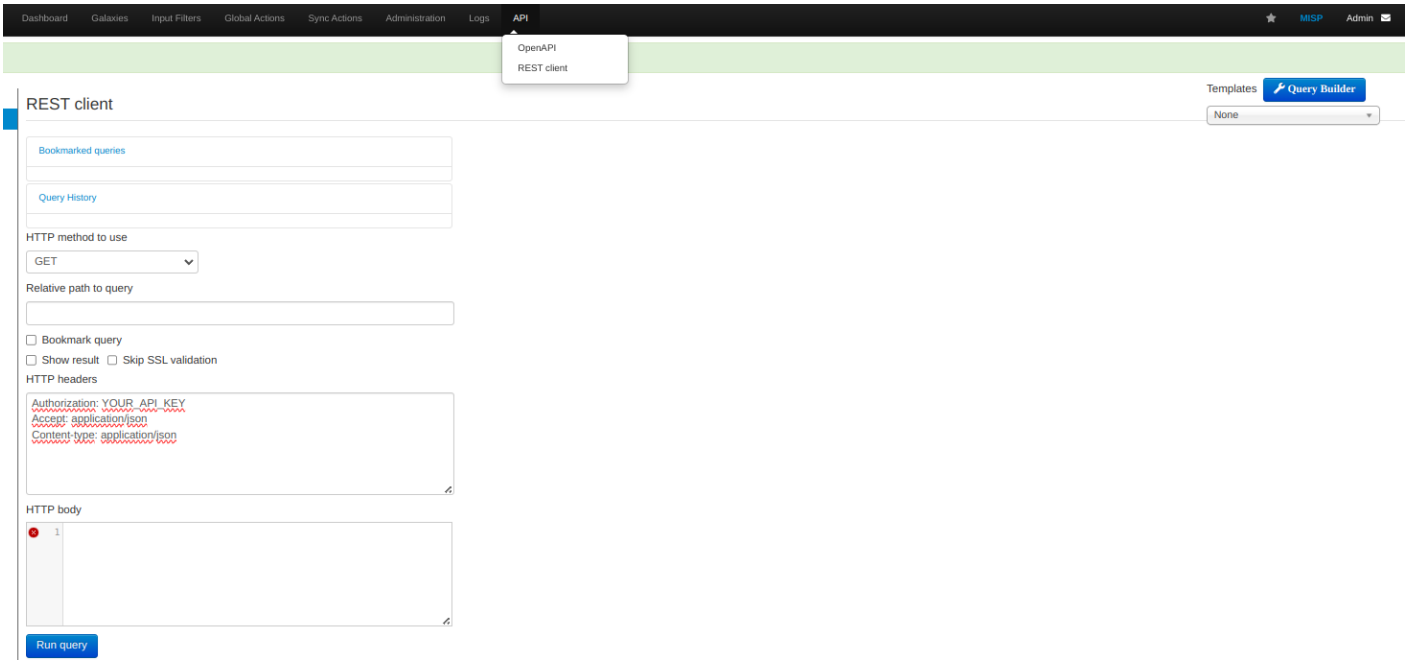
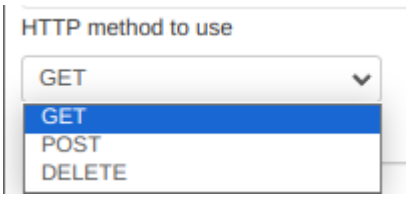
REST Client ile API İstekleri



- **Bookmarked Queries (Yer İmlenmiş Sorgular):**
  - Bookmarked Queries bölümü, daha önceden yapılan ve kullanıcı tarafından kaydedilmiş API sorgularını görüntülemek için kullanılır. Bu özellik, sıkça kullanılan sorguları kolayca erişilebilir bir konumda tutmak için kullanışlıdır.
- **Query History (Sorgu Geçmişi):**
  - Query History bölümü, daha önceden yapılan tüm API sorgularının geçmişini görüntülemek için kullanılır. Bu özellik, geçmişte yapılan sorguların takibini sağlar ve hata ayıklama veya tekrar kullanım için kullanıcıya referans oluşturur.
- **HTTP Method to Use (Kullanılacak HTTP Metodu):**
  - Bir API isteğinin hangi HTTP metodu kullanılarak yapılacağını belirler. Örneğin, GET, POST, PUT, DELETE gibi metotlardan biri seçilir. Bu, isteğin amacına ve API'nin desteklediği operasyonlara bağlıdır.



    - **GET**:
      - GET metodu, bir kaynağın okunması için kullanılır. Sunucudan belirtilen kaynağı almak için kullanılır. Örneğin, bir web sayfasını veya bir dosyayı almak için GET isteği yapılır.
    - **POST**:

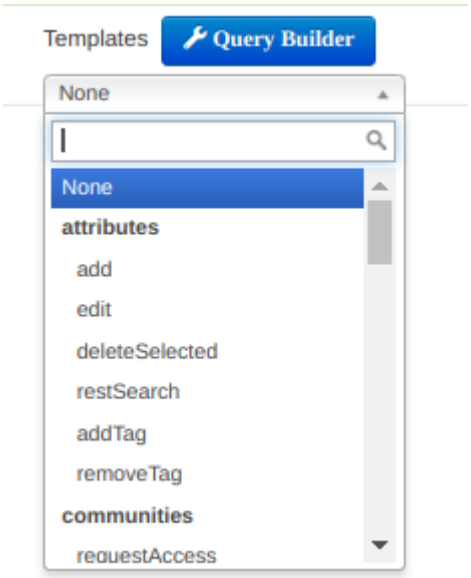
        - POST metodu, bir kaynağa veri göndermek için kullanılır. Genellikle bir form gönderirken veya sunucuya veri kaydetmek için kullanılır. Örneğin, bir form doldurulduğunda ve gönderildiğinde, bu bilgiler POST isteğiyle sunucuya iletilir.
    - **DELETE**:
        - DELETE metodu, bir kaynağı silmek veya kaldırmak için kullanılır. Belirtilen kaynağın sunucu tarafından silinmesini istemek için kullanılır. Örneğin, bir dosyanın veya kaydın silinmesi için DELETE isteği yapılır.
- **Relative Path to Query (Sorgulanacak İlgili Yol):**
    - Bir isteğin gönderileceği URL'nin kök URL (Root URL)'ye göre konumunu belirten kısaltılmış bir yol ifadesidir. Bu, isteğin hangi endpoint'e yönlendirileceğini belirlemek için kullanılır. Göreceli yol (Relative Path), isteğin tam URL'sini oluşturmak için kök URL ile birleştirilir ve istenen endpoint veya kaynağın konumunu belirtir.
- **Bookmark Query (Sorguyu Yer İmleri Ekle):**
    - Bookmark Query özelliği, işaretlendiği takdirde oluşturulan bir sorgunun yer imlerine eklenmesini sağlar. Böylece, sıkça kullanılan veya önemli sorguların kolayca erişilebilir olmasını sağlar.
- **Show Result (Sonucu Göster):**
    - Show Result, API isteğinin sonucunun görüntülenmesini sağlar. işaretlendiği takdirde isteğin başarıyla tamamlanıp tamamlanmadığını ve alınan yanıtın içeriğini kullanıcıya gösterir.
- **Skip SSL Validation (SSL Doğrulamasını Atla):**
    - Skip SSL Validation özelliği, SSL sertifikası doğrulamasının atlanmasını sağlar. Bu, güvenli olmayan bir ortamda veya test amaçlı kullanımlarda gerekebilir, ancak genellikle tavsiye edilmez.
- **HTTP Headers (HTTP Başlıkları):**
    - HTTP headers, bir HTTP isteği veya yanıtı iletilirken kullanılan başlık alanlarıdır. Bu başlıklar, isteğin veya yanıtın ne olduğunu, nasıl işlendiğini ve ne tür veri içerdiğini belirtir. RESTful API'lerde, HTTP başlıkları önemli bilgiler içerebilir ve belirli işlevlerin gerçekleştirilmesini sağlar.

        - **"Authorization"** başlığı, bir API'ye yetkilendirme bilgilerini eklemek için kullanılır. Bu başlık, isteği gönderenin kimliğini doğrulamak ve yetkilendirilmiş erişim sağlamak için kullanılır.
        Örneğin, bir API'ye erişmek için gereken API anahtarını (API key) belirtmek için kullanılır:
        `Authorization: YOUR_API_KEY`

- **"Accept"** başlığı, istemci tarafından sunulan veri biçimini belirlemek için kullanılır. Sunucuya, istemcinin hangi medya türlerini kabul edebileceğini bildirmek için kullanılır. Bu, sunucunun yanıt olarak hangi medya türlerini gönderebileceğini belirler.
  Örneğin, istemci uygulama JSON formatında veri almak istiyorsa, aşağıdaki gibi bir başlık kullanır:
  `Accept: application/json`
- **"Content-Type"** başlığı, istemcinin sunucuya gönderdiği verinin türünü belirtmek için kullanılır. Sunucuya gönderilen verinin hangi medya türünde olduğunu belirtir. Örneğin, istemci bir POST, PUT veya DELETE isteği gönderirken veriyi JSON formatında kodladıysa, aşağıdaki gibi bir başlık kullanır:
  `Content-Type: application/json`

- **HTTP Body (HTTP Gövdesi):**
  - HTTP Body, isteğin gövdesine eklenen veri veya bilgileri sunucuya iletmektir. Bu veri, genellikle istemcinin sunucuya iletmek istediği kaynakların detaylarını veya işlem yapılacak verileri içerir.
    Örneğin, bir tehdit istihbaratı raporu oluşturmak için MISP API'ye bir POST isteği gönderirken, HTTP gövdesi bu raporun içeriğini taşır.
    Rapor JSON formatında gönderilmek istenirse, HTTP gövdesi aşağıdaki gibi bir yapıda olabilir:

```
{
 "title": "Örnek Rapor",
 "description": "Bu bir örnek rapordur.",
 "tags": ["tehdit", "analiz"],
 ...
}
```

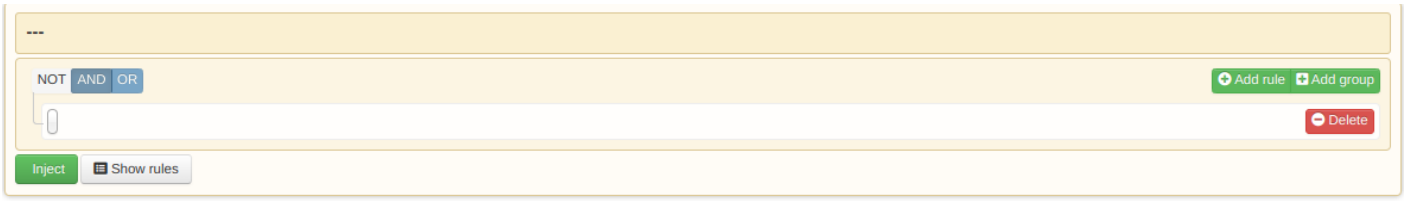


- **Templates (Şablonlar):**
  - Şablonlar, daha önceden tanımlanmış ve genellikle yaygın olarak kullanılan API isteklerinin önceden yapılandırılmış formatlarını içerir. Kullanıcılar, belirli bir şablonu seçerek, istek için gereken parametreleri doldurarak hızlıca bir API isteği

oluşturabilirler. Örneğin, "Event Search", "Attribute Creation" gibi şablonlar bulunabilir.

- **Query Builder (Sorgu Oluşturucu):**
  - Sorgu Oluşturucu, kullanıcıların bir API isteği için gerekli olan parametreleri adım adım seçmelerine olanak tanır. Kullanıcılar, belirli bir endpoint veya operasyon için gerekli olan parametreleri seçerek veya doldurarak isteklerini yapılandırabilirler. Kullanıcıların API'yi kullanırken gereken parametreleri hatırlamalarına veya manuel olarak yazmalarına gerek kalmadan kolayca istek oluşturmalarını sağlar.

> Sorgu oluşturucu butonuna tıklandığı zaman yeni bir alan açılır.



Kullanıcının daha karmaşık ve özelleştirilmiş sorgular oluşturmasına olanak tanıyan bir araçtır. Bu alanda kullanıcılar, isteklerini daha fazla filtrelemek veya belirli koşulları karşılayan verileri sorgulamak için kapsamlı sorgu kuralları oluşturabilirler.

Örneğin, bir kullanıcı belirli bir tarihten sonra oluşturulan etkinlikleri veya belirli bir tehdit seviyesine sahip olanları filtrelemek istiyorsa, query builder aracını kullanarak bu koşulları belirtebilirler. Ayrıca, bu araç sayesinde birden fazla koşulu birleştirerek daha karmaşık sorgular da oluşturulabilir.

Bu yeni alan, kullanıcılara API isteklerini daha esnek ve özelleştirilmiş bir şekilde oluşturma imkanı sunar ve istenen verilere daha doğru bir şekilde erişmelerini sağlar.

Revision #8
Created 11 April 2024 14:42:52 by İlayda Durlanık
Updated 14 April 2024 16:12:32 by İlayda Durlanık