

# EventFilter

No description

```
input EventFilter {  
  sender: SuiAddress  
  transactionDigest: String  
  emittingModule: String  
  eventType: String  
}
```

## Fields

`EventFilter.sender` • `SuiAddress` **scalar**

Filter down to events from transactions sent by this address.

`EventFilter.transactionDigest` • `String` **scalar**

Filter down to the events from this transaction (given by its transaction digest).

`EventFilter.emittingModule` • `String` **scalar**

Events emitted by a particular module. An event is emitted by a particular module if some function in the module is called by a PTB and emits an event.

Modules can be filtered by their package, or package::module. We currently do not support filtering by emitting module and event type at the same time so if both are provided in one filter, the query will error.

`EventFilter.eventType` • `String` **scalar**

This field is used to specify the type of event emitted.

Events can be filtered by their type's package, package::module, or their fully qualified type name.

Generic types can be queried by either the generic type name, e.g. `0x2::coin::Coin`, or by the full type name, such as `0x2::coin::Coin<0x2::sui::SUI>`.

## Member Of

[events](#) **query**



© 2025 SUI FOUNDATION | DOCUMENTATION DISTRIBUTED UNDER CC BY 4.0