

Modify Response Body

! INFO

- Only requests triggered by XHR/fetch can be modified by the browser extension. For other type of requests, use [Desktop App](#)
- Response Modifications will not show up in the browser network devtools due to technical constraints. Try [this](#) to test whether Response Rule is working or not.

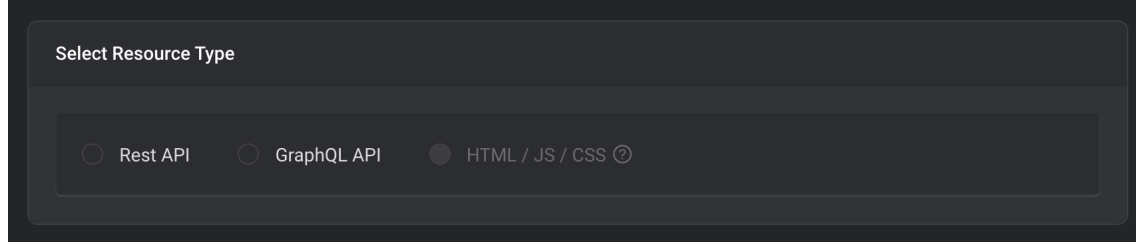
The Modify Response Rule allows you to mock the response body of an HTTP request.

Popular use cases

- You want to work on front-end while back-end is not available or ready yet.
- You want to test application behaviour when provided altered data.
- You want to simulate errors by returning different status codes.
- You want to modify API responses but don't have access to the back-end.

Modify API Response using Requestly Chrome Extension





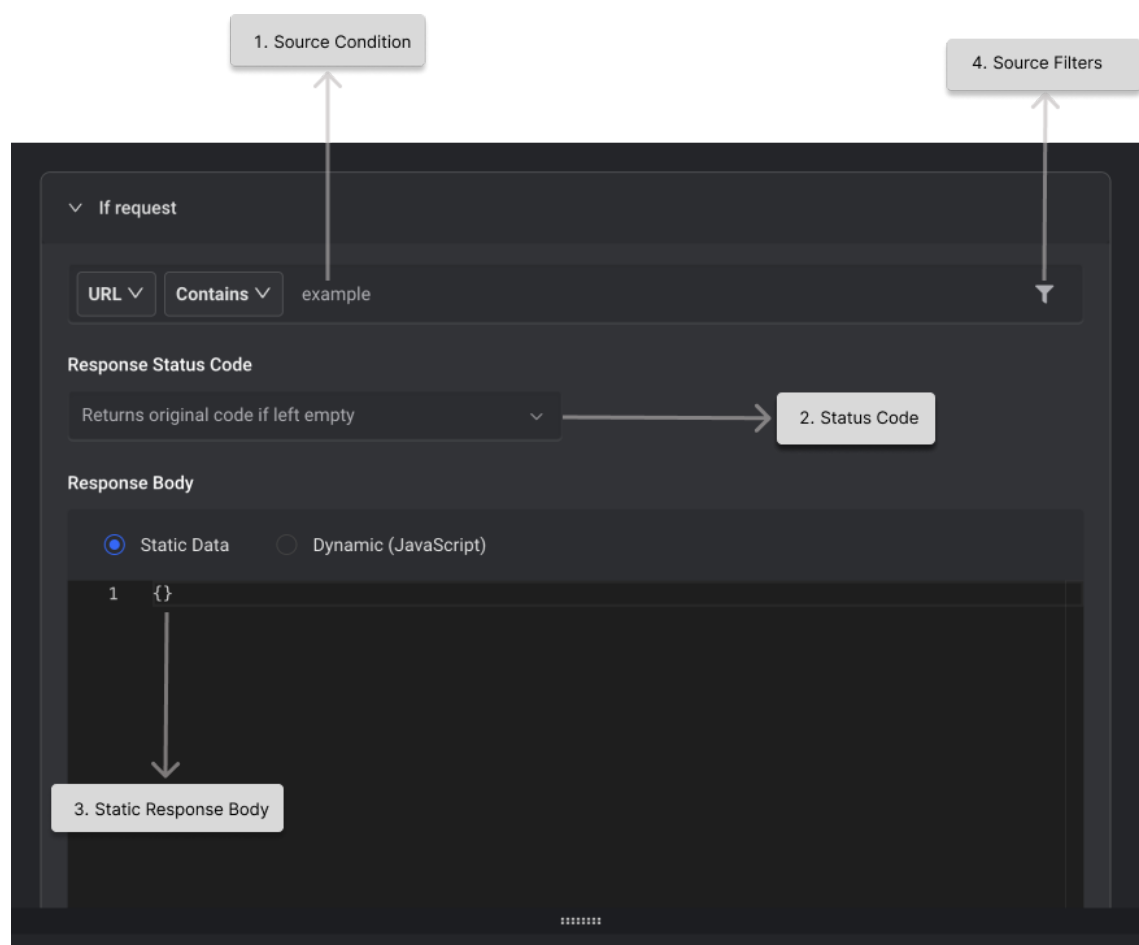
Response Rule resource types

- Rest API: Supports modifying XHR/fetch requests.
- GraphQL API: Target & modify GraphQL api using operation.
- HTML/JS/CSS: Only available in Desktop app

Types of Modifications

Static Response Modification

In this mode, you can enter the response body which you want as the response.



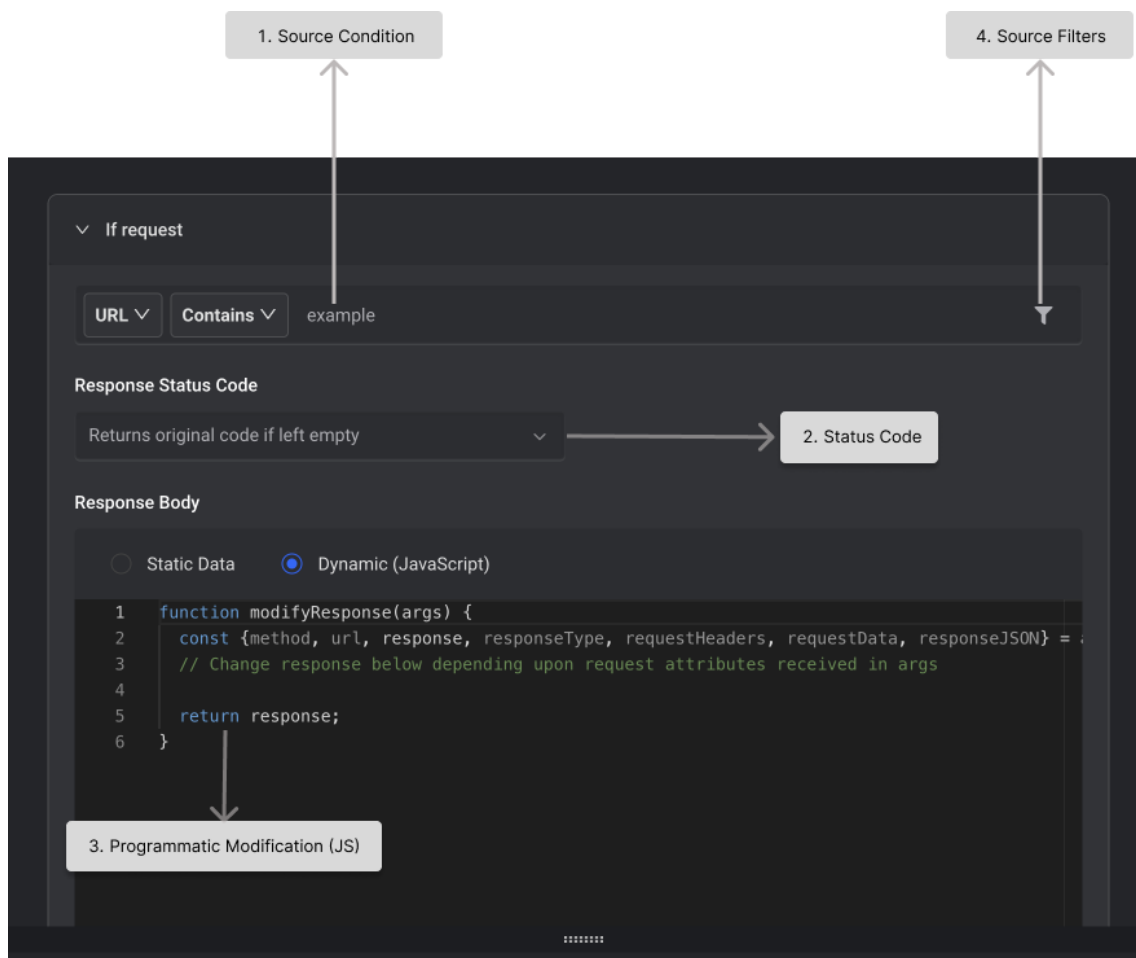
Response Rule - Static Body

1. **Source Condition** : If the request url matches this *source condition*, then only the rule is applied. You can use `URL`, `Host` or `Path` with `Regex`, `Contains`, `Wildcard` or `Equals` to match the source request. Learn more about source conditions [here](#).
2. **Status Code** : Update the Response status code. (2XX, 3XX, 4XX, 5XX ...). If left empty, original status code is returned.

3. **Static Response Body** : Define the updated response body which needs to be returned.
4. **Source Filters** : This helps you define better targeting conditions and restrict rules to be applied on specific webpages (or domains), request types, request methods, or request payload. Learn more about source filters [here](#).

Programmatic Response Modification

Programmatic Modification Script (JS) : Write a JavaScript script which can modify the existing response programmatically.



Arguments of `modifyResponse`

1. `method` (string)- The [HTTP method](#) of the request. `GET | POST | PUT | DELETE` etc.
2. `url` (string) - The request URL.
3. `response` (string)- The original response object represented as string. Eg:

```
'{"id":1,"app":"requestly","feature":"modify-request"}'
```

4. `responseType` (string)- The [content-type](#) of the HTTP response.
5. `requestHeaders` (Object<string, string>)- The [request headers](#) sent to the server.
6. `requestData` (string)- The HTTP request payload in case of `POST | PUT | PATCH` requests.
7. `responseJSON` (JSON object)- The original response represented as JSON Object:

```
{
  "id":1,
  "app":"requestly",
  "feature":"modify-request"
}
```

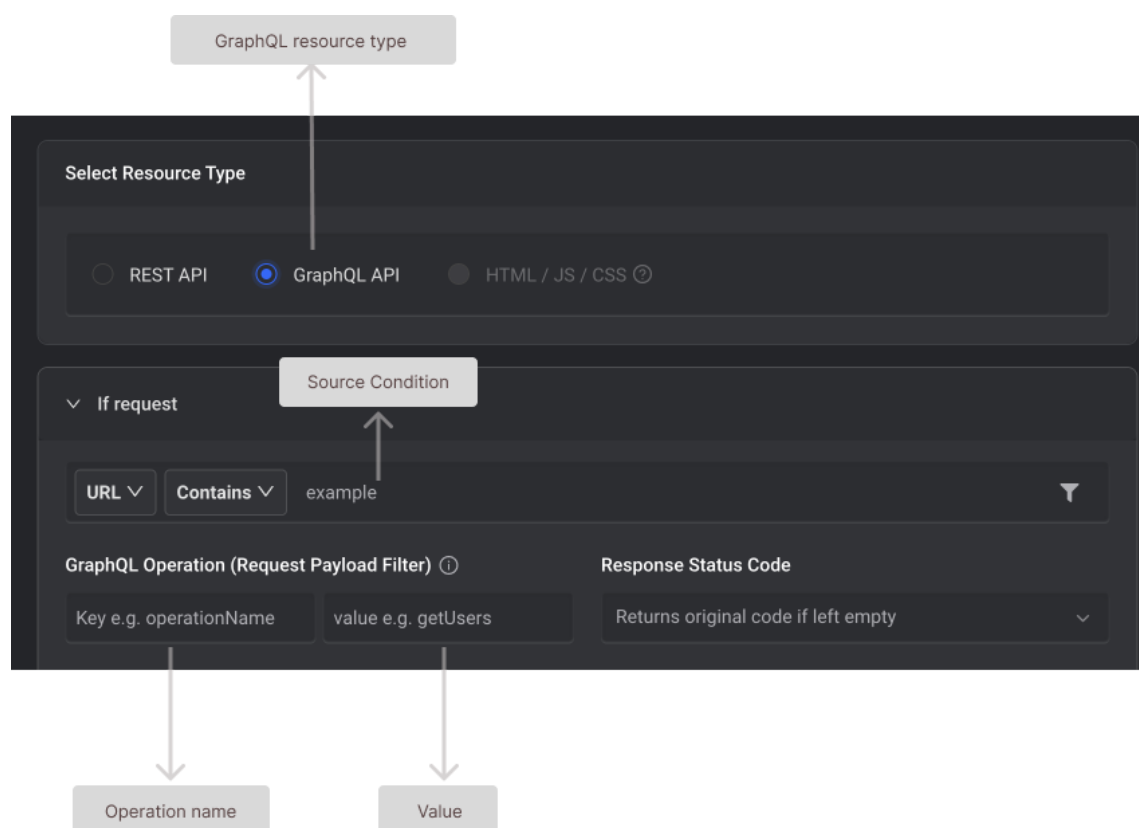
Return type of `modifyResponse` (`string` or `JSON object`)

You can modify both response or responseJSON and return any of them. Response Rule will automatically handle the response returned for the request.

Targeting GraphQL requests

You can target GraphQL requests using the operation name in request body. To do so add:

- Payload JSON Key, e.g., `operationName`
- Value, e.g., `getUsers`



To learn more about this checkout [GraphQL Support](#).

Guides

- [Modifying Response asynchronously](#): Async modifications lets you perform some long running tasks before returning the response

- [Modifying Non Existent Domain](#): Helpful for frontend developers when the backend isn't ready yet.

FAQ

▶ Response Rule not working

▶ What happens if i leave the status code field empty

▶ Where would ``console.log`` show up?

▶ How to target ``query`` in graphql requests

 [Edit this page](#)