January 23, 2017

Building a sports stats chatbot using DialogFlow (API.AI)

(Note: this tutorial will be most useful if you know the rules of cricket, and you are a software developer. I have created this as a tutorial for software developers who wish to create chatbots on the API.AI platform. I would love to get your comments on how I can improve this article. Also, scroll to the end of post if you would like to get the ZIP file for the agent so you can play around with it within your API.AI console)

I am a big fan of the sport of cricket. Actually, I am really good at watching the game, and spouting off random stats, but not so good at playing the game. 🙂

So there is this tremendous live cricket scores/commentary website called Cricinfo, which was acquired by ESPN many moons ago and is now called ESPNCricinfo. The site has pretty much every statistic which might interest a cricket fan like me. On their site, they have something called StatsGuru (SG), which basically acts like a faceted search library for cricket related data.

The challenge with the StatsGuru interface

While the SG interface is really comprehensive, it is not super intuitive. It is a little tedious to use. Converting your natural language query to a format that SG can understand can be very helpful.

Enter the chatbot

I have been learning about chatbots recently, so I decided to create a bot which can take question in English as an input and convert it into the right query structure that you can u SG. To do this, I decided to use API.AI. The choice for API.AI was based on this quora answer as I wanted to start off with something easy to create and deploy even if it meant it was not a super accurate chatbot.

Modeling the problem

To create a chatbot on API.AI, you need to have an understanding of the following core ideas: entities and intents.

Entities are data models, sort of like class definitions and example class instances rolled into one.

Intents are triggers, that is, it is a "type of question" (asked by the bot consumer) which will be answered with bot-creator defined "type of answer". Intents themselves comprise of four parts - *user says, contexts, actions and responses.*

User says are the questions. You can specify different example ways of asking the same question, and the API.AI's machine learning/natural language processing engine is smart enough to not only use the templates you provide in your examples, but it can also infer slight variations of those templates. API.AI has two modes to input the user questions: the example mode and the template mode. In the example mode, you will provide multiple variants of the user says phrases by including entity values. In the template mode, you will provide variants of the user says phrases by using only placeholders.

Contexts refer to the current state of the bot. Suppose you have a phone tree like decision the chatbot has to make. Based on the user's prior response, (E.g. select 1 for checking account, select 2 for savings account), you can set the context based on the response. In turn, the next time around, the chatbot will see what the current context is and choose an action based on that.

Actions are a way to represent the chatbot's understanding of what the user said. You can use actions, for example, to trigger specific custom code. When the bot doesn't really understand what the user says, no action will be generated.

(A quick note: a bot doesn't understand questions the way humans do. We are not quite at that point yet. Here, when I say "didn't really understand", I explicitly mean that the bot cannot map the question to a response because it cannot find the pattern it is looking for in the question. Without going too deep, the bot doesn't understand because its pattern matching capabilities were not sufficient to extract the expected pattern).

Responses are the responses displayed to the bot consumer. A very important property of API.AI responses is that they support dynamic values based on the entities you have defined - that is, you can customize the response to the question with dynamic values. Again, this will be explained in detail in a later section.

Cricinfo StatsGuru URL structure

The chatbot will be returning links to web pages on Cricinfo based on the user's questions. Let us look at the URL format of some of these pages.

Player pages on Cricinfo

A typical player page on cricinfo has the following URL structure:

http://www.espncricinfo.com/ci/content/player/35320.html

The link above is for Sachin Tendulkar's page on Cricinfo. Sachin Tendulkar is one of the most popular cricket players (and that is something of an understatement).

So Cricinfo assigns a player ID to each player - note that we will only be focusing on international players here.

Team record pages on Cricinfo

You can also get team records by using a specific URL structure on Cricinfo:

http://stats.espncricinfo.com/ci/engine/records/index.html?id=2;type=team

The above URL shows all the records specific to the Australian team.

Player record pages on Cricinfo

And probably one of the most interesting types of stats we will be looking for is an individual player's records. The following is the query builder page:

http://stats.espncricinfo.com/ci/engine/player/34102.html?class=2;orderby=batted_score;type=allround;view=innings

Using the query builder page, you can select/deselect the appropriate checkboxes and it will result in query results page with a structure such as:

http://stats.espncricinfo.com/ci/engine/player/34102.html?class=2;filter=advanced;orderby=batted_score;template=results;type=allround;view=innings

The above URL will show Rohit Sharma's ODI scores ordered by highest to lowest.

Defining the entities

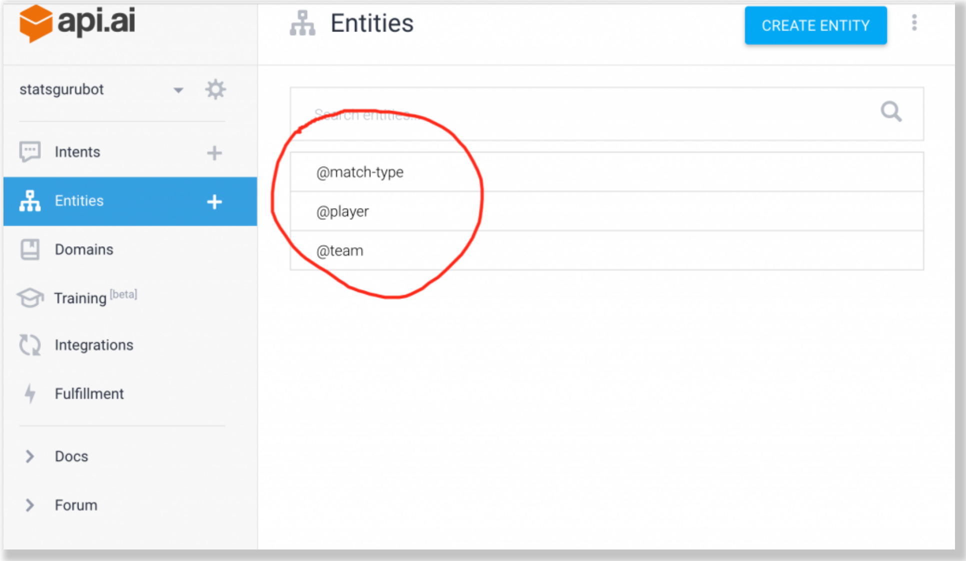
So we can see the following types of entities would be needed to get basic queries to work:

1. Player
2. Team
3. Game-type

The API.AI interface for entities allows us to name an entity, and then provide sample values for the entity so that it can do the mapping when it receives queries. For any given entity "instance", it is best to provide as many variations of it as you can especially because one of the strongest signals for the bot's pattern matching is a user input which perfectly matches the entity's value.

For the StatsGuru bot, these are the entities I have defined. Note that when you have a bi like for example the list of all international cricket players, you can simply upload a CSV or

JSON file which has the appropriate data. For the list of international cricketers, I just wrote a script and created a JSON file with all the international cricketers.



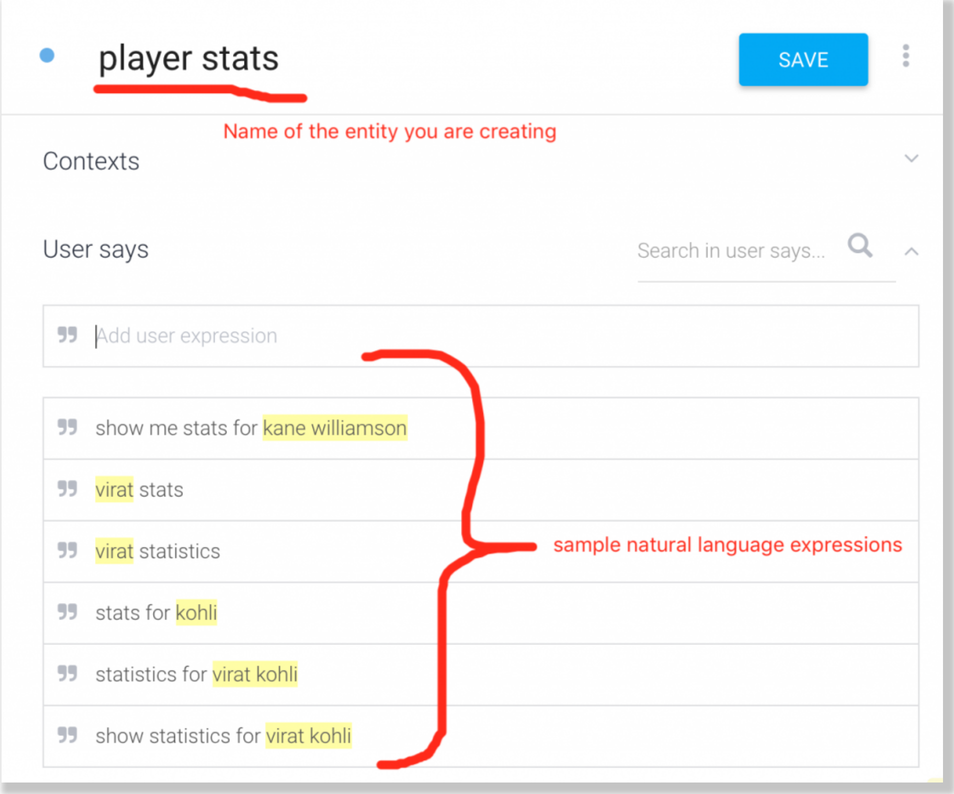
Defining an intent for the player page

After defining and populating the entities, we are now ready to create intents. I will start by demonstrating how to create an intent for the player page. If you remember, the player page has the following URL structure:

http://www.espncricinfo.com/ci/content/player/35320.html

On the Intents tab inside API.AI, create an intent (named "player stats") in this example. You need to do three steps to define an intent fully.

First, you need to provide a few variations of the query by putting yourself in the shoes of your chatbot user and imagining how they are likely to phrase a given question. See the image for the list of variations I defined for this particular intent.

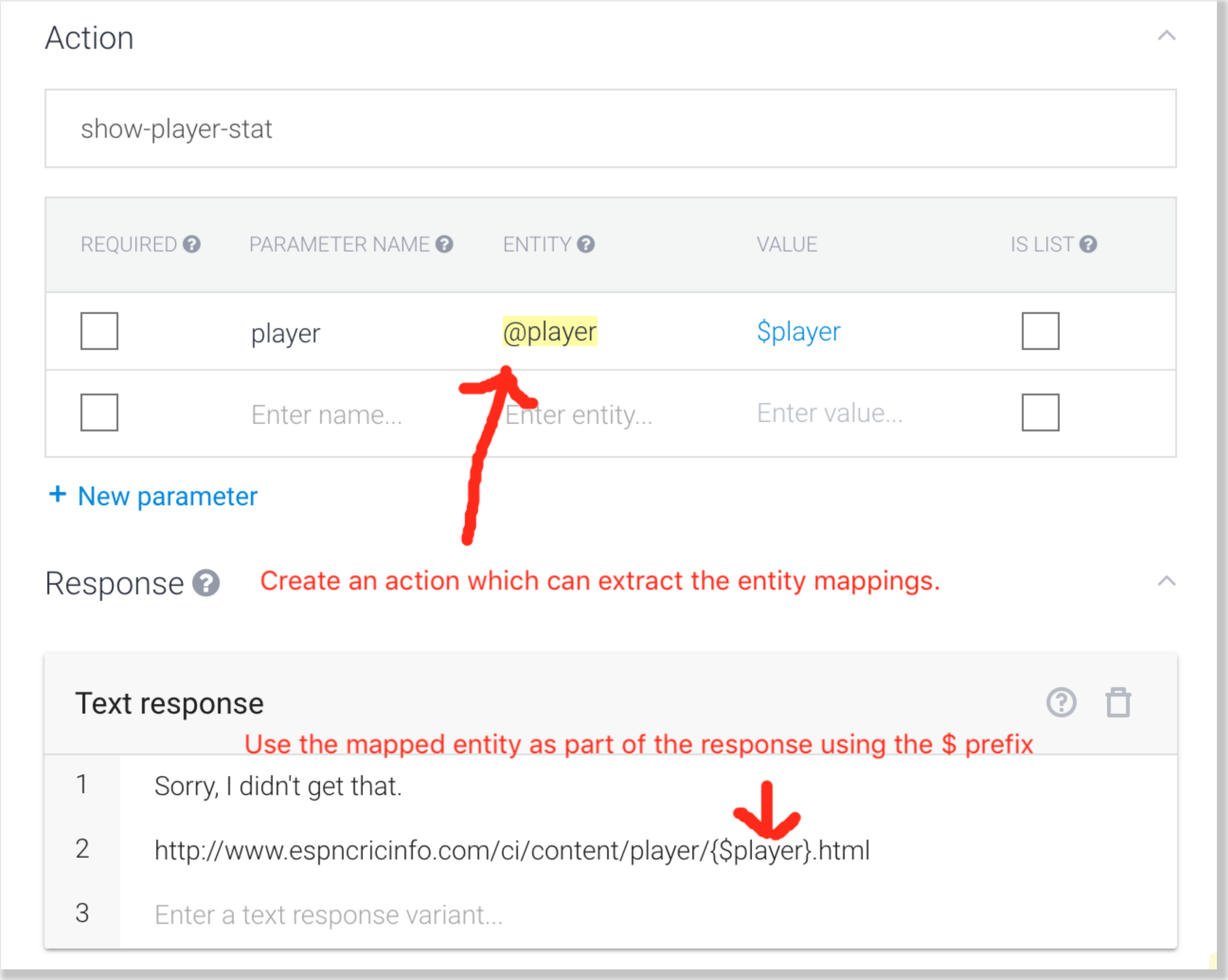


show me stats for kane williamson
virat stats
virat statistics
stats for kohli
statistics for virat kohli
show statistics for virat kohli

Second, you need to "map" the query to entities you have already defined by using an Action. For example, in this case the only entity we are interested in is the player entity and so we add it under the Action called "show-player-stat". The parameter name is a user friendly name for you to remember, but the Entity column has to match an existing entity you have created (when you type the @ sign, API.AI will try to autocomplete with existing entities) and the Value column is the output format which you will use in the Response.

Third, the Response will be specified based on how you want the chat bot to respond to this particular query. For example, you could have a chat bot which just plays the game Personality by answering Yes/No (albeit a little hard to create that - I should probably use it as a future project). In that case, the bot will just parse the query and respond with one of two predefined values.

In our case, however, the response has to be customized to the player whose statistics has been requested. So use the "value" you extracted from the query by using the format as shown in the figure below. From what I understand, the value text (i.e. $player) has to match the entity text (@player) exactly.



Action

show-player-stat

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	player	@player	$player	□
□	Enter name...	Enter entity...	Enter value...	□

+ New parameter

Response

Create an action which can extract the entity mappings.

Text response

Use the mapped entity as part of the response using the $ prefix

1 Sorry, I didn't get that.

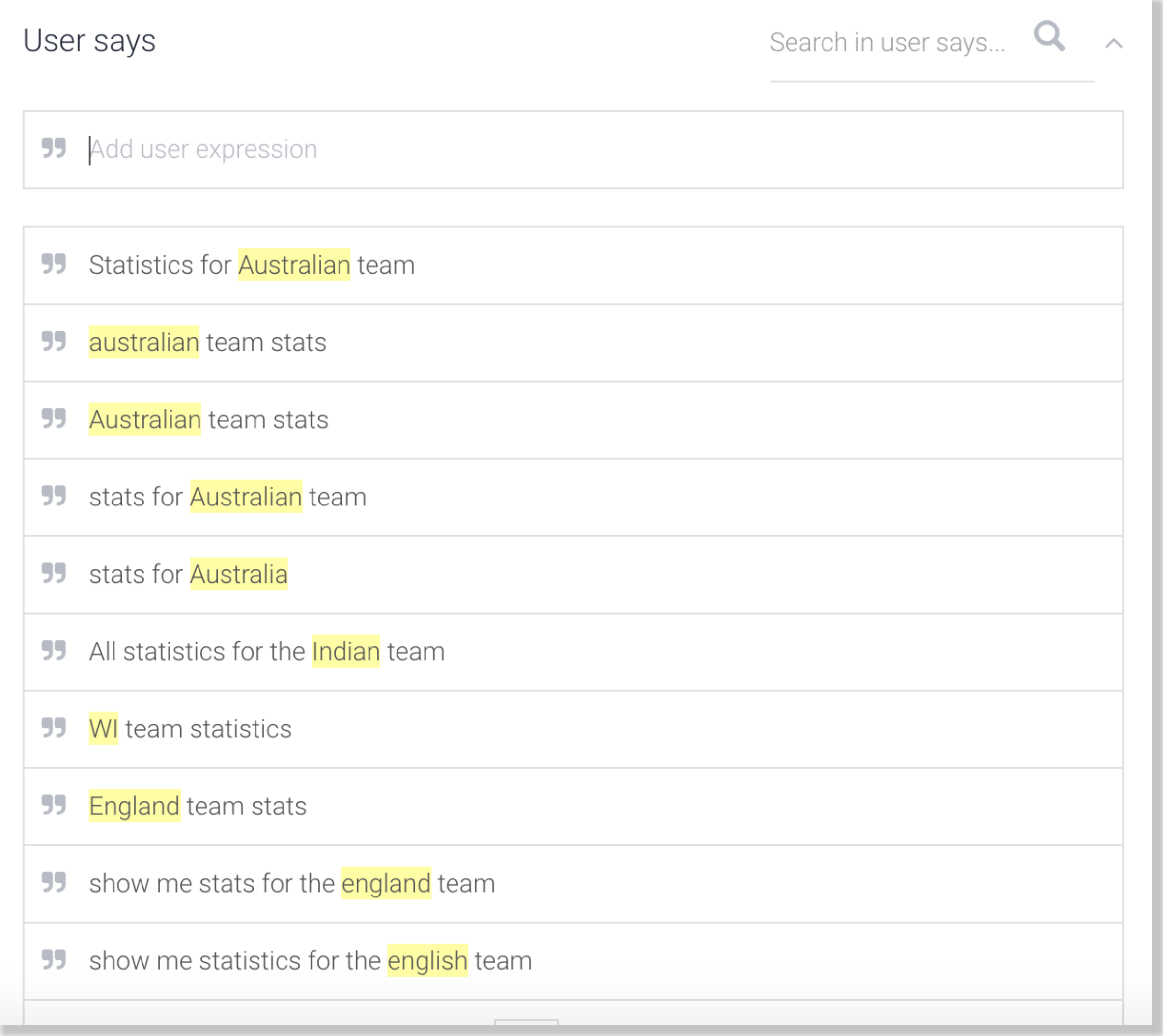
2 http://www.espncricinfo.com/ci/content/player/{$player}.html

3 Enter a text response variant...

Now Save the intent, and we are ready to query the chatbot.

Defining an intent for the team stats page

This is similar to defining an intent for the player stats. See screenshots below for the definition of the intent.



Statistics for Australian team

australian team stats

Australian team stats

stats for Australian team

stats for Australia

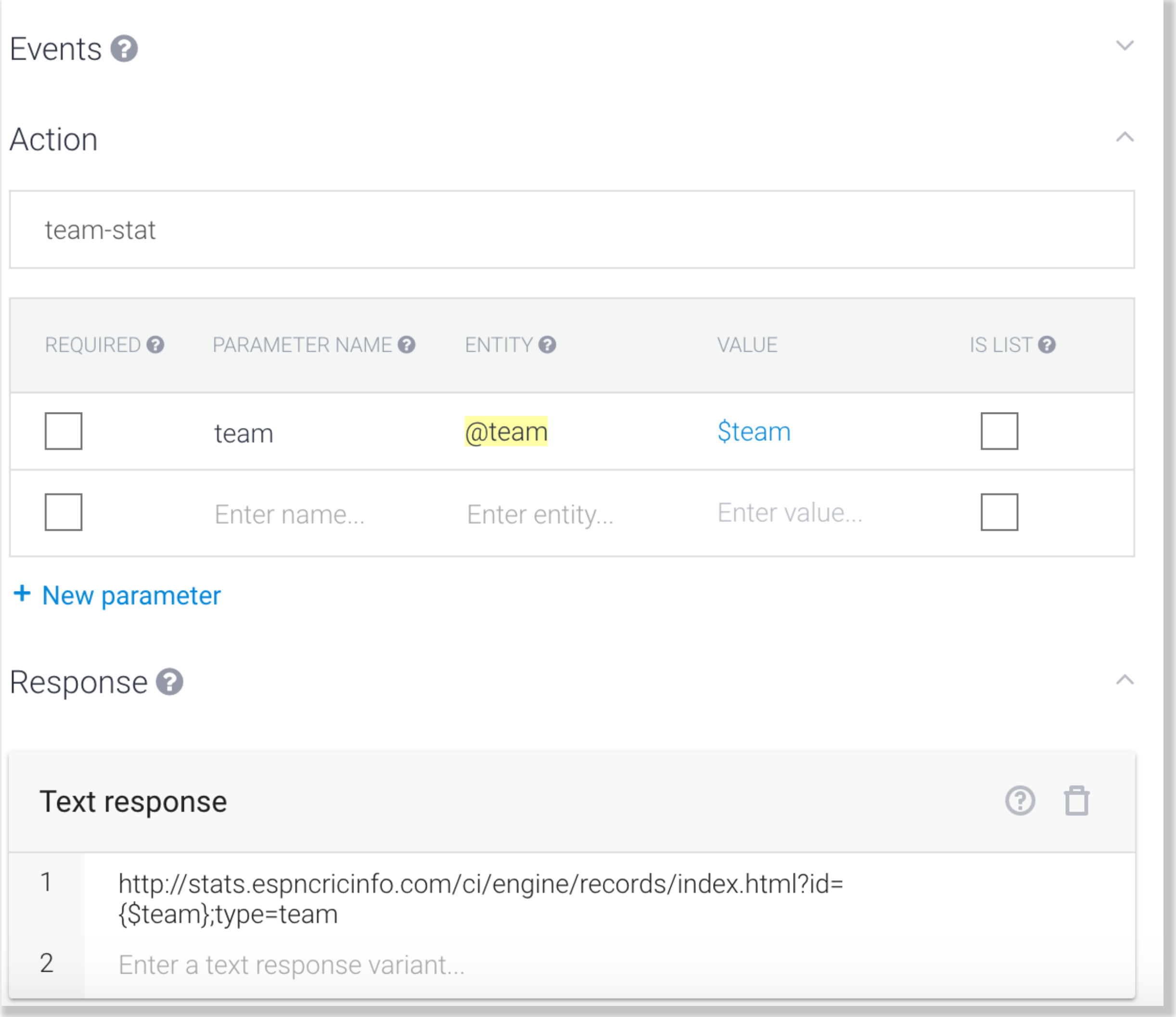
All statistics for the Indian team

WI team statistics

England team stats

show me stats for the england team

show me statistics for the english team



team-stat

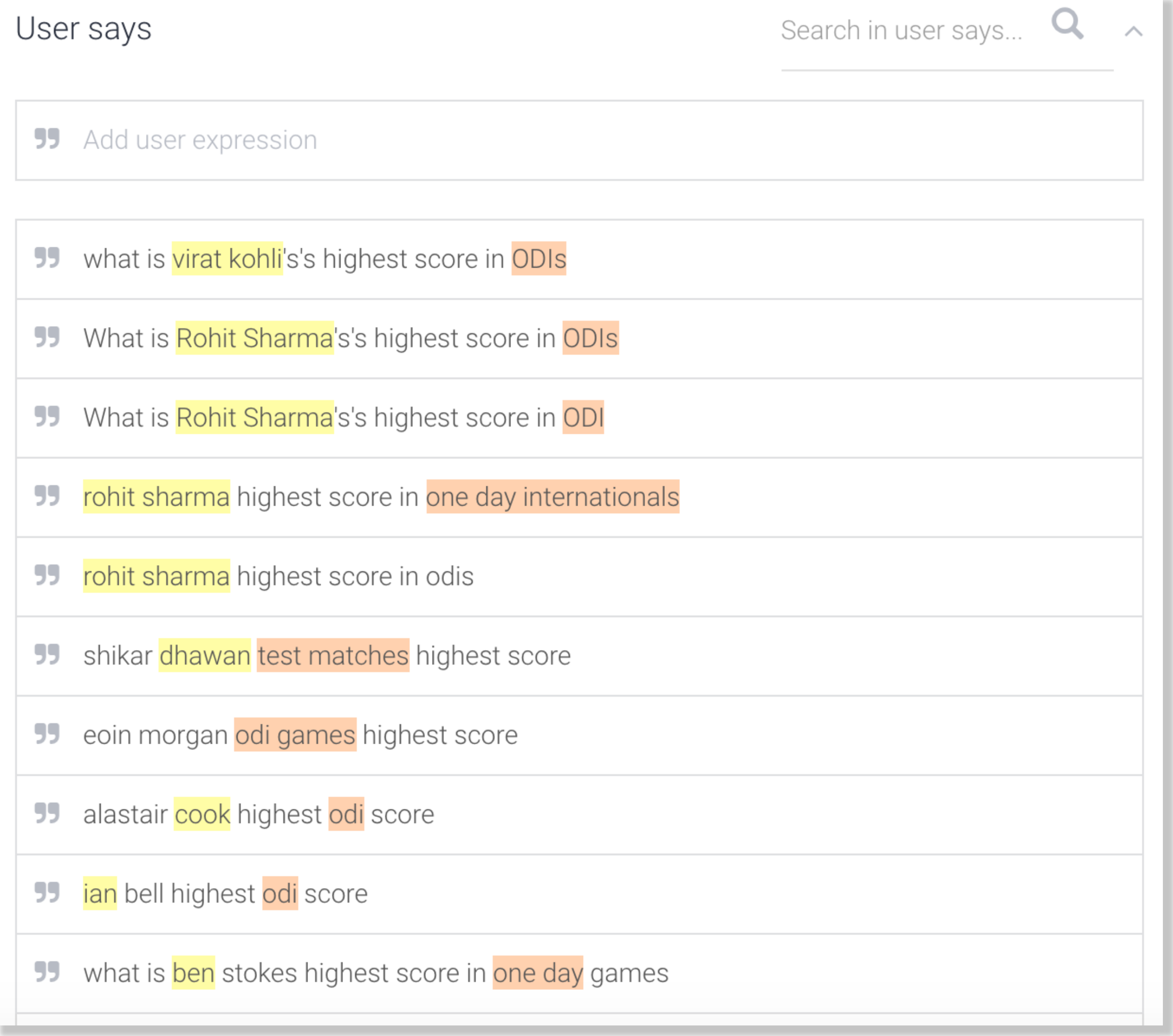
REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	team	@team	$team	□
□	Enter name...	Enter entity...	Enter value...	□

1 http://stats.espncricinfo.com/ci/engine/records/index.html?id={$team};type=team

2 Enter a text response variant...

Defining an intent for the highest Test/ODI/T20 score for a player

Notice that we need to extract two entities - the player name and the match type (Test vs ODI vs T20). See screenshots below.



User says

what is virat kohli's's highest score in ODIs

What is Rohit Sharma's's highest score in ODIs

What is Rohit Sharma's's highest score in ODI

rohit sharma highest score in one day internationals

rohit sharma highest score in odis

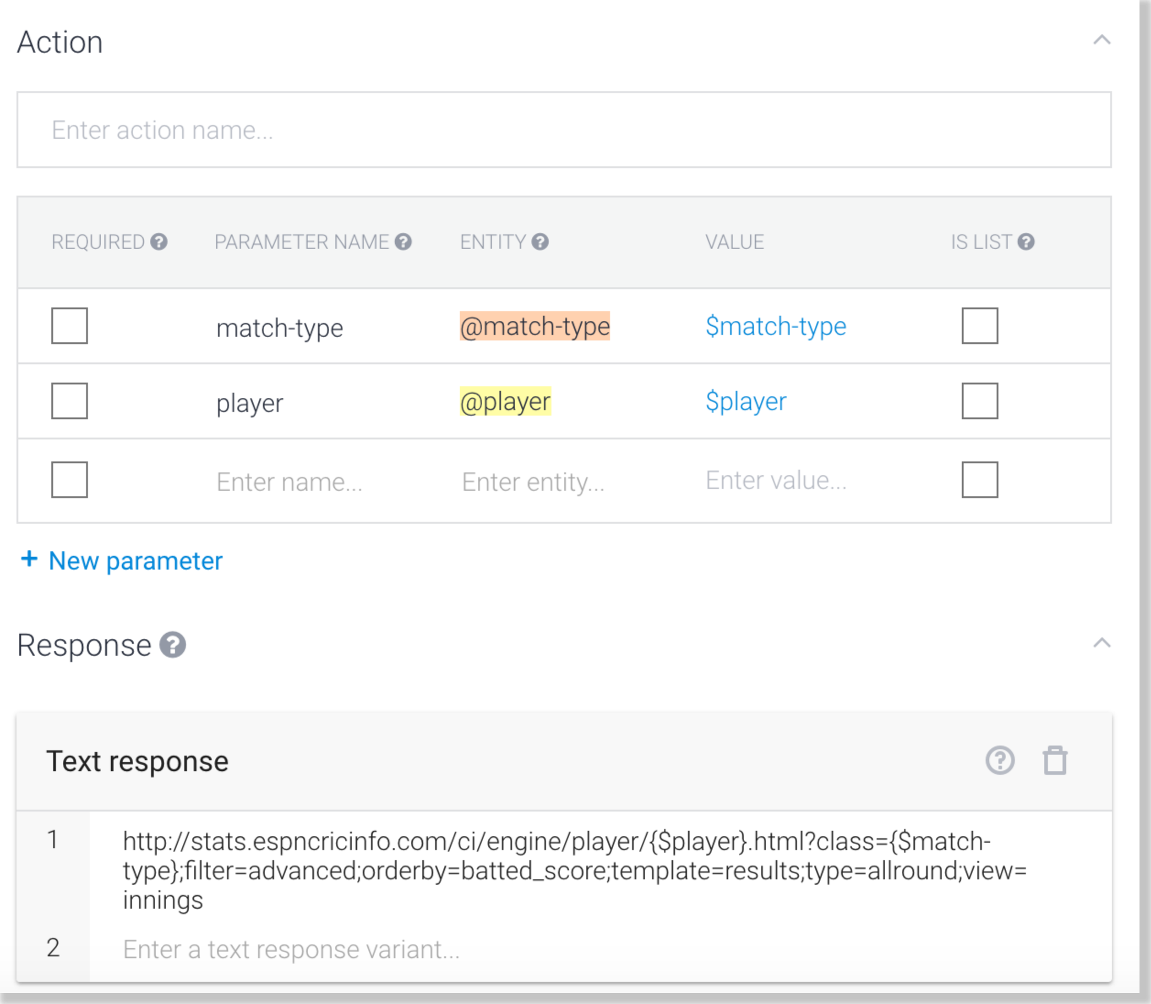
shikar dhawan test matches highest score

eoin morgan odi games highest score

alastair cook highest odi score

ian bell highest odi score

what is ben stokes highest score in one day games



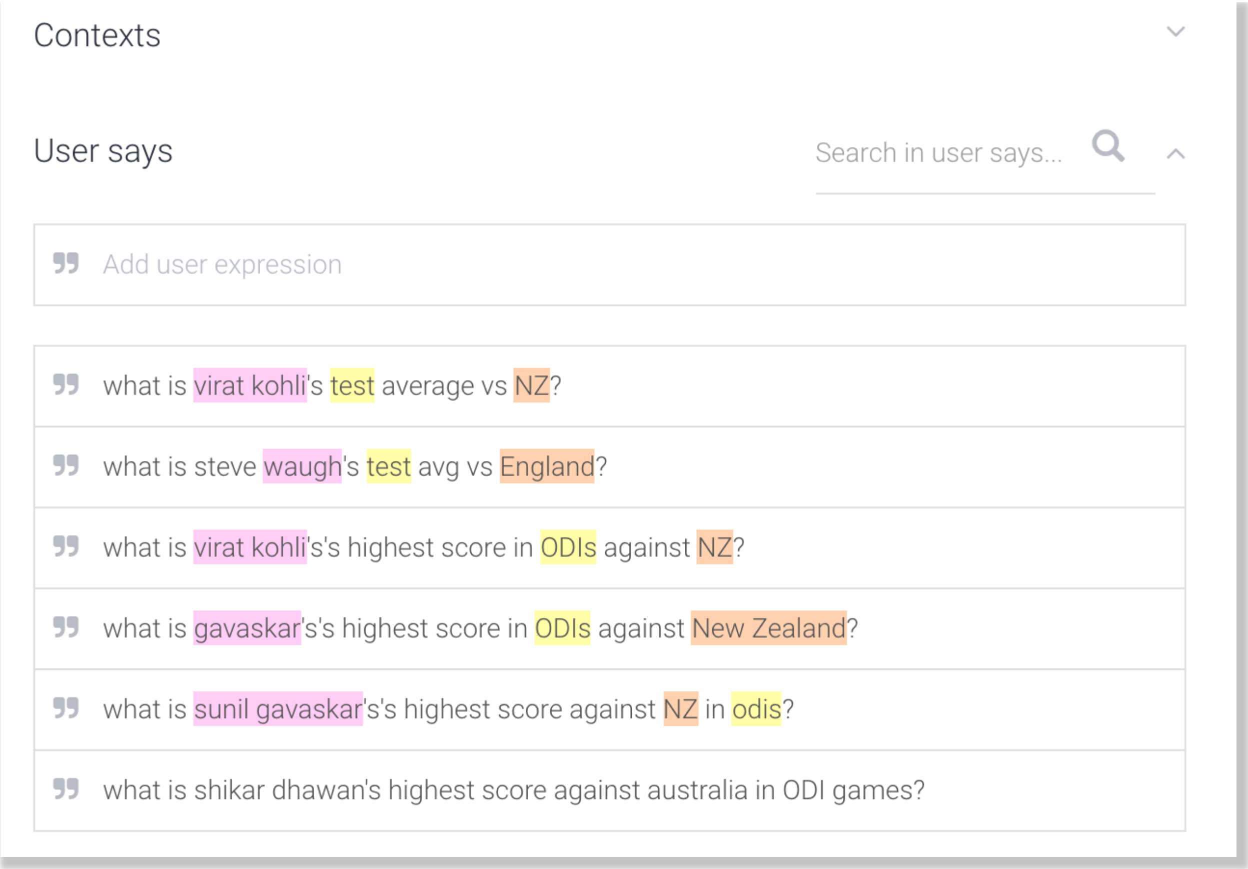
REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	match-type	@match-type	$match-type	□
□	player	@player	$player	□
□	Enter name...	Enter entity...	Enter value...	□

1 http://stats.espncricinfo.com/ci/engine/player/{$player}.html?class={$match-type};filter=advanced;orderby=batted_score;template=results;type=allround;view=innings

2 Enter a text response variant...

Defining an intent for the highest Test/ODI/T20 score for a player against a particular opponent

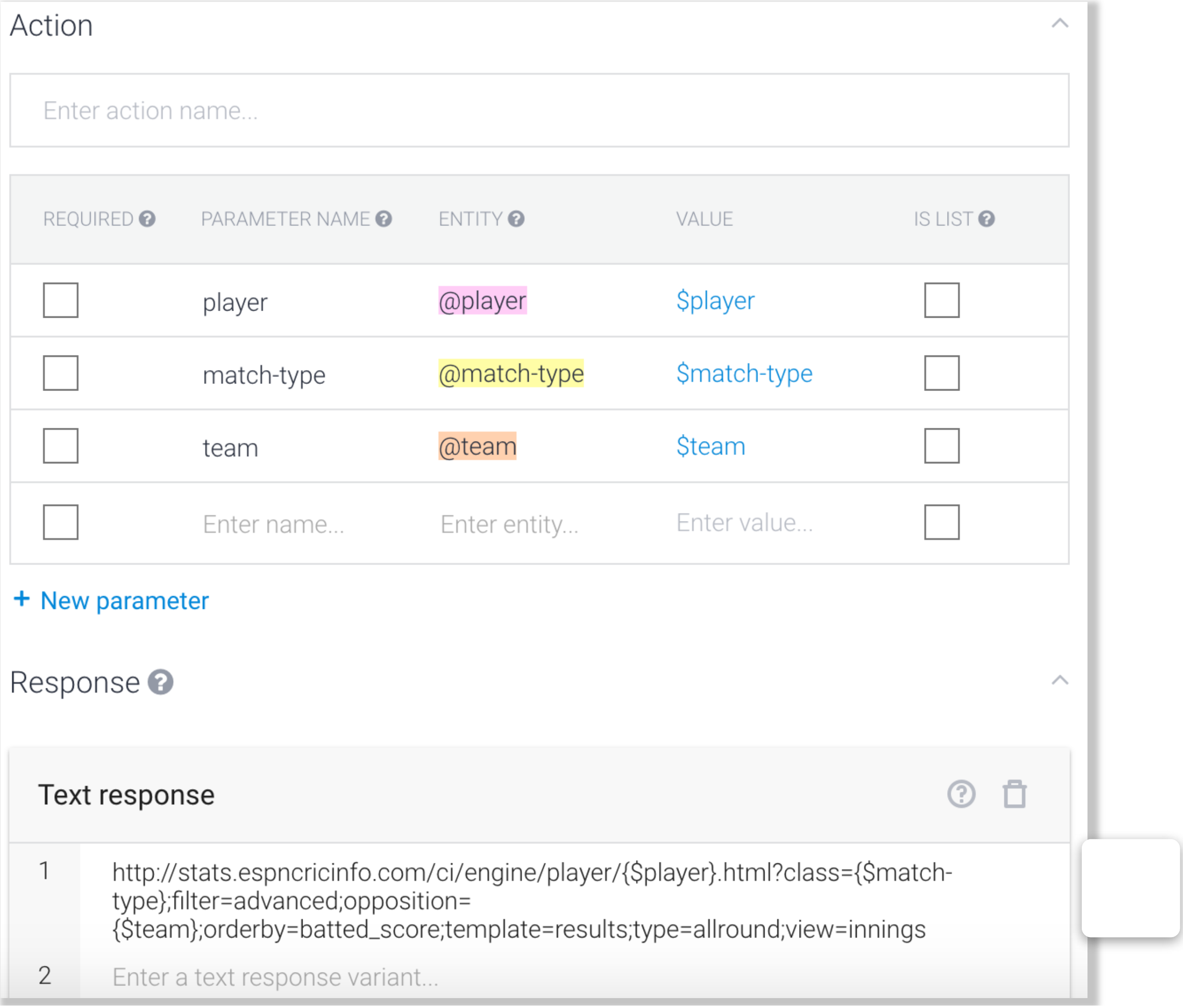
Here, we need to extract three entities - player, game type as well as opponent (team). See screenshots below:



Contexts

User says

- what is virat kohli's test average vs NZ?
- what is steve waugh's test avg vs England?
- what is virat kohli's's highest score in ODIs against NZ?
- what is gavaskar's's highest score in ODIs against New Zealand?
- what is sunil gavaskar's's highest score against NZ in odis?
- what is shikar dhawan's highest score against australia in ODI games?



Action

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	player	@player	$player	□
□	match-type	@match-type	$match-type	□
□	team	@team	$team	□
□	Enter name...	Enter entity...	Enter value...	□

Response

Text response

1 http://stats.espncricinfo.com/ci/engine/player/{$player}.html?class={$match-type};filter=advanced;opposition={$team};orderby=batted_score;template=results;type=allround;view=innings

2 Enter a text response variant...

Challenges in the current API.AI system

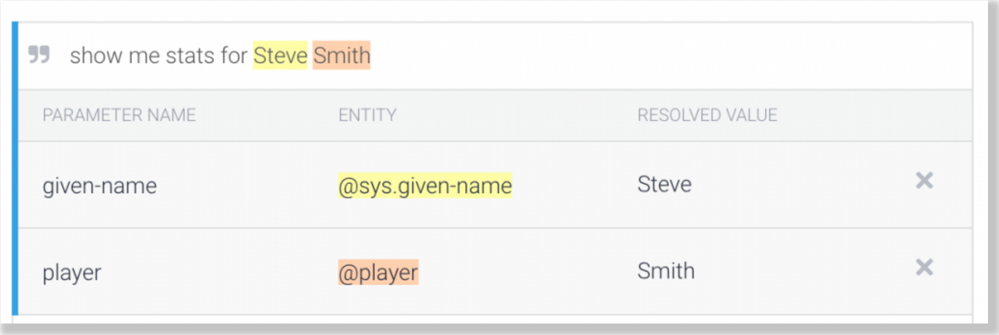
Of course, there are some challenges in the current implementation of API.AI. Here are three issues that I faced:

1 Resolving ambiguity in player names

One of the most famous players in cricket today in Steve Smith. Unfortunately, since both Steve and Smith are very common names, the chat bot cannot identify this name and instead gets the player stats for a different player.

2 Interference from built in entities

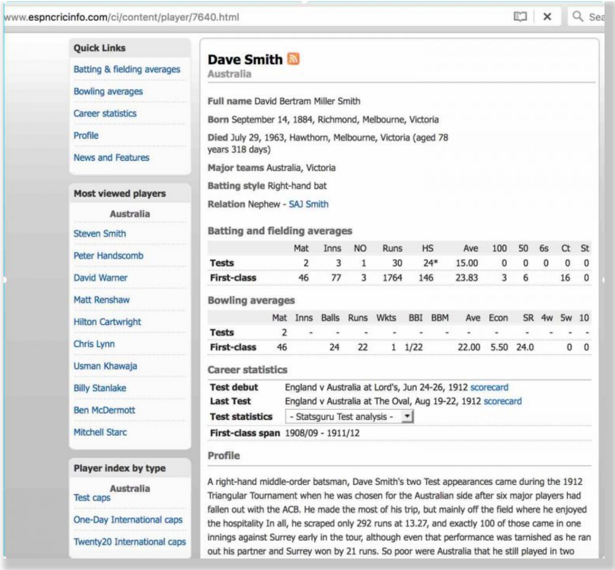
This is actually related to the previous point. For very common names, API.AI maps the name to its internally defined value. See screenshot below for Steve Smith.



show me stats for Steve Smith

PARAMETER NAME	ENTITY	RESOLVED VALUE
given-name	@sys.given-name	Steve
player	@player	Smith

As you can see, the name Steve Smith is mapped to given-name (API.AI internal entity) where @sys.given-name is Steve and the player is mapped to Smith. But which Smith? There are many Smiths in cricket. The actual Smith it returns is Dave Smith who is not a contemporary player.



www.espncricinfo.com/ci/content/player/7640.html

Quick Links

- Batting & fielding averages
- Bowling averages
- Career statistics
- Profile
- News and Features

Most viewed players

Australia

- Steven Smith
- Peter Handscomb
- David Warner
- Matt Renshaw
- Hilton Cartwright
- Chris Lynn
- Usman Khawaja
- Billy Stanlake
- Ben McDermott
- Mitchell Starc

Player index by type

Australia

- Test caps
- One-Day International caps
- Twenty20 International caps

Dave Smith

Australia

Full name David Bertram Miller Smith

Born September 14, 1884, Richmond, Melbourne, Victoria

Died July 29, 1963, Hawthorn, Melbourne, Victoria (aged 78 years 318 days)

Major teams Australia, Victoria

Batting style Right-hand bat

Relation Nephew - SAJ Smith

Batting and fielding averages

	Mat	Inns	NO	Runs	HS	Ave	100	50	6s	Ct	St
Tests	2	3	1	30	24*	15.00	0	0	0	0	0
First-class	46	77	3	1764	146	23.83	3	6		16	0

Bowling averages

	Mat	Inns	Balls	Runs	Wkts	BBI	BBM	Ave	Econ	SR	4w	5w	10
Tests	2	-	-	-	-	-	-	-	-	-	-	-	-
First-class	46		24	22	1	1/22		22.00	5.50	24.0		0	0

Career statistics

Test debut	England v Australia at Lord's, Jun 24-26, 1912 scorecard
Last Test	England v Australia at The Oval, Aug 19-22, 1912 scorecard
Test statistics	- Statsguru Test analysis -
First-class span	1908/09 - 1911/12

Profile

A right-hand middle-order batsman, Dave Smith's two Test appearances came during the 1912 Triangular Tournament when he was chosen for the Australian side after six major players had fallen out with the ACB. He made the most of his trip, but mainly off the field where he enjoyed the hospitality In all, he scraped only 292 runs at 13.27, and exactly 100 of those came in one innings against Surrey early in the tour, although even that performance was tarnished as he ran out his partner and Surrey won by 21 runs. So poor were Australia that he still played in two

Why did it choose this player? Unfortunately, it is something of a black box.

So there doesn't seem to be a way to "bias" the results in favor of, for e.g., current vs past players.

3 No hyperlinks within the chat box

I believe this is just a limitation of the chat box, but in my specific use case it certainly adds friction to the experience because you cannot click through from the URL in the chat box directly to the page of interest.

Want to replicate this bot?

If you would like to import this bot into API.AI to dig further, sign up below.

Get the agent ZIP file

- Sign up to download the agent ZIP file for the StatsGuruBot
- Import it into your own API.AI console

- Learn how to create a similar bot by reading the intents and entities

Enter Email

Name

SUBSCRIBE

Summary

I found the overall process of building a chat bot using API.AI both pleasant and intuitive. Given the competition it is facing from newer players such as converse.ai, I expect there will be plenty of updates to their system and (hopefully) they will act quickly to address some of these issues.

To see the chatbot in action, click here.

P.S. I have started writing a step by step guide to API.AI if that is something which would interest you.

Article by aravindmc / ChatBots / Leave a Comment