured items>**

**Webhook:**

* + Perform these actions on **Ticket**
  + Choose action as **Trigger Webhook**
  + **Request Type:** POST
  + **URL:** <End Point of Docker or Staging or Prod>/fsadmin/fsbot/global\_endpoint
  + **Authorization**: Use**I have API key ?**and Enter the API key for freshsales authentication.
  + **Encoding: JSON** & sub division as **Advanced**.
  + **JSON Body:**
  + {  
        "ticket\_id" : "{{ticket.id}}",  
        "requester\_email" : "{{ticket.from\_email}}",  
        "req\_items" : "{{ticket.service\_request\_item\_names}}",  
        "fields" : "{{ticket.service\_item\_fields}}",  
        "file" : "<corresponding script path in S3 bucket>.rb"  
    }
  + The file present in the S3 bucket should be **fsbot\_class\_name.rb** format only. Should be in underscore format. It will be **camelized** while accessing.

4) **Example Email as a Screenshot:**

