## Match

An entire session recording.

| Field | Type | Description |
| --- | --- | --- |
| auto\_scored | boolean | Whether SwingVision’s auto scoring feature was used |
| games\_per\_set | int | How many games were required to win each set, takes on values of: *[6, 10, 3, 9, 8, 4]* |
| gid | int | The service game number within the match, starting from 1 for the first service game. |
| guest\_id | int | Identifier for the opponent in SwingVision’s database. *A guest\_id of nil indicates that a user was not tagged as an opponent in the session.* |
| guest\_right\_handed | boolean | Indicator for right-handedness of guest |
| guest\_shot\_count | int | How many shots were hit by the guest |
| host\_id | int | Identifier for the primary player in SwingVision’s database |
| host\_right\_handed | boolean | Indicator for right-handedness of host |
| host\_shot\_count | int | How many shots were hit by the host |
| id | int | Identifier for the match in SwingVision’s database |
| is\_ad | boolean | Indicator for whether advantage points were played |
| is\_super\_tiebreak | boolean | Indicator for whether the super tiebreak format was used for the deciding set |
| is\_tiebreak | boolean | Indicator for whether tiebreaks were played |
| original\_score\_tier | string | Original scoring mode set by the user at the beginning of the match, takes on values with these mappings:  *{‘beginner’ : ‘game by game’, ‘intermediate’ : ‘point by point’, ‘advanced’ : ‘point by point +’}* |
| score\_tier | string | Final scoring mode after auto scoring is run, takes on values: *[‘beginner’ : ‘game by game’, ‘intermediate’ : ‘point by point’, ‘advanced’ : ‘point by point +’]* |
| sets\_per\_match | int | Which “Best of x” format was used for the match, takes on values of: *[5, 3, 1]* |
| total\_points | int | Total number of points in the match |

## 

## Game

An individual service game within a match. Each game is played with a scoring progression of 0, 15, 30, 40, Ad, Game, etc. with the exception of tiebreaks, which have a scoring progression of 0, 1, 2, 3, etc.

| Field | Type | Description |
| --- | --- | --- |
| did\_host\_win\_game | boolean | Indicator for whether the host player ultimately won the service game |
| ended\_at | string | UTC Time at which point ended, in format: '2023-05-29T11:13:14.946000Z' |
| gid | int | The service game number within the match, starting from 1 for the first service game. |
| is\_tiebreak | boolean | Indicator for whether the service game is a tiebreak |
| match\_id | float | Identifier for the match in SwingVision’s database |
| points\_per\_game | int | If the service game is a tiebreak, the number of points required to win the game: *[0, 7, 10]* |
| prev\_guest\_games | int | The number of games won in the current set by the guest player, as of the beginning of the service game |
| prev\_host\_games | int | The number of games won in the current set by the host player, as of the beginning of the service game |
| sid | int | The set number within the match, starting from 1 for the first set. |
| started\_at | string | UTC Time at which point started, in format: '2023-05-29T11:13:14.946000Z' |
| total\_points | int | Number of points played within the service game |
| win\_margin | int | If the service game is a tiebreak, the minimum difference in the number of points (between the winning team and losing team) required to win the game (typically **2**) |

## 

## Point

An individual rally that consists of shots hit back and forth. Every point starts with a serve or feed and ends with a winner or error.

| Field | Type | Description |
| --- | --- | --- |
| detail | string | String that indicates how the point ended, takes values of: *['normal', 'double\_fault', 'service\_winner', 'forehand\_winner', 'backhand\_unforced\_error', 'backhand\_winner', 'forehand\_forced\_error', 'forehand\_unforced\_error', 'backhand\_forced\_error', 'ace']* |
| did\_host\_win\_point | boolean | Indicates whether the host won the point |
| ended\_at | string | UTC Time at which point ended, in format: '2023-05-29T11:13:14.946000Z' |
| game\_id | int | The service game number within the match, starting from 1 for the first service game. *A game\_id of 0 indicates that the point/rally was played during the warmup prior to the start of the match.* |
| is\_rally | boolean | Indicator of whether the point is a rally or a scored point  *Note: A value of FALSE indicates that this was a rally played in a non-scoring setting, such as a practice session.* |
| match\_id | int | Identifier for the match in SwingVision’s database |
| num\_of\_serves | integer | Number of serves hit during point (*0*, *1* or *2*)  *Note: num\_of\_serves value of 0 indicates that the point started with a non-serve shot such as a feed* |
| pid | int | The rally number within the match, starting from 1 for the first rally. |
| prev\_guest\_points | int | The number of points won in the current service game by the guest player, as of the beginning of the point |
| prev\_host\_points | int | The number of points won in the current service game by the host player, as of the beginning of the point |
| rally\_length | int | The number of shots in the rally |
| server | string | Classification of the server, takes on values of: *['host', 'guest']* |
| set\_id | int | The set number within the match, starting from 1 for the first set. *A set\_id of 0 indicates that the point/rally was played during the warmup prior to the start of the match.* |
| started\_at | string | UTC Time at which point started, in format: '2023-05-29T11:13:14.946000Z' |
| was\_break\_point | boolean | Indicates whether the point was a break point (e.g. “15-40”) |
| was\_set\_point\_for\_guest | boolean | Indicates whether it was set point for the guest |
| was\_set\_point\_for\_host | boolean | Indicates whether it was set point for the host (e.g. “40-15 in the 5-4 game of the first set”) |

## 

## Shot

An individual shot hit by a player within a point.

| Field | Type | Description |
| --- | --- | --- |
| bounce\_court\_side | string | Side of the net, relative to the camera, where the ball **landed** on the court, takes on values of: *['far', 'near', 'net']* |
| bounce\_location | list (float) | 3D Coordinate of where the shot bounced, takes format of *[float, float, float]*  *Note: The center of the near baseline is (0,0,0). X is positive to the right, Y is positive away from the camera, and Z is positive above the court’s surface.* |
| bounce\_location\_lat | string | Name of the location where the ball **landed** on the court longitudinally, takes on values of: *['ad', 'deuce', 'ad\_alley', 'ad\_out', 'deuce\_alley', 'deuce\_out', 'center\_line']* |
| bounce\_location\_long | string | Name of the location where the ball **landed** on the court laterally, takes on values of: *['service\_box', 'no\_mans\_land', 'out']* |
| game\_id | int | The game number within the match, starting from 1 for the first game of the set.  *game\_id of 0 indicates shot was hit during the warmup prior to the start of the match*  *Note: game\_id value is carried over between sets*  *(eg. If the 1st set is 6-0 and the 2nd set is 3-0 when the shot is hit, game\_id will be 9.)* |
| hit\_court\_side | string | Side of the net that the shot was hit relative to the camera, takes values of: *[‘near', 'far']* |
| hit\_location | list (float) | 3D Coordinate of the where the shot was hit, takes format of *[float, float, float]*  *Note: The center of the near baseline is (0,0,0). X is positive to the right, Y is positive away from the camera, and Z is positive above the court’s surface.* |
| hit\_location\_lat | string | Name of location where the shot was **hit** on the court longitudinally, takes on values of: *['ad', 'deuce', 'ad\_alley', 'ad\_out', 'deuce\_alley', 'deuce\_out', 'center\_line']* |
| hit\_location\_long | string | Name of location where the shot was **hit** on the court laterally, takes on values of: *['service\_box', 'no\_mans\_land', 'out']* |
| hit\_type | string | Classification of the stroke type, takes on values of: *['feed', 'ground\_stroke', 'volley', 'serve', 'first\_serve', 'second\_serve', 'overhead']* |
| hit\_velocity | list (float) | Velocity of the shot in 3D space, takes format of: *[float, float, float]*  *Note: each float in hit\_velocity is the velocity of the ball with respect to each axis in 3D space (x, y, z)* |
| hit\_wing | string | The side of the player’s body that the shot was hit from, takes on values of: *['right', 'left', 'top']*  *Note: Only overheads can take on hit\_wing of ‘top’* |
| match\_id | int | Identifier for the match in SwingVision’s database |
| net\_type | string | Indicator for whether the ball hit the net or successfully traveled over the net, takes on values of: *['over', 'net']*  *Note: Shots that hit the net and subsequently land on the other side will have net\_type of ‘over’ while shots that don’t will have net\_type of ‘net’.* |
| pid | int | The rally number within the match, starting from 1 for the first rally. |
| player | string | Classification of player who hit the shot, takes on values of: *['host', 'guest']* |
| set\_id | int | The set number within the match, starting from 1 for the first set.  *A set\_id of 0 indicates that the shot was hit during the warmup prior to the start of the match.* |
| shot\_type | string | Type of the shot within the rally/point, takes values of: *['none', 'serve', 'first\_serve', 'second\_serve', 'return', 'serve\_plus\_one', 'return\_plus\_one', 'in\_play']*  *Note: values of ‘none’ are typically used for feeds* |
| sid | int | The shot number within the match, starting from 1 for the first shot. |
| user\_id | float | Identifier for the user in SwingVision’s database |

## 