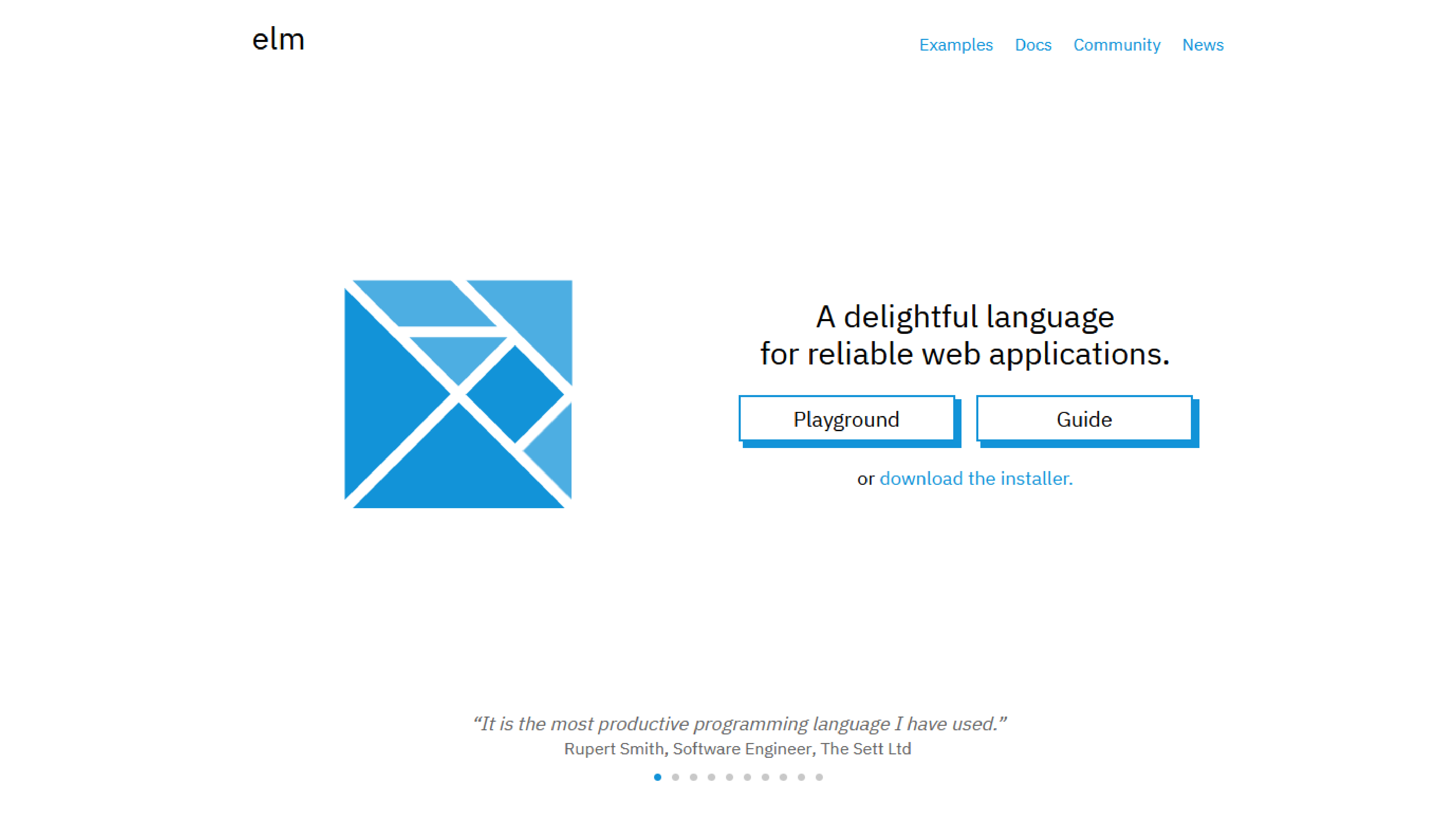
Elm SDK – your favourite SDK you’ll never use

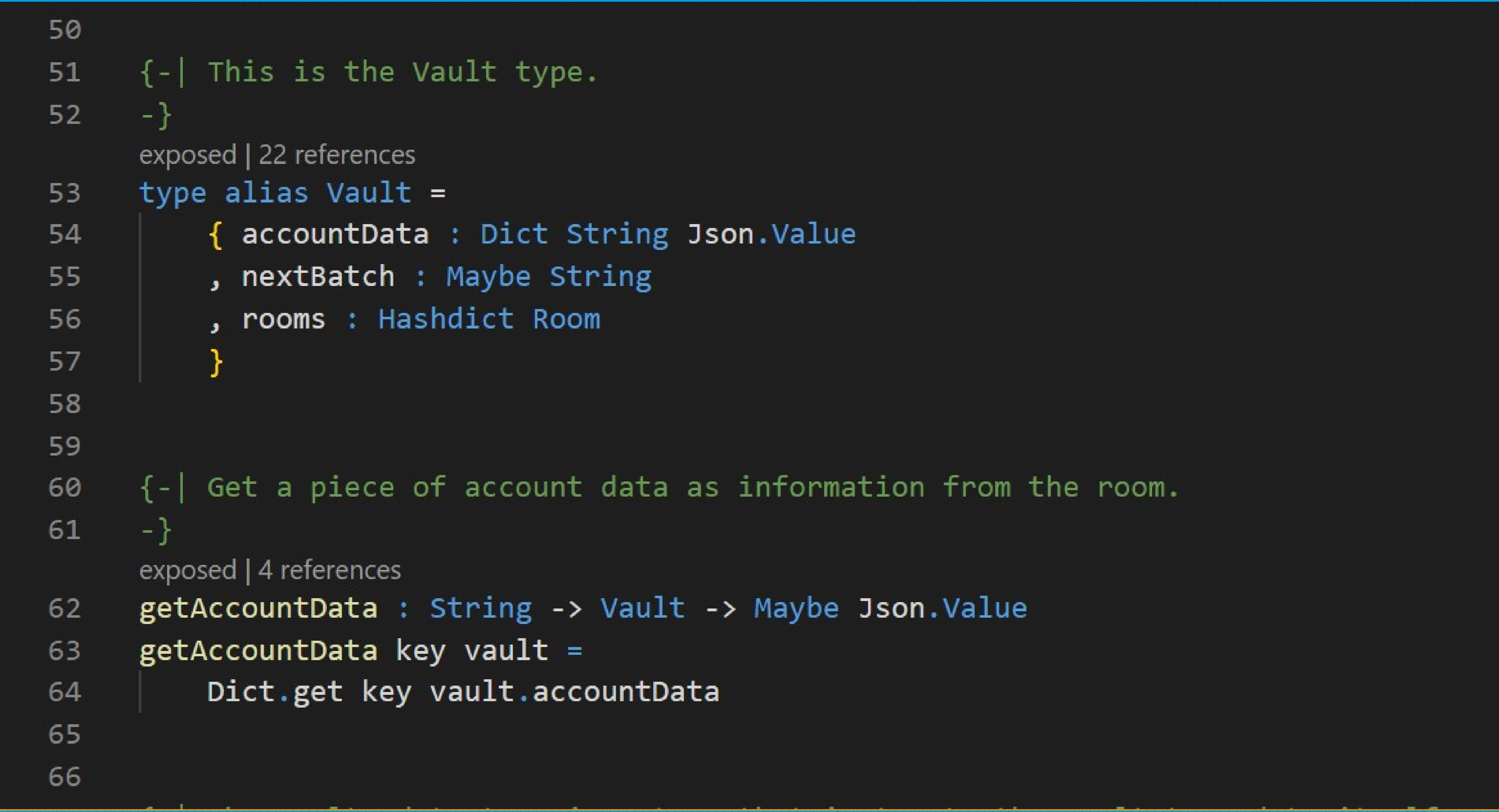
By Bram Noordstar



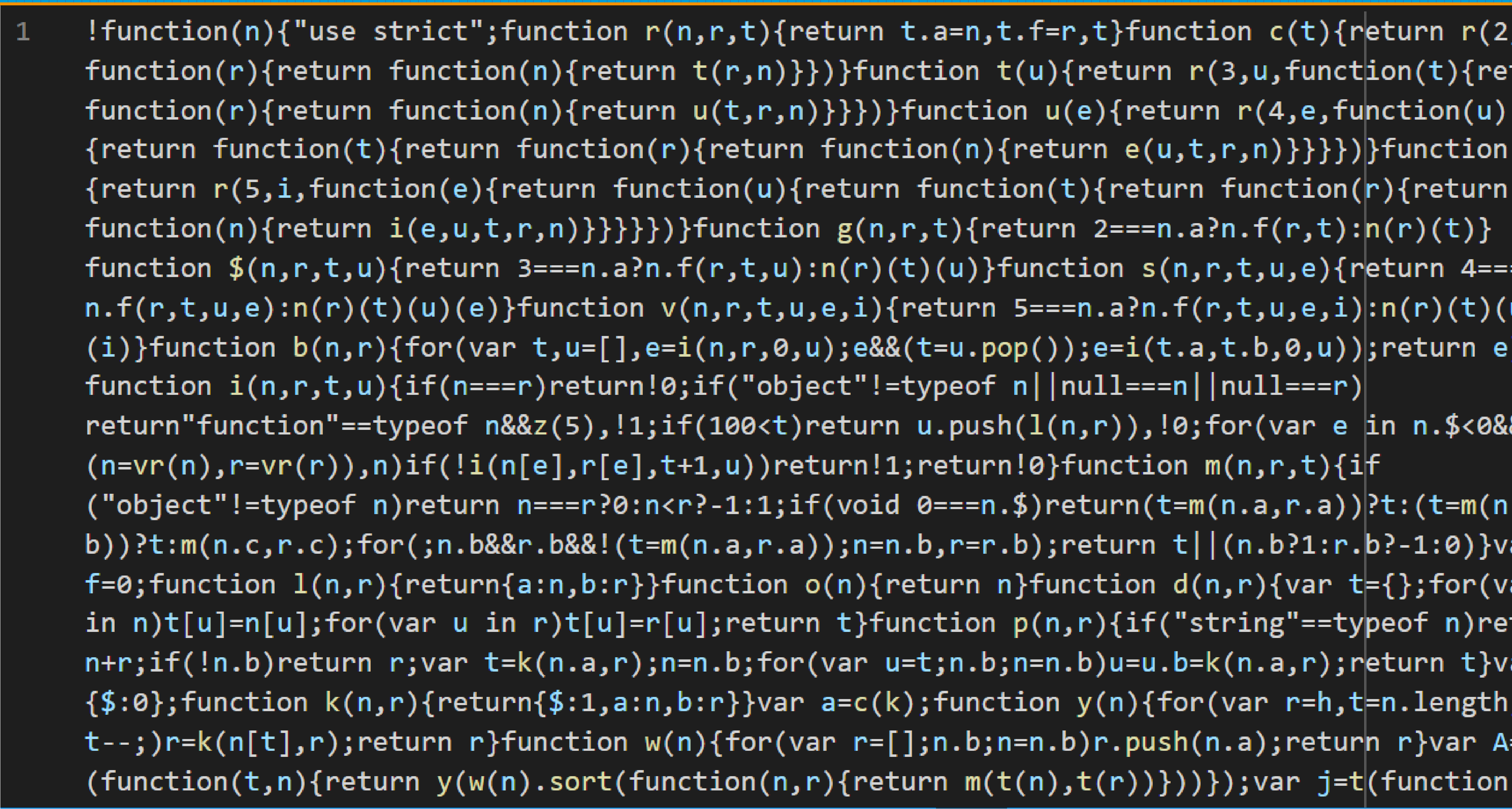
A delightful language
for reliable web applications.

or download the installer.

“It is the most productive programming language I have used.”
Rupert Smith, Software Engineer, The Sett Ltd

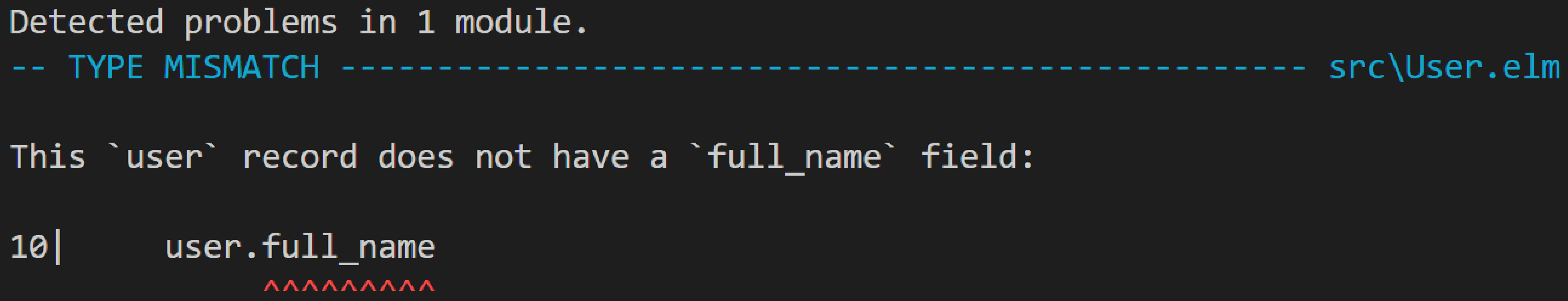


```elm

{-| This is the Vault type.
-}
exposed | 22 references
type alias Vault =
    { accountData : Dict String Json.Value
    , nextBatch : Maybe String
    , rooms : Hashdict Room
    }


{-| Get a piece of account data as information from the room.
-}
exposed | 4 references
getAccountData : String -> Vault -> Maybe Json.Value
getAccountData key vault =
    Dict.get key vault.accountData


```

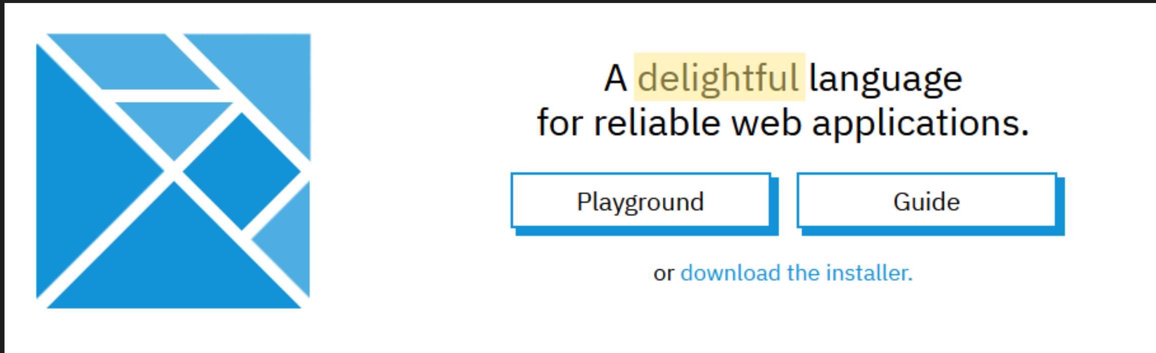


```
!function(n){"use strict";function r(n,r,t){return t.a=n,t.f=r,t}function c(t){return r(2,
function(r){return function(n){return t(r,n)}})}function t(u){return r(3,u,function(t){ret
function(r){return function(n){return u(t,r,n)}}})}function u(e){return r(4,e,function(u)
{return function(t){return function(r){return function(n){return e(u,t,r,n)}}}})}function
{return r(5,i,function(e){return function(u){return function(t){return function(r){return
function(n){return i(e,u,t,r,n)}}}}})}function g(n,r,t){return 2===n.a?n.f(r,t):n(r)(t)}
function $(n,r,t,u){return 3===n.a?n.f(r,t,u):n(r)(t)(u)}function s(n,r,t,u,e){return 4===
n.f(r,t,u,e):n(r)(t)(u)(e)}function v(n,r,t,u,e,i){return 5===n.a?n.f(r,t,u,e,i):n(r)(t)(
(i)}function b(n,r){for(var t,u=[],e=i(n,r,0,u);e&&(t=u.pop());e=i(t.a,t.b,0,u));return e
function i(n,r,t,u){if(n===r)return!0;if("object"!=typeof n||null===n||null===r)
return"function"==typeof n&&z(5),!1;if(100<t)return u.push(l(n,r)),!0;for(var e in n.$<0&
(n=vr(n),r=vr(r)),n)if(!i(n[e],r[e],t+1,u))return!1;return!0}function m(n,r,t){if
("object"!=typeof n)return n===r?0:n<r?-1:1;if(void 0===n.$)return(t=m(n.a,r.a))?t:(t=m(n
b))?t:m(n.c,r.c);for(;n.b&&r.b&&!(t=m(n.a,r.a));n=n.b,r=r.b);return t||(n.b?1:r.b?-1:0)}va
f=0;function l(n,r){return{a:n,b:r}}function o(n){return n}function d(n,r){var t={};for(va
in n)t[u]=n[u];for(var u in r)t[u]=r[u];return t}function p(n,r){if("string"==typeof n)ret
n+r;if(!n.b)return r;var t=k(n.a,r);n=n.b;for(var u=t;n.b;n=n.b)u=u.b=k(n.a,r);return t}va
{$:0};function k(n,r){return{$:1,a:n,b:r}}var a=c(k);function y(n){for(var r=h,t=n.length
t--;)r=k(n[t],r);return r}function w(n){for(var r=[];n.b;n=n.b)r.push(n.a);return r}var A
(function(t,n){return y(w(n).sort(function(n,r){return m(t(n),t(r))}))});var j=t(function
```

Delightful

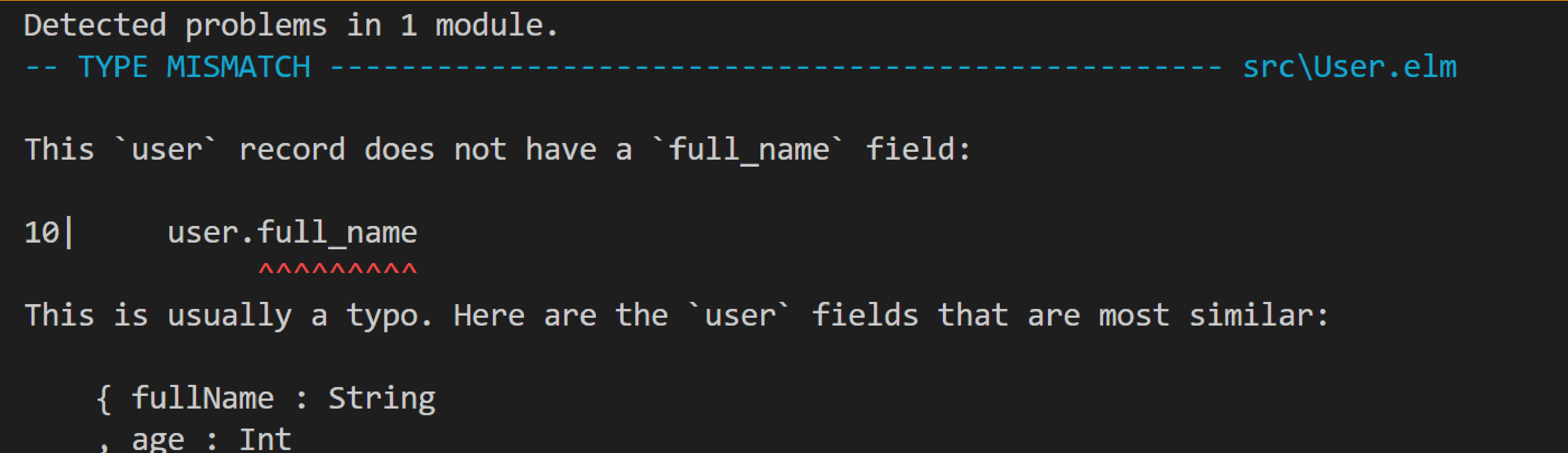


```
Detected problems in 1 module.
-- TYPE MISMATCH --------------------------------- src\User.elm

This `user` record does not have a `full_name` field:

10|      user.full_name
              ^^^^^^^^^
```

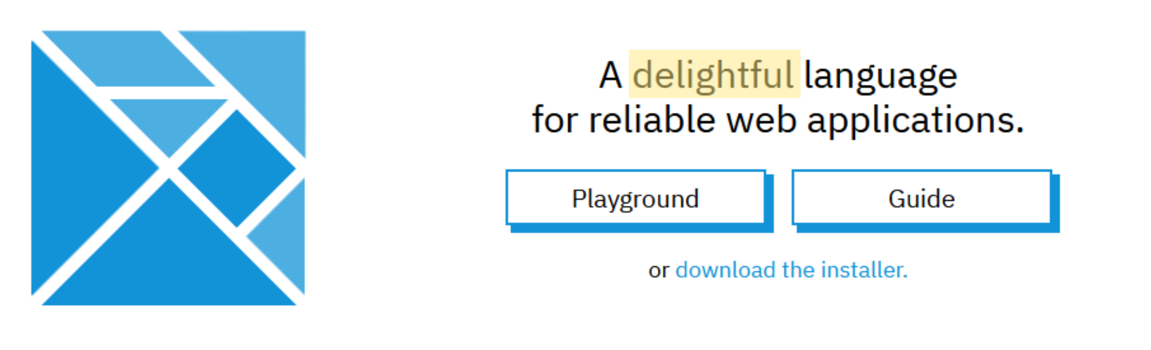


A delightful language for reliable web applications.

Delightful



```
Detected problems in 1 module.
-- TYPE MISMATCH -------------------------------- src\User.elm

This `user` record does not have a `full_name` field:

10|      user.full_name
              ^^^^^^^^^

This is usually a typo. Here are the `user` fields that are most similar:

    { fullName : String
    , age : Int
    }
```

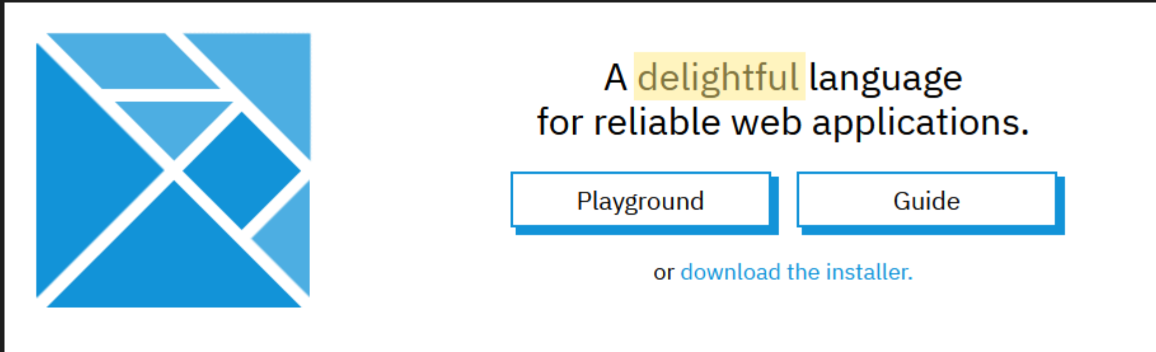


A delightful language
for reliable web applications.

or download the installer.

Delightful

```
Detected problems in 1 module.
-- TYPE MISMATCH ---------------------------------- src\User.elm

This `user` record does not have a `full_name` field:

10|     user.full_name
             ^^^^^^^^^
This is usually a typo. Here are the `user` fields that are most similar:

    { fullName : String
    , age : Int
    }

So maybe full_name should be fullName?
```

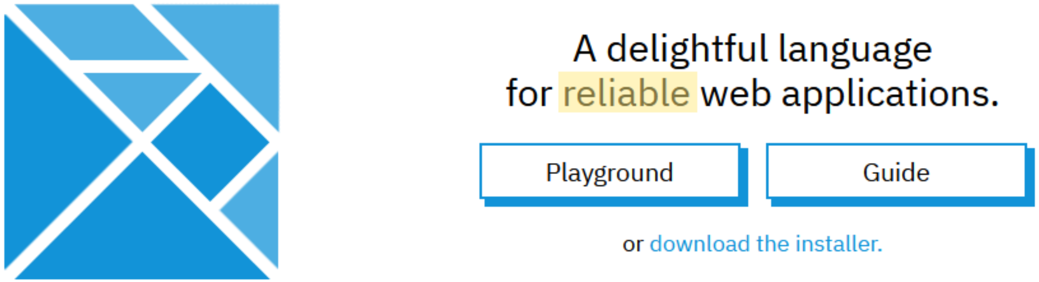


A delightful language
for reliable web applications.

or download the installer.

Reliable

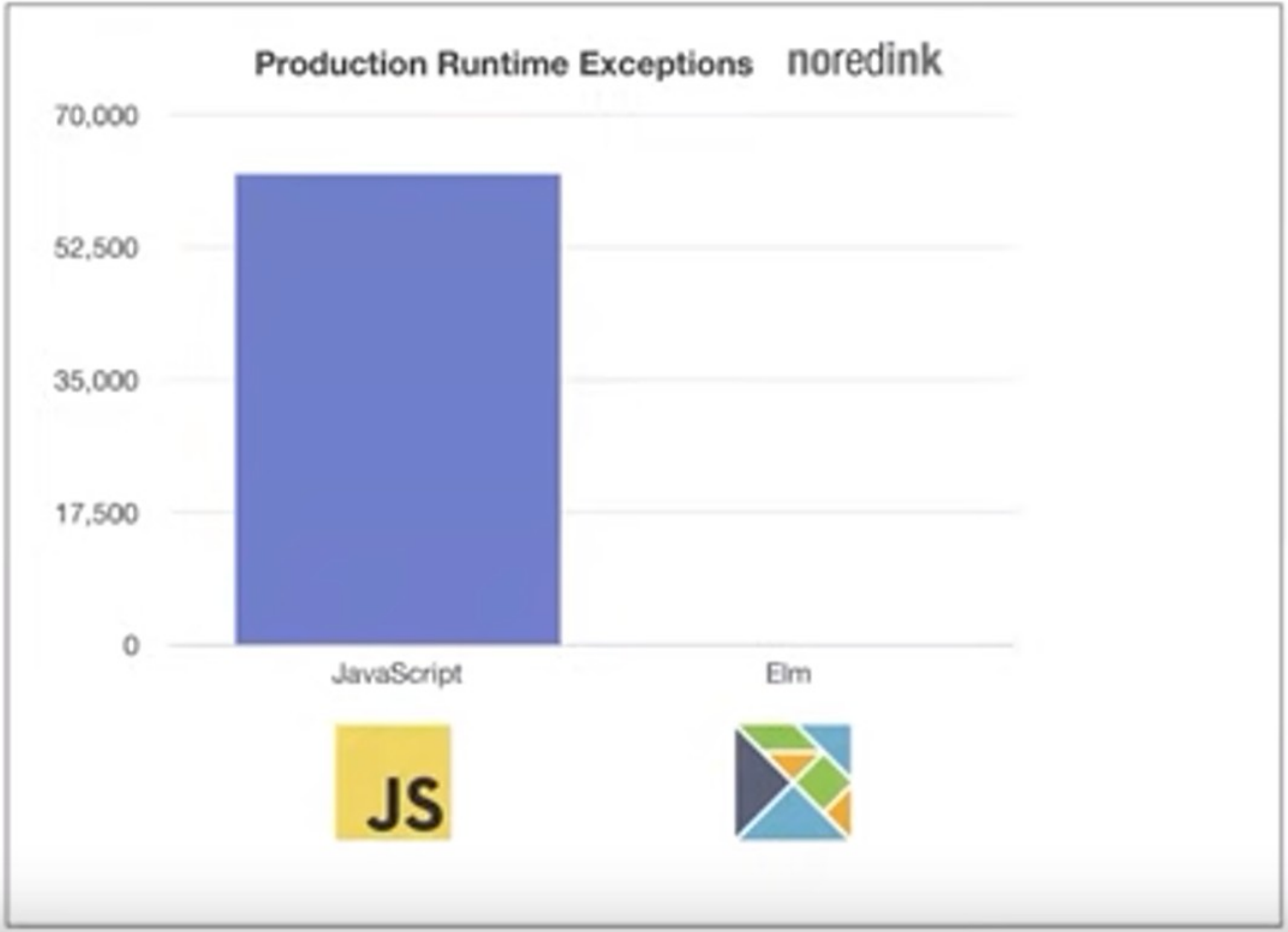
- Sound type system
- No “any” type
- No “billion dollar mistake”
- Immutable
- Pure
- No runtime exceptions



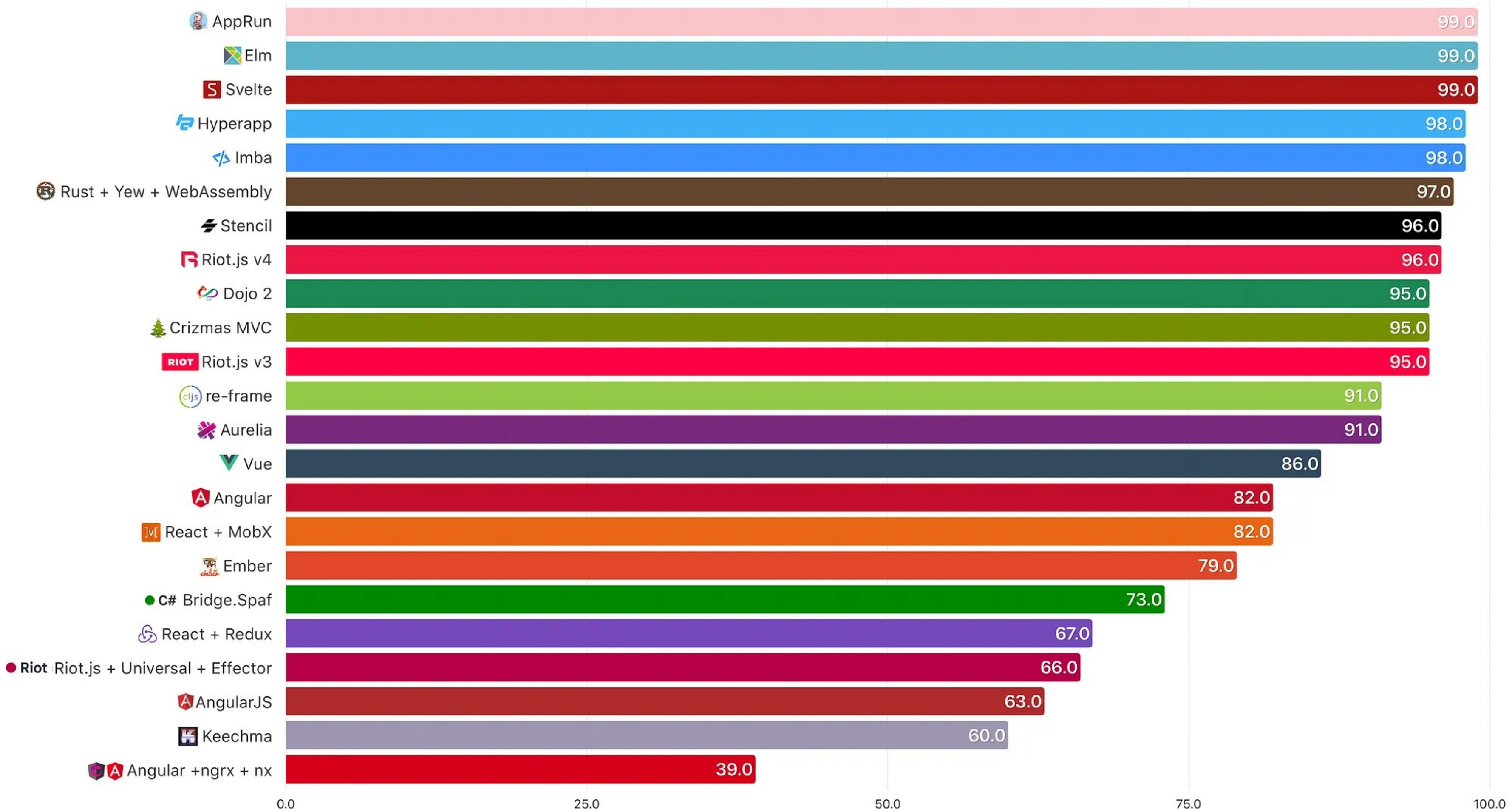
A delightful language for reliable web applications.

or download the installer.

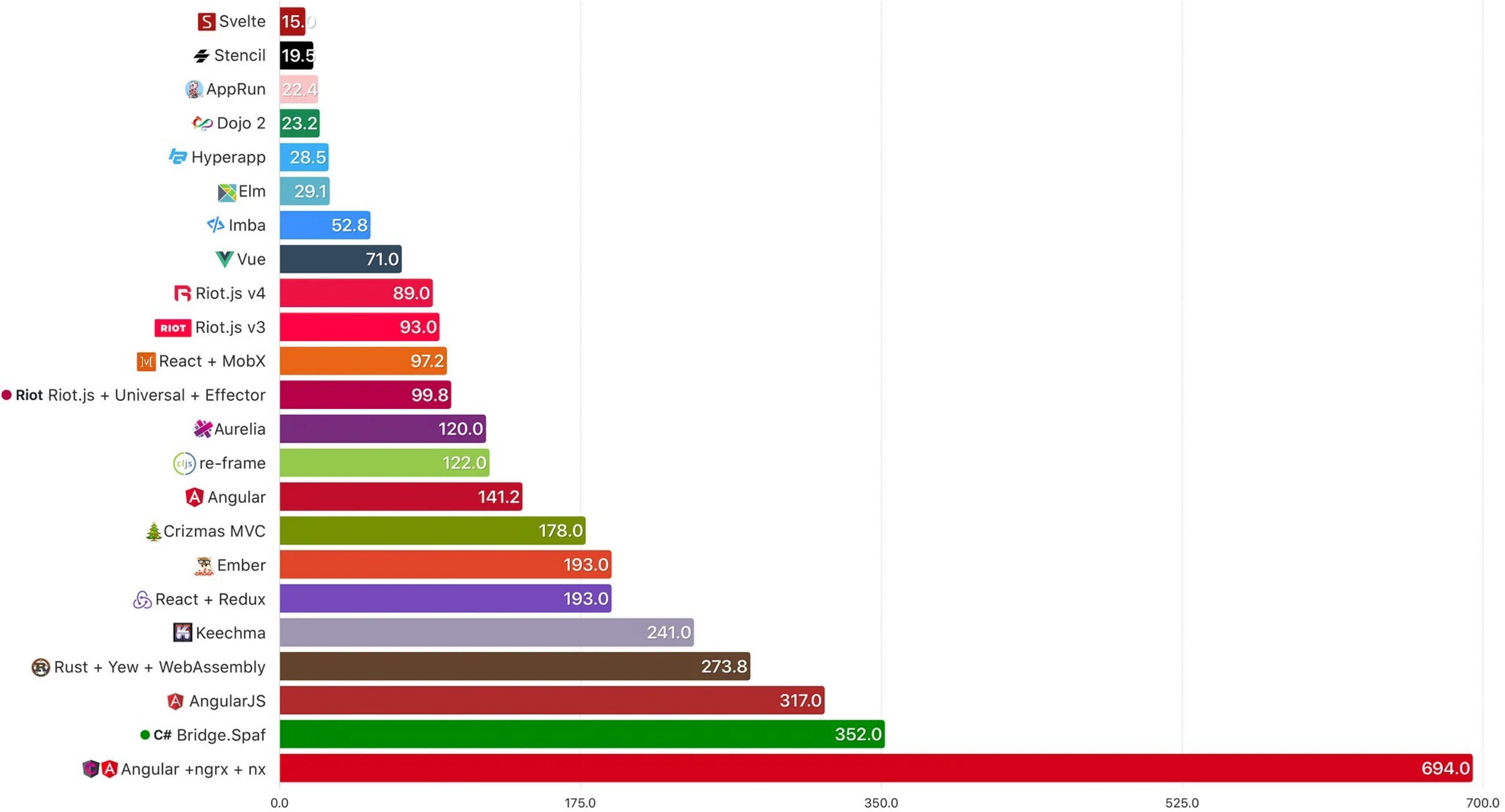
No runtime exceptions?

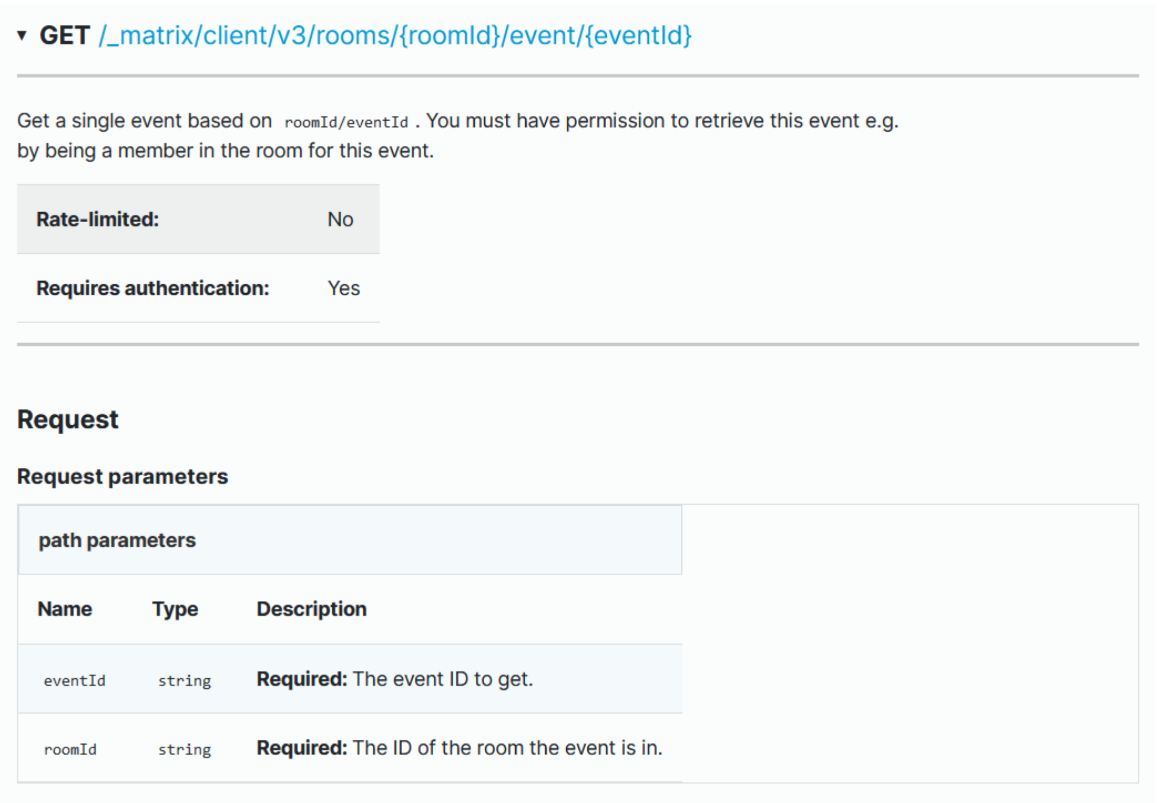


Performance



Package size





▾ **GET** /_matrix/client/v3/rooms/{roomId}/event/{eventId}

Get a single event based on `roomId/eventId`. You must have permission to retrieve this event e.g. by being a member in the room for this event.

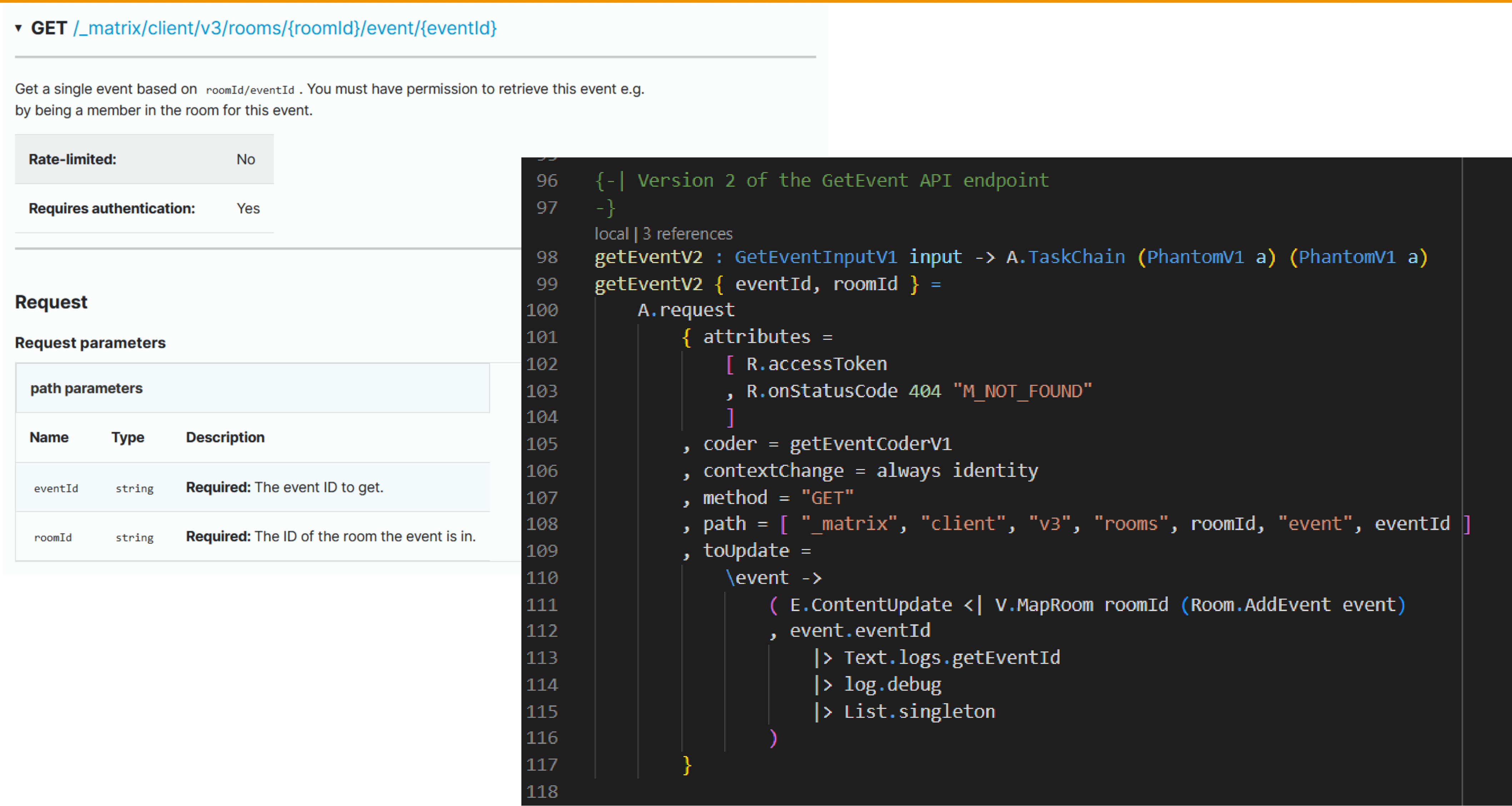
Rate-limited:	No
Requires authentication:	Yes

Request

Request parameters

path parameters

Name	Type	Description
`eventId`	`string`	**Required:** The event ID to get.
`roomId`	`string`	**Required:** The ID of the room the event is in.



▾ **GET** /_matrix/client/v3/rooms/{roomId}/event/{eventId}

Get a single event based on `roomId/eventId`. You must have permission to retrieve this event e.g. by being a member in the room for this event.

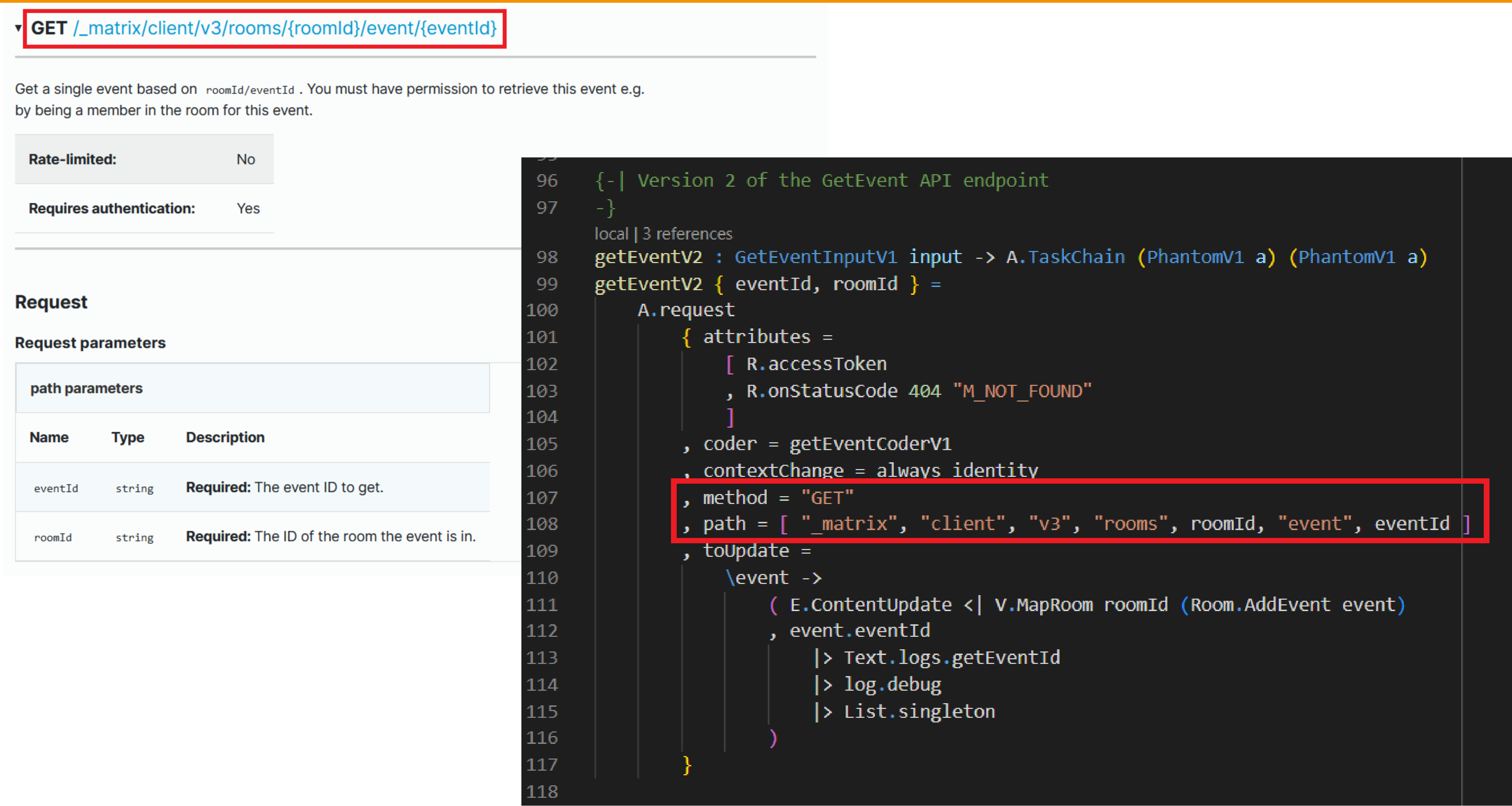
Rate-limited:	No
Requires authentication:	Yes

Request

Request parameters

path parameters		
Name	**Type**	**Description**
`eventId`	`string`	**Required:** The event ID to get.
`roomId`	`string`	**Required:** The ID of the room the event is in.

```
{-| Version 2 of the GetEvent API endpoint
-}
local | 3 references
getEventV2 : GetEventInputV1 input -> A.TaskChain (PhantomV1 a) (PhantomV1 a)
getEventV2 { eventId, roomId } =
    A.request
        { attributes =
            [ R.accessToken
            , R.onStatusCode 404 "M_NOT_FOUND"
            ]
        , coder = getEventCoderV1
        , contextChange = always identity
        , method = "GET"
        , path = [ "_matrix", "client", "v3", "rooms", roomId, "event", eventId ]
        , toUpdate =
            \event ->
                ( E.ContentUpdate <| V.MapRoom roomId (Room.AddEvent event)
                , event.eventId
                    |> Text.logs.getEventId
                    |> log.debug
                    |> List.singleton
                )
        }

```



▾ **GET** /_matrix/client/v3/rooms/{roomId}/event/{eventId}

Get a single event based on `roomId/eventId`. You must have permission to retrieve this event e.g. by being a member in the room for this event.

Rate-limited:	No
Requires authentication:	Yes

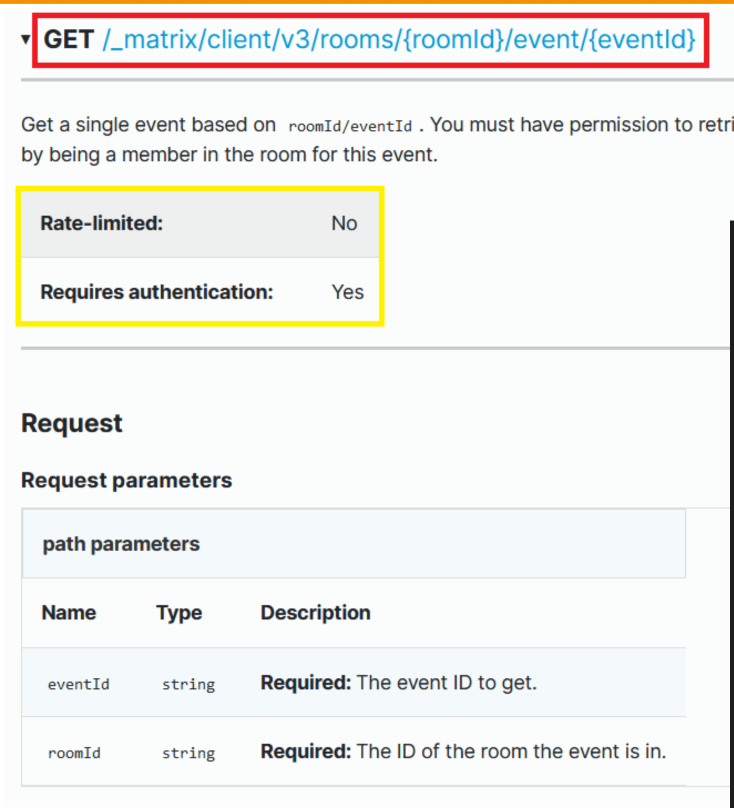
Request

Request parameters

path parameters

Name	Type	Description
`eventId`	`string`	**Required:** The event ID to get.
`roomId`	`string`	**Required:** The ID of the room the event is in.

```
{-| Version 2 of the GetEvent API endpoint
-}
local | 3 references
getEventV2 : GetEventInputV1 input -> A.TaskChain (PhantomV1 a) (PhantomV1 a)
getEventV2 { eventId, roomId } =
    A.request
        { attributes =
            [ R.accessToken
            , R.onStatusCode 404 "M_NOT_FOUND"
            ]
        , coder = getEventCoderV1
        , contextChange = always identity
        , method = "GET"
        , path = [ "_matrix", "client", "v3", "rooms", roomId, "event", eventId ]
        , toUpdate =
            \event ->
                ( E.ContentUpdate <| V.MapRoom roomId (Room.AddEvent event)
                , event.eventId
                    |> Text.logs.getEventId
                    |> log.debug
                    |> List.singleton
                )
        }
```



▾ GET /_matrix/client/v3/rooms/{roomId}/event/{eventId}

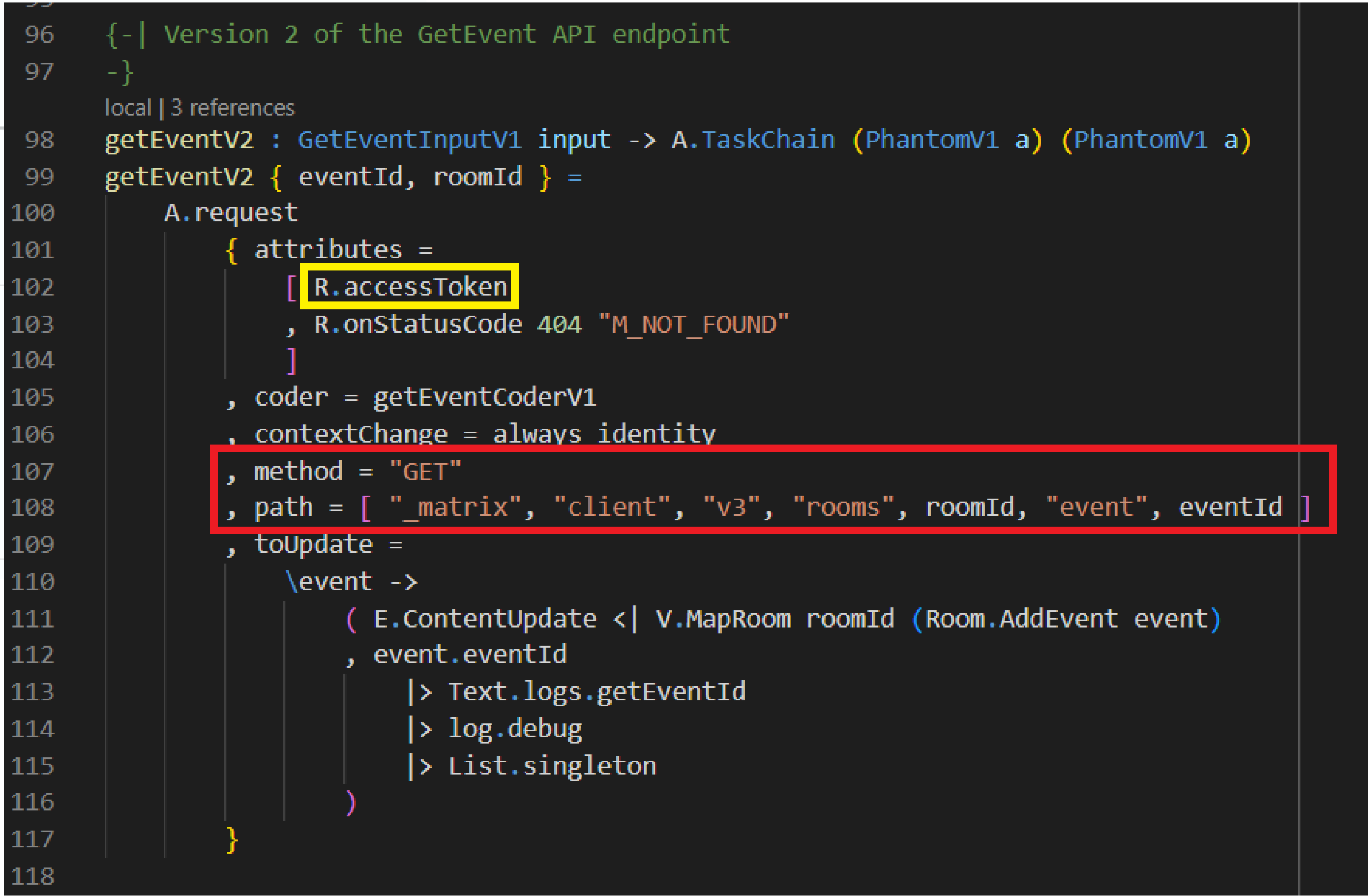
Get a single event based on `roomId/eventId`. You must have permission to retrieve this event e.g. by being a member in the room for this event.

Rate-limited:	No
Requires authentication:	Yes

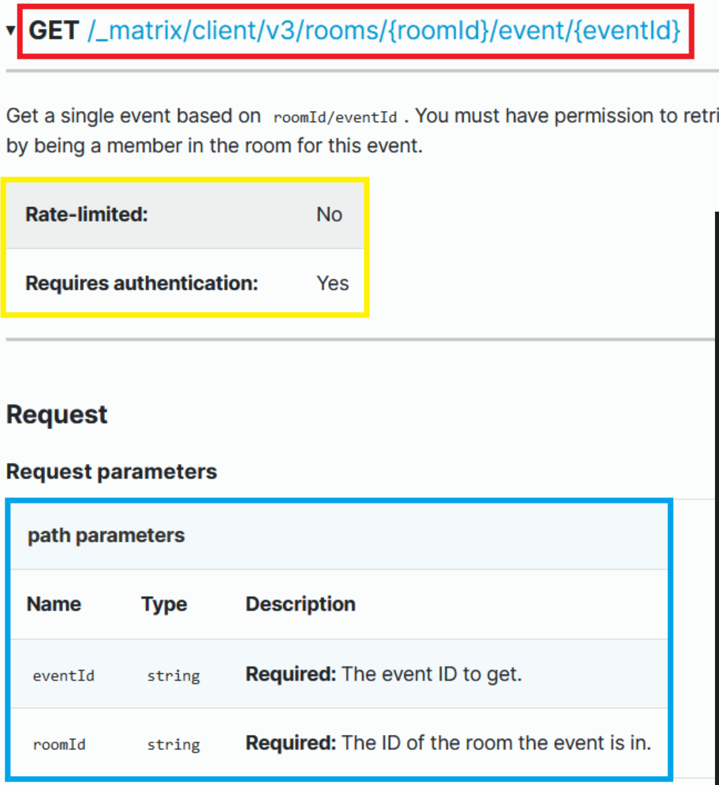
Request

Request parameters

path parameters		
Name	**Type**	**Description**
`eventId`	`string`	**Required:** The event ID to get.
`roomId`	`string`	**Required:** The ID of the room the event is in.



```
{-| Version 2 of the GetEvent API endpoint
-}
local | 3 references
getEventV2 : GetEventInputV1 input -> A.TaskChain (PhantomV1 a) (PhantomV1 a)
getEventV2 { eventId, roomId } =
  A.request
    { attributes =
      [ R.accessToken
      , R.onStatusCode 404 "M_NOT_FOUND"
      ]
    , coder = getEventCoderV1
    , contextChange = always identity
    , method = "GET"
    , path = [ "_matrix", "client", "v3", "rooms", roomId, "event", eventId ]
    , toUpdate =
      \event ->
        ( E.ContentUpdate <| V.MapRoom roomId (Room.AddEvent event)
        , event.eventId
          |> Text.logs.getEventId
          |> log.debug
          |> List.singleton
        )
    }
```



▾ GET /_matrix/client/v3/rooms/{roomId}/event/{eventId}

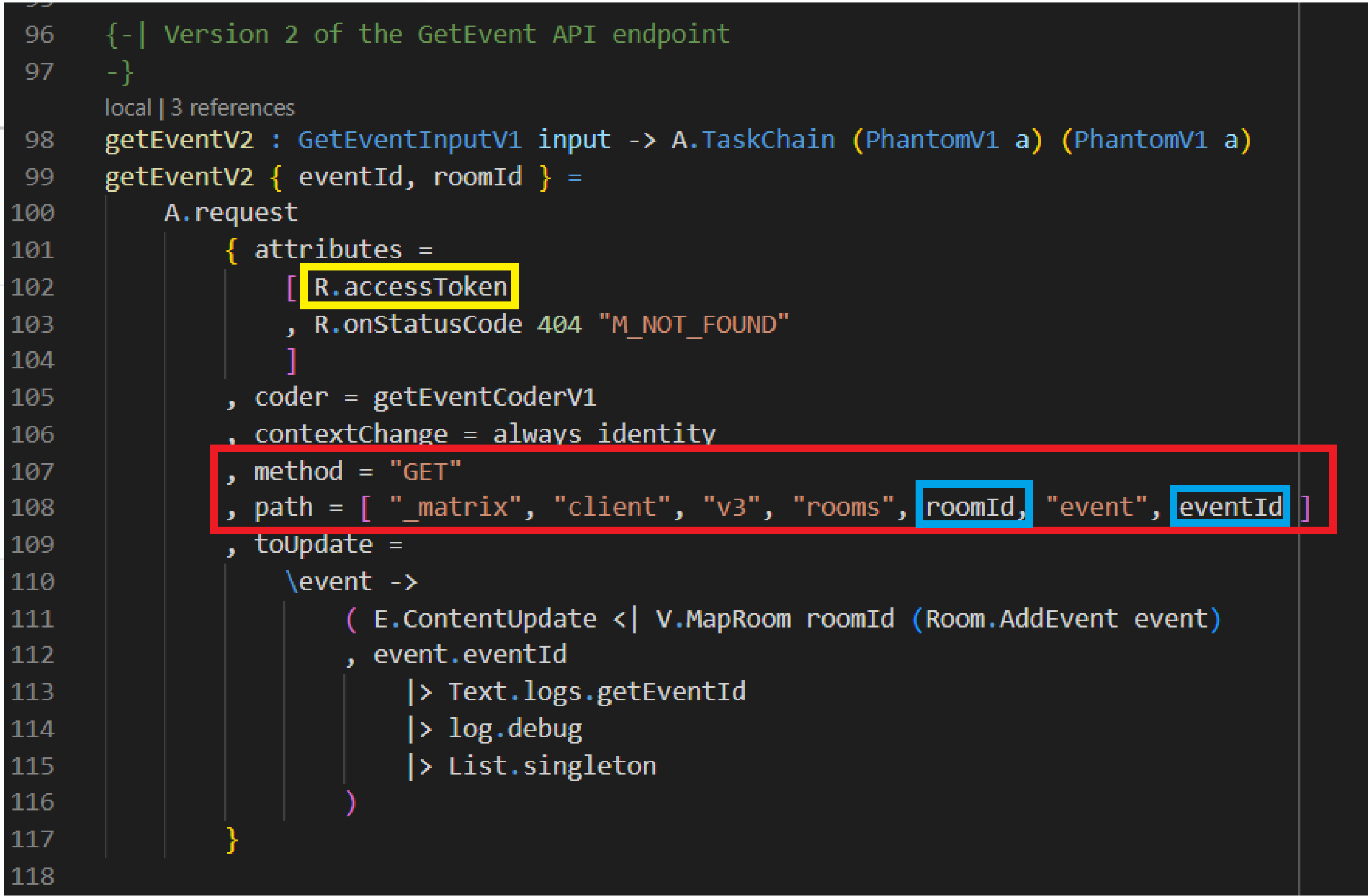
Get a single event based on `roomId/eventId`. You must have permission to retrieve this event e.g. by being a member in the room for this event.

Rate-limited:	No
Requires authentication:	Yes

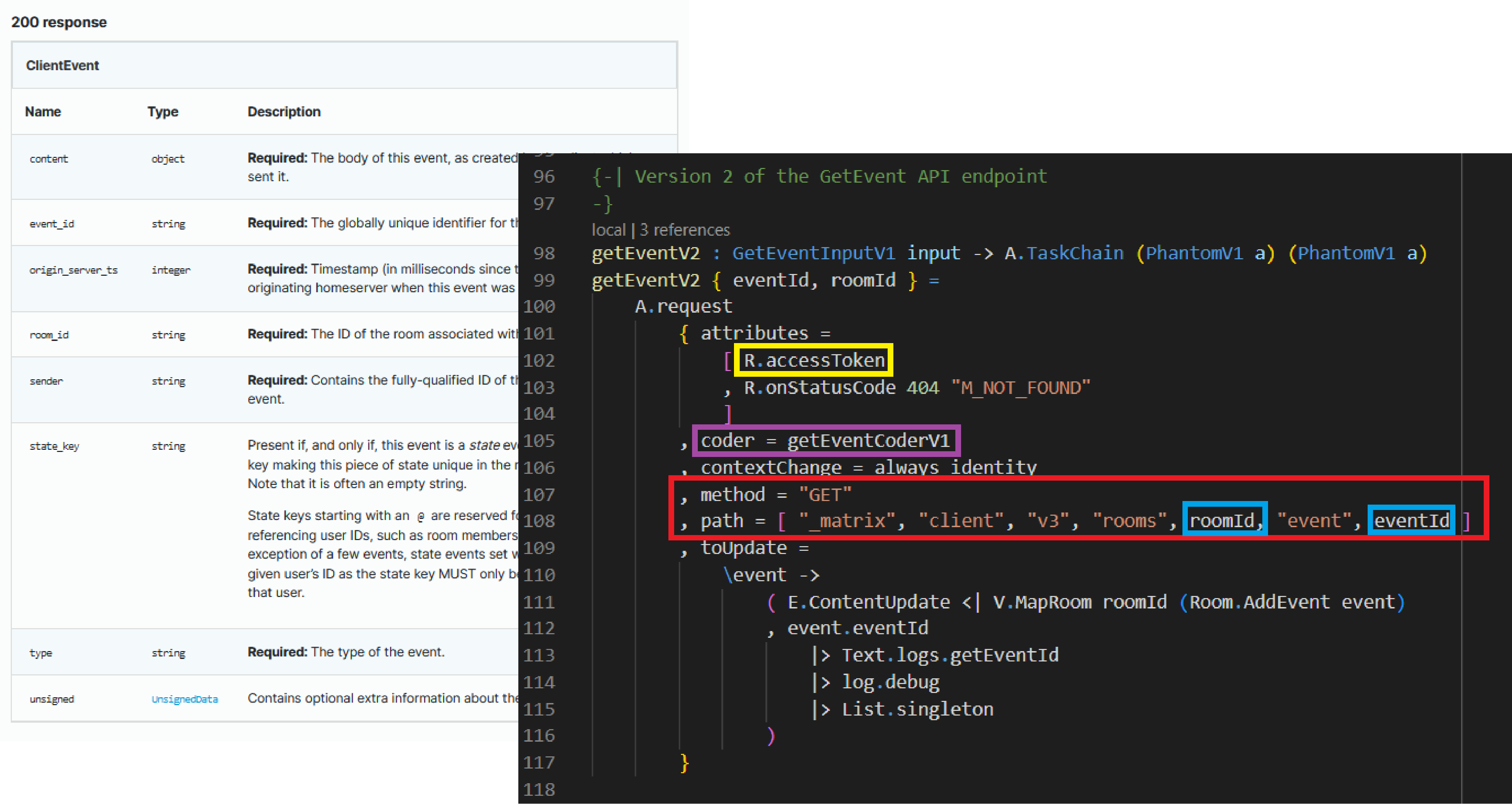
Request

Request parameters

path parameters		
Name	**Type**	**Description**
`eventId`	`string`	**Required:** The event ID to get.
`roomId`	`string`	**Required:** The ID of the room the event is in.



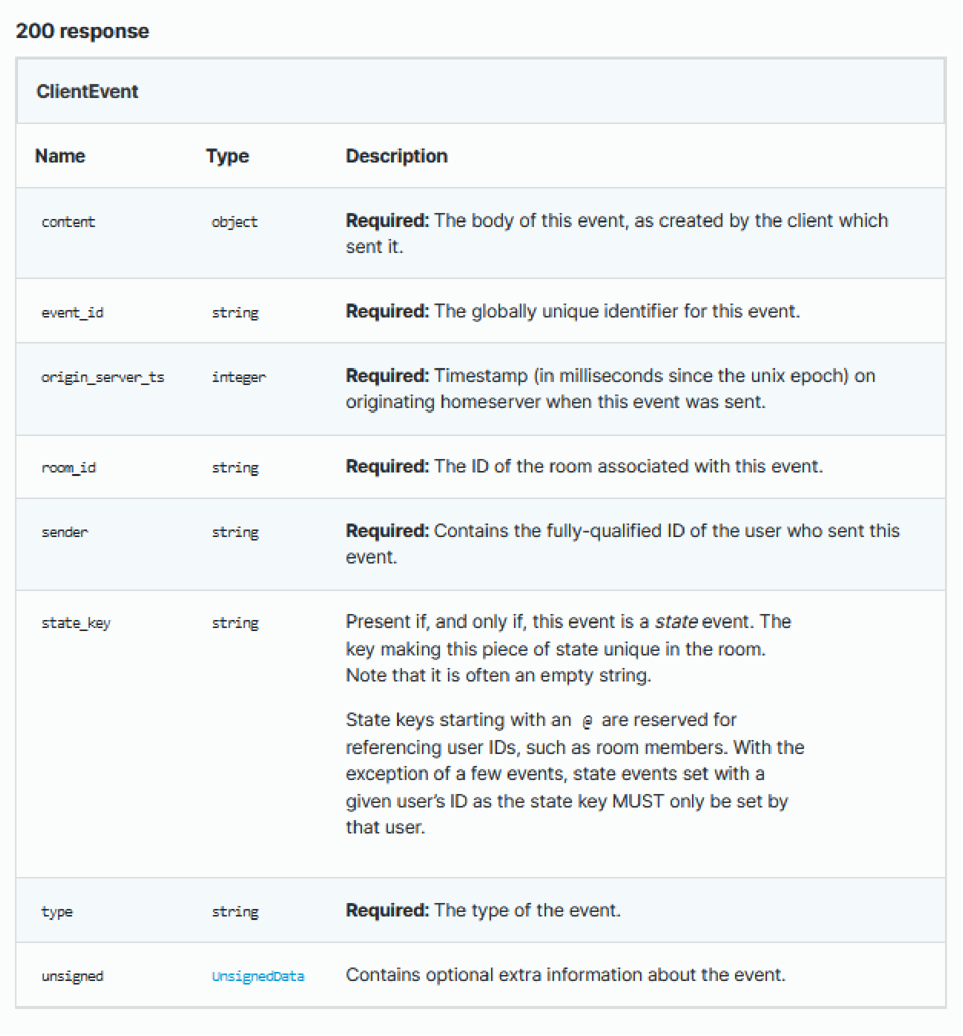
```
{-| Version 2 of the GetEvent API endpoint
-}
local | 3 references
getEventV2 : GetEventInputV1 input -> A.TaskChain (PhantomV1 a) (PhantomV1 a)
getEventV2 { eventId, roomId } =
    A.request
        { attributes =
            [ R.accessToken
            , R.onStatusCode 404 "M_NOT_FOUND"
            ]
        , coder = getEventCoderV1
        , contextChange = always identity
        , method = "GET"
        , path = [ "_matrix", "client", "v3", "rooms", roomId, "event", eventId ]
        , toUpdate =
            \event ->
                ( E.ContentUpdate <| V.MapRoom roomId (Room.AddEvent event)
                , event.eventId
                    |> Text.logs.getEventId
                    |> log.debug
                    |> List.singleton
                )
        }
```



200 response

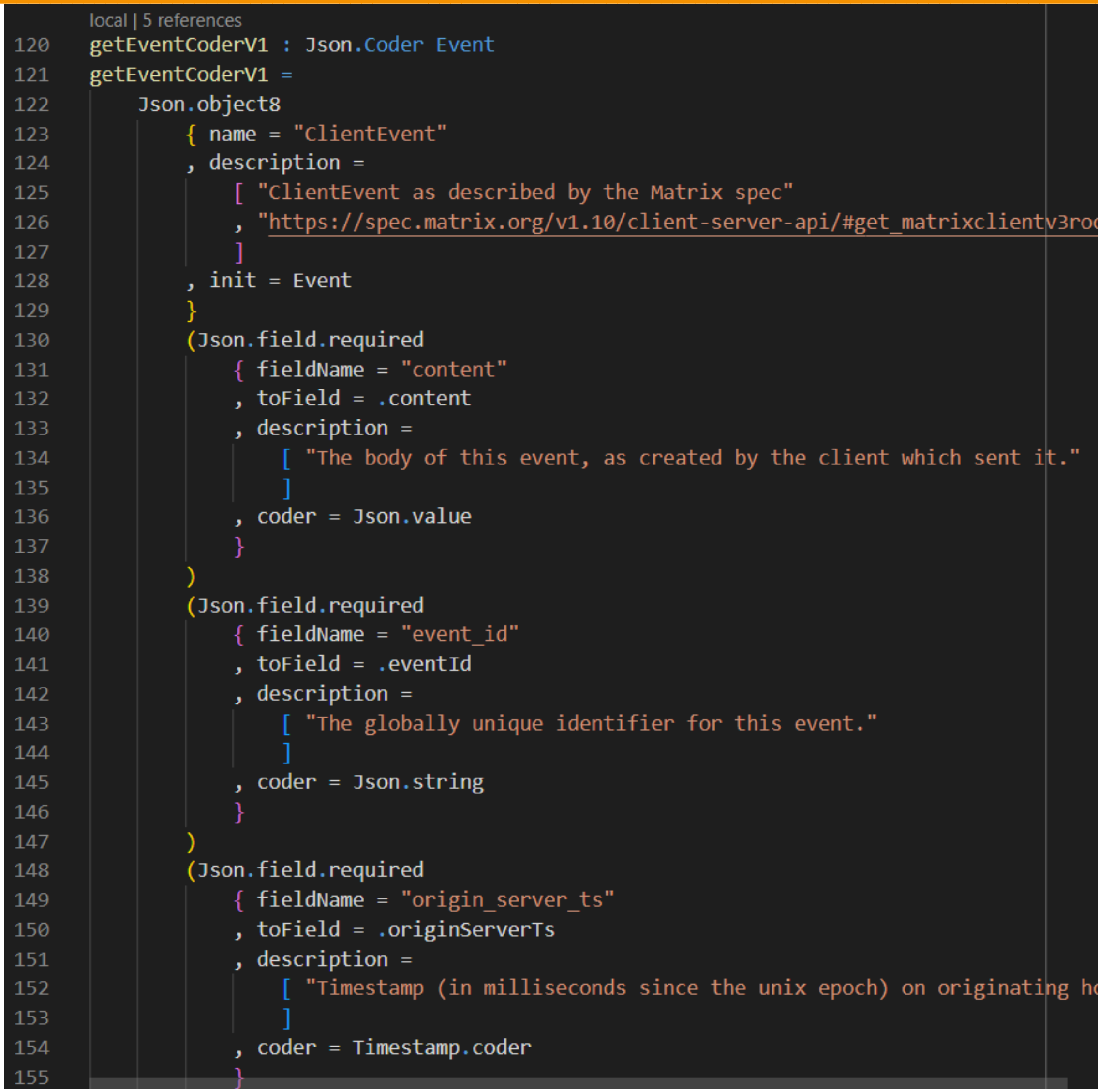
ClientEvent		
Name	Type	Description
content	object	**Required:** The body of this event, as created sent it.
event_id	string	**Required:** The globally unique identifier for th
origin_server_ts	integer	**Required:** Timestamp (in milliseconds since t originating homeserver when this event was
room_id	string	**Required:** The ID of the room associated wit
sender	string	**Required:** Contains the fully-qualified ID of th event.
state_key	string	Present if, and only if, this event is a *state* ev key making this piece of state unique in the r Note that it is often an empty string. State keys starting with an `@` are reserved fo referencing user IDs, such as room members exception of a few events, state events set w given user's ID as the state key MUST only b that user.
type	string	**Required:** The type of the event.
unsigned	UnsignedData	Contains optional extra information about the

```
{-| Version 2 of the GetEvent API endpoint
-}
local | 3 references
getEventV2 : GetEventInputV1 input -> A.TaskChain (PhantomV1 a) (PhantomV1 a)
getEventV2 { eventId, roomId } =
    A.request
        { attributes =
            [ R.accessToken
            , R.onStatusCode 404 "M_NOT_FOUND"
            ]
        , coder = getEventCoderV1
        , contextChange = always identity
        , method = "GET"
        , path = [ "_matrix", "client", "v3", "rooms", roomId, "event", eventId ]
        , toUpdate =
            \event ->
                ( E.ContentUpdate <| V.MapRoom roomId (Room.AddEvent event)
                , event.eventId
                    |> Text.logs.getEventId
                    |> log.debug
                    |> List.singleton
                )
        }

```

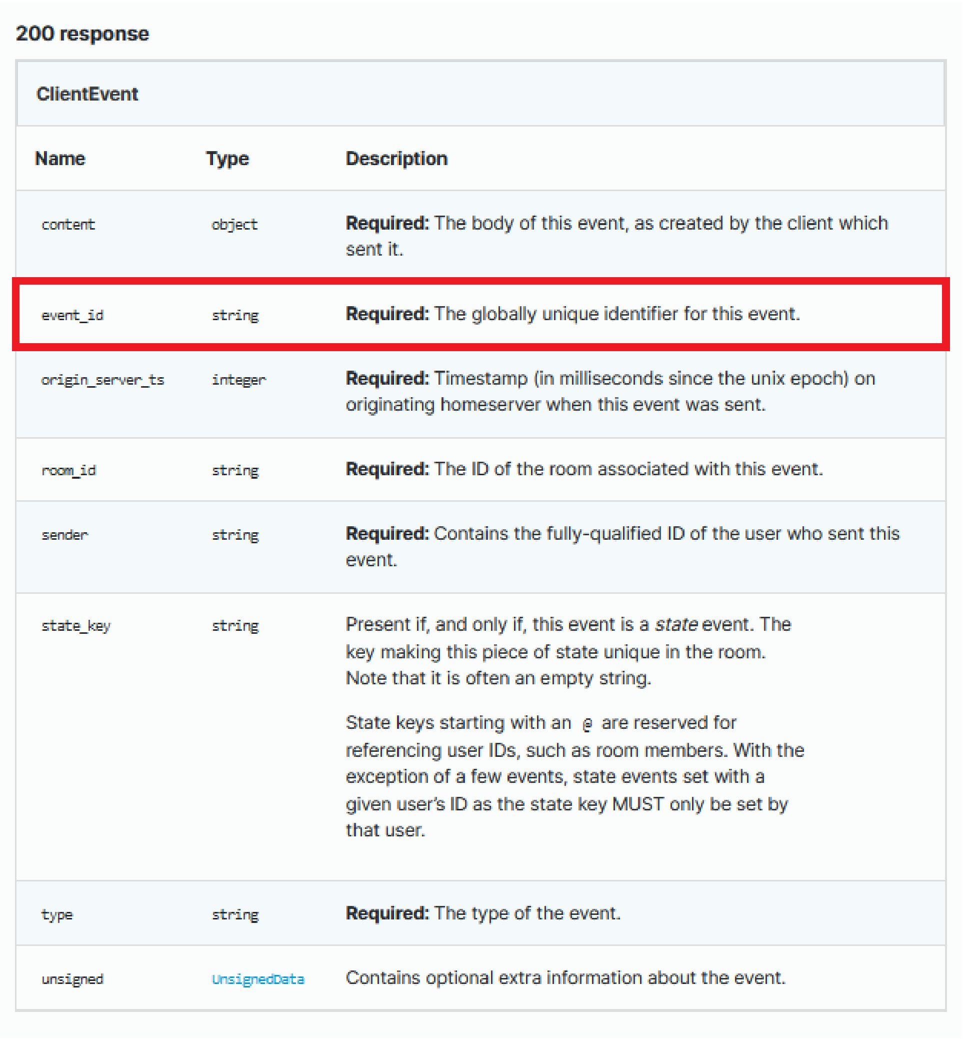


200 response

ClientEvent		
Name	**Type**	**Description**
content	object	**Required:** The body of this event, as created by the client which sent it.
event_id	string	**Required:** The globally unique identifier for this event.
origin_server_ts	integer	**Required:** Timestamp (in milliseconds since the unix epoch) on originating homeserver when this event was sent.
room_id	string	**Required:** The ID of the room associated with this event.
sender	string	**Required:** Contains the fully-qualified ID of the user who sent this event.
state_key	string	Present if, and only if, this event is a *state* event. The key making this piece of state unique in the room. Note that it is often an empty string. State keys starting with an `@` are reserved for referencing user IDs, such as room members. With the exception of a few events, state events set with a given user's ID as the state key MUST only be set by that user.
type	string	**Required:** The type of the event.
unsigned	UnsignedData	Contains optional extra information about the event.



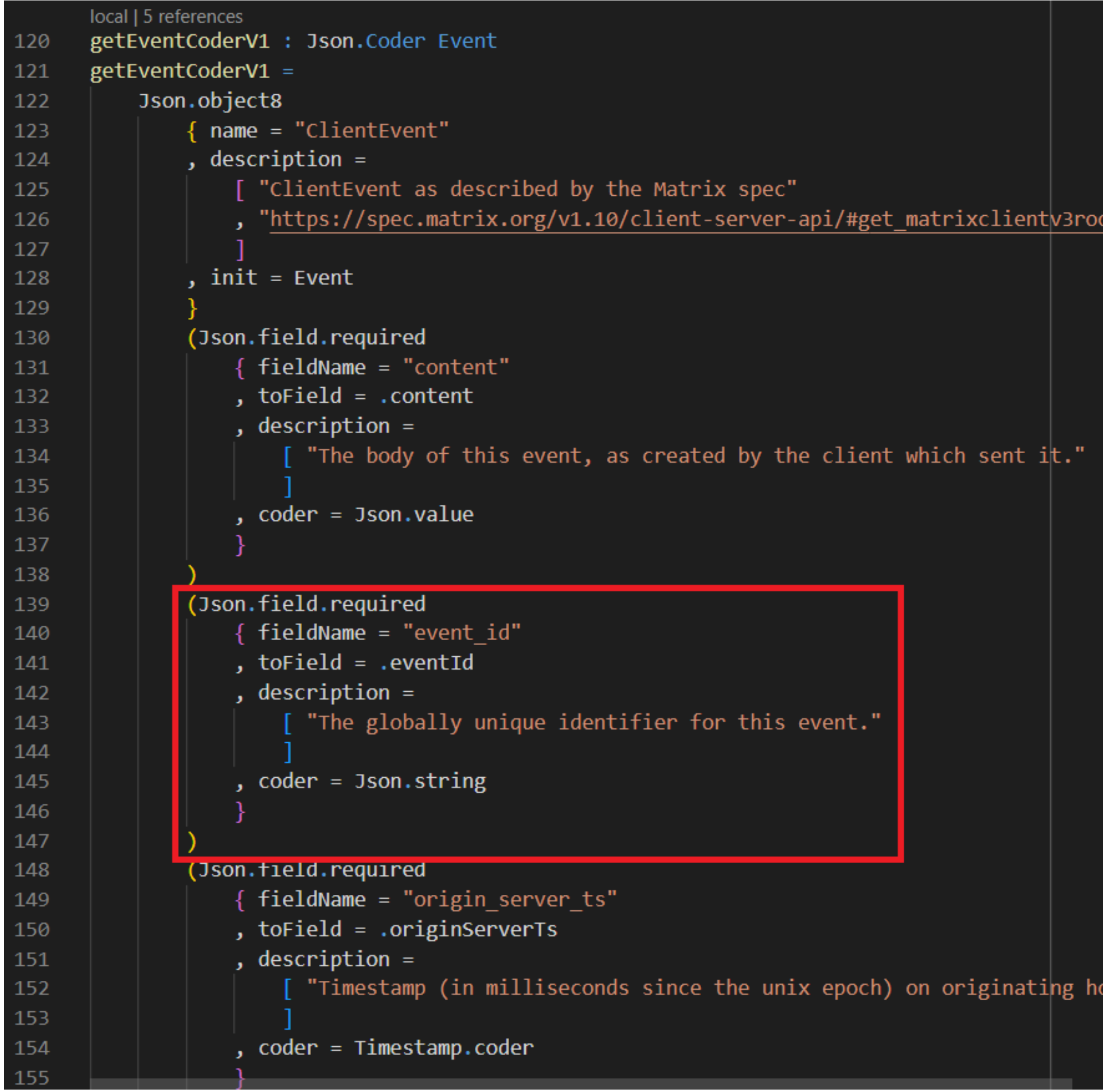
```
local | 5 references
getEventCoderV1 : Json.Coder Event
getEventCoderV1 =
    Json.object8
        { name = "ClientEvent"
        , description =
            [ "ClientEvent as described by the Matrix spec"
            , "https://spec.matrix.org/v1.10/client-server-api/#get_matrixclientv3roo
            ]
        , init = Event
        }
        (Json.field.required
            { fieldName = "content"
            , toField = .content
            , description =
                [ "The body of this event, as created by the client which sent it."
                ]
            , coder = Json.value
            }
        )
        (Json.field.required
            { fieldName = "event_id"
            , toField = .eventId
            , description =
                [ "The globally unique identifier for this event."
                ]
            , coder = Json.string
            }
        )
        (Json.field.required
            { fieldName = "origin_server_ts"
            , toField = .originServerTs
            , description =
                [ "Timestamp (in milliseconds since the unix epoch) on originating ho
                ]
            , coder = Timestamp.coder
            }
```



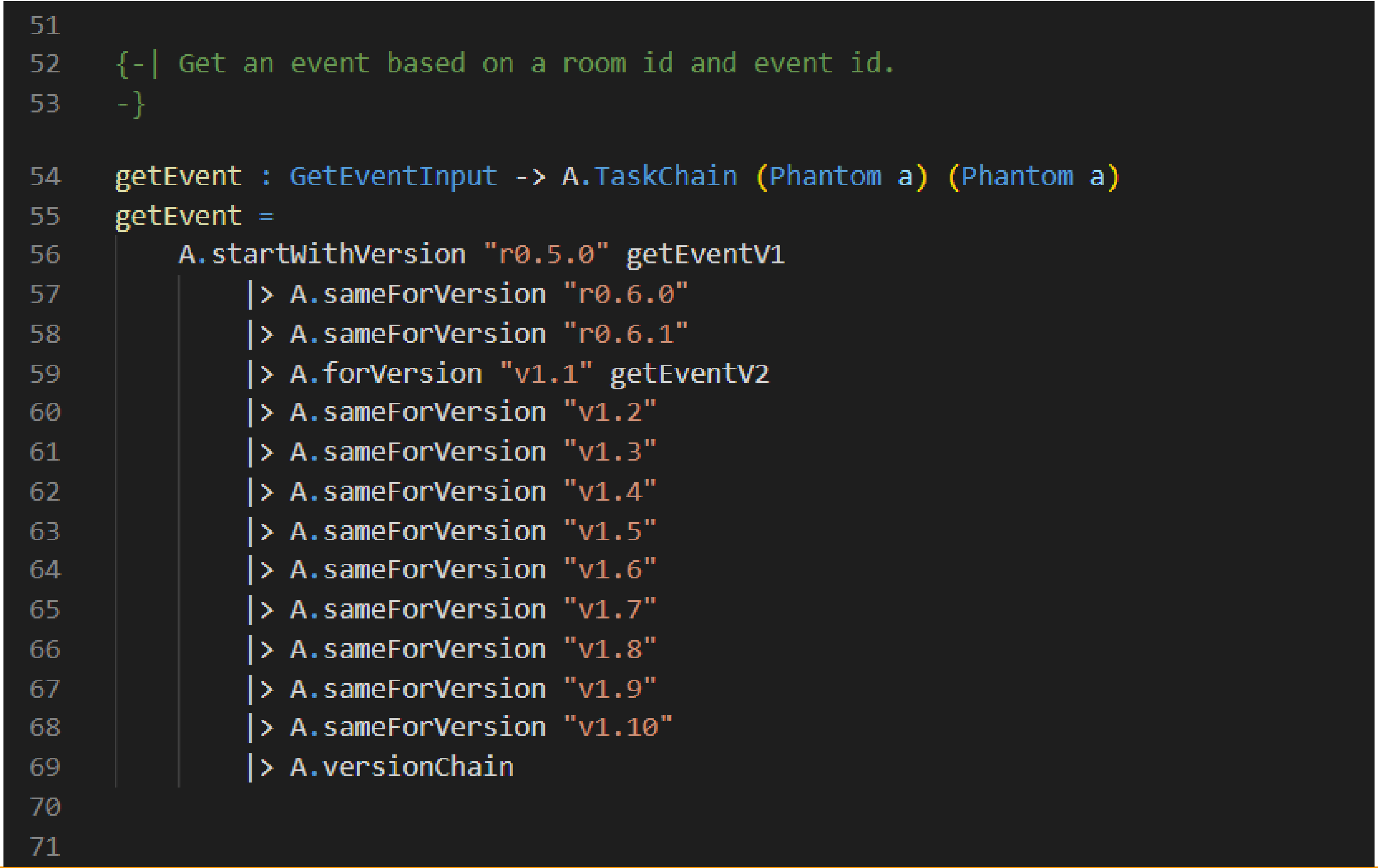
200 response

ClientEvent

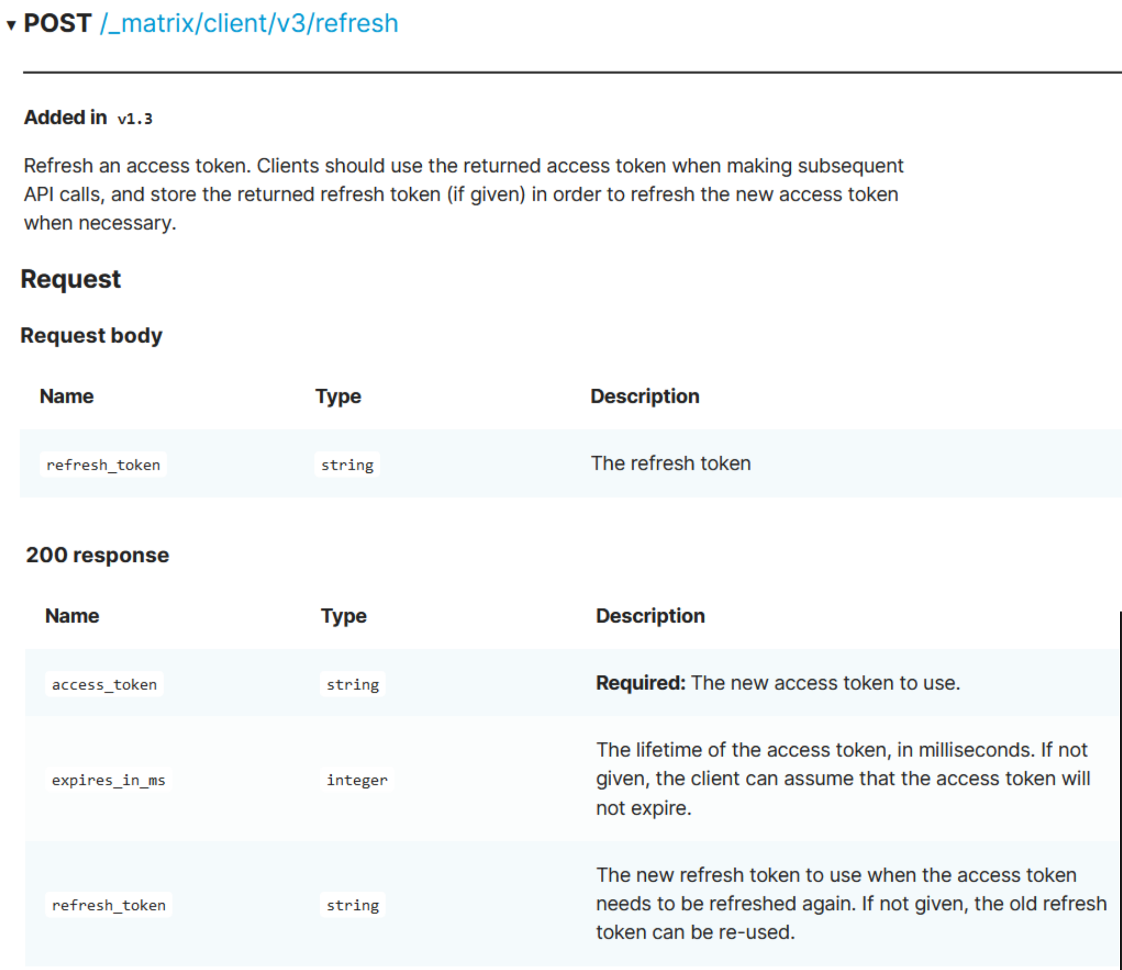
Name	Type	Description
content	object	**Required:** The body of this event, as created by the client which sent it.
event_id	string	**Required:** The globally unique identifier for this event.
origin_server_ts	integer	**Required:** Timestamp (in milliseconds since the unix epoch) on originating homeserver when this event was sent.
room_id	string	**Required:** The ID of the room associated with this event.
sender	string	**Required:** Contains the fully-qualified ID of the user who sent this event.
state_key	string	Present if, and only if, this event is a *state* event. The key making this piece of state unique in the room. Note that it is often an empty string. State keys starting with an `@` are reserved for referencing user IDs, such as room members. With the exception of a few events, state events set with a given user's ID as the state key MUST only be set by that user.
type	string	**Required:** The type of the event.
unsigned	UnsignedData	Contains optional extra information about the event.



```
getEventCoderV1 : Json.Coder Event
getEventCoderV1 =
    Json.object8
        { name = "ClientEvent"
        , description =
            [ "ClientEvent as described by the Matrix spec"
            , "https://spec.matrix.org/v1.10/client-server-api/#get_matrixclientv3roo
            ]
        , init = Event
        }
        (Json.field.required
            { fieldName = "content"
            , toField = .content
            , description =
                [ "The body of this event, as created by the client which sent it."
                ]
            , coder = Json.value
            }
        )
        (Json.field.required
            { fieldName = "event_id"
            , toField = .eventId
            , description =
                [ "The globally unique identifier for this event."
                ]
            , coder = Json.string
            }
        )
        (Json.field.required
            { fieldName = "origin_server_ts"
            , toField = .originServerTs
            , description =
                [ "Timestamp (in milliseconds since the unix epoch) on originating ho
                ]
            , coder = Timestamp.coder
            }
```



```
{-| Get an event based on a room id and event id.
-}
getEvent : GetEventInput -> A.TaskChain (Phantom a) (Phantom a)
getEvent =
    A.startWithVersion "r0.5.0" getEventV1
        |> A.sameForVersion "r0.6.0"
        |> A.sameForVersion "r0.6.1"
        |> A.forVersion "v1.1" getEventV2
        |> A.sameForVersion "v1.2"
        |> A.sameForVersion "v1.3"
        |> A.sameForVersion "v1.4"
        |> A.sameForVersion "v1.5"
        |> A.sameForVersion "v1.6"
        |> A.sameForVersion "v1.7"
        |> A.sameForVersion "v1.8"
        |> A.sameForVersion "v1.9"
        |> A.sameForVersion "v1.10"
        |> A.versionChain
```



▾ **POST** /_matrix/client/v3/refresh

Added in `v1.3`

Refresh an access token. Clients should use the returned access token when making subsequent API calls, and store the returned refresh token (if given) in order to refresh the new access token when necessary.

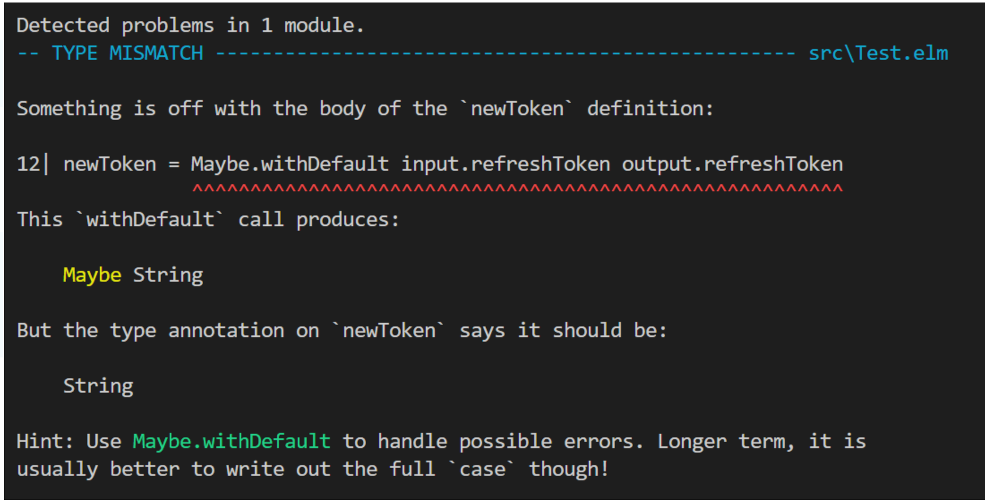
Request

Request body

Name	Type	Description
`refresh_token`	`string`	The refresh token

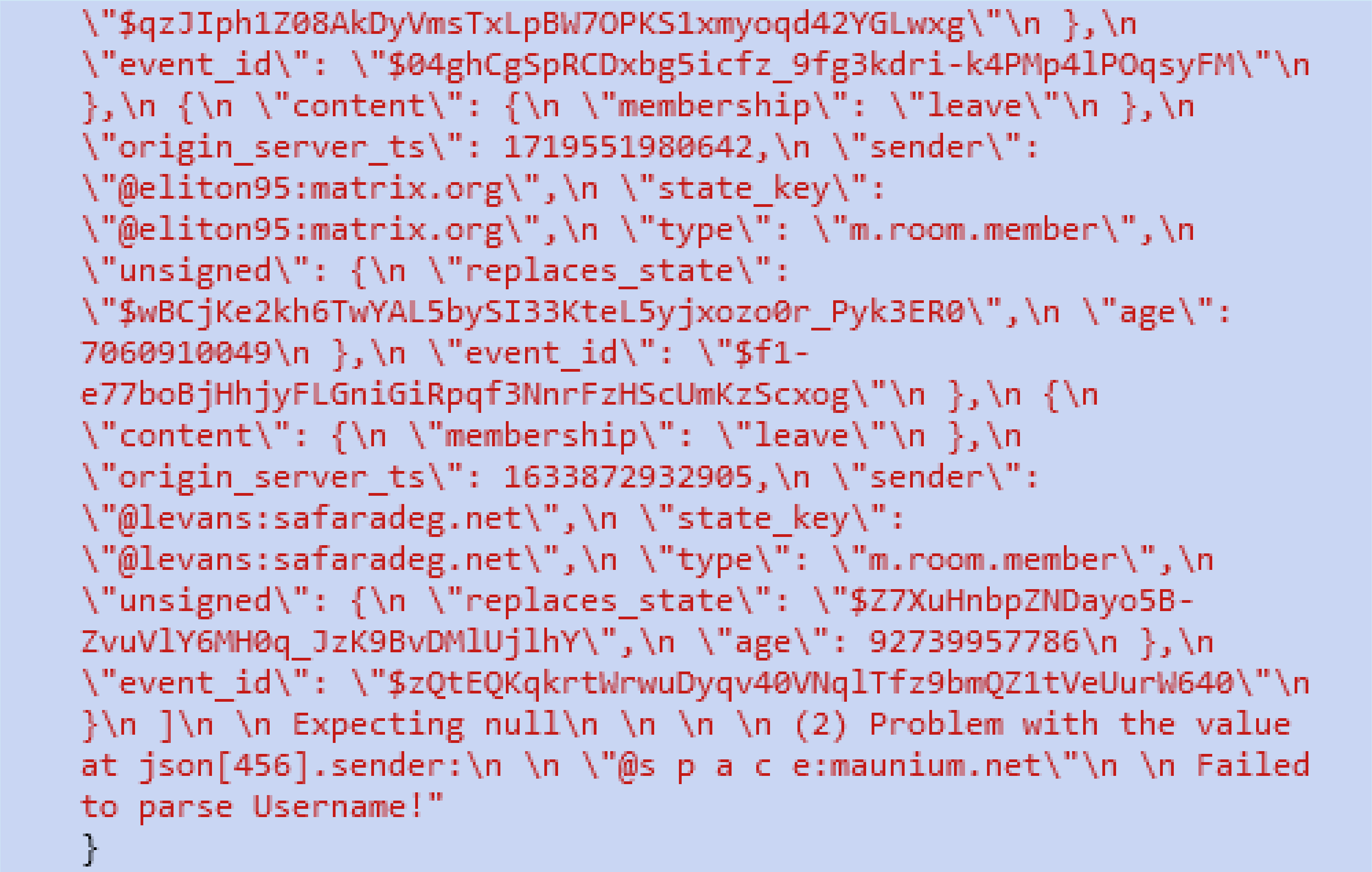
200 response

Name	Type	Description
`access_token`	`string`	**Required:** The new access token to use.
`expires_in_ms`	`integer`	The lifetime of the access token, in milliseconds. If not given, the client can assume that the access token will not expire.
`refresh_token`	`string`	The new refresh token to use when the access token needs to be refreshed again. If not given, the old refresh token can be re-used.



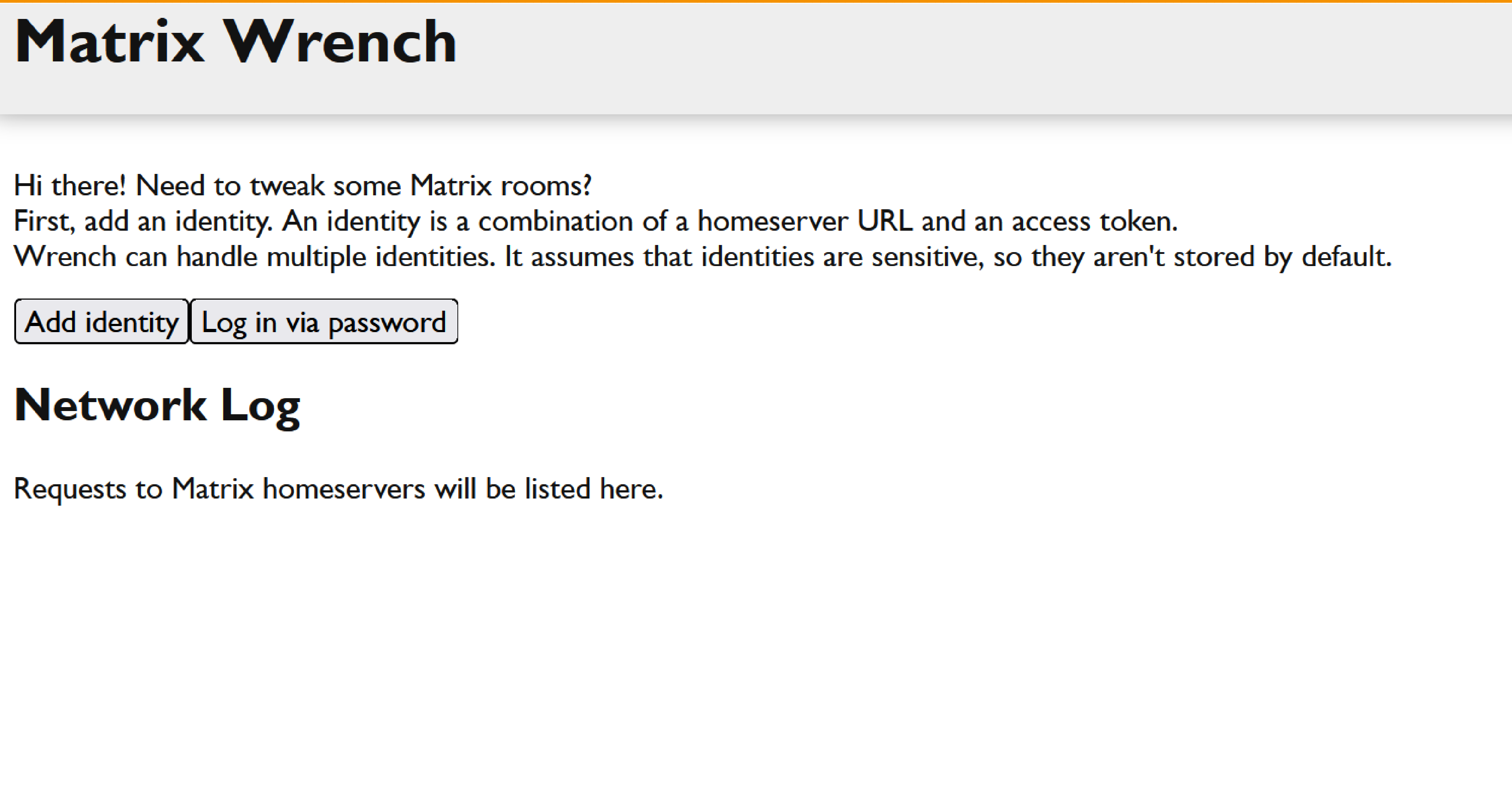
```
Detected problems in 1 module.
-- TYPE MISMATCH -------------------------------------------------- src\Test.elm

Something is off with the body of the `newToken` definition:

12| newToken = Maybe.withDefault input.refreshToken output.refreshToken
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
This `withDefault` call produces:

    Maybe String

But the type annotation on `newToken` says it should be:

    String

Hint: Use Maybe.withDefault to handle possible errors. Longer term, it is
usually better to write out the full `case` though!
```

Pain points



```
\"$qzJIph1Z08AkDyVmsTxLpBW7OPKS1xmyoqd42YGLwxg\"\n },\n
\"event_id\": \"$04ghCgSpRCDxbg5icfz_9fg3kdri-k4PMp41POqsyFM\"\n
},\n {\n \"content\": {\n \"membership\": \"leave\"\n },\n
\"origin_server_ts\": 1719551980642,\n \"sender\":
\"@eliton95:matrix.org\",\n \"state_key\":
\"@eliton95:matrix.org\",\n \"type\": \"m.room.member\",\n
\"unsigned\": {\n \"replaces_state\":
\"$wBCjKe2kh6TwYAL5bySI33KteL5yjxozo0r_Pyk3ER0\",\n \"age\":
7060910049\n },\n \"event_id\": \"$f1-
e77boBjHhjyFLGniGiRpqf3NnrFzHScUmKzScxog\"\n },\n {\n
\"content\": {\n \"membership\": \"leave\"\n },\n
\"origin_server_ts\": 1633872932905,\n \"sender\":
\"@levans:safaradeg.net\",\n \"state_key\":
\"@levans:safaradeg.net\",\n \"type\": \"m.room.member\",\n
\"unsigned\": {\n \"replaces_state\": \"$Z7XuHnbpZNDayo5B-
ZvuVlY6MH0q_JzK9BvDMlUjlhY\",\n \"age\": 92739957786\n },\n
\"event_id\": \"$zQtEQKqkrtWrwuDyqv40VNqlTfz9bmQZ1tVeUurW640\"\n
}\n ]\n \n Expecting null\n \n \n \n (2) Problem with the value
at json[456].sender:\n \n \"@s p a c e:maunium.net\"\n \n Failed
to parse Username!"
}
```

Why does it exist?

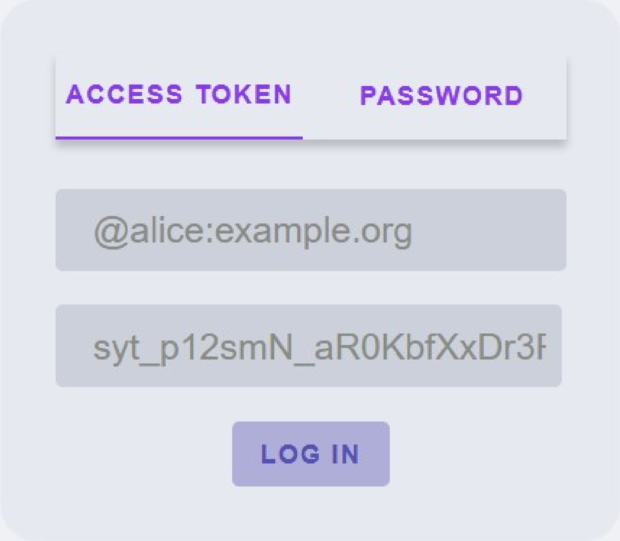


Matrix Wrench

Hi there! Need to tweak some Matrix rooms?
First, add an identity. An identity is a combination of a homeserver URL and an access token.
Wrench can handle multiple identities. It assumes that identities are sensitive, so they aren't stored by default.

Network Log

Requests to Matrix homeservers will be listed here.



@alice:example.org

syt_p12smN_aR0KbfXxDr3

Specs

- Standalone HTML + debugger: 1.048 KB
- Standalone HTML: 899 KB
- Highly optimized JS: 204 KB
- Gzipped minified JS: 64KB

Why should I pay attention?

- Small-scale clients optimize for a custom need
    - Matrix Wrench
    - Moderation client
    - Sysadmin client
- FOSS static HTML files of max. 1 MB
- More rigorous matrix-js-sdk
- Descriptive implementation of spec

Thanks for your attention!

https://matrix.to/#/#elm-sdk:matrix.org