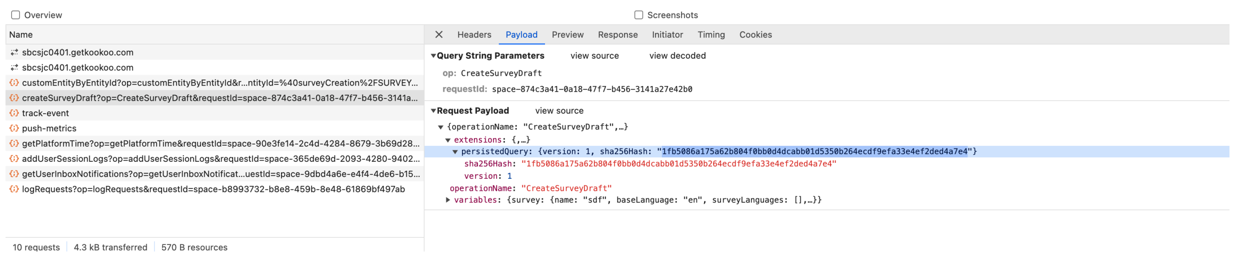
Graphql query interceptor

Graphql Persisted Query Cache clearing at runtime

16. `GRAPHQL_PERSISTED_QUERY_CACHE` we clear this cache to see the actual query but it has some drawbacks-
    a. Reloading whole page
    b. Query is visible only single time
17. So we do have other method also to view the query
    a. We will be intercepting the calls from web browser and then we will change the cache key and sent the query, Now the server will get an invalid Persistent query cache so it will return error status code and ui code will be sending the whole query next time.
    b. Step1 . Copy the cache key from network tab
    c.



```
sbcsjc0401.getkookoo.com
sbcsjc0401.getkookoo.com
customEntityByEntityId?op=customEntityByEntityId&r...ntityId=%40surveyCreation%2FSURVEY...
createSurveyDraft?op=CreateSurveyDraft&requestId=space-874c3a41-0a18-47f7-b456-3141a...
track-event
push-metrics
getPlatformTime?op=getPlatformTime&requestId=space-90e3fe14-2c4d-4284-8679-3b69d28...
addUserSessionLogs?op=addUserSessionLogs&requestId=space-365de69d-2093-4280-9402...
getUserInboxNotifications?op=getUserInboxNotificat...uestId=space-9dbd4a6e-e4f4-4de6-b15...
logRequests?op=logRequests&requestId=space-b8993732-b8e8-459b-8e48-61869bf497ab

10 requests | 4.3 kB transferred | 570 B resources
```

```
▼Query String Parameters      view source      view decoded
  op: CreateSurveyDraft
  requestId: space-874c3a41-0a18-47f7-b456-3141a27e42b0

▼Request Payload      view source
 ▼ {operationName: "CreateSurveyDraft",…}
   ▼ extensions: {,…}
     ▼ persistedQuery: {version: 1, sha256Hash: "1fb5086a175a62b804f0bb0d4dcabb01d5350b264ecdf9efa33e4ef2ded4a7e4"}
        sha256Hash: "1fb5086a175a62b804f0bb0d4dcabb01d5350b264ecdf9efa33e4ef2ded4a7e4"
        version: 1
     operationName: "CreateSurveyDraft"
   ▶ variables: {survey: {name: "sdf", baseLanguage: "en", surveyLanguages: [],…}}
```

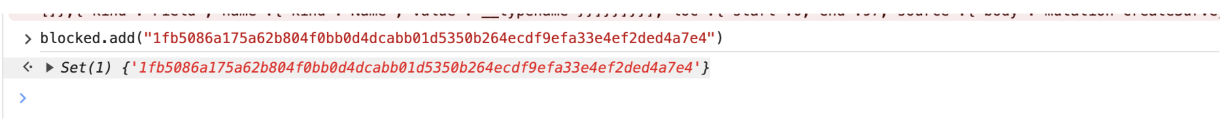
    d. Step2: - paste this code in the console which creates an JavaScript Set with name "blocked" which should contain the sha which needs to be altered and changed. Since it will create a wrapper on

```
const originalFetch = window.fetch;
const blocked = new Set();
// Override fetch method
window.fetch = async function (url, options = {}) {
  // Modify request body using the function
  if (options && options.body) {
    options.body = modifyRequestBody({
      method: options.method || "GET",
      url: url,
      body: options.body,
      bodyAsJson: tryParseJson(options.body),
    });
  }

  // Call the original fetch with modified options
  return originalFetch(url, options);
};

// Helper function to parse JSON safely
function tryParseJson(body) {
  try {
    return JSON.parse(body);
  } catch (e) {
    console.log("failed to parse json");
    return null;
  }
}

function modifyRequestBody({ method, url, body, bodyAsJson }) {
  let sha = bodyAsJson?.extensions?.persistedQuery;
  if (sha && blocked.has(sha.sha256Hash)) {
    console.log(
      "Blocked request",
      bodyAsJson.extensions.persistedQuery.sha256Hash
    );
    bodyAsJson.extensions.persistedQuery.sha256Hash = "1";
    return JSON.stringify(bodyAsJson);
  }
  return body;
}
```

a. Step4: - Now we need to do something which so resent the same query but without refreshing the whole page.
   eg refresh the widget, resent the failed request

b.

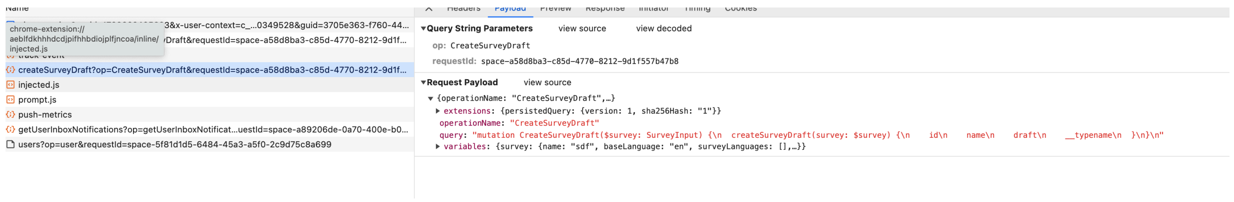
c.



```
> blocked.add("1fb5086a175a62b804f0bb0d4dcabb01d5350b264ecdf9efa33e4ef2ded4a7e4")
<· ▶ Set(1) {'1fb5086a175a62b804f0bb0d4dcabb01d5350b264ecdf9efa33e4ef2ded4a7e4'}
>
```

adding sha which i needed to be blocked

d.

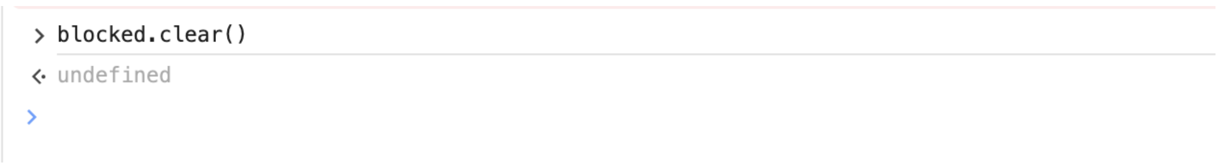


```
chrome-extension://
aeblfdkhhhdcdjpifhhbdiojplfjncoa/inline/
injected.js
…3&x-user-context=c_…0349528&guid=3705e363-f760-44…
…yDraft&requestId=space-a58d8ba3-c85d-4770-8212-9d1f…
createSurveyDraft?op=CreateSurveyDraft&requestId=space-a58d8ba3-c85d-4770-8212-9d1f…
injected.js
prompt.js
push-metrics
getUserInboxNotifications?op=getUserInboxNotificat…uestId=space-a89206de-0a70-400e-b0…
users?op=user&requestId=space-5f81d1d5-6484-45a3-a5f0-2c9d75c8a699
```

```
▼Query String Parameters     view source     view decoded
   op: CreateSurveyDraft
   requestId: space-a58d8ba3-c85d-4770-8212-9d1f557b47b8
▼Request Payload     view source
 ▼ {operationName: "CreateSurveyDraft",…}
   ▶ extensions: {persistedQuery: {version: 1, sha256Hash: "1"}}
     operationName: "CreateSurveyDraft"
     query: "mutation CreateSurveyDraft($survey: SurveyInput) {\n  createSurveyDraft(survey: $survey) {\n    id\n    name\n    draft\n    __typename\n  }\n}\n"
   ▶ variables: {survey: {name: "sdf", baseLanguage: "en", surveyLanguages: [],…}}
```

Resent Request using refreshing the widget

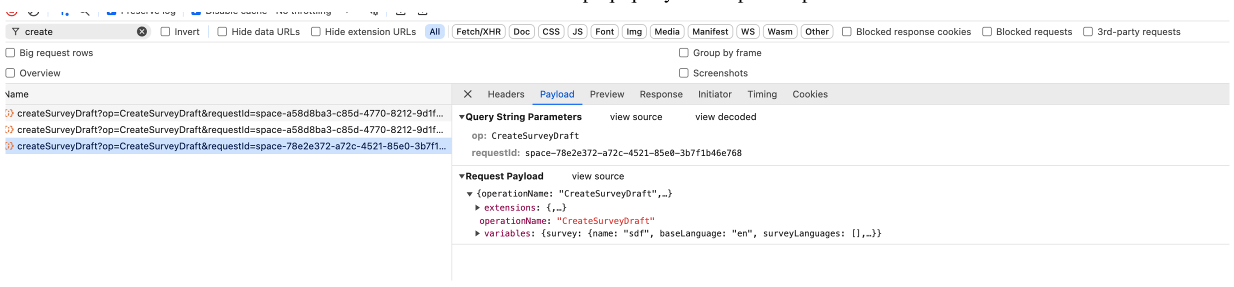
e.



```
> blocked.clear()
<· undefined
>
```

Clearing blocked Set post getting the query

f.



```
create  Invert  Hide data URLs  Hide extension URLs  All  Fetch/XHR  Doc  CSS  JS  Font  Img  Media  Manifest  WS  Wasm  Other  Blocked response cookies  Blocked requests  3rd-party requests
Big request rows                                   Group by frame
Overview                                           Screenshots

Name
createSurveyDraft?op=CreateSurveyDraft&requestId=space-a58d8ba3-c85d-4770-8212-9d1f...
createSurveyDraft?op=CreateSurveyDraft&requestId=space-a58d8ba3-c85d-4770-8212-9d1f...
createSurveyDraft?op=CreateSurveyDraft&requestId=space-78e2e372-a72c-4521-85e0-3b7f1...

Headers  Payload  Preview  Response  Initiator  Timing  Cookies

▼Query String Parameters    view source    view decoded
  op: CreateSurveyDraft
  requestId: space-78e2e372-a72c-4521-85e0-3b7f1b46e768

▼Request Payload    view source
 ▼ {operationName: "CreateSurveyDraft",…}
   ▶ extensions: {,…}
    operationName: "CreateSurveyDraft"
   ▶ variables: {survey: {name: "sdf", baseLanguage: "en", surveyLanguages: [],…}}
```

Sending query again after clearing the blocked set