



genius sports LiveStats - In-Arena Feed

Documentation - Basketball

[Introduction](#)[Connection Parameters](#)

Message Types

[ping](#)[status](#)[setup](#)[matchInformation](#)[teams](#)[officials](#)[boxscore](#)[action](#)[playbyplay](#)[summary](#)[standings](#)[competitionStatistics](#)[schedule](#)

Definitions

[Action Types](#)[Playing Surface Definitions](#)

Introduction

This document explains the TV Feed from the LiveStats software. This feed is provided in venue by means of a LAN connection. It provides both individual actions and aggregated data.

The TV feed is a simple TCP socket connecting to specified port. By default the port number is **7677** on the computer running the LiveStats software, however this can be changed in venue by the statistician. For correct values, you will need to confirm with the statistician in venue.

Character Set

All data provided to these API functions is required to be in UTF-8 format.

All responses will be provided as UTF-8 strings.

Message Format

Messages are sent in JSON structures.

Each message content differs depending on the type of the message, but will contain at least the following keys.

- **type** This field indicates the type of message. The valid types are each explained in more detail in this documentation.

```
{
  "type": "action",
  ...
  ...
}
```

The body of the response is a series of messages (defined below). Each message is delimited by a carriage return/line-feed `\r\n` series of characters (in hex, 0x0D 0x0A).

From time to time new key values/structures will be added to this protocol. Any client consuming this data should ignore anything they do not understand. New fields/values/structures should not cause a failure or crash in the client system.

Not all keys will exist in all messages for all events. No value present for a key can be sent either as a value of `null` or the key not being present in the data.

Connection Parameters

On connection you must send a parameters message specifying the data you want to be sent in the stream. If you do not send this message within 5 seconds of connection, the connection will be closed automatically. If you do not want to set a parameter it does not need to be included in the message. If no parameters need to be set then a blank message with only the **type** parameter set should be sent.

type

enum

The type of the message

parameters**types**

string(100)

A comma separated value of the message types to be sent through the feed. If not set, then the following value is assumed.

se,ac,mi,te,box,bbp**st**

status

se

setup

ac

action

mi

match information

te

teams

of

officials

su

summary

box

boxscore - Statistics for teams and players for the match

bbp

playbyplay

sd

standings

sc

schedule

cs

competition statistics

playbyplayOnConnect

enum

If this parameter is set to 1, then the playbyplay section will only be sent at the start of the connection not on every action. If the **types** parameter is used and **bbp** is not specified then this parameter has no effect as the playbyplay will not be sent at all.

0

No (Default)

1

Yes

fromMessageId
integer

When connecting to the service, previous messages occurring in the game will not be sent, only new messages. By specifying a messageId with this paramter, messages with a messageId greater than (or equal to) the number specified here will be resent. 0 is valid and will resend all messages on connection.

Example

```
{
  "type": "parameters",
  "types": "se,ac,mi,te,sc,box"
}
```

Message Types

ping

This message type is always sent regardless of the values of the types URL parameter. If no other message is available, this message type will be sent every 10 seconds to maintain the connection.

type enum	The type of the message ping
timestamp date	The current date/time Format is: YYYY-MM-DD HH:MM:SS:CC

Example

```
{
  "type": "ping",
  "timestamp" : "2017-03-31 16:56:34:01"
}
```

status

This message type has the current status of the match.

type enum	The type of the message status
---------------------	--

status enum	The status of the match. READY WARMUP PREMATCH ANTHEM ONCOURT COUNTDOWN INPROGRESS PERIODBREAK INTERRUPTED CANCELLED FINISHED PROTESTED COMPLETE RESCHEDULED DELAYED
-----------------------	---

+ period object

periodStatus enum	The status of the period. PENDING STARTED ENDED CONFIRMED
-----------------------------	---

clock string	The current time on the clock. Format is: MM:SS:CC
------------------------	---

shotClock string	The current time on the shot clock. Format is: MM:SS:CC
----------------------------	--

clockRunning enum	Is the clock running? 0 No 1 Yes
possession integer	The number of the team in the match, 1 or 2 that currently has possession. 0 is valid.
possessionArrow integer	The number of the team in the match, 1 or 2 to which the possession Arrow is currently pointing. 0 is valid.
+ scores array	An array of team records

Example

```
{
  "type": "status",
  "period": {
    "current": 1,
    "periodType": "REGULAR"
  },
  "clock": "01:55:99",
  "clockRunning": 1,
  "shotClock": "00:00:00",
  "status": "INPROGRESS",
  "periodStatus": "STARTED",
  "scores": [{
    "teamNumber": 1,
    "score": 14
  }, {
    "teamNumber": 2,
```

setup

This message type has information about the how the match will be run.

type enum	The type of the message setup
---------------------	--------------------------------------

⊕ periods

object

shotClock

The maximum value of the shot clock, in seconds.

integer

foulsPersonal

The number of personal fouls allowed.

integer

foulsTechnical

The number of technical fouls allowed.

integer

foulsBeforeBonus

The number of fouls allowed before a bonus is given.

integer

maxFoulsPersonal

The maximum number of personal fouls allowed

integer

maxFoulsTechnical

The maximum number of technical fouls allowed

integer

⊕ timeouts

object

Example

```
{
  "type": "setup",
  "timeouts": {
    "extratime": 1,
    "period4": 2,
    "period3": 2,
    "period2": 2,
    "period1": 2,
    "half2": 3,
    "half1": 2,
    "style": "period"
  },
  "periods": {
    "breakPeriod": 2,
    "breakHalftime": 2,
    "length": 10.
  }
}
```

matchInformation

This message type has information about the match. It is generally only sent at the start of the connection.

type

enum

The type of the message

matchInformation**+ competition**

object

+ venue

object

+ league

object

+ match

object

+ pool

object

Example

```
{
  "match": {
    "matchName": "",
    "matchTime": "2017-02-24 12:00:00",
    "matchNumber": 89,
    "phaseName": "",
    "poolNumber": 0,
    "matchId": 2,
    "externalId": "89",
    "roundDescription": "1",
    "matchType": "REGULAR",
    "twitterHashtag": "",
    "roundNumber": "1"
  },
  "competition": {
    "competitionName": "National Season 2017".
```

teams

This message type contains information about the teams and players involved in this game

type enum	The type of the message teams
messageId integer	Unique identifier of the message being sent
+ teams array	An array of team records

Example

```
{
  "messageId": 23,
  "teams": [{
    "players": [{
      "active": 1,
      "pno": 1,
      "firstName": "Natalie",
      "personId": 485217,
      "shirtNumber": "2",
      "playingPosition": "",
      "height": 0,
      "familyName": "Barstow",
      "starter": 1,
      "captain": 0
    }, {
      "active": 1.
```

officials

This message type contains information about the match officials assigned to the match

type enum	The type of the message officials
messageId integer	Unique identifier of the message being sent
+ commissioner object	

+ referee1
object

+ referee2
object

+ referee3
object

+ statisticians An array of statisticians
array

+ technicalDelegate
object

boxscore

This message type contains box score information.

type The type of the message
enum

 boxscore

+ teams An array of teams
array

Example

```
{
  "teams": [{
    "teamNumber": 1,
    "total": {
      "team": {
        "sAssists": 4,
        "sMinutes": 0,
        "sPointsInThePaint": 2,
        "sPointsInThePaintMade": 1,
        "sPointsSecondChance": 4,
        "sReboundsDefensive": 7,
        "sReboundsOffensive": 3,
        "sFreeThrowsMade": 3,
        "sSecondChancePointsMade": 2,
        "sSteals": 0,
        "sThreePointersAttempted": 7.
```

action

This message type contains an individual game action.

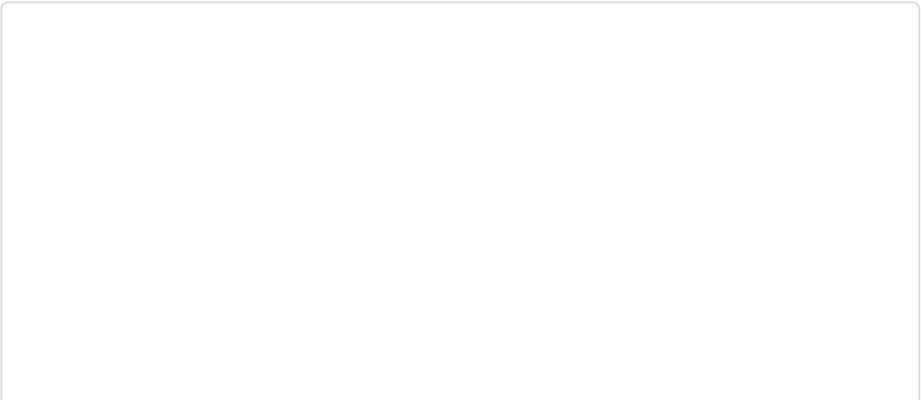
Actions come in two types, administrative or sport. A sport type action is related to the sport aspect of the match, eg substitution or foul. An administrative type action is related to the process of publishing the game eg. period confirmed. Only sport type actions are given an actionNumber. The action number is a sequential integer that should uniquely refer to that action. The actionNumber is repeated to indicate an edit or delete on the original action.

type enum	The type of the message action
messageId integer	Unique identifier of the message being sent
* actionNumber integer	Unique sequence number for this action
teamNumber integer	The number of the team in the match, 1 or 2.
pno integer	The number of the player in the match. This is a sequence number not a jersey number. This is generally 1 to the number of total players in the team. It is not the personId field.
clock * string	The time on the clock when this action occurred. Format is: MM:SS:CC
shotClock string	The time on the shot clock when this action occurred. Format is: MM:SS:CC
timeActual datetime	The date/time when this action occurred in UTC. Format is: YYYY-MM-DD HH:MM:SS
period integer	What period of the match is this statistic related to. 0 is valid and represents the value for the total match. Overtime periods should start at 1.

periodType enum	The type of period REGULAR <i>(Default)</i> OVERTIME
actionType * string(20)	A code representing the action type. See Action Types for the list of options.
success * enum	Was this action completed successfully? Generally all actions except scoring actions are successful. 1 Yes 0 No
subType string(20)	The subordinate type of the action. Used to further define the action. See Action Types for the list of options.
qualifiers array	An array of qualifiers to the action. See Action Types for the list of options.
value string(1000)	A text field. Some action types require a value. This is where that value will be placed. See Action Types for the list of actions that require this field.
previousAction integer	The actionNumber of the base/primary action if this action relates to another action. eg. For a freethrow/freekick, this may be the actionNumber of the original foul.
officialId integer	The unique identifier for the official calling the action
x float	X coordinate on the playing surface where the action occurred. Defined in Playing Surface Definition.
y float	Y coordinate on the playing surface where the action occurred. Defined in Playing Surface Definition.
area string	The area of the playing surface where the action occurred. Defined in Playing Surface Definition.

side * enum	Indicates which side of the playing surface the player's/team's goal is. (blank) left Team's goal is to the left right Team's goal is to the right
score1 * string	The score of team 1 after this action is complete
score2 * string	The score of team 2 after this action is complete
edited datetime	If the action has been edited (after being initially sent) this field contains the last time it was edited (in UTC). If the action has not been edited, then this field is not present. Format is: YYYY-MM-DD HH:MM:SS
inserted datetime	If the action has been inserted (added out of sequence) this field contains the time it was inserted (in UTC). If the action has not been inserted, then this field is not present. Format is: YYYY-MM-DD HH:MM:SS
deleted datetime	If the action has been deleted (after being initially sent) this field contains the time it was deleted (in UTC). If the action has not been deleted, then this field is not present. Format is: YYYY-MM-DD HH:MM:SS
origMessageId integer	If this action is a delete/update then this field contains the messageId of the original message.

Example



```
{
  "pno": 15,
  "period": 1,
  "actionNumber": 16,
  "success": 1,
  "previousAction": 0,
  "messageId": 34,
  "periodType": "REGULAR",
  "score2": 2,
  "score1": 0,
  "y": 42.857,
  "x": 81.239,
  "subType": "layup",
  "qualifiers": ["pointsinthepaint"],
  "timeActual": "2017-03-05 13:01:30",
  "shotClock": "00:00:00".
}
```

playbyplay

This message type contains an array of 'sport' actions. This list of actions is the current state of the game and is the result of any inserts,edits,deletes. It does not contain any 'administrative' type actions.

type enum	The type of the message playbyplay
+ actions array	Array of play by play actions, sorted ascending

Example



```
{
  "type": "playbyplay",
  "actions": [{
    "pno": 1,
    "period": 1,
    "actionNumber": 1,
    "success": 1,
    "previousAction": 0,
    "periodType": "REGULAR",
    "score2": 0,
    "score1": 0,
    "subType": "in",
    "timeActual": "2017-03-05 12:50:03",
    "shotClock": "00:00:00",
    "teamNumber": 1,
    "clock": "10:00:00".
  ]
}
```

summary

This message type has information about the how the match was run. Usually it will only be sent once at the end of the match.

type enum	The type of the message summary
messageId integer	Unique identifier of the message being sent
timeActual datetime	The actual start date/time of the match in the UTC timezone Format is: YYYY-MM-DD HH:MM:SS
timeEndActual datetime	The actual end date/time of the match in the UTC timezone Format is: YYYY-MM-DD HH:MM:SS
duration integer	The duration (in minutes) of the match, excluding any pregame activities
durationActual integer	The actual duration of the match (mins) including timeouts, stoppages and period breaks.

temperature integer	The temperature during the match (degrees Celsius)
-------------------------------	--

attendance integer	The number of people who attended the match
------------------------------	---

Example

```
{
  "durationActual": 97,
  "timeEndActual": "2017-03-05 14:38:01",
  "messageId": 923,
  "timeActual": "2017-03-05 13:01:11",
  "duration": 40,
  "attendance": 250,
  "temperature": 25,
  "type": "summary"
}
```

standings

This message type contains the current standings for the competition

type enum	The type of the message standings
---------------------	---

+ standings array	Array of standings records
-----------------------------	----------------------------

competitionStatistics

This message type has information about statistics for the competition. These are statistics that are current before the match begun. It is generally only sent at the start of the connection.

type enum	The type of the message competitionStatistics
messageId integer	Unique identifier of the message being sent
+ teams array	An array of team records

schedule

This message type has information about the schedule for the competition. This schedule contains all matches for +/- 24 hours and are current before the match begun. It is generally only sent at the start of the connection.

type enum	The type of the message schedule
+ matches array	An array of matches

Definitions

Action Types

For a particular action an event can have 1 type, 1 subtype and any number of qualifiers

If qualifiers are able to be used for a particular action/subType, then any number of qualifiers may be applied to the action.

If the subType indicates (blank) this means that the action can be used without a subType.

Sport Action Types

type	subType	qualifiers
game		

type	subType	qualifiers
	end	<ul style="list-style-type: none">confirmed
period	start end	<ul style="list-style-type: none">confirmed
2pt	(blank) dunk layup fadeaway tipin jumpshot alleyoop drivinglayup hookshot	Applies to all 2pt subTypes: <ul style="list-style-type: none">2ndchancefastbreakfromturnoverpointsinthepaintblocked
	floatingjumpshot stepbackjumpshot pullupjumpshot turnaroundjumpshot wrongbasket	
3pt		Applies to all 3pt subTypes:

type	subType	subTypes: qualifiers
	(blank) fadeaway jumpshot floatingjumpshot stepbackjumpshot pullupjumpshot turnaroundjumpshot wrongbasket	2ndchance <ul style="list-style-type: none">fastbreakfromturnoverblocked
freethrow	1of1 1of2 1of3 2of2 2of3	Applies to all freethrow subTypes: <ul style="list-style-type: none">2ndchancefastbreakfromturnover
	3of3	
jumpball	(blank) startperiod unclearposs lodgedball	

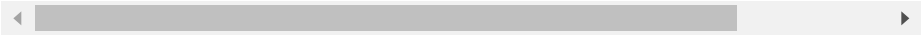
type	subType	qualifiers
	blocktieup outofbounds outofboundsrebound doubleviolation won lost	
assist		
block		
rebound	defensive offensive	Applies to all sub-types: <ul style="list-style-type: none">teamdeadball
foul		Applies to all foul subTypes: <ul style="list-style-type: none">1freethrow2freethrow3freethrowoneandone

	offensive personal	Applies to offensive subType: <ul style="list-style-type: none">blockchargedoubleflagrant1flagrant2 Applies to all personal foul subType:
--	---------------------------	--

type	subType	qualifiers
		<ul style="list-style-type: none">shootinglooseball
		<ul style="list-style-type: none">doubleflagrant1flagrant2
	technical	Applies to technical subType: <ul style="list-style-type: none">doubleclassaflagrant2contactdeadball
	unsportsmanlike	
	disqualifying	
	benchTechnical	Applies to benchTechnical subType: <ul style="list-style-type: none">doubleclassa
	coachTechnical	Applies to coachTechnical subType: <ul style="list-style-type: none">doubleclassadirect
	adminTechnical	Applies to adminTechnical subType: <ul style="list-style-type: none">administrativeclassbbenchclassbcoachclassbcoachindirect
	coachDisqualifying	

timeout		
type	subType	qualifiers
timeout	<div>full</div> <div>short</div> <div>officials</div> <div>commercial</div>	<div>Applies to official subType:</div> <div><ul style="list-style-type: none">bloodrulecallreview</div>
steal		
turnover	<div>(blank)</div> <div>offensivegoaltending</div> <div>laneviolation</div> <div>ballhandling</div> <div>dribbling</div> <div>badpass</div> <div>lostball</div> <div>overandback</div> <div>backcourt</div> <div>doubledribble</div> <div>outofbounds</div>	
	<div>travel</div> <div>shotclock</div> <div>offensive</div> <div>3sec</div> <div>5sec</div>	<div>Applies to 5sec subType:</div> <div><ul style="list-style-type: none">team</div>

type	8sec subType	qualifiers
	10sec	Applies to 10sec subType: <ul style="list-style-type: none">team
	24sec	Applies to other subType: <ul style="list-style-type: none">team
	other	
substitution	in out	



Administrative Action Types

type	subType	qualifiers	
status			Indicating a change in the status of the record.
	delayed	<ul style="list-style-type: none">prevgameteamlateearly	The scheduled time of the match has been changed. The clock should be stopped when the play has been delayed. Negative delay indicates early start.
	loaded		The record has been loaded into the software.

type	subType	qualifiers	
	inprogress		<p>The r ready proce the s and t team been</p> <p>The r curre progr</p>
	periodbreak		<p>The r curre progr but ir breal betw peric</p>
	interrupted	<ul style="list-style-type: none">• equipment• venue• weather• spectator• medical	<p>The r has b stop temp and v resur a del delay minu shou pass the v field. quali can b to giv reasc the d</p>
	cancelled		<p>The r has b stop will n resta</p>
	rescheduled		<p>The r has b</p>

type	subType	qualifiers	
			stop has n start will n start. It wil playe at an time.
	finished		The r has c the e
	protested		The r has c the e the r has b prote
	complete		The r comp score been confi and r offici
periodstatus	pending		Indic chan; the p statu Indic peric abou start
	started		Indic peric start
	ended		Indic peric ende

type	subType confirmed	qualifiers	end
			Indic that s have confi for th peric
shotclock			

	adjustment	<ul style="list-style-type: none">kickedballfoul	The r of ser adjus shou passe the v field. Nega value allow
possessionchange			
possessionarrowchange			
capturestatus	unreliable		Indic that t state game sent softw differ signif from actu situa
	reliable		Indic that t state game sent softw

type	subType	qualifiers	now matc actual situa
clock	start stop		

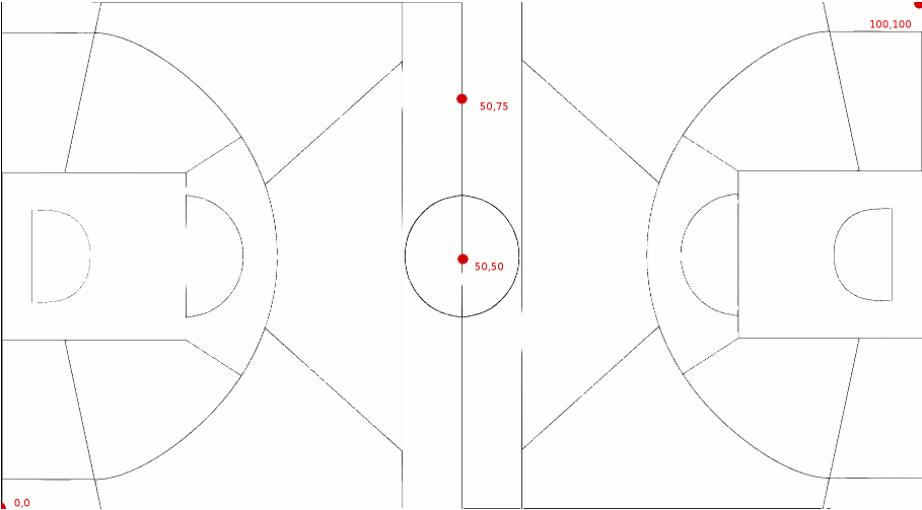
	adjustment	<ul style="list-style-type: none">second_plussecond_minusminute_plusminute_minus	The r of se adju shou pass the v field. Nega value allow
--	------------	---	---



Playing Surface Definitions

Coordinates

area , x , y and side are defined based upon the diagram below.

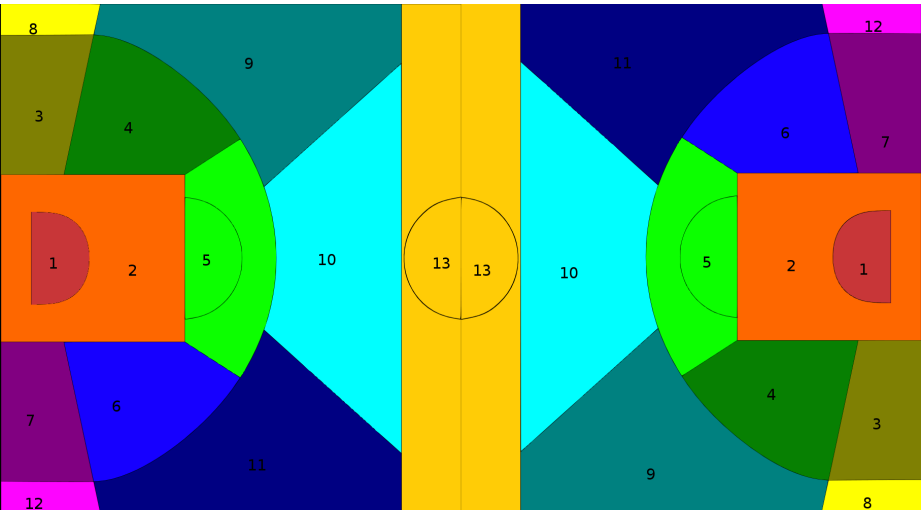


(basketball_coords.png)

x and **y** are defined as percentage measurments of the total width and height of the basketball court. The origin (0,0) is the bottom left corner of the image meaning the top right point is (100,100).

Example:
A FIBA Standard Basketball Court length is 28m and the 3pt line is at 6.75m
Therefore the x coordinate of the 3pt line = $6.75/28 * 100 = 24.1$
and the x coordinate of the other 3pt line = $(28-6.75)/28 * 100 = 75.89$

Action Areas



(basketball_courtmap.png)

area is defined by which of the defined boxes the player was in when the action occurred.

The value of **area** in reference to the diagram are as follows:

Number on diagram	Value
1	underbasket
2	inthepaint
3	insiderrightwing
4	insiderright
5	insidecenter
6	insideleft
7	insideleftwing
8	outsiderrightwing
9	outsiderright
10	outsidecenter
11	outsideleft
12	outsideleftwing
13	backcourt

Court Types(svg)

FIBA Court(svg)



NCAA Old Court(svg)

width 15.2m (50 feet)
length 28.7m (94 feet)
3pt line distance 6.32m (20 feet 9 inches)
 (basketball_court-2_(NCAAOld).svg)

NCAA New Court(svg)

width 15.2m (50 feet)
length 28.7m (94 feet)
3pt line distance 6.75m (22.15 feet)
 (basketball_court-3_(NCAANew).svg)