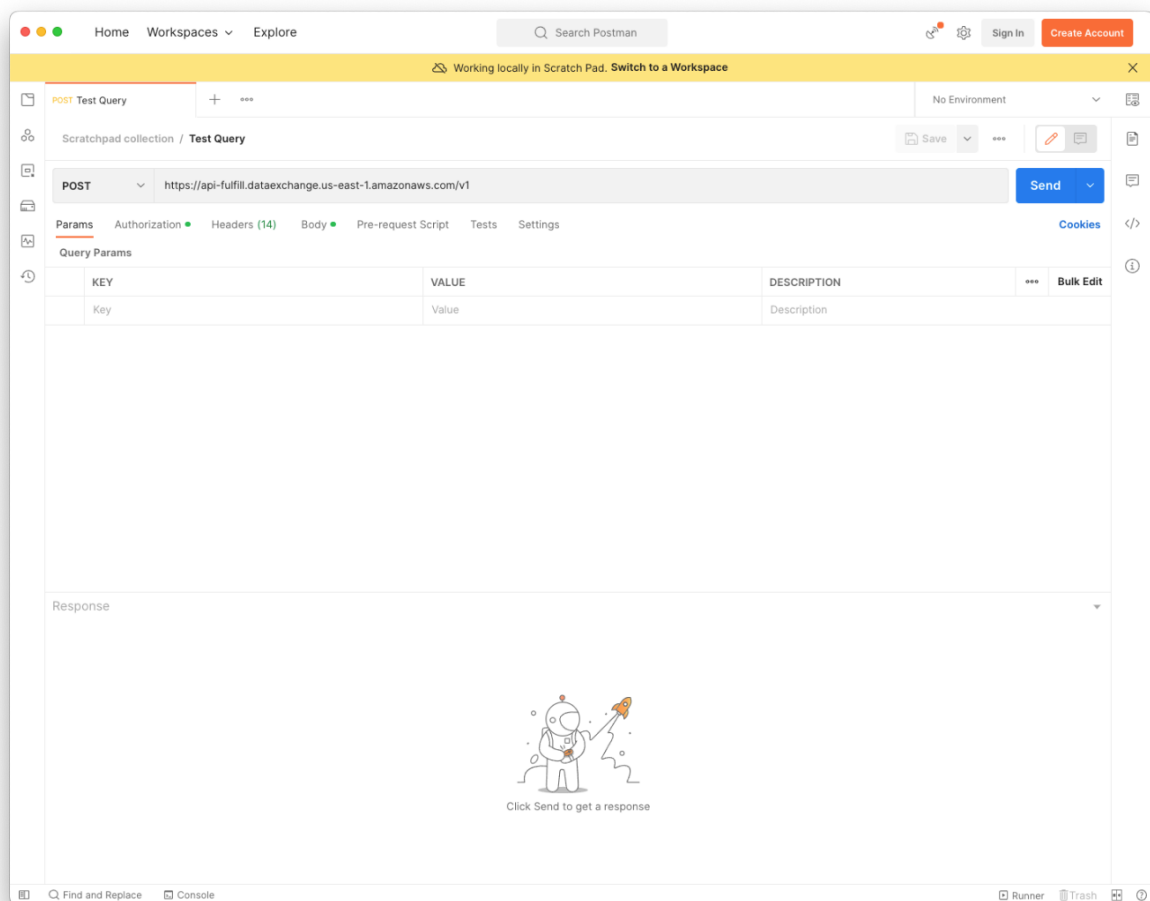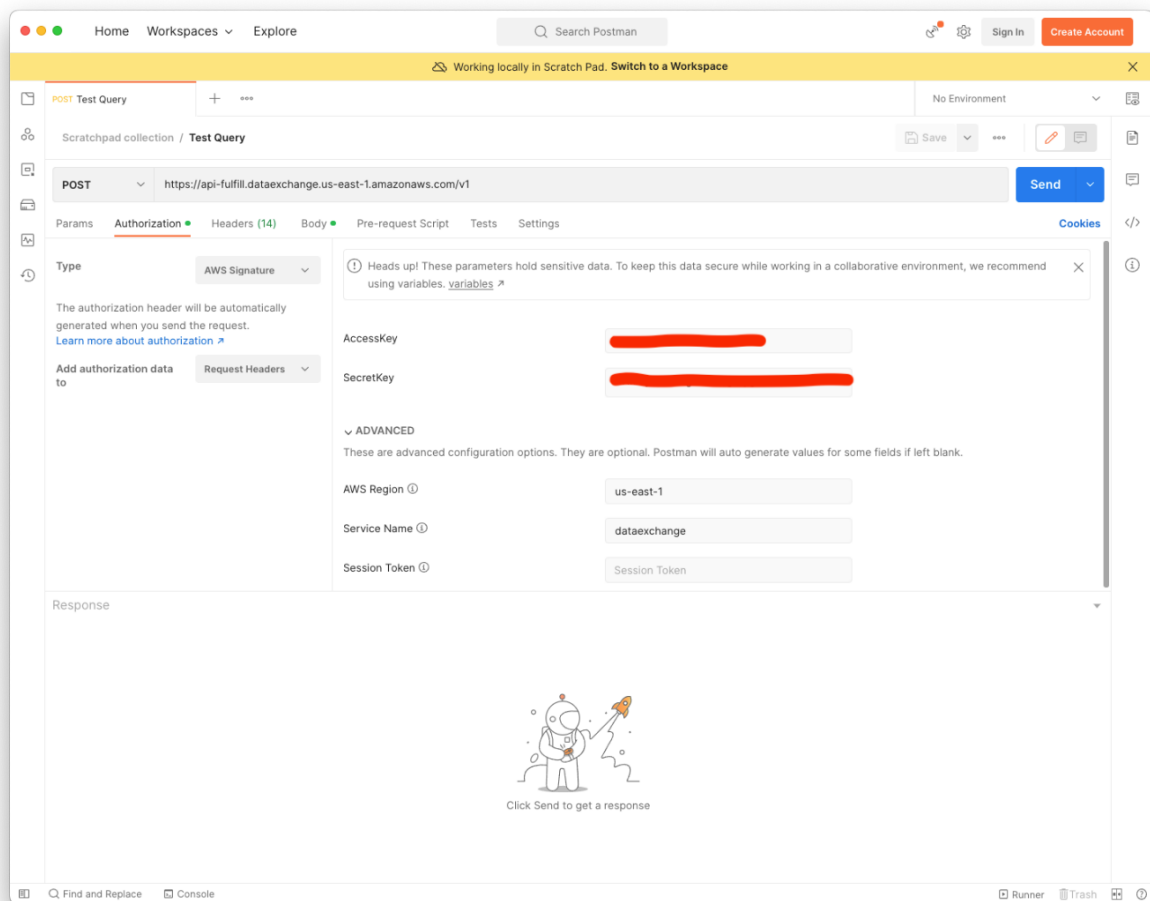# Sending requests to IMDb's Real-time API via Postman

To be able to send requests to IMDb's Real-time API via Postman you have to use AWS Sigv4 signatures to sign your requests. The following guide will help you set the parameters in Postman correctly so you could use it to send queries to the API.

- Before you begin: Have your AWS Access Key ID and AWS Secret Access Key ready.
  - *Please be aware that saving your AWS Access Key ID and AWS Secret Access Key in Postman is considered a security risk and you should fully protect your Postman workspace/collection/request from unauthorized access to avoid the disclosure of your AWS credentials.*
- Create a new request in Postman
  - Set the Method to **POST**
  - The request URL should be **https://api-fulfill.dataexchange.us-east-1.amazonaws.com/v1**

- On the **Authorization** tab
  - Set Type to be **AWS Signature**
  - Set Add authorization data to to be **Request Headers**
  - Set AccessKey to your **AWS Access Key ID**
  - Set SecretKey to your **AWS Secret Access Key**
  - Set AWS Region to **us-east-1**
  - Set Service Name to **dataexchange**
  - Leave Session Token empty
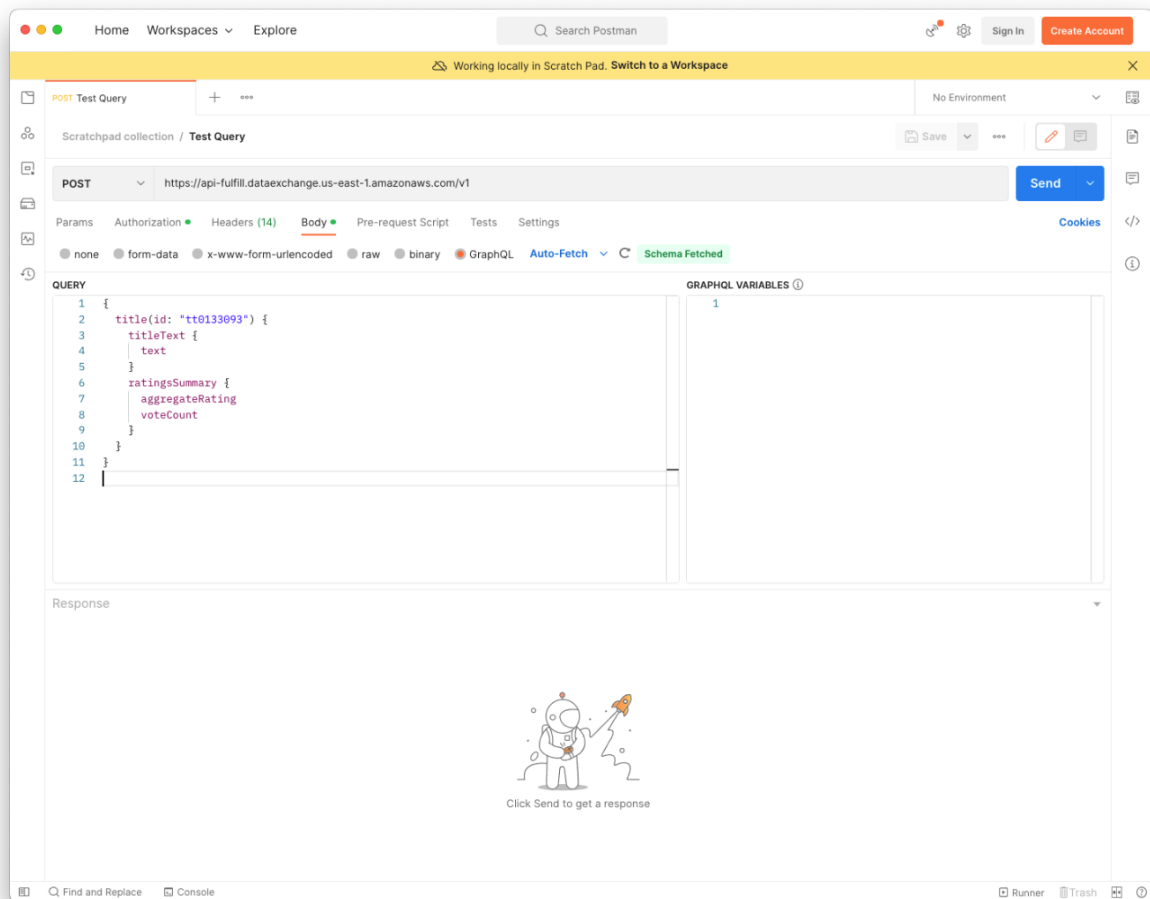


- On the **Body** tab
  - Select **GraphQL**
  - And paste this into the text editor:
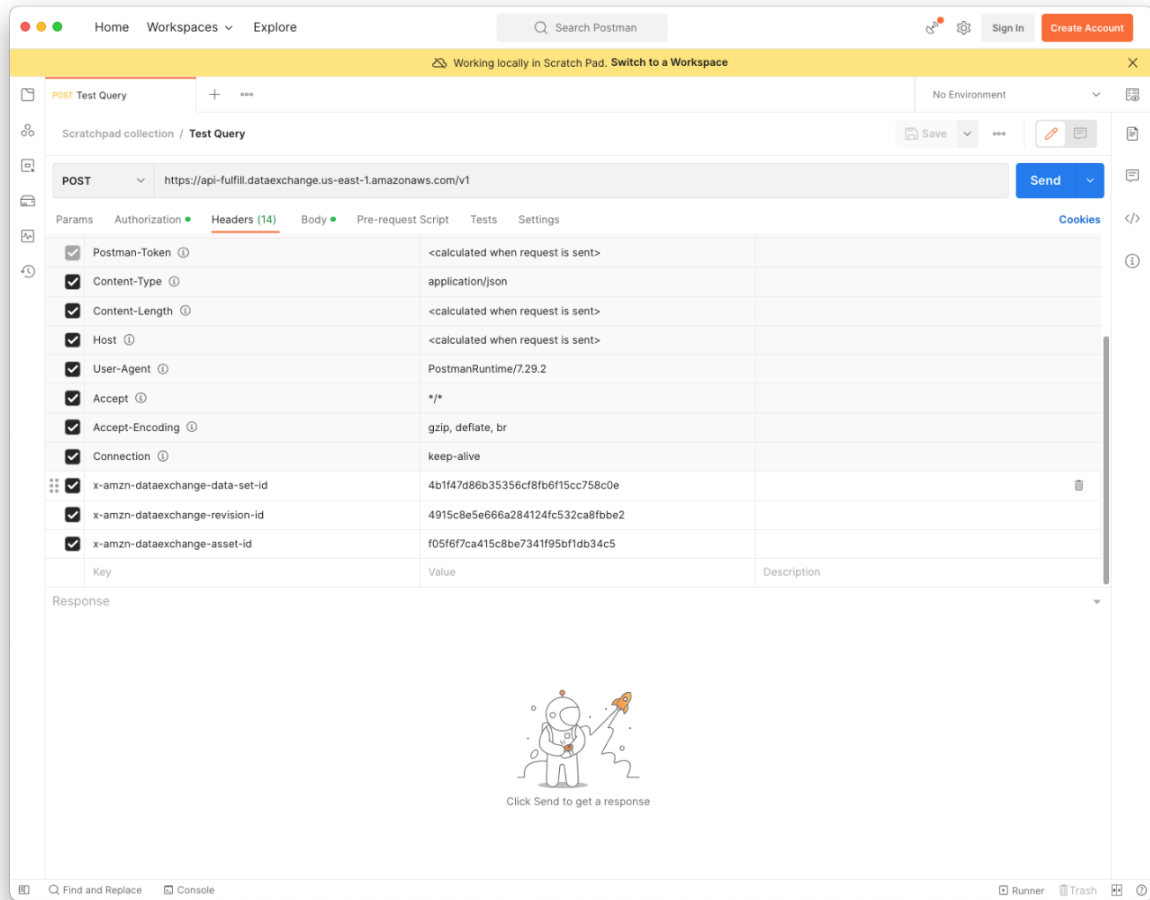
```
{
  title(id: "tt0133093") {
    titleText {
      text
    }
    ratingsSummary {
      aggregateRating
      voteCount
    }
  }
}
```



- On the **Headers** tab
  - Make sure **Content-Type** is set to **application/json**
  - Add a new key **x-amzn-dataexchange-data-set-id** with a value of **4b1f47d86b35356cf8fb6f15cc758c0e**
  - Add a new key **x-amzn-dataexchange-revision-id** with a value of **4915c8e5e666a284124fc532ca8fbbe2**
  - Add a new key **x-amzn-dataexchange-asset-id** with a value of **f05f6f7ca415c8be7341f95bf1db34c5**

Now you can press **Send** and you should see a response similar to the below.

```json
{
    "data": {
        "title": {
            "titleText": {
                "text": "The Matrix"
            },
            "ratingsSummary": {
                "aggregateRating": 8.7,
                "voteCount": 1900035
            }
        }
    },
    "extensions": {
        "disclaimer": "Any use of IMDb Content accessed through this API is subject to
    }
}
```

Working locally in Scratch Pad. **Switch to a Workspace**

POST Test Query    +    •••          No Environment ∨

Scratchpad collection / **Test Query**          Save ∨    •••

POST ∨    https://api-fulfill.dataexchange.us-east-1.amazonaws.com/v1          **Send** ∨

Params    Authorization •    Headers (14)    Body •    Pre-request Script    Tests    Settings          **Cookies**

○ none    ○ form-data    ○ x-www-form-urlencoded    ○ raw    ○ binary    ● GraphQL    Auto-Fetch ∨    ⟳    Schema Fetched

**QUERY**          **GRAPHQL VARIABLES** ⓘ

```
 1  {
 2    title(id: "tt0133093") {
 3      titleText {
 4        text
 5      }
 6      ratingsSummary {
 7        aggregateRating
 8        voteCount
 9      }
10    }
11  }
12
```

```
 1
```

Body    Cookies    Headers (20)    Test Results          Status: **200 OK**    Time: **227 ms**    Size: **1.2 KB**    **Save Response** ∨

Pretty    Raw    Preview    Visualize    JSON ∨

```
 1  {
 2    "data": {
 3      "title": {
 4        "titleText": {
 5          "text": "The Matrix"
 6        },
 7        "ratingsSummary": {
 8          "aggregateRating": 8.7,
 9          "voteCount": 1900035
10        }
11      }
12    },
13    "extensions": {
14      "disclaimer": "Any use of IMDb Content accessed through this API is subject to any applicable license agreement between you and IMDb ..."
```

Find and Replace    Console          Runner    Trash