F09 Football Match Statistics

Feed Information

This feed is a step up from the F8 feed and offers subscribers a complete match statistics package. It includes all of the information provided by F7 & F8 on Team Line-Up, Team Statistics, Bookings, Goals, and Substitutions but then also a further tier of information on Individual Player Statistics. For each player there is a full breakdown of passes, shots, crosses, dribbles, tackles, interceptions and much more.
A full list of the statistics provided can be found in the appendix to this document.

Delivery Timings

Once a production environment is established between Opta and the subscriber, Opta delivers SRML MatchResults files per game. The first file should be expected anytime between 1 hour to 5 minutes before the fixture is due to kick off. This depends on how quickly team lineup information is known.
This feed is updated upon stops in play within the game including the following types of events:-

- Goal
- Shot at Goal
- Free Kick
- Offside Given
- Cards
- Start / End Half
- Corner won/lost
- Substitution

The feed should update approximately every 90 seconds and will be pushed out automatically in event of key actions as above. There will however always be a gap of at least 30 seconds between files unless there is a goal scored, in which case a file will be produced immediately. Please note that there may be a gap of several minutes between feeds if no key events take place.

Postponed Games

In a pre-match scenario where a match is postponed, Opta will produce and deliver a file which looks as per below:

```
<?xml version="1.0" encoding="UTF-8"?> <SoccerFeed
TimeStamp="20101217T133514+0000"> <SoccerDocument Type="Latest"
uID="f321827"> <Competition uID="c8"> <Country>England</Country>
<Name>English Barclays Premier League</Name> <Stat
Type="season_id">2010</Stat> <Stat Type="season_name">Season
2010/2011</Stat> <Stat Type="symid">EN_PR</Stat> <Stat
```

```
Type="matchday">18</Stat> </Competition> <MatchData> <MatchInfo
MatchType="Regular" Period="PreMatch">
<Date>20101218T150000+0000</Date> <Result Type="Postponed" />
</MatchInfo> <TeamData Score="0" Side="Home" TeamRef="t3"> </TeamData>
<TeamData Score="0" Side="Away" TeamRef="t110"> </TeamData>
</MatchData> </SoccerDocument> </SoccerFeed>
```

The Opta F1 file will also present the specific game as being postponed and an updated F1 file will be delivered. The relevant match data will look something like this within the feed:

```
<MatchData uID="g321827"> <MatchInfo MatchDay="18" Period="Postponed"
Venue_id="3250"> <Date>2010-12-18 15:00:00</Date> <TZ>GMT</TZ>
<</MatchInfo> <Stat Type="Venue">Emirates Stadium</Stat> <Stat
Type="City">London</Stat> <Stat Type="Postponed">Frozen Pitch</Stat>
<TeamData Score="" Side="Home" TeamRef="t3" /> <TeamData Score=""
Side="Away" TeamRef="t110" /> </MatchData>
```

When the rescheduled fixture's time and date is announced, the match ID will remain the same, as will the matchday, but the game date will be updated and the status will revert to pre-match:

```
<MatchData uID="g321827"> <MatchInfo MatchDay="18" Period="PreMatch"
Venue_id="3250"> <Date>2011-02-23 19:45:00</Date> <TZ>GMT</TZ>
</MatchInfo> <Stat Type="Venue">Emirates Stadium</Stat> <Stat
Type="City">London</Stat> <TeamData Score="" Side="Home" TeamRef="t3"
/> <TeamData Score="" Side="Away" TeamRef="t110" /> </MatchData>
```

However we will not retrigger a new F9 file whilst the game remains in its postponed state – so this postponed feed will still exist on your server as the latest file for this mach.

On the day of the rescheduled match, the first F9 feed that we deliver will be in-line with a normal F9 delivery ie displaying the team line-ups prior to the match kicking off.

Abandoned games

In the live-match scenario where a game kicks off and has to be abandoned part-way through, you will have already been receiving a series of F9 feeds with live stats; however to notify you that the game has been abandoned during play, the value of the 'Type' attribute within the 'Result' element in the F9 feed will change as below:

<Result Type="Abandoned" />

The Opta F1 file will also present the specific game as being abandoned and an updated F1 file will be delivered. The relevant match data will look something like this within the feed:

```
<MatchData uID="g321827"> <MatchInfo MatchDay="18" Period="Abandoned"
Venue_id="3250"> <Date>2010-12-18 15:00:00</Date> <TZ>GMT</TZ>
</MatchInfo> <Stat Type="Venue">Emirates Stadium</Stat> <Stat
Type="City">London</Stat> <Stat Type="Abandoned">Fog</Stat> <TeamData
Score="" Side="Home" TeamRef="t3" /> <TeamData Score="" Side="Away"
TeamRef="t110" /> </MatchData>
```

When the rescheduled fixture's time and date is announced, the match ID will remain the same, as will the matchday, but the game date will be updated and the status will revert to pre-match:

<MatchData uID="g321827"> <MatchInfo MatchDay="18" Period="PreMatch" Venue_id="3250"> <Date>2011-02-23 19:45:00</Date> <TZ>GMT</TZ> </MatchInfo> <Stat Type="Venue">Emirates Stadium</Stat> <Stat Type="City">London</Stat> <TeamData Score="" Side="Home" TeamRef="t3" /> <TeamData Score="" Side="Away" TeamRef="t110" /> </MatchData>
Dependent on the competition that the abandoned game relates to, the rescheduled game will either restart from the beginning at 0-0 (eg English Premier League), or it will restart at the minute of abandonment and continue on the same score (eg Italian Serie A). This restart structure is decided by the official league body and not by Opta.

Depending on which of those two restart rules the competition employs, it will determine whether you will receive a brand new F9 file like any other non-abandoned match (eg Premier League), or a partially complete F9 file that picks up from where the previously abandoned F9 feed left off.

How to determine whether a match has finished?

When a match is in play, the top part of the feed containing the match information will typically look as follows (highlighted relevant information):
<?xml version="1.0" encoding="ISO-8859-1"?> <SoccerFeed TimeStamp="20120314T091010+0000"> <SoccerDocument Type="Latest" uID="f360733"> <Competition uID="c8"> <Country>England</Country> <Name>English Barclays Premier League</Name> <Stat Type="season_id">2011</Stat> <Stat Type="season_name">Season 2011/2012</Stat> <Stat Type="symid">EN_PR</Stat> <Stat Type="matchday">27</Stat> </Competition> <MatchData> <MatchInfo MatchType="Regular" Period="FirstHalf" TimeStamp="20120304T135845+0000"> <Attendance>52388</Attendance> <Date>20120304T120000+0000</Date> <Result Type="" /> </MatchInfo>
Where <SoccerDocument Type="Latest" illustrates that this is the latest data from a live match and Period="FirstHalf" tells you the period of the game; in this case you can see the data relates to the latest information available from the first half of the match.

When a match reaches full time, those two highlighted elements will change to the following:
SoccerDocument Type="Result"
Period="FullTime"

***Note**, the time taken for you to receive the final feed containing this information that the game has finished, will take around 5 minutes after the final whistle; this is due to final checks by our collection team.

Is a game going to extra-time and/or penalty shoot-out?

When a match is being played in a competition that can go to extra-time and/or a penalty shoot-out, you shoudl check the period values for end of regular time and end of extra time.
When a game ends but is going to extra time it will always have the following status:
FullTime90
When a game ends but is going to penalties it will always have the following status:
FullTimePens

You will only ever see a fulltime status if the match is not going to extra time or penalties.

Second leg of a two-legged match

If a match is the second leg of a two legged game (eg knock-out stages of the Champions League) then in addition to receiving the usual F8 feed information in the match file, we also include some information from the first leg, which is positioned at the end of the feed.

Within the first <SoccerDocument> tag of the feed, you will see something like the following:

<PreviousMatch MatchRef="f350712" MatchType="1st Leg" VenueRef="v1377" />
This indicates that there is a previous match relevant to the second leg (ie a first leg).

When this is the case, we include a second <SoccerDocument> tag at the end of the feed which follows directly on from the first closing </SoccerDocument> tag.
Therefore, in the feed that is a second leg of a two legged match, there will be two <SoccerDocument> tags – the first containing the information for the relevant match (ie the second leg) and the second tag will contain information for the first leg.

This information for the first leg will be similar to a usual F8 feed apart from a restricted amount of data in the <MatchData> and <TeamData> tags.
You will still be able to determine the winner (if there was one) from the first leg however by the usual means of:

<Result Type="NormalResult" Winner="t8" />
If you want to find more information relating to this match, you can trace the full feed from the first leg from our previous deliveries to you by taking note of the MatchRef=" " attribute within the <PreviousMatch> tag, eg:
<PreviousMatch MatchRef="f350712">

Understanding how formations work

Click here for appendix "Formation Explained" with diagrams to illustrate all formation types.

We display the formations that the teams are using within the <TeamData> tag. This element mostly contains the cumulative team data – split into first half, second half and total – but after this cumulative data has been inputted, towards the end of this element's data, the team's formation used in the match is displayed, eg:
<Stat Type="formation_used">4411</Stat>
The value (in this example, '4411') can be seen illustrated in the diagrams in the appendix "Formations Explained" here, where you'll see that this relates to Team Formation "6".
This is the formation that the team uses at the start the match and is not updated (regardless of tactical changes, substitutions, sending offs, injuries etc.).
The diagrams in the appendix "Formations Explained" show all of the possible formation numbers and the layout of players in each formation – note that team formation #1 is not used. These are all the formations that Opta can provide. Please email customerservice-uk@optasportsdata.com to find out a little more information if these tactical formations are not understood.

File naming convention

The file naming convention used for this feed is the following:
srml-{competition_id}-{season_id}-f{game_id}-matchresults.xml

Feed samples

Please click below to view a sample feed:
F9 example latest
F9 example with extra time & penalties and updated match times
F9 example of abandoned match
F9 example of suspended match
F9 example with red card (from second yellow)
F9 example with delayed match start
F9 example with retired player

Elements/attribute/value descriptions (expand all)

Tables detailing all elements, attributes and values:

Element	`<Soccerfeed>`
Nesting	`SoccerFeed`
Description	Root Element of all related fields
Attribute	**`TimeStamp`**
Description	Indicates when the file was created
Data type	Date/Time
Values	Dynamic (YYYYMMDDThhmmss+0000)

Element	`<SoccerDocument>`
Nesting	`SoccerFeed/SoccerDocument`
Description	Root Element of all related fields
Attribute	**`Type`**
Description	The status of the match; in play or finished
Data type	String

Values	• "Result" - The game has officially finished • "Latest" - Collection of data has not fully finished

Attribute	**`uID`**
Description	Unique ID of this fixture (**Note:** in the case of a match played over 2 legs, a 2nd Soccer Document node will show the first leg data.)
Data type	Positive Integer
Values	Dynamic

Element	`<Competition>`
Nesting	`SoccerFeed/SoccerDocument/Competition`
Description	Root of competition data

Attribute	**`uID`**
Description	Unique ID of this competition
Data type	Positive Integer
Values	Dynamic

Element	`<Name>`
Nesting	`SoccerFeed/SoccerDocument/Competition`
Description	Name of the competition

Element	`<Round>`
Nesting	`SoccerFeed/SoccerDocument/Competition`
Description	Root of round info

Attribute	**`RoundNumber`**
Description	Unique ID of this competition
Data type	Positive Integer
Values	Dynamic

Attribute	**`name`**
Description	Name of the round
Data type	String
Values	Dynamic

Attribute	**`Pool`**
Description	Group name/number that the specified game is from
Data type	String
Values	Dynamic

Element	`<MatchData>`
Nesting	`SoccerFeed/SoccerDocument/MatchData`
Description	Root of all the dynamic data within the game

Element	`<MatchInfo>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/MatchInfo`
Description	Root of Match Info

Attribute	**`Period`**
Description	Status of the match
Data type	String
Values	• "PreMatch" - This game is yet to start - i.e. the line-ups have just been announced • "FirstHalf" - The game is currently in the 1st half • "HalfTime" - Match is currently at half time • "SecondHalf" - Match is currently in the 2nd period • "ExtraFirstHalf" - Match is currently in the first period of extra time • "ExtraSecondHalf" - Match is currently in the second period of extra time • ExtraHalfTime - Match is currently in half time of the extra time period • "ShootOut" - The game is currently within a penalty shoot-out • "FullTime" - The referee has now stopped this game • "FullTime90" - The game has ended but is now going to extra time • "FullTimePens" - The game has ended but is now going to penalties

Attribute	**`MatchType`**
Description	The type of the match, regular, cup etc.
Data type	String

Values	• Regular - Typical 90 minute fixture - all league games will have this • Cup - Cup game which can have all 5 periods of the match being played in • Cup Gold - Golden Goal can be played • Replay - Cup game which was tied and is then replayed in a 2nd match • Cup English - Game can go to the end of the 2nd half of extra time but no penalty shoot-out • Cup Short - Game goes straight to penalties if teams are level after the second half • 2nd Leg Cup Short - 2nd Leg of game that goes staright to penalties if match is level • 1st Leg - game is the 1st leg of a 2 legged tie • 2nd Leg - game is the 2nd leg of a 2 legged tie and will potentially go to penalties if the teams are drawing after the 2nd half of the second leg • 2nd Leg Away Goal - Game is the 2nd leg of a 2-legged tie and will potentially go to penalties if the two teams cannot be separated by the away goal rule

Attribute	**`Attendance`**
Description	Shows the number of people in the crowd - currently available for English Leagues and German Bundesliga only
Data type	Positive Integer
Values	Dynamic

Element	`<MatchOfficial>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/MatchOffical`
Description	Root of match official data

Attribute	**`uID`**
Description	Opta's unique ID for the official
Data type	String
Values	N/A

Element	`<OfficialRef>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/ MatchOffical/OfficialRef`

Description	Root of match official data
Attribute	`Type`
Description	Currently Opta only records the main referee
Data type	String
Values	Main

Element	`<OfficialName>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/MatchOffical/OfficialName`
Description	Root of official name with two sub nodes (First and Last). Please note: there is no such thing as a "Known Name" for a referee in the Opta database

Element	`<Date>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/MatchInfo/Date`
Description	The date and time of this fixture is in the following format: "YYYYMMDDThhmmss+0000"

Element	`<PreviousMatch>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/PreviousMatch`
Description	Root of previous match data
Attribute	`MatchRef`
Description	Unique game ID for the 1st leg. This attribute only appears for matches played over 2 legs
Data type	String: "f" followed by integer
Values	Dynamic
Attribute	MatchType
Description	Regular, 2nd Leg etc.
Data type	String

Values	• Regular - Typical 90 minute fixture - all league games will have this • Cup - Cup game which can have all 5 periods of the match being played in • Cup Gold - Golden Goal can be played • Replay - Cup game which was tied and is then replayed in a 2nd match • Cup English - Game can go to the end of the 2nd half of extra time but no penalty shoot-out • Cup Short - Game goes straight to penalties if teams are level after the second half • 1st Leg - game is the 1st leg of a 2 legged tie • 2nd Leg - game is the 2nd leg of a 2 legged tie and will potentially go to penalties if the teams are drawing after the 2nd half of the second leg • 2nd Leg Away Goal - Game is the 2nd leg of a 2-legged tie and will potentially go to penalties if the two teams cannot be separated by the away goal rule

Attribute	VenueRef
Description	Unique ID of the venue where the 1st leg was played ("v" followed by positive integer)
Data type	String
Values	Dynamic

Element	`<Stat>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Stat`
Description	Root Element

Attribute	**Type**
Description	The match time attributes
Data type	Positive integer
Values	• match_time - The total length of the match • first_half_time - The total length of the first half • second_half_time - The total length of the second half • first_half_extra_time - The total length of the first half of extra time • second_half_extra_time - The total length of the second half of extra time

Attribute	**`Type="match_state"`**
Description	If there is a game affecting suspension, the feed will update during the suspension with this attribute. The attribute will disappear when the game has restarted
Data type	String
Values	When present the match_state will always be Suspended ***To be released 18th July 2016***
Attribute	**`Type="match_state_reason"`**
Description	The reason for the game suspension
Data type	String
Values	• Awaiting officials decision - used if the reason is unknown • Crowd trouble • Drinks break • Fight • Fire • Floodlight failure • Object thrown on pitch • Other reason • Referee injury • Spectator on pitch • Weather problem ***To be released 18th July 2016***
Description	Timestamp
Attribute	The time the period started
Data type	Date/Time
Values	first_half_start (YYYYMMDDThhmmss+0000) first_half_stop (YYYYMMDDThhmmss+0000) second_half_start (YYYYMMDDThhmmss+0000) second_half_stop (YYYYMMDDThhmmss+0000) first_half_extra_start (YYYYMMDDThhmmss+0000) first_half_extra_stop (YYYYMMDDThhmmss+0000) second_half_extra_start (YYYYMMDDThhmmss+0000) second_half_extra_stop (YYYYMMDDThhmmss+0000)

Element	`<Result>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/MatchInfo/Result`
Description	Root of previous stat data
Attribute	`Type`
Description	See below

Data type	String
Values	• NormalResult - A game finished in typcial circumstances (90min match) • Aggregate - this is displayed for matches played over 2 legs when there is a winner based on the total score over the 2 games. Note: it will only appear if Match Type = 2nd leg • AwayGoals - this is displayed for matches played over 2 legs when the two teams are level on goals scored, but tie is decided by the away goals rule. Note: it will only appear if Match Type = 2nd leg • Penalty Shootout - A game's result was decided after a penalty shoot-out had taken place • AfterExtraTime - A game's result was decided after extra time had been played • GoldenGoal - A game was decided on the golden goal Ruling • Abandoned - A game had been abandoned midway throught he play • Postponed - A game was postponed before the game was started • "Void" - The match has been deemed as void (e.g. team going into administration) • Delayed - The start of the match has been delayed
Attribute	`Winner`
Description	The team id of the winner of the tie goes here
Data type	Positive integer
Values	Dynamic
Attribute	`Minutes`
Description	The length of the delay in minutes
Data type	Positive integer
Values	The length the game has been delayed
Attribute	`Reason`
Description	These are the reasons which can lead to a postponement or abandonment of a match.
Data type	String

Values	• Crowd • Floodlight Failure • Frozen Pitch • Insufficient Players • Other • Suspended • Unknown • Waterlogged Pitch • Weather
Element	`<AssistantOfficials>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/AssistantOfficials`
Description	Root element of Assistant Officials

Element	`<AssistantOfficial>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/AssistantOfficials/AssistantOfficial`
Description	Root of previous stat data
Attribute	`FirstName`
Description	First name of official
Data type	String
Values	Dynamic
Attribute	`LastName`
Description	Surname of official
Data type	String
Values	Dynamic
Attribute	`Type`
Description	Type of assistant official
Data type	String
Values	• Linesman 1 • Linesman 2 • Fourth official • Additional assistant referee 1 • Additional assistant referee 2
Attribute	`uID`

Description	Official's unique ID
Data type	Positive Integer
Values	Dynamic

Element	`<TeamData>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData`
Description	Root of team data

Attribute	`Score`
Description	The score for the specified team. If the game was decided on penalty shoot-out this will be still set to the score that the game would have finished at after extra time
Data type	String
Values	Dynamic

Attribute	`Side`
Description	N/A
Data type	String
Values	"Home" or "Away"

Attribute	`TeamRef`
Description	This value will have the unique team ID. We recommend that you remove the 't' from the string and use as an integer as other feeds that Opta provide may represent this as the integer only
Data type	Positive integer
Values	Dynamic

Attribute	`ShootOutScore`
Description	The score that the team achieved within the penalty shoot-out period
Data type	Positive integer
Values	Dynamic

Element	`<Goal>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/Goal`
Description	Root of previous stat data

Attribute	`Period`
Description	These are the 4 distinct periods thata goal can be scored in. If you would like to see the score of a penalty shoot-out, you should use the ShootOut element below
Data type	String
Values	• FirstHalf • SecondHalf • ExtraFirstHalf • ExtraSecondHalf
Attribute	`PlayerRef`
Description	This value will be the ID of the player that scored the goal. To get the corresponding player names, you will need to receive the F40 squads feed
Data type	Positive integer
Values	Dynamic
Attribute	`Type`
Description	See below
Data type	String
Values	• Goal - Any typical goal • Own - denotes an own goal • Penalty - Denotes goal scored from a penalty. Even in a penalty shoot-out
Attribute	`Eventnumber`
Description	Unique within the game. The first digit indicates the half/period. The middle digits indicate the event minute. the last indicates the event number (from cards, goals and subs) which are processed in blocks indepedently of chronological order; cards first, then goals, then substitutes
Data type	Positive Integer
Values	Dynamic
Attribute	`EventID`
Description	Unique ID within the entriety of Opta's all-time event database
Data type	Positive Integer (From early 2019 this will change to Positive BIGINT)
Values	Dynamic

Attribute	`Time`
Description	Minute that the goal was scored in
Data type	Positive integer
Values	Dynamic
Attribute	`Timestamp`
Description	ISO timestamp indicating the actual time of the event. This will indicate GMT time and whether or not BST has been added
Data type	String
Values	Dynamic (YYYYMMDDThhmmss+hhhh (GMT))
Attribute	`uID`
Description	Unique Goal Event within Game
Data type	Positive integer
Values	Dynamic
Attribute	`SoloRun`
Description	When a player takes five or more touches of the ball from receiving the ball to scoring, this attribute will be present
Data type	Positive integer
Values	When present it will always be set to "1" SoloRun is only collected for games with detail_id= 1 or 2 ***To be released 18th July 2016***

Element	`<Assist>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Teamdata/Goal/Assist`
Description	Root of previous stat data
Attribute	`PlayerRef`
Description	Unique ID of the player who made the assisting pass for the goal
Data type	Positive integer
Values	Dynamic

Element	`<2ndAssist>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/Goal/2ndAssist`
Description	Root of previous stat data

Attribute	`PlayerRef`
Description	Unique ID of the player if there was a 2nd assist, i.e pass to create the opportunity for the player making the assist
Data type	Positive integer
Values	Dynamic

Element	`<MissedPenalty>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/MissedPenalty`
Description	Root of previous stat data

Attribute	`EventNumber`
Description	Same as above
Data type	Positive integer
Values	Dynamic

Attribute	`Period`
Description	4 distinct periods
Data type	String
Values	• FirstHalf • SecondHalf • ExtraFirstHalf • ExtraSecondHalf

Attribute	`PlayerRef`
Description	Unique ID of the player missing the penalty
Data type	Positive integer
Values	Dynamic

Attribute	`Time`
Description	Match time that penalty was missed - note this node only appears for regular play penalties, not shoot-outs
Data type	Positive integer
Values	Dynamic

Attribute	`EventID`
Description	Unique ID within the enitrety of Opta's all-time database

Data type	Positive integer (From early 2019 this will change to Positive BIGINT)
Values	Dynamic
Attribute	`uID`
Description	ID of the missed penalty for the team taking it within this game.
Data type	Positive integer
Values	Mp{teamID}-number
Attribute	Type
Description	Type of penalty miss
Data Type	String
Values	• missed - ball went wide of target or over the bar • post - ball hit post or woodwork • saved - ball was saved by goalkeeper

Element	`<Booking>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/Booking`
Description	Root of previous stat data
Attribute	`Card`
Description	See below
Data type	String
Values	• Yellow - The booking was a yellow card • Red - The booking resulted in the player being sent off
Attribute	`CardType`
Description	See below
Data type	String
Values	• Yellow - The booking was a yellow card • SecondYellow - The booking was second yellow card • StraightRed - The booking was a straight red card
Attribute	`Period`
Description	These are the 5 distinct periods in which a booking can be made
Data type	String

Values	• FirstHalf • SecondHalf • ExtraFirstHalf • ExtraSecondHalf • ShootOut
Attribute	`Reason`
Description	See values below
Data type	String
Values	
Attribute	`Time`
Description	Minute of booking
Data type	Positive integer
Values	Dynamic
Attribute	`uID`
Description	Unique booking ID within the game
Data type	Positive integer
Values	Dynamic
Attribute	`EventID`
Description	Unique ID within the entriety of Opta's all-time event database.
Data type	Positive integer (From early 2019 this will change to Positive BIGINT)
Values	Dynamic
Attribute	`EventNumber`
Description	Unique event number within game, definition above on goal element
Data type	positive integer
Values	Dynamic

Element	`<Substitution>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/PlayerLineUp/Substitution`
Description	Root of previous stat data
Attribute	`Period`

Description	See below
Data type	String
Values	• 1 - First Half • 2 - Second Half • 3 - Extra FIrst Half • 4 - Extra Second Half
Attribute	`Reason`
Description	See below
Data type	String
Values	• Injury - Player was substitued due to an injury • Tactical - Player was substitued for tactical reasons
Attribute	`SubOff`
Description	Unique Player ID of player being substituted out of the game
Data type	String
Values	Dynamic
Attribute	`SubOn`
Description	Unique Player ID of player being brought into the game
Data type	String
Values	Dynamic
Attribute	`SubstitutePosition`
Description	See below
Data type	String
Values	• 1 - Goalkeeper • 2 - Defender • 3 - Midfielder • 4 - Striker
Attribute	`uID`
Description	Unique substitution event within the game
Data type	Positive integer
Values	Dynamic
Attribute	`EventID`
Description	Unique ID within the entriety of Opta's all-time event database.

Data type	Positive integer (From early 2019 this will change to Positive BIGINT)
Values	Dynamic
Attribute	`EventNumber`
Description	Unique event number within game, definition above on goal element
Data type	Positive integer
Values	Dynamic
Attribute	**`Retired`**
Description	This attribute will be shown if a player is forced to leave the field after his team have made their allocated substitutions, and he is unable to complete the game. It will not show otherwise
Data type	Positive integer
Values	Always set to "1" ***To be released 18th July 2016***

Element	`<ShootOut>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/ShootOut`
Description	Root of previous stat data
Attribute	`FirstPenalty`
Description	See below
Data type	Positive integer
Values	• 1 - Indictaes team who took 1st penalty • 0 - Indicates team who took 2nd penalty

Element	`<PenaltyShot>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/PenaltyShot`
Description	Root of penalty shot data
Attribute	`Outcome`
Description	See below
Data type	String
Values	• Scored - Goal Scored by player • Missed - Penalty Off Target • Saved - Penalty Saved by the goalkeeper

Attribute	`Event Number`
Description	The unique number of the event within the game
Data type	Positive integer
Values	Dynamic
Attribute	`PlayerRef`
Description	The unique ID of the player taking the penalty
Data type	Positive integer
Values	Dynamic
Attribute	`uID`
Description	The unique ID of the penalty within the penalty shoot-out of this game
Data type	Positive integer
Values	Dynamic

Element	`<PlayerLineUp>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/PlayerLineup`
Description	Root of all players in the line-up within a team

Element	`<MatchPlayer>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/PlayerLineUp/MatchPlayer`
Description	Root of matchplayer data
Attribute	`PlayerRef`
Description	Player ID for the specified player
Data type	String
Values	Dynamic
Attribute	`Position`
Description	Starting position of the player
Data type	String

Values	• Goalkeeper • Defender • Midfielder • Striker • Substitute
Attribute	`ShirtNumber`
Description	Shirt number assigned to the player at the start of the game. If the shirt number is unknown it will appear as ShirtNumber=""
Data type	String
Values	Dynamic
Attribute	`Status`
Description	See below
Data type	String
Values	• Start - player started the game • Sub - Player started on the pitch
Attribute	`SubPosition`
Description	The position the substitute usually plays in
Data type	String
Values	• Goalkeeper • Defender • Midfielder • Striker • Substitute
Attribute	`Captain`
Description	Attribute designating the captain within the team
Data type	Positive integer
Values	1- The player is the captain of the team This attribute will not appear for players that are not the captain

Element	`<Stat>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/TeamData/Stat`
Description	Root of of all stat data
Attribute	`Type`

Description	See below
Data type	String
Values	• List of values - This list can be found in the Appendix of this document (see link at top of page) • formation_place - See the formation description part of the Appendix document (see link at top of page)
Attribute	FH
Description	• Additional attributes showing data per half, only to appear for team data • Data from the first half only
Data type	String
Values	Dynamic
Attribute	SH
Description	Data from the second half only
Data type	String
Values	Dynamic
Attribute	EFH
Description	Data from the first half of extra time
Data type	String
Values	Dynamic
Attribute	ESH
Description	Data from second half of extra time
Data type	String
Values	Dynamic

Element	<Team>
Nesting	SoccerFeed/SoccerDocument/MatchData/team
Description	Root of of all team data
Attribute	uID
Description	The unqiue ID for a team
Data type	Positive integer
Values	Dynamic

Element	`<Name>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Team/Name`
Description	Root of all name data
Attribute	`N/A`
Description	The official name of this team. This will be the lcoalised team name when of a domestic league competition.
Data type	String
Values	Dynamic

Element	`<Player>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Team/Player`
Description	Root of all name data
Attribute	`Position`
Description	Starting position of the palyer
Data type	String
Values	• Goalkeeper • Defender • Midfielder • Striker • Substitute
Attribute	`uID`
Description	Unique ID of the player
Data type	Positive integer
Values	Dynamic

Element	`<PersonName>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Team/PersonName`
Description	Root of player name data
Attribute	`First`
Description	Player's first name
Data type	String
Values	N/A

Attribute	`Last`
Description	Player's last name
Data type	String
Values	N/A
Attribute	`Known`
Description	Player's known name or nick-name. Note: Please ignore first and last name if this node exists. For example, Ronaldinho will have a known name
Data type	String
Values	N/A

Element	`<TeamOfficial>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Team/ /MatchOfficial`
Description	Root of of all team official data
Attribute	`Type`
Description	See below
Data type	N/A
Values	• Manager - Name of person in overall charge of the team • Matchday Manager - Name of person in charge of the team for given match if different from Manager • uID - unique id for the managers

Element	`<PersonName>`
Nesting	`SoccerFeed/SoccerDocument/MatchData/Team/ TeamOfficial/PersonName`
Description	Root of of all team official name data
Attribute	`First`
Description	First Name
Data type	String
Values	N/A
Attribute	`Last`

Description	Surname
Data type	String
Values	N/A
Attribute	`Known`
Description	Known Name. Note: Please ignore First and Last name if this node exists. For example, Ronaldinho will have a known name.
Data type	String
Values	N/A

Element	`<Venue>`
Nesting	`SoccerFeed/SoccerDocument/Venue`
Description	Root of of all venue data
Attribute	`uID`
Description	Unique ID for the venue where the match will be held
Data type	Positive integer
Values	Dynamic

Element	`<Name>`
Nesting	`SoccerFeed/SoccerDocument/Venue/Name`
Description	Official name of the venue

Element	`<Country>`
Nesting	`SoccerFeed/SoccerDocument/Venue/Country`
Description	Country the venue is in