

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Making Interactive Experiences

/ Networking and Multiplayer

/ Replay System

/ DemoNetDriver and Streamers

/ HTTP Streamer REST API

# HTTP Streamer REST API

Overview of the REST API for the HTTP Streamer



The following requests will be made by the **HTTP Streamer** and must be handled by your replay server. In most cases, simply recording the information given in the appropriate location, or providing that information back to a viewer upon request, will be sufficient. It is good to note at this point that replays are not generally stored as a single file. Each individual chunk of a replay generally occupies its own file. Event data, although associated with a replay, is stored separately, since events can be searched without knowing what replay they belong to. Even event groups should be stored as separate files with lists of all events (across all replays) belonging to that group.

The HTTP Streamer's REST API provides functionality for streaming replays from the game server to the replay server (uploading), for streaming live or pre-recorded replays from the replay server to the game, or viewer, server (downloading), or for querying information about the replays available on the server (searching). To implement your own replay server, you will need to respond to the following HTTP requests, as well as any additional requests specific to your game.

## Upload Requests

# Start Streaming

Sends a request to begin uploading a replay stream.

## Syntax

```
1 <Server URL>replay/<Session Name Override>?app=<Replay Version
  AppString>&version=<Replay Network Version>&cl=<Replay CL>&friendlyName=
  <Platform Friendly Name>
2
```

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	application/json

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name Override	(Optional, including the leading "/".) Any name for the replay. Usually includes a GUID.	The replay server is expected to recognize this name when a replay is being requested. It should be returned as the session name, although it can be ignored

Element	Value	Intended Use / Notes
		and the streamer will respect the returned name.
Replay Version AppString	A user-defined string that describes the game being played.	This value is expected to be used for filtering when the HTTP Streamer requests a list of replays.
Replay Network Version	A user-defined string that describes the version of the game being played.	This value is also used for filtering lists of replays, so it should be stored with the replay data.
Replay CL	The changelist of the build being played.	This value is used for filtering lists of replays.
Platform Friendly Name	The platform on which the game is being played.	This value might be displayed to users, and is expected to be returned when lists of replays are requested. However, it will not be sent to the game server with requests for lists of replays and therefore is not expected to be used as a filter.

## Response

Expects a response of type `FNetworkReplayStartUploadingResponse`, which contains JSON data with the string "sessionId" defined. This value will be used as the **Session Name** in future communication from the HTTP Streamer. If the optional `<Session Name Override>` was provided, the returned "sessionId", and thus the ultimate `<Session Name>`, should be the same value. This is the game's way of demanding the session name.

Name	Type	Content
Users	String Array	List of users present in the game. Should be stored as tags on the replay.

## Upload Header

Sends a request to upload the binary data constituting the archive header, which should be recorded in a header file associated with **Session Name**.

## Syntax

```
1 <Server URL>replay/<Session Name>/file/replay.header?numChunks=<Stream Chunk
  Index>&time=<Total Demo Time In MS>
2
```

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	application/octet-stream

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.

Element	Value	Intended Use / Notes
Session Name	Name of the replay being uploaded.	This name will match the value that the replay server provided in response to the initial upload request.
Stream Chunk Index	The index of the current chunk of replay data being sent.	Stream chunks are not uniform in size, but they are strictly ordered by this value.
Total Demo Time In MS	The total running time of the demo up to the end of the current stream chunk.	Tells how much replay data, in milliseconds, has been uploaded so far without having to interpret binary replay data.

## Flush Stream

Sends a request to flush the stream. The content is binary replay data, which can be compressed. This data represents the chunk indicated by **Stream Chunk Index**. Since chunks can vary in data size as well as the length of time they cover, it is recommended that each chunk be stored in a separate file. The HTTP Streamer will attempt to flush its current stream data via this system every ten seconds. The interval between flushes can be adjusted by changing the console variable `httpReplay.ChunkUploadDelayInSeconds`

To enable compression, three functions must be overridden in a child class of

`FHttpNetworkReplayStreamer`:

- `SupportsCompression` must return `true`.
- `CompressBuffer` must perform compression and return `true`, except in the case of an error.
- `DecompressBuffer` must perform decompression and return `true`, except in the case of an error.

## Syntax

```
1 <Server URL>replay/<Session Name>/file/stream.<Stream Chunk Index>?numChunks=  
  <Total Chunks>&time=<Total Demo Time In MS>&mTime1=<Start Time In MS>&mTime2=  
  <End Time In MS>&absSize=<Total Uploaded Bytes>  
2
```

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	application/octet-stream

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being uploaded.	This name will match the value that the replay server provided in response to the initial upload request.
Stream Chunk Index	The index of the current chunk of replay data being sent.	Stream chunks are not uniform in size, but they are strictly ordered by this value.
Total Chunks	The total number of chunks sent so far.	This is always one greater than <b>Stream Chunk Index</b> .

Element	Value	Intended Use / Notes
Total Demo Time In MS	The total running time of the demo up to the end of the current stream chunk.	Tells how much replay data, in seconds, has been uploaded so far without having to interpret binary replay data.
Start Time In MS	The total running time as of the point where this stream chunk begins.	Time is in milliseconds. For streaming replays, can be used to identify which chunk a scrubbing viewer needs.
End Time In MS	The total running time as of the point where this stream chunk ends.	Time is in milliseconds. For streaming replays, can be used to identify which chunk a scrubbing viewer needs.
Total Uploaded Bytes	The total size of the demo up to the end of the current stream chunk.	Tells how much replay data, in bytes, has been uploaded so far.

## Add/Update Event

Sends a request to add or update a replay event. Content contains binary event data to be recorded. This data can be sent through DemoNetDriver and is be user-defined. The second version of this HTTP request can be used to modify or update an existing event, whereas the first version is used to create a new event.



Note that events are meant to be stored separately from replays, which is why event names use GUIDs. Each event knows what replay it belongs to, which is why events are not embedded in the replays themselves.

## Syntax

```
1 <Server URL>replay/<Session Name>/event?group=<Group Name>&time1=<Event Time  
  In MS>&time2=<Event Time In MS>&meta=<Meta Tag>&incrementSize=false  
2  
3 <Server URL>replay/<Session Name>/event/<Session Name>_<Event Name>?group=  
  <Group Name>&time1=<Event Time In MS>&time2=<Event Time In MS>&meta=<Meta  
  Tag>&incrementSize=false
```

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	application/octet-stream

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being uploaded.	This name will match the value that the replay server provided in response to the initial upload request.
Group Name	The name of the event group containing this event.	Events can be put into groups for filtering lists, such as grouping all penalty calls in a sports title.



Element	Value	Intended Use / Notes
Event Name	The GUID of the event being added or updated.	(Optional.) Information about specific, named events may be requested by viewers. If not specified, the event can still be found by listing the events in its group.
Event Time In MS	The time in the replay when this event takes place.	Time is in milliseconds. Although "time1" and "time2" can be used as a begin time and an end time respectively, they are always the same value in the default HTTP Streamer.
Meta Tag	Meta tag associated with this event.	Replay lists can be filtered by meta tag, if the request specifies any. <b>Meta Tag</b> will be URL-encoded by the HTTP Streamer during transmission.

## Stop Streaming

Sends a request to indicate that the replay is done uploading.

## Syntax

```
<Server URL>replay/<Session Name>/stopUploading?numChunks=<Total Num Chunks>&t
```

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	application/octet-stream (no content is sent)

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being uploaded.	This name will match the value that the replay server provided in response to the initial upload request.
Total Num Chunks	The number of chunks in the finished replay.	
Total Time In MS	The total running time, in milliseconds, of the replay.	
Total Bytes Uploaded	The total size, in bytes, of the replay.	

## Download Requests

### Request Start Download

The purpose of this request is to start downloading a replay from the replay server by establishing a connection based on the user name. Immediately after this request goes

through successfully, the HTTP Streamer will send a request to download the replay's header, and then another to enumerate all checkpoints in the replay.

## Syntax

<Server URL>replay/<Session Name>/startDownloading?user=<User Name>

 Copy full snippet

## Details

Item	Value
Request Method	POST
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being downloaded.	This name will match the value that the server provided when the replay was originally uploaded.
User Name	Unique Net Id (GUID)	The preferred network GUID that is logged into the client machine. This value comes from the engine and is not intended to be user-supplied.

# Response

The expected response to this request is of type

FNetworkReplayStartDownloadingResponse

, which contains JSON data.

Name	Type	Content
State	String	The word "Live", to indicate that the game being viewed is happening live. Otherwise, the replay is assumed to be pre-recorded.
NumChunks	Integer	This is expected to be nonzero, and can be updated as streaming continues (used for live replays).
Time	Integer	The total time (in MS) of the replay. Subject to change if <b>State</b> indicated that this is a live stream.
ViewerId	String	A unique, name assigned by the replay server that the game server will use when requesting further chunks of replay data. Often referred to as <b>Viewer Name</b> in this document.

# Request Header

## Syntax

<Server URL>replay/<Session Name>/file/replay.header

 Copy full snippet

## Details

Item	Value
Request Method	GET

Item	Value
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being downloaded.	This name will match the value that the replay server provided when the replay was originally uploaded.

## Response

Expects a response with the replay's header in the content portion, as binary data.

## Request Checkpoint Data

### Syntax

```
<Server URL>replay/<Session Name>/event?group=checkpoint
```

 Copy full snippet

## Details

Item	Value
Request Method	GET

Item	Value
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being downloaded.	This name will match the value that the replay server provided when the replay was originally uploaded.

## Response

Expects a response of type `FHttpResponsePtr` containing checkpoint data as a JSON-formatted string.

Name	Type	Content
events	Array	Contains array entries which consist of all the other fields in this table.
id	String	The name (a GUID) of the checkpoint.
group	String	The group to which the event belongs, if any.
meta	String	Meta Tag string associated with this event, if any.
time1	Integer	Start time (in MS) of the event.

Name	Type	Content
time2	Integer	End time (in MS) of the event. Often the same as the start time.



Note that this is technically a generic request for a group of events. While "checkpoint" is explicitly specified here, the replay server should be able to handle other values for "group" and report all events for any given group.

# Request Stream Chunk

This is a request for a specific stream chunk, named by the viewer. This request will not be sent while there is a chunk already being downloaded.

## Syntax

```
<Server URL>replay/<Session Name>/file/stream.<Stream Chunk Index>
```

Copy full snippet

## Details

Item	Value
Request Method	GET
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Session Name	Name of the replay being downloaded.	This name will match the value that the replay server provided when the replay was originally uploaded.
Stream Chunk Index	The index of the current chunk of replay data being requested.	Stream chunks are not uniform in size, but they are strictly ordered by this value.

## Response

The expected response has a header with some information about the chunk, and its content is a binary chunk of replay data. The binary data can be compressed, if compression is supported. The following fields are expected in the header:

Header Field Name	Description
NumChunks	The total number of chunks in the stream. This can be useful with streamed games, as it lets the user know about new chunks that have been created while viewing.
Time	The total length of the replay (in MS). As with NumChunks, this value can change during the viewing session if the replay is live.
State	This variable has one recognized value, "Live". If this variable is missing or is a different value, the streamer will switch to non-live streaming mode.
MTime1	The start time of the stream, in MS. Can be non-zero for live replay streams. This variable will only be checked if a chunk of data sent, and either this is the first chunk the streamer has downloaded, or the stream is live.



Header Field Name	Description
MTime2	The end time of the stream, in MS. Can change as more chunks come in. This variable will always be checked if a chunk of data is sent, or if the replay is live.

To enable compression, three functions must be overridden in a child class of

`FHttpNetworkReplayStreamer`:

- `SupportsCompression` must return `true`.
- `CompressBuffer` must perform compression and return `true`, except in the case of an error.
- `DecompressBuffer` must perform decompression and return `true`, except in the case of an error.

## Enumerate Events for Replay

Can be called directly from DemoNetDriver, meaning your game project can do this at any time. This request asks the replay server to send a list of all events pertaining to a specific replay (identified by **Session Name**) and filtered by a certain event group. It is recommended that empty group names, as well as a lack of the "group" argument, be supported.

## Syntax

```
1 <Server URL>replay/<Session Name>/event?group=<Group Name>"
2
```

 Copy full snippet

## Details

Item	Value
Request Method	GET
Content Type	None

## Response

Content received in response to this request is expected to be a JSON-formatted string corresponding to `FReplayEventList` and `FReplayEventListItem`.

Name	Type	Content
events	Array	Contains elements which consist of all the other fields in this table.
id	String	The name (a GUID) of the checkpoint.
group	String	The group to which the event belongs, if any.
meta	String	Meta Tag string associated with this event, if any.
time1	Integer	Start time (in MS) of the event.
time2	Integer	End time (in MS) of the event. Often the same as the start time.

## Refresh Viewer

This request is a heartbeat, sent every ten seconds. This frequency is currently hard-coded, but will likely be made user-configurable without code changes in the future. If a viewer fails to send this request for a long period of time (perhaps one minute), then the viewer has most likely disconnected unexpectedly and it would be reasonable to disconnect them. If the "final=true" version is sent, the user is indicating that this is the final heartbeat, and the user has stopped watching the replay.

# Syntax

```
1 <Server URL>replay/<Session Name>/viewer/<Viewer Name>
2
3 <Server URL>replay/<Session Name>/viewer/<Viewer Name>?final=<Final
  Heartbeat>
4
```

 Copy full snippet

# Details

Item	Value
Request Method	POST
Content Type	None

# Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Final Heartbeat	"true" or "false"	By default, "final" will not be sent unless <b>Final Heartbeat</b> is true. If true, the viewer has stopped watching and no further data should be sent.

# Search Requests

# Search Replays

Request a list of all replays that the replay server can send. These replays are expected to be pre-filtered by the user's criteria in the URL. The replay server should be able to handle some of the filter fields being empty or missing.

## Syntax

```
1 <Server URL>replay?app=<Replay Version AppString>&cl=<Replay CL>&version=
2 <Replay Network Version>&meta=<Meta Tag>&user=<User Name>&recent=<In Recent
3 Viewer>"
```

 Copy full snippet

## Details

Item	Value
Request Method	GET
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Replay Version AppString	A user-defined string that describes the game being played.	This value was sent when the user began uploading the replay, and is now being used for filtering the list of replays sent to the potential viewer.

Element	Value	Intended Use / Notes
Replay CL	The changelist of the build being played.	This value is used for filtering lists of replays.
Replay Network Version	A user-defined string that describes the version of the game being played.	This value should reflect changes in your game's data formats as they affect replay. It is also used for filtering lists of replays.
Meta	Tags applied to this replay.	User-applied meta tags can be used for filtering games that have certain features. This could be anything from "all Capture The Flag matches" to "games that went into double-overtime".
User	Unique Net Id (GUID)	This is used to filter games based on their inclusion of a specific player.
Recent	"true" or "false"	If true, only recently-uploaded replays should be shown. What defines "recent" is up to the replay server.

## Response

Expects a response with content in the form of a JSON-formatted string.

Name	Type	Content
replays	Array	Contains elements which consist of all the other fields in this table.
AppName	String	The <b>Replay Version AppString</b> that was sent when the replay was uploaded. Should match the

Name	Type	Content
		parameter you sent with the request.
SessionName	String	The name (GUID) of the replay.
FriendlyName	String	The <code>&lt;Platform Friendly Name&gt;</code> that was provided when the replay was uploaded.
Timestamp	DateTime	The time and date associated with the replay (i.e. when this game happened).
SizeInBytes	Integer	Total size, in bytes, of the replay.
DemoTimeInMs	Integer	Running length, in MS, of the replay.
NumViewers	Integer	Number of users currently watching this replay.
IsLive	Boolean	True if the game in the replay is still being played.
Changelist	Integer	The changelist (CL) of the engine running the game when this replay was recorded. Corresponds to the <b>Replay CL</b> argument sent when uploading the replay. Should match the parameter you sent with the request.
shouldKeep	Boolean	True if the debug "shouldKeep" request was made.

## Search Replays by Event

Sends a request for a list of replays from the server. These replays are expected to be pre-filtered to include only those that contain at least one event in the named **Event Group**.

## Syntax

```
1 <ServerURL>event?group=<Event Group>
2
```

 Copy full snippet

## Details

Item	Value
Request Method	GET
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Event Group	The name of the event group containing this event.	Events can be put into groups for filtering lists, such as grouping all penalty calls in a sports title.

## Response

Expects a response with content as a JSON-formatted string containing a list of replays from the server.

Name	Type	Content
replays	Array	Contains elements which consist of all the other fields in this table.

Name	Type	Content
AppName	String	The <b>Replay Version AppString</b> that was sent when the replay was uploaded.
SessionName	String	The name (GUID) of the replay.
FriendlyName	String	The <code>&lt;Platform Friendly Name&gt;</code> that was provided when the replay was uploaded.
Timestamp	DateTime	The time and date associated with the replay (i.e. when this game happened).
SizeInBytes	Integer	Total size, in bytes, of the replay.
DemoTimeInMs	Integer	Running length, in MS, of the replay.
NumViewers	Integer	Number of users currently watching this replay.
blsLive	Boolean	True if the game in the replay is still being played.
Changelist	Integer	The changelist (CL) of the engine running the game when this replay was recorded. Corresponds to the <b>Replay CL</b> argument sent when uploading the replay.
shouldKeep	Boolean	True if the debug "shouldKeep" request was made.

## Request Replay

This is called directly from the DemoNetDriver, so it can be done from the game project at any time, but will usually be called while viewing a replay with an **Event Name** retrieved from a list provided by a previous request.

## Syntax



```
1 <Server URL>event/<Event Name>
2
```

 Copy full snippet

## Details

Item	Value
Request Method	GET
Content Type	None

## Parameters

Element	Value	Intended Use / Notes
Server URL	Replay server URL. Drawn from <code>DefaultEngine.ini</code> , unmodified.	The final "/" in the URL must be included - it will not be added automatically.
Event Name	The GUID of the event requested.	Since events are identified by GUID, the event can be from any replay on file. Make sure that the event you request is from a list of events pertaining to the replay you are currently watching.

## Response

Expects a response with Content as a binary stream from the replay server.