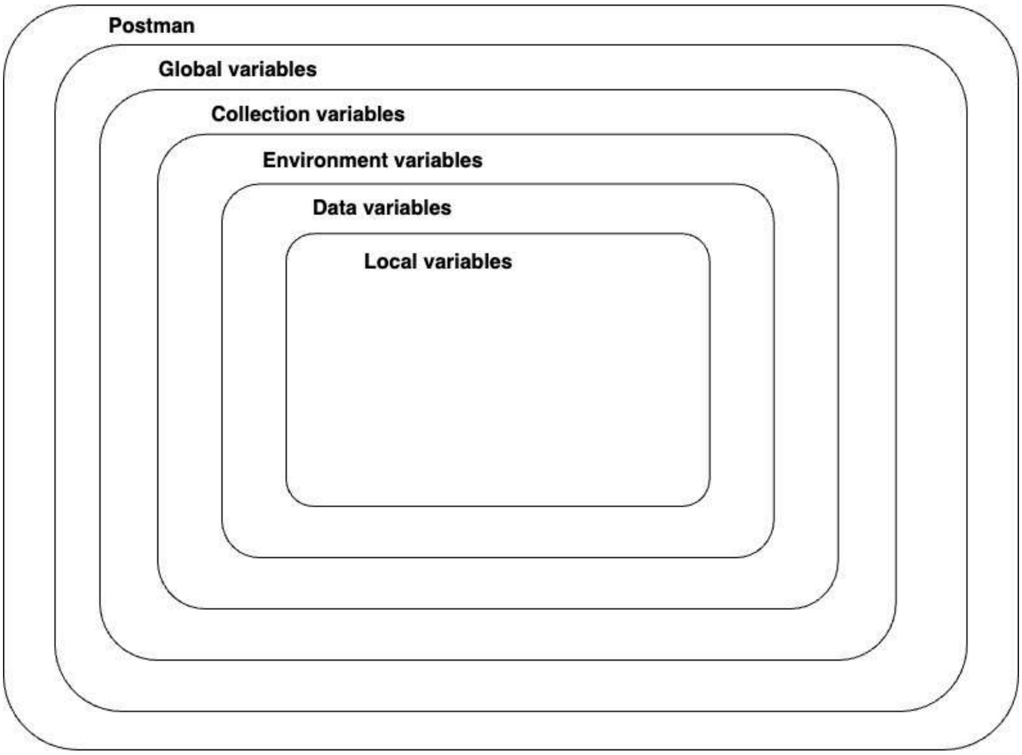
4. Variables in Postman

Variables in Postman

- ➔ Variables are placeholders for values used in requests, scripts, or collections.
  - ◆ Example → urls', API Keys, Request Payload/Body, headers, etc
- ➔ We use Variables
  - ◆ To avoid hardcoding values.
  - ◆ To make scripts reusable and dynamic.
  - ◆ To manage configurations for different environments.

Types and Scope of Variables in Postman

1. **Global Variable** → Accessible in workspace i.e, in Multiple collections but outside not possible.
   a. url_global = https://reqres.in
2. **Environment Variable** → Specific to an environment like QA or Prod or UAT
   a. url_qa_env = https://reqres.in
3. **Collection Variable** → Specific to a collection of requests. Outside collection we cannot access
   a. url_collect = https://reqres.in
4. **Local Variable** → Specific to a single request in a collection
   a. url_local = https://reqres.in
5. **Data Variable** → Values provided through CSV/JSON files in the collection runner for DDT.



Note

- If a variable with the same name is declared in two different scopes, the value stored in the variable with narrowest scope will be used.
  - For example, if there is a global variable named username and a local variable named username, the local value will be used when the request runs.

Accessing Variables in URLs

- Use **{{variable_name}}** to reference variables in the request URL.
- **Example**
  - {{url_global}}/api/users?page=2
  - {{url_qa_env}}/api/users?page=2
  - {{url_collect}}/api/users?page=2
  - {{url_local}}/api/users?page=2

Note

- Each variable has an **Initial value** and **Current value**
  - **Initial Value (value set during creation of variable)**
    - Set when defining a variable in a collection, environment, or globals.
    - Synced to Postman's servers and shared with your team.
    - Useful for sharing elements with teammates but be cautious as it shares data with others.
    - Use Postman Vault for sensitive data to keep it private and local.
  - **Current Value (Temporarily value for testing, can be different from the initial value)**
    - Used when sending requests.
    - Stored locally and not synced to Postman's servers.
    - Changes to current values aren't saved in shared collections, environments, or globals.
- Global and environment variables can also be classified as **Default type** and **Secret type**.
  - **Default Type**
    - Automatically assigned to variables.
    - Shown as plain text without additional properties.
  - **Secret Type**
    - Masks initial and current values for all workspace members.
    - Prevents unintentional disclosure of sensitive data like API secrets, passwords, tokens, and keys.

- **Recommendation**
    - Store sensitive data in **Postman Vault** to keep it private.
    - Share it with others by using a **secret type variable** in an environment.

Workout in Postman Tool

➔ Create a collection **Day4_VariablesAndScripts** in workspace in Postman Tool

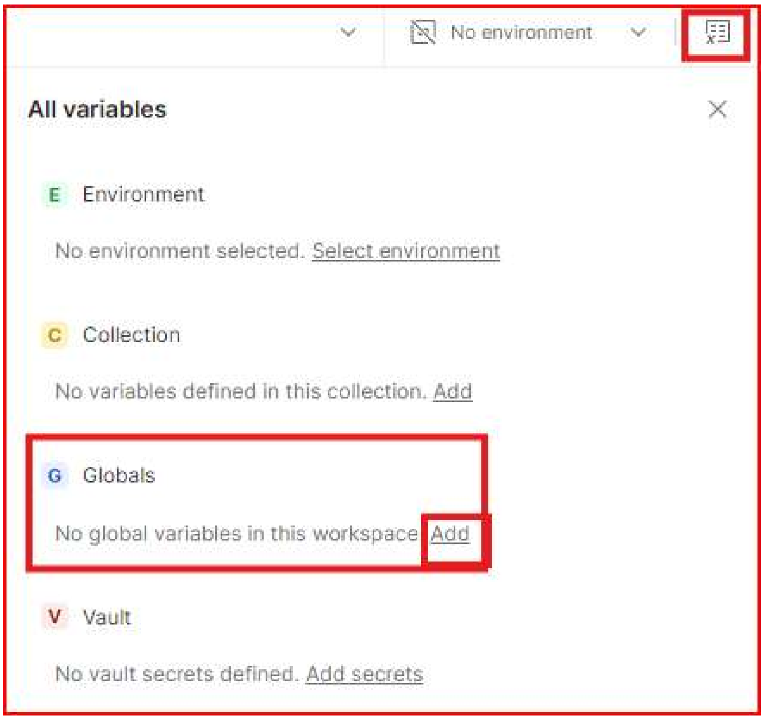
➔ Observe any one request in **https://reqres.in/** → **https://reqres.in/api/users?page=2**

- **Base URL** → https://reqres.in
- **Path** → /api/users
- **Query Parameter** → ?page=2

➔ Like above request for multiple requests Base URL is same only Path and Query Parameters will change.

Global Variable Creation and Usage

1. We will make Base URL as a Global variable.
    a. Variables → Globals → Add



No environment

All variables

E Environment

No environment selected. Select environment

C Collection

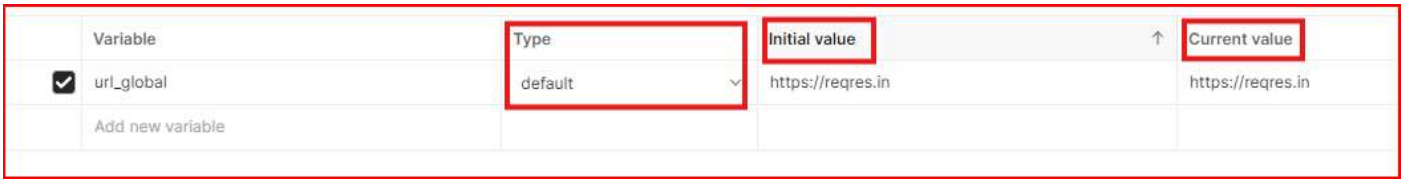
No variables defined in this collection. Add

G Globals

No global variables in this workspace Add

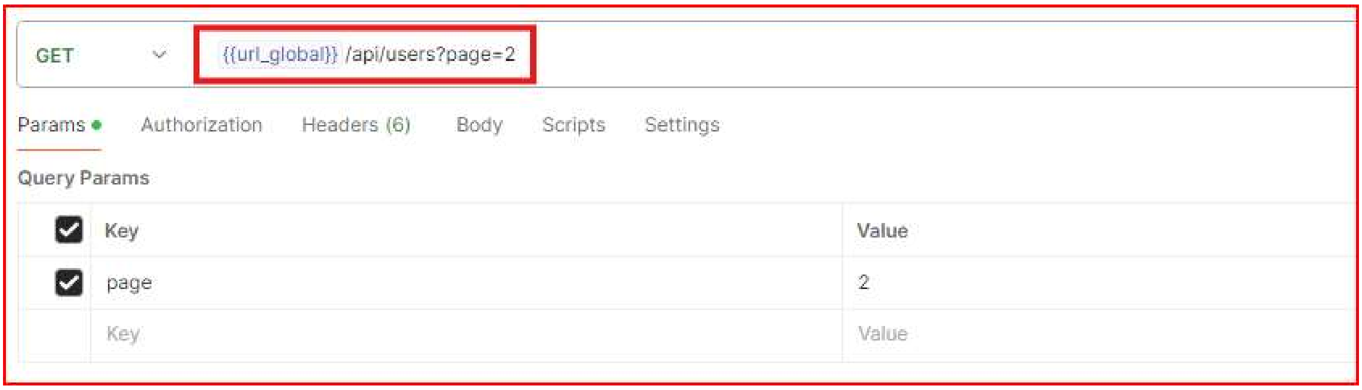
V Vault

No vault secrets defined. Add secrets



	Variable	Type	Initial value	Current value
☑	url_global	default	https://reqres.in	https://reqres.in
	Add new variable			

2. Create a Request with name **Get Users_globalVar** in Day4_Variables Collection
3. Replace Base url with Global variable as below.
    a. **{{url_global}}**/api/users?page=2



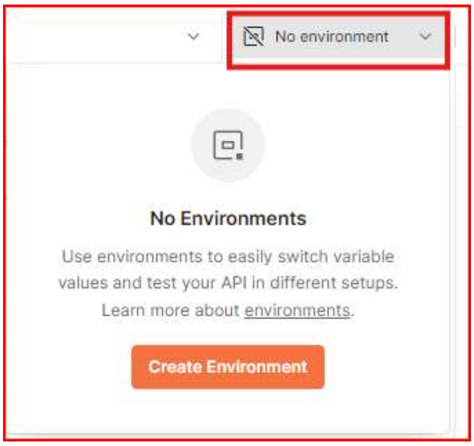
GET {{url_global}} /api/users?page=2

Query Params

☑	Key	Value
☑	page	2
	Key	Value

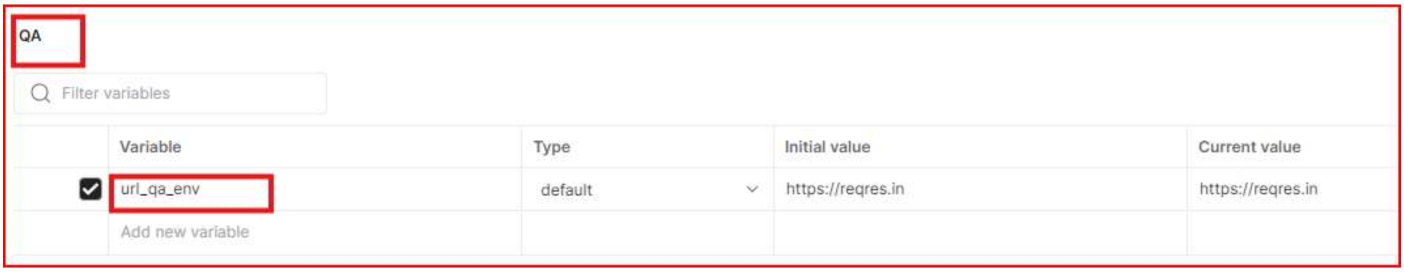
Environmental Variable Creation and Usage

1. First we will Create Environment and then we will create Environmental Variable under Environment.



No Environments

Use environments to easily switch variable values and test your API in different setups. Learn more about environments.



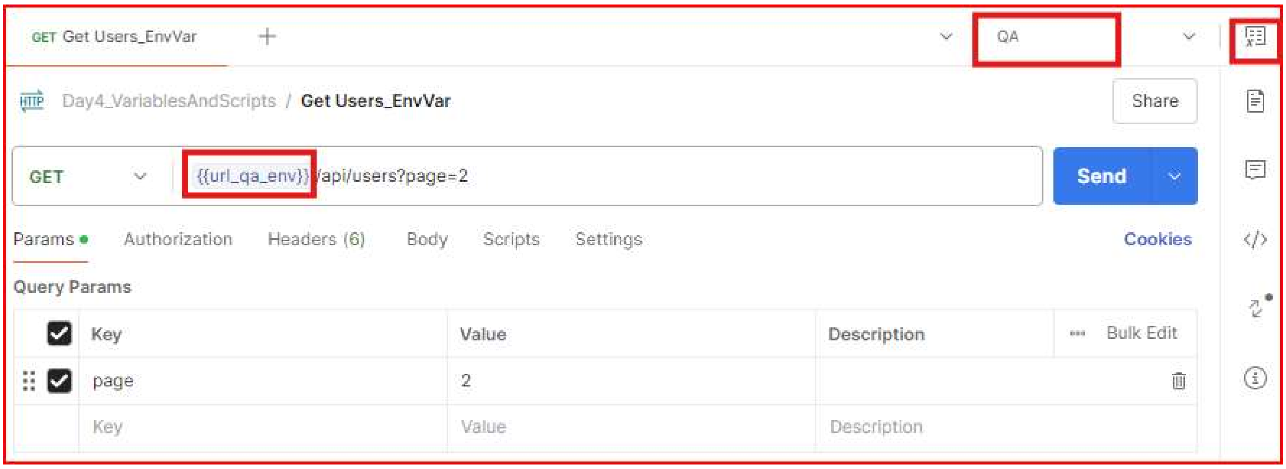
QA

	Variable	Type	Initial value	Current value
☑	url_qa_env	default	https://reqres.in	https://reqres.in
	Add new variable			

2. Create a Request with name **Get Users_EnvVar** in Day4_Variables Collection

3. Replace Base url with Environmental variable as below by **selecting that enviornment**

   a. **{{url_qa_env}}**/api/users?page=2



GET {{url_qa_env}}/api/users?page=2

Query Params

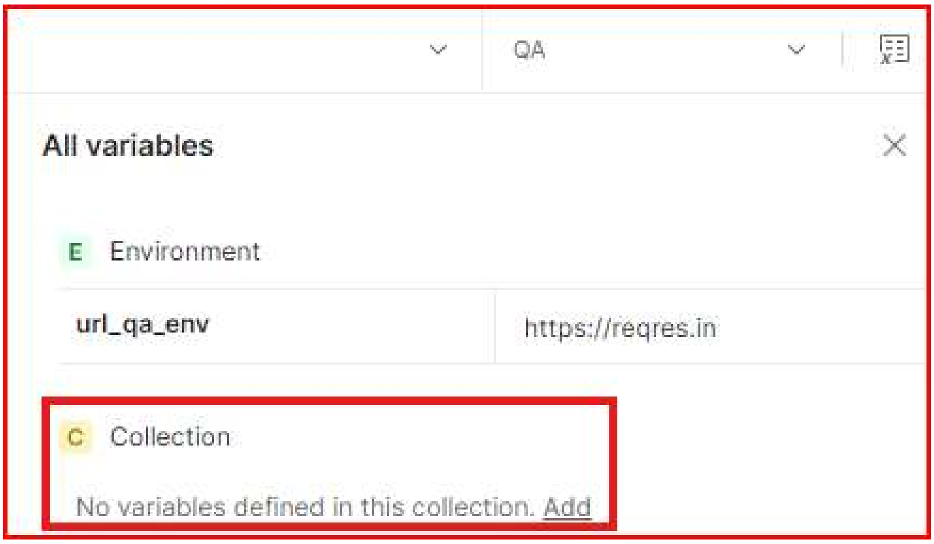
☑	Key	Value	Description	••• Bulk Edit
☑	page	2		
	Key	Value	Description	

Note

➔ If the correct environment is not selected or switched for the environmental variable, an error will occur.

Collection Variable Creation and Usage

1. We will make Base URL as a Collection variable.
   a. Variables → Collection → Add → Variable



QA

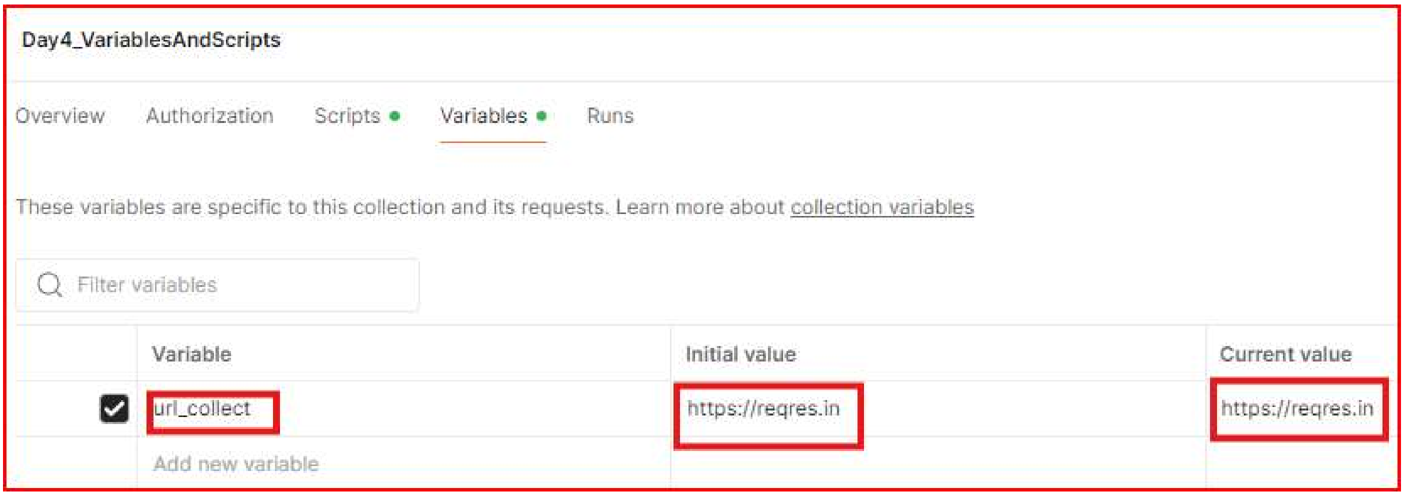
All variables

E Environment

url_qa_env	https://reqres.in

C Collection

No variables defined in this collection. Add



Day4_VariablesAndScripts

Overview Authorization Scripts Variables Runs

These variables are specific to this collection and its requests. Learn more about collection variables

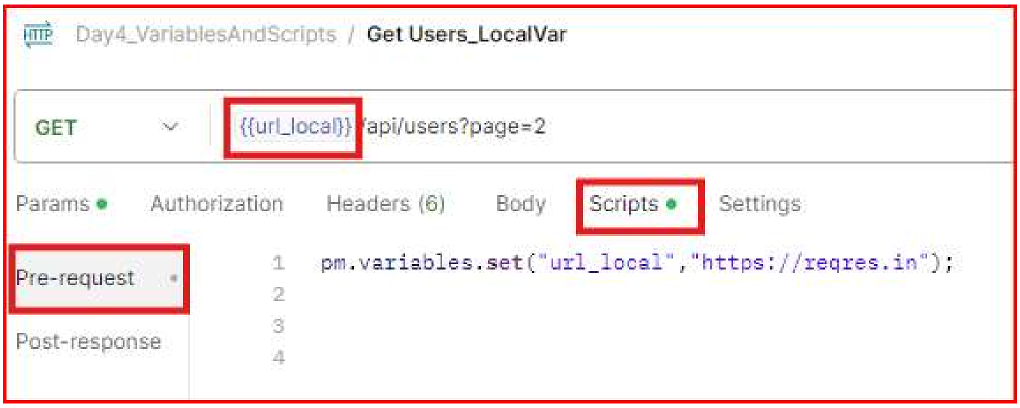
Filter variables

	Variable	Initial value	Current value
☑	url_collect	https://reqres.in	https://reqres.in
	Add new variable		

2. Create a Request with name **Get Users_CollectVar** in Day4_Variables Collection
3. Replace Base url with Collection variable as below by selecting **No enviornment**
   a. **{{url_collect}}**/api/users?page=2

Local Variable Creation and Usage

1. We will now make Base URL as a Local variable.
2. Create a Request with name **Get Users_LocalVar** in Day4_Variables Collection
3. Local Variables are created under **Pre-request Script in Scripts Tab** of Request using JavaScript
   a. **pm.variables.set("url_local","https://reqres.in");**



{{url_local}}/api/users?page=2

```
pm.variables.set("url_local","https://reqres.in");
```

Note

➔ In Real time we will create any type of variable using JavaScript Statements or using snippets instead of using **UI options**.