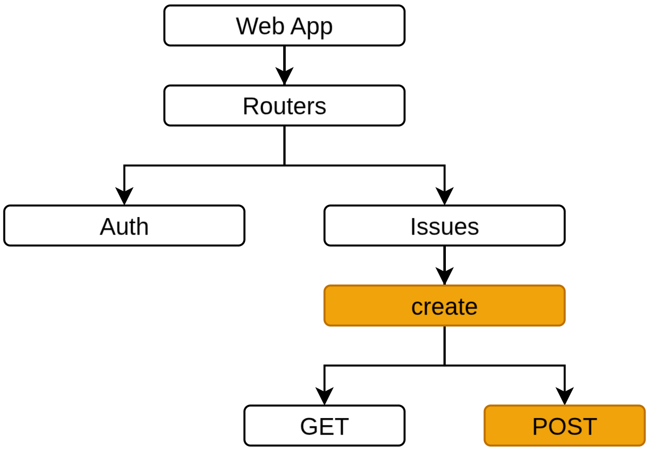
Contributors: Issues::Create::POST↗

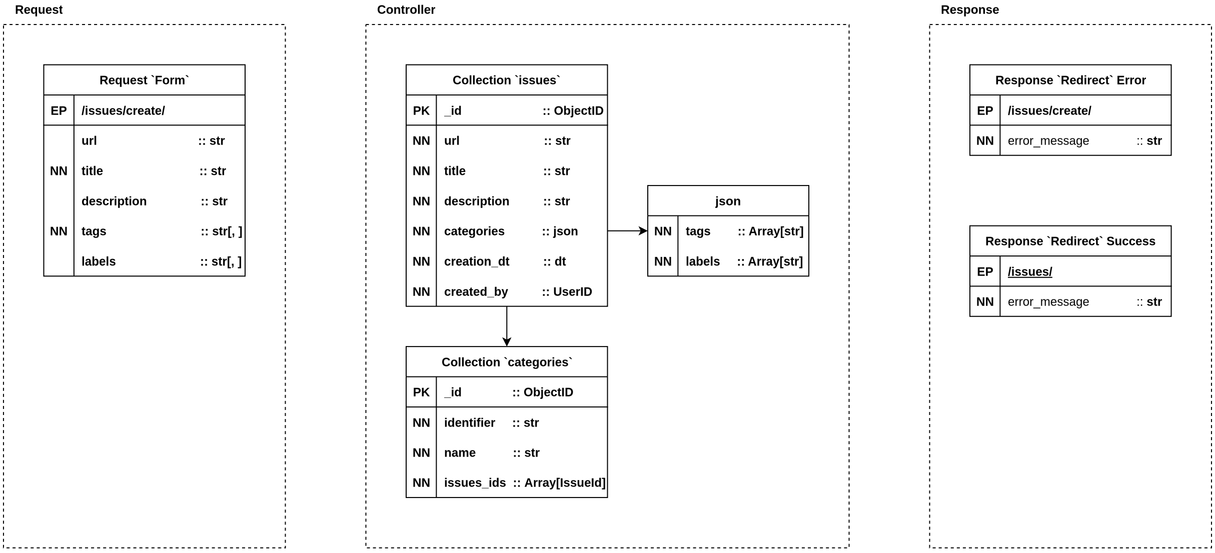
YóUnǎi ⊶ ames0k0
EMail↗ LinkedIn↗ Github↗

Abstract

Contributors is a web application designed to streamline open-source contribution by improving how developers discover and share coding tasks. Built to address the limitations of issue discovery on code hosting platforms with integrated issue trackers, it enables users to curate, tag, and submit contribution-friendly issues—helping developers quickly find clear, actionable opportunities to collaborate when they're ready to contribute.
Tech Stack: *Python*, *FastAPI*, *MongoDB*



Ticket



Request

Request `Form`		
EP	/issues/create/	
	url	:: str
NN	title	:: str
	description	:: str
NN	tags	:: str[,]
	labels	:: str[,]

Controller

Collection `issues`		
PK	_id	:: ObjectID
NN	url	:: str
NN	title	:: str
NN	description	:: str
NN	categories	:: json
NN	creation_dt	:: dt
NN	created_by	:: UserID

json		
NN	tags	:: Array[str]
NN	labels	:: Array[str]

Collection `categories`		
PK	_id	:: ObjectID
NN	identifier	:: str
NN	name	:: str
NN	issues_ids	:: Array[IssueId]

Response

Response `Redirect` Error		
EP	/issues/create/	
NN	error_message	:: str

Response `Redirect` Success		
EP	/issues/	
NN	error_message	:: str

1. `Request`
2. Form Validation
3. Data Validation
4. Data Insertion
5. `Response`

1. Request

```
+! CurrentUser
|> cookies[AccessToken]
```

2. Form Validation

```
+? F.url        +! F.title        +? F.description  +! F.tags         +? F.labels
|> str.Strip    |> str.Strip      |> str.Strip      |> str.Strip      |> str.Strip
|> SupportedURL |> orelse raise                     |> str.Split[,]   |> str.Split[,]
|> orelse raise                                     |> orelse raise
```

3. Data Validation

```
+? F.url
|> db.exists(issues.url)
|> raise
```

4. Data Insertion

```
+! F.@ + CurrentUser
|> db.insert(issues)

+! F.tags
|> db.exists(categories.identifier && categories.name)
|> db.update(categories.issues_ids)
|> orelse db.insert(categories)

+? F.labels
|> db.exists(categories.identifier && categories.name)
|> db.update(categories.issues_ids)
|> orelse db.insert(categories)
```

5. Response

```
+? Error                          +? Success
|> entryPoint(/issues/create/)    |> entryPoint(/issues/)
|> error_message                  |> error_message
```