Oracle MOOC: Oracle Intelligent Bots

Session 1

Create Chatbots with Intents, Entities, Utterances, and Flows

In this lab, you'll define a simple financial chatbot and test it just to see how it works. After that, you'll begin a second, more complete chatbot that you'll complete over the course of these labs.

For your first chatbot, you'll create an intent and add a few utterances. Next, you'll create an entity and associate it with your intent. You will define the flow using BotML. With these pieces in place, you'll then train the chatbot and test it.

With the first chatbot done, you'll move on to create the second chatbot. To jumpstart your development of this more complex chatbot, you'll import its intent definitions. You'll also use this chatbot to find out about the different ways that you can train your chatbot.

What Do You Need?

For this lab, you'll need the following files from the `labfiles/code` directory of the `labfiles.zip`:

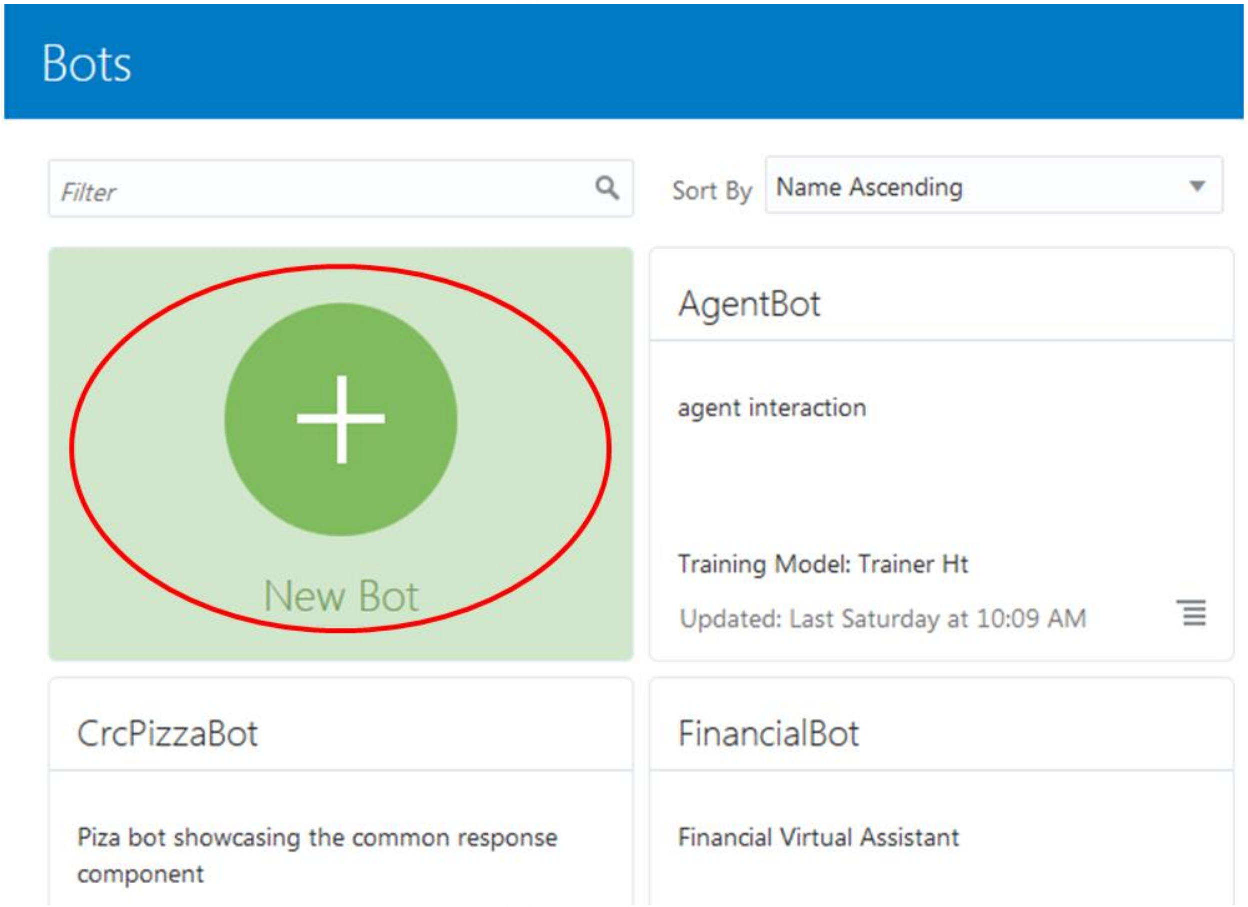
- `FirstBotYAML.txt`
- `MasterBot-Intents.csv`

Step 1: Create a Simple Banking Chatbot

In this section, you create a simple banking chatbot and examine the main artifact types.

1. Open your instance of Oracle Intelligent Bots and then click **New Bot**.

Oracle MOOC: Oracle Intelligent Bots



AgentBot

agent interaction

Training Model: Trainer Ht

Updated: Last Saturday at 10:09 AM

CrcPizzaBot

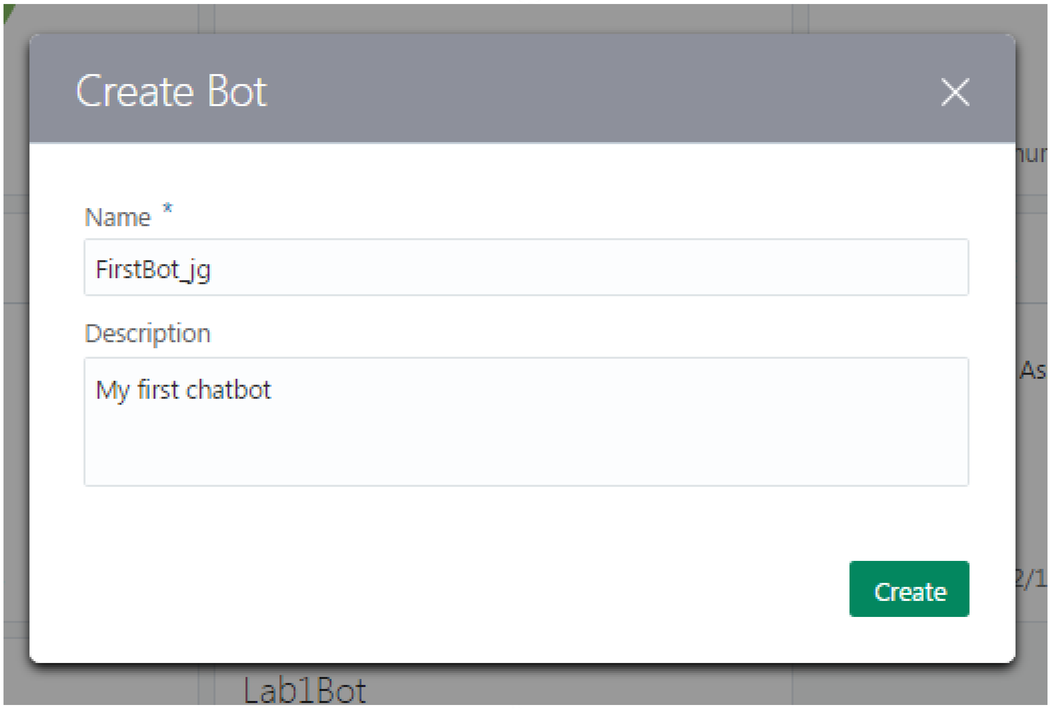
Piza bot showcasing the common response component

FinancialBot

Financial Virtual Assistant

2. In the Create Bot dialog, enter *FirstBot_XX*, where *XX* are your initials. Next, add a description and then click **Create**.

Oracle MOOC: Oracle Intelligent Bots



Create Bot

Name *

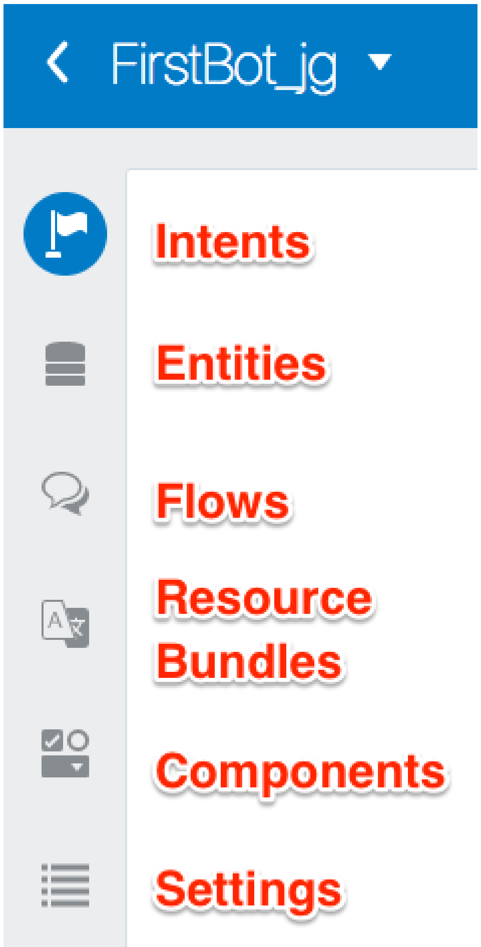
FirstBot_jg

Description

My first chatbot

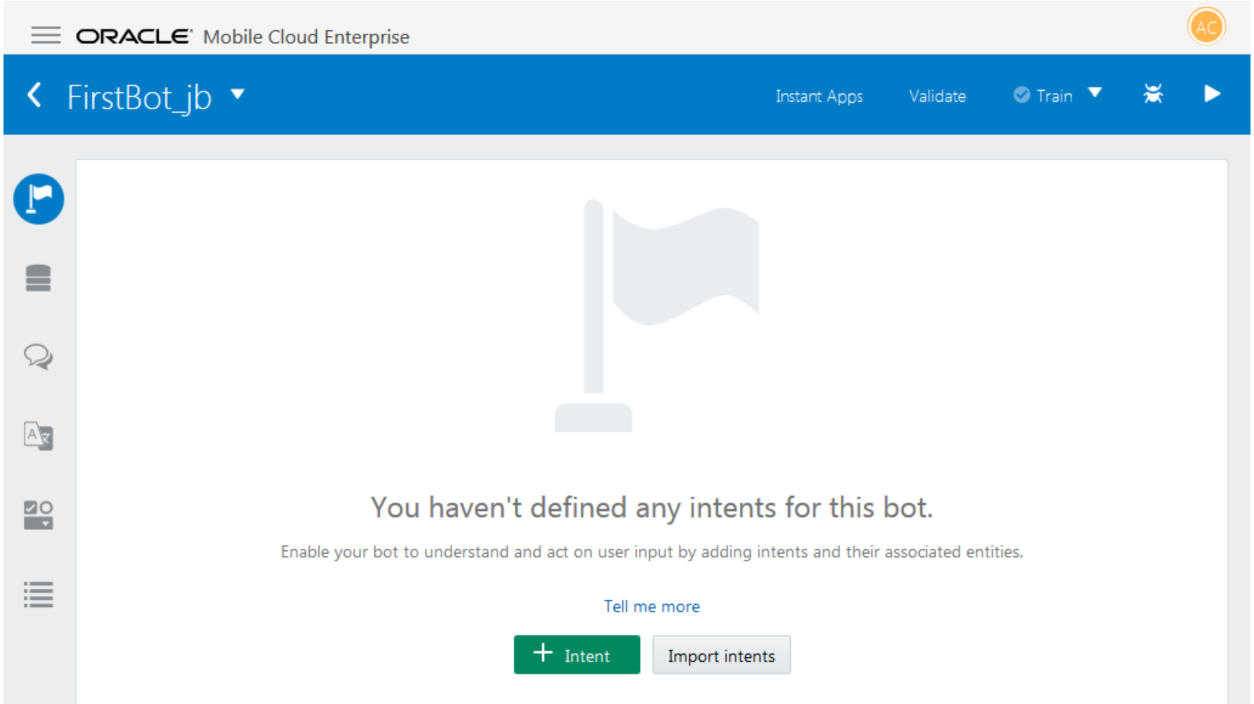
Oracle MOOC: Oracle Intelligent Bots

3. In the left navbar, you can see a list of icons that you use to navigate to your intents, entities, dialog flow, components, resource bundles, and settings.



4. By default, the Intents page is open, but as of this moment, you don't have any intents.

Oracle MOOC: Oracle Intelligent Bots

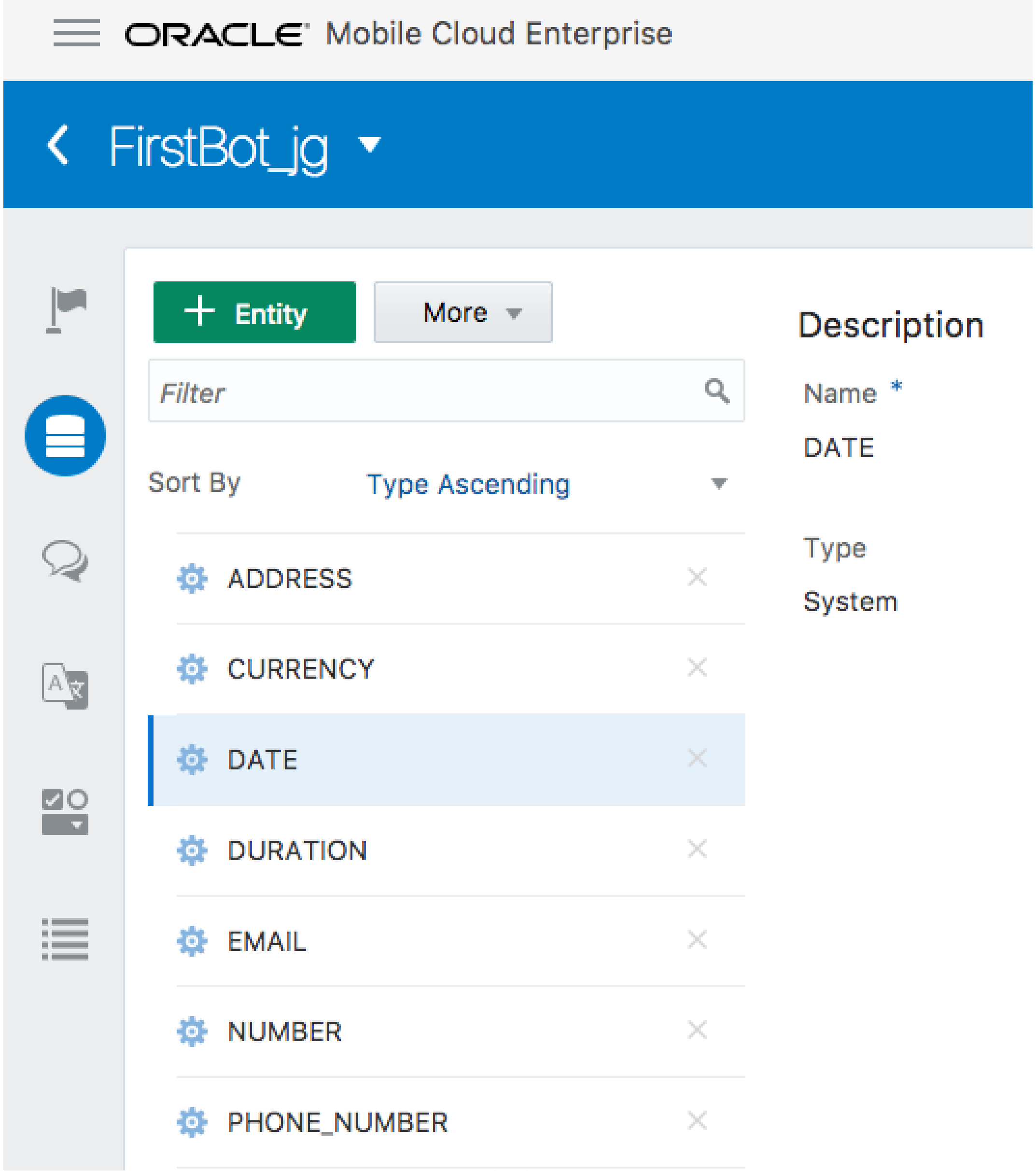


You haven't defined any intents for this bot.

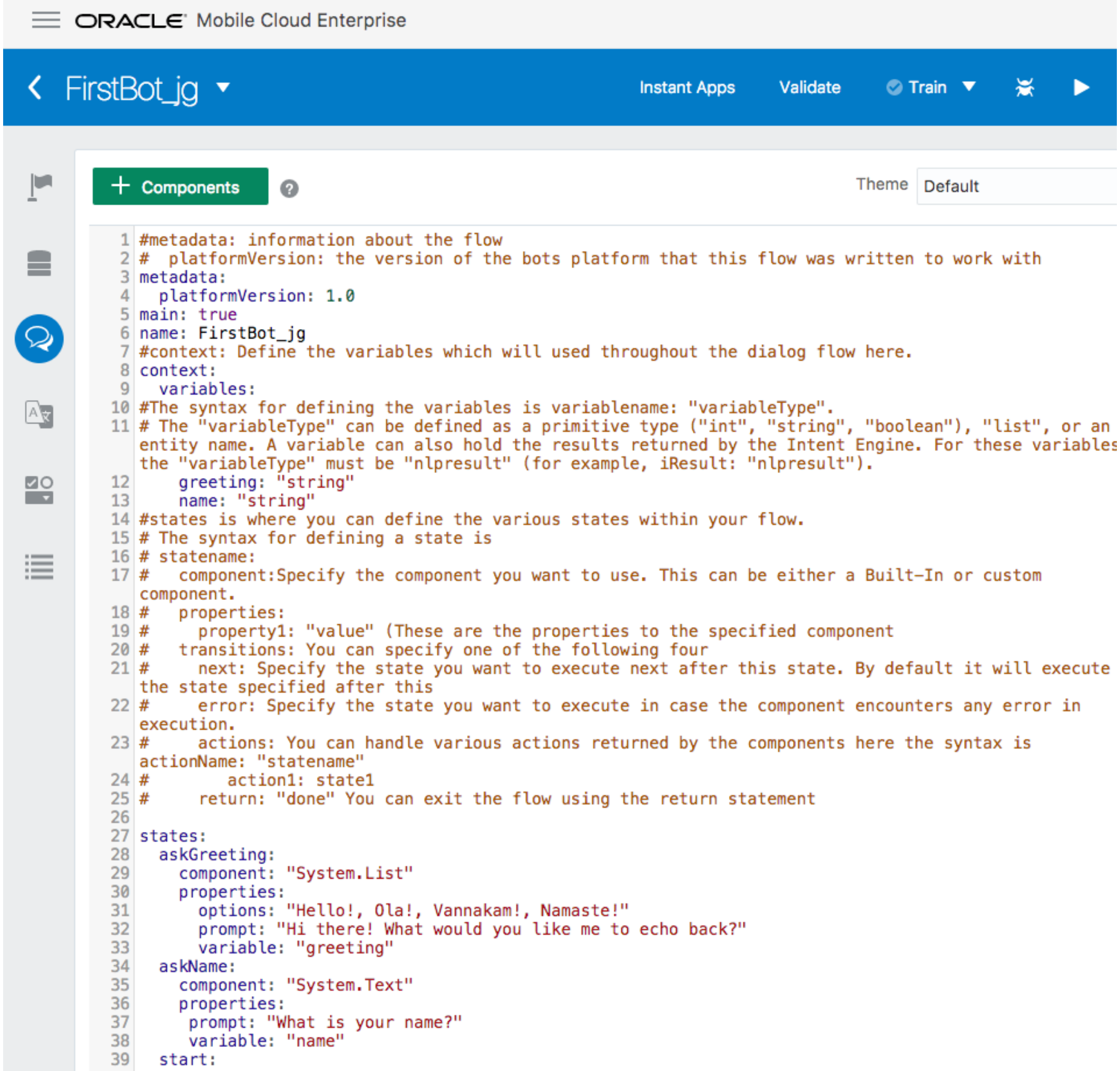
Enable your bot to understand and act on user input by adding intents and their associated entities.

5. Click **Entities** (the second icon down) and notice that it's pre-populated system entities. These are standard entities that you can use in your chatbot without having to explicitly define them.

Oracle MOOC: Oracle Intelligent Bots



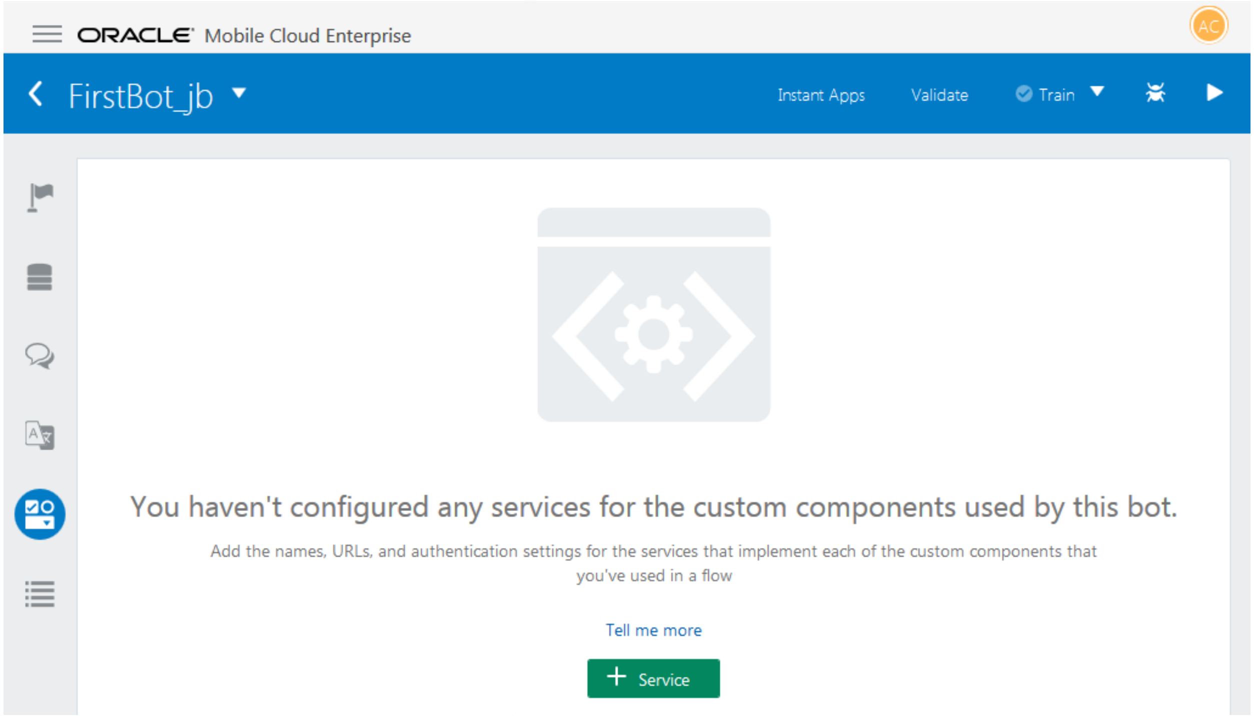
6. Next, click the **Flows** icon. Notice that it’s pre-populated with code that enables the chabot to output a “hello” message. Don't worry about the code for the flow right now--you'll make modifications to it later.



```
#metadata: information about the flow
#  platformVersion: the version of the bots platform that this flow was written to work with
metadata:
  platformVersion: 1.0
main: true
name: FirstBot_jg
#context: Define the variables which will used throughout the dialog flow here.
context:
  variables:
#The syntax for defining the variables is variablename: "variableType".
# The "variableType" can be defined as a primitive type ("int", "string", "boolean"), "list", or an entity name. A variable can also hold the results returned by the Intent Engine. For these variables the "variableType" must be "nlpresult" (for example, iResult: "nlpresult").
    greeting: "string"
    name: "string"
#states is where you can define the various states within your flow.
# The syntax for defining a state is
# statename:
#   component:Specify the component you want to use. This can be either a Built-In or custom component.
#   properties:
#     property1: "value" (These are the properties to the specified component
#   transitions: You can specify one of the following four
#     next: Specify the state you want to execute next after this state. By default it will execute the state specified after this
#     error: Specify the state you want to execute in case the component encounters any error in execution.
#     actions: You can handle various actions returned by the components here the syntax is actionName: "statename"
#        action1: state1
#     return: "done" You can exit the flow using the return statement

states:
  askGreeting:
    component: "System.List"
    properties:
      options: "Hello!, Ola!, Vannakam!, Namaste!"
      prompt: "Hi there! What would you like me to echo back?"
      variable: "greeting"
  askName:
    component: "System.Text"
    properties:
     prompt: "What is your name?"
     variable: "name"
  start:
```

7. Now, click the **Components** icon. Later on, you'll see your custom components that provide your chatbot with various functions and data. But since you've just started, there are no predefined services that obtain the custom components.

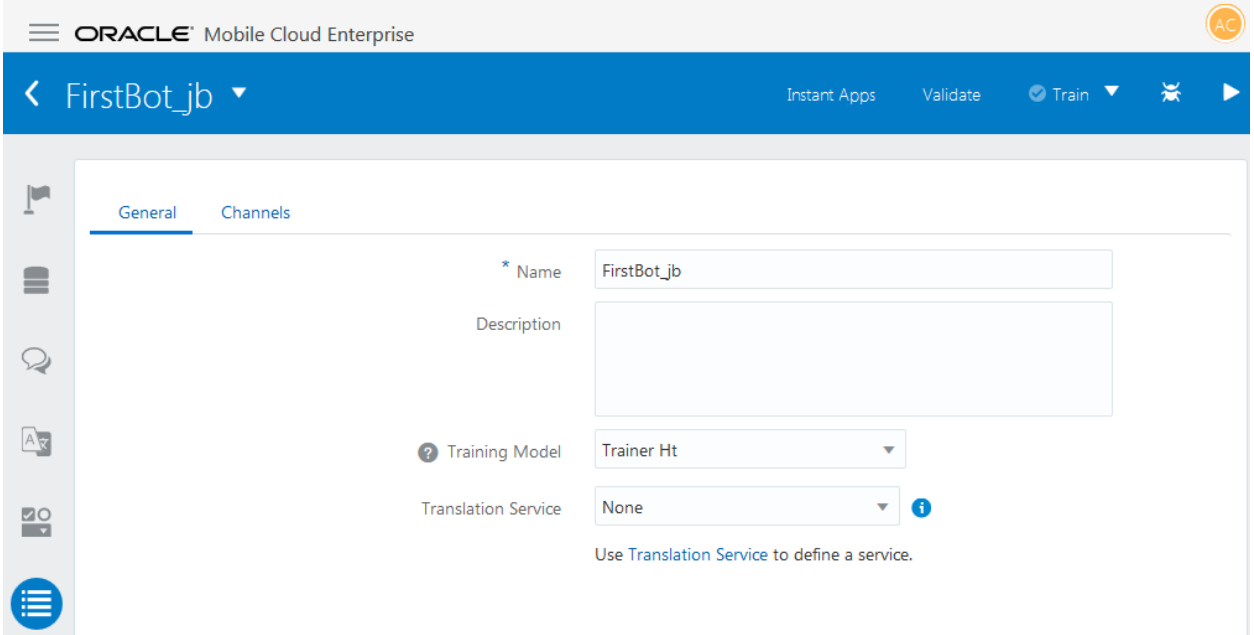


You haven't configured any services for the custom components used by this bot.

Add the names, URLs, and authentication settings for the services that implement each of the custom components that you've used in a flow

8. Finally, click the Settings icon. Notice its tabs: General and Channels.
9. The General tab contains general details about the chatbot and some properties that influence how the chatbot is trained. You'll find out more about that later.
10. The Channels tab is where you'll publicize your bot by hooking it up to Facebook Messenger. That too is something that you'll do in another lab.

Oracle MOOC: Oracle Intelligent Bots



* Name: FirstBot_jb

Description:

Training Model: Trainer Ht

Translation Service: None

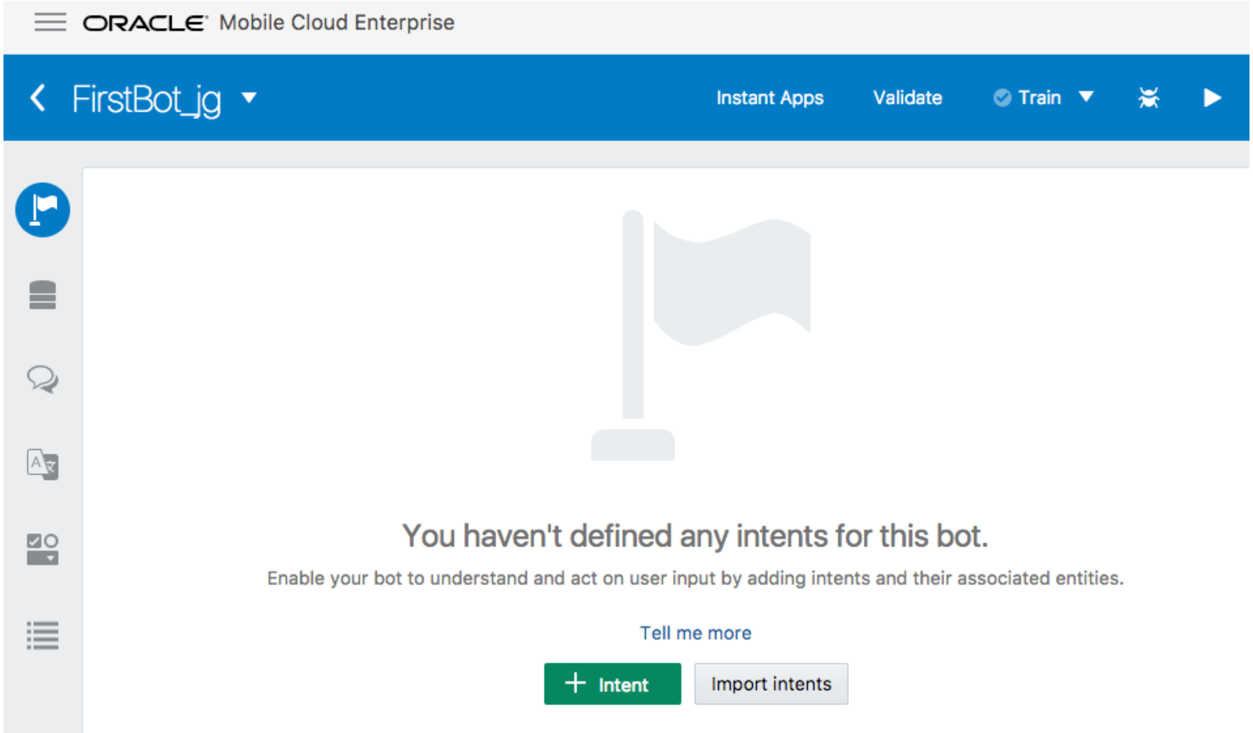
Use Translation Service to define a service.

In the next section, you add artifacts to make the chatbot work.

Step 2: Add an Intent, Entity, and Flow to Your ChatBot

In this section, you will create an intent and some utterances. Next, you will add an entity and some BotML code to control the flow. Finally, you will train the chatbot and test it out.

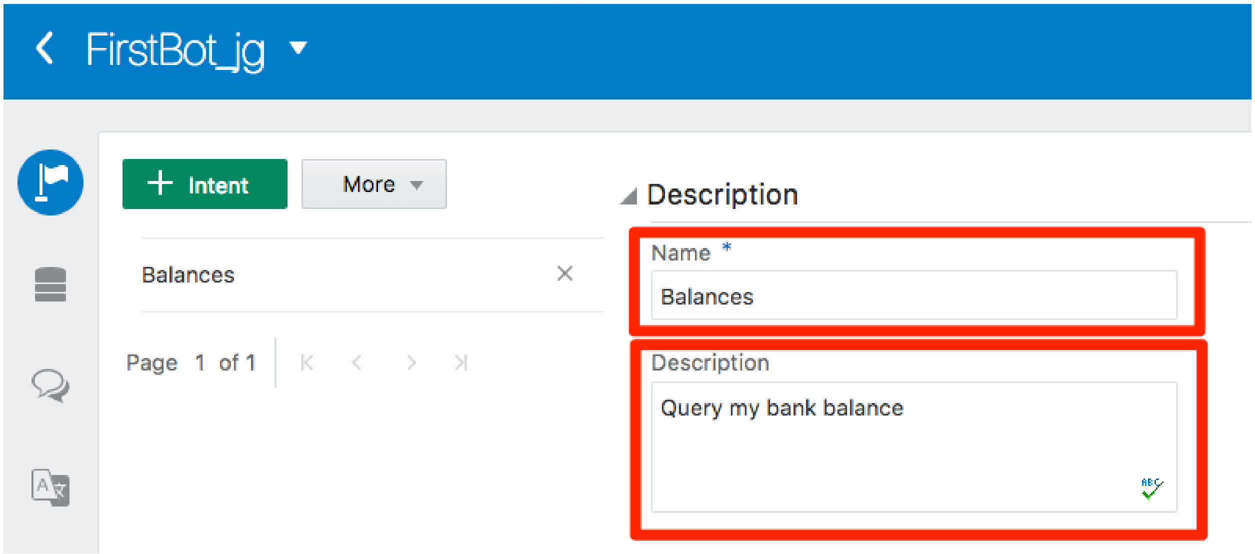
1. Click the **Intents** icon in the left navbar and then click the green **Add Intent** button.



You haven't defined any intents for this bot.

Enable your bot to understand and act on user input by adding intents and their associated entities.

2. This intent will be used to find out your bank balance, so replace *Intent1* in the Name field with *Balances* and then provide a description. These values are saved automatically, so you do not need to explicitly save them. As you create artifacts, you may notice a message in the lower right corner that tells you that your work has been saved.

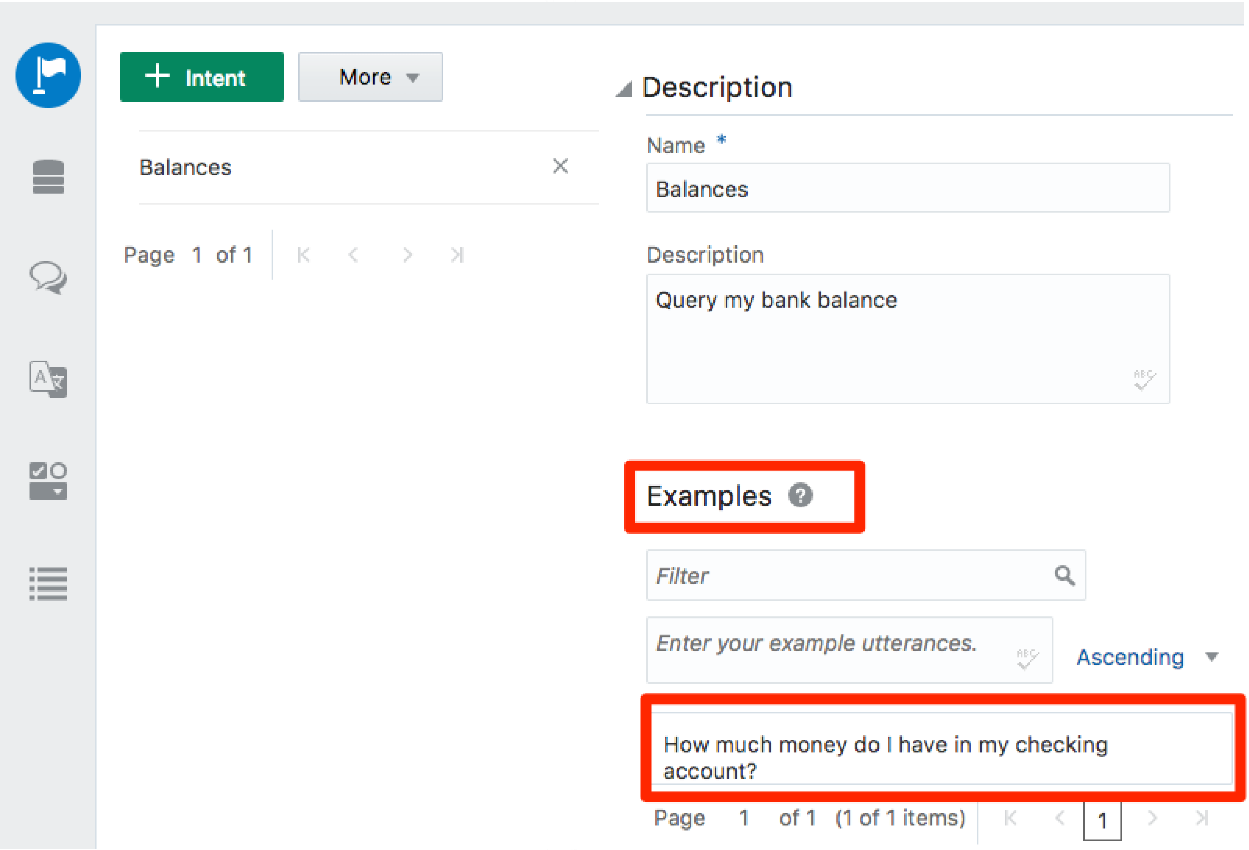


Name *: Balances

Description: Query my bank balance

Oracle MOOC: Oracle Intelligent Bots

3. Now that you have an intent, you need some example phrases, or utterances, that express what a checking an account balance means. In the Examples area, add the following text: *How much money do I have in my checking account?* and then press Return.



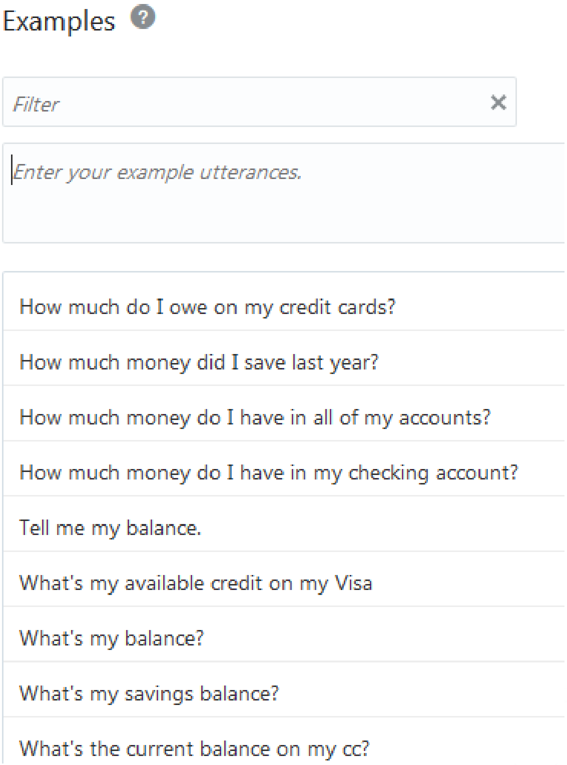
4. Add another example utterance, *How much do I owe on all my credit cards?* and then press Return.

5. Add the following list of utterances to your intent, each followed by a return. Notice that the examples don't need to be in the form of a question; they can be a statement.

- How much money did I save last year?
- How much money do I have in all of my accounts?
- Tell me my balance.
- What's my savings balance?
- What's my available credit on my Visa?

Oracle MOOC: Oracle Intelligent Bots

- What’s my balance?
- What’s the current balance on my cc?



Examples

Filter

Enter your example utterances.

How much do I owe on my credit cards?

How much money did I save last year?

How much money do I have in all of my accounts?

How much money do I have in my checking account?

Tell me my balance.

What's my available credit on my Visa

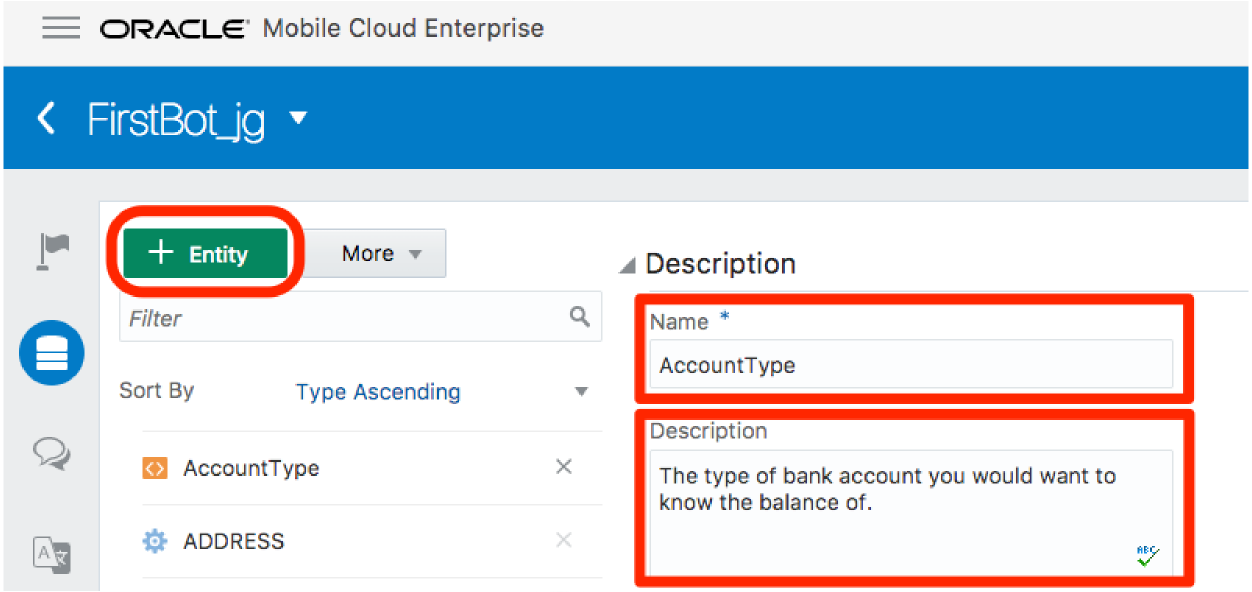
What's my balance?

What's my savings balance?

What's the current balance on my cc?

6. Now that you’ve created the intent, you can add an entity to it. Click the **Entities** icon in the left navbar.
7. If you want to request the balance of an account, you would probably need to know the kind of account that returns that balance. To identify this information, you need to create an entity that defines different types of accounts.

   Click the green **Add Entity** button, replace *Entity1* in the Name field with *AccountType* and then add a short description.



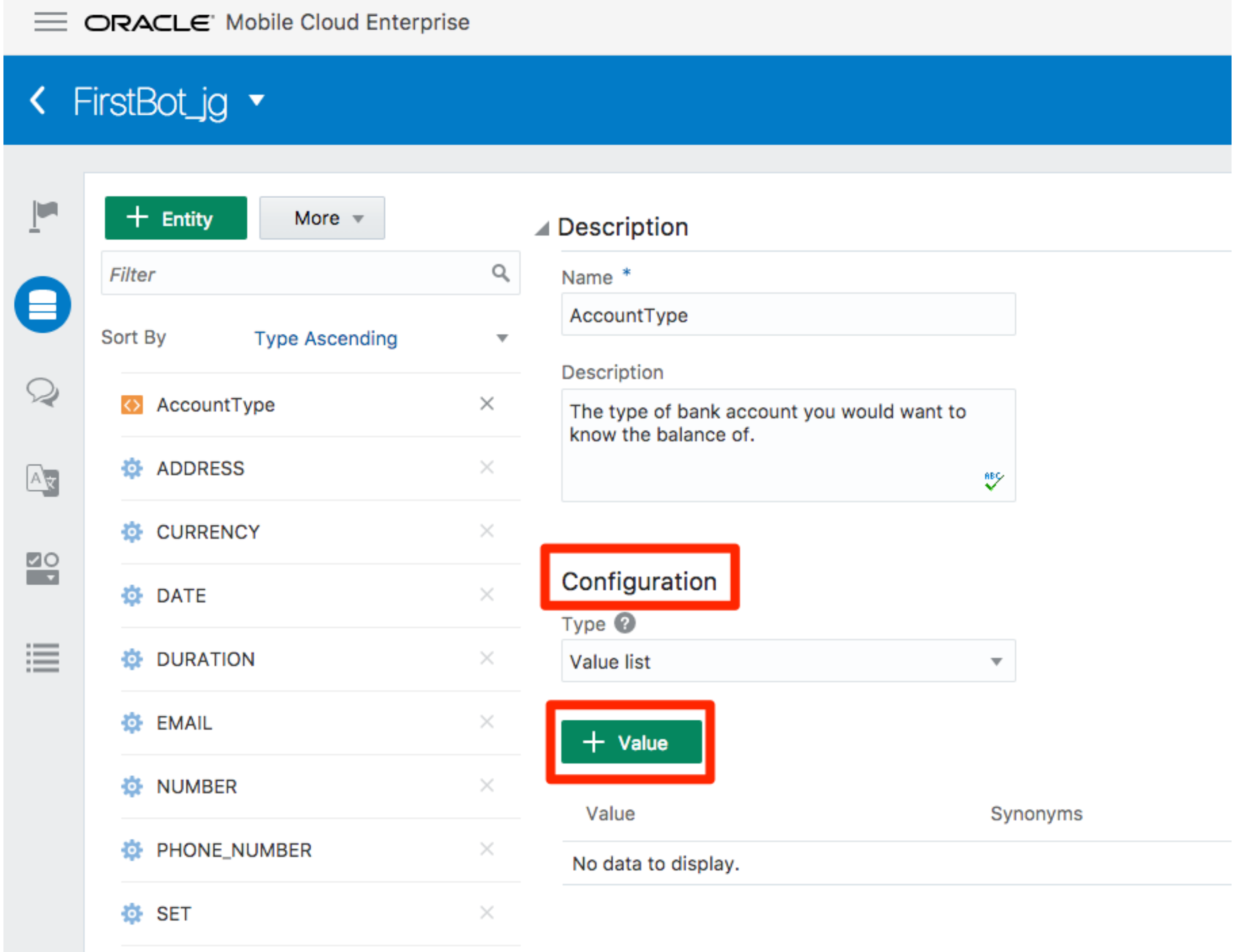
Name *
AccountType

Description
The type of bank account you would want to know the balance of.

8. Now that you have an entity, you need to provide some values that it will use to identify key words from the user input. In the case of the account type, you need to add values that represent the various types of accounts that you could query for a balance.

9. In the Configuration area, be sure that the Type property is set to **Value list** and then click the green **Add Value** button.

Oracle MOOC: Oracle Intelligent Bots



Name * AccountType

Description: The type of bank account you would want to know the balance of.

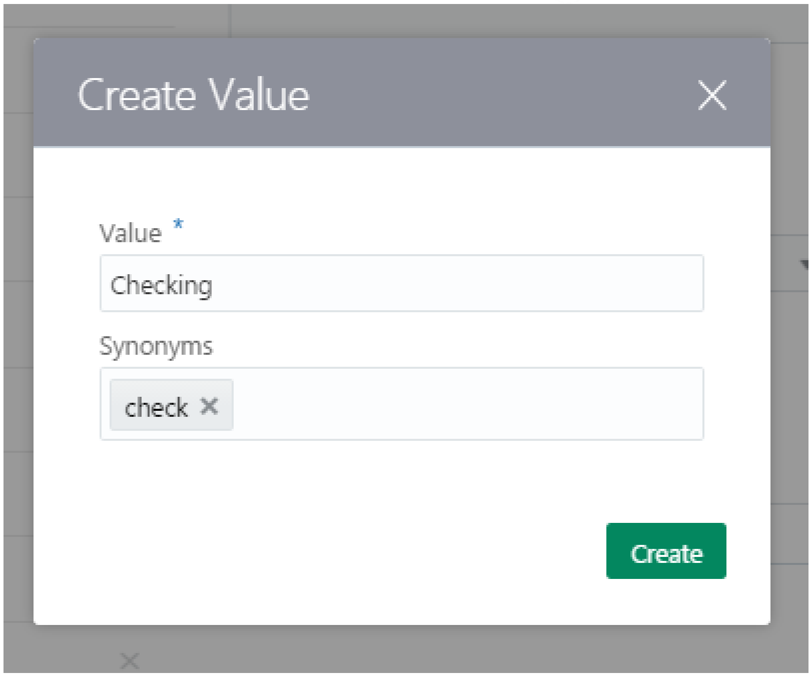
Type: Value list

No data to display.

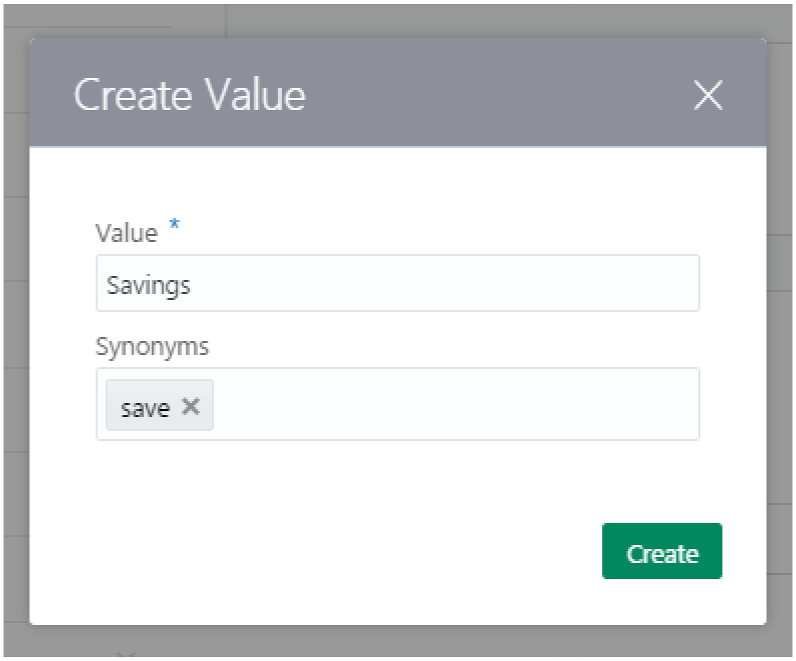
10. In the popup dialog, enter *Checking* as the value and *check* as a synonym (followed by a return).

11. Click **Create**.

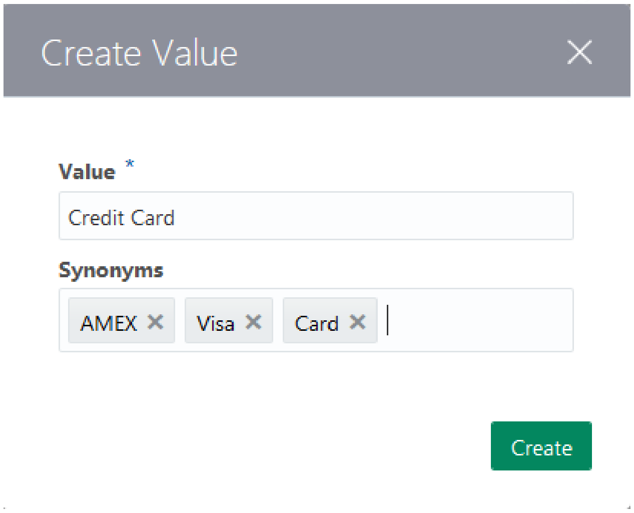
Oracle MOOC: Oracle Intelligent Bots



12. Next, add a second value called *Savings* and then add *save* as the synonym.



13. Finally, add a third value named *Credit Card*. Enter *AMEX*, *Visa*, and *Card* as its synonyms.



Create Value

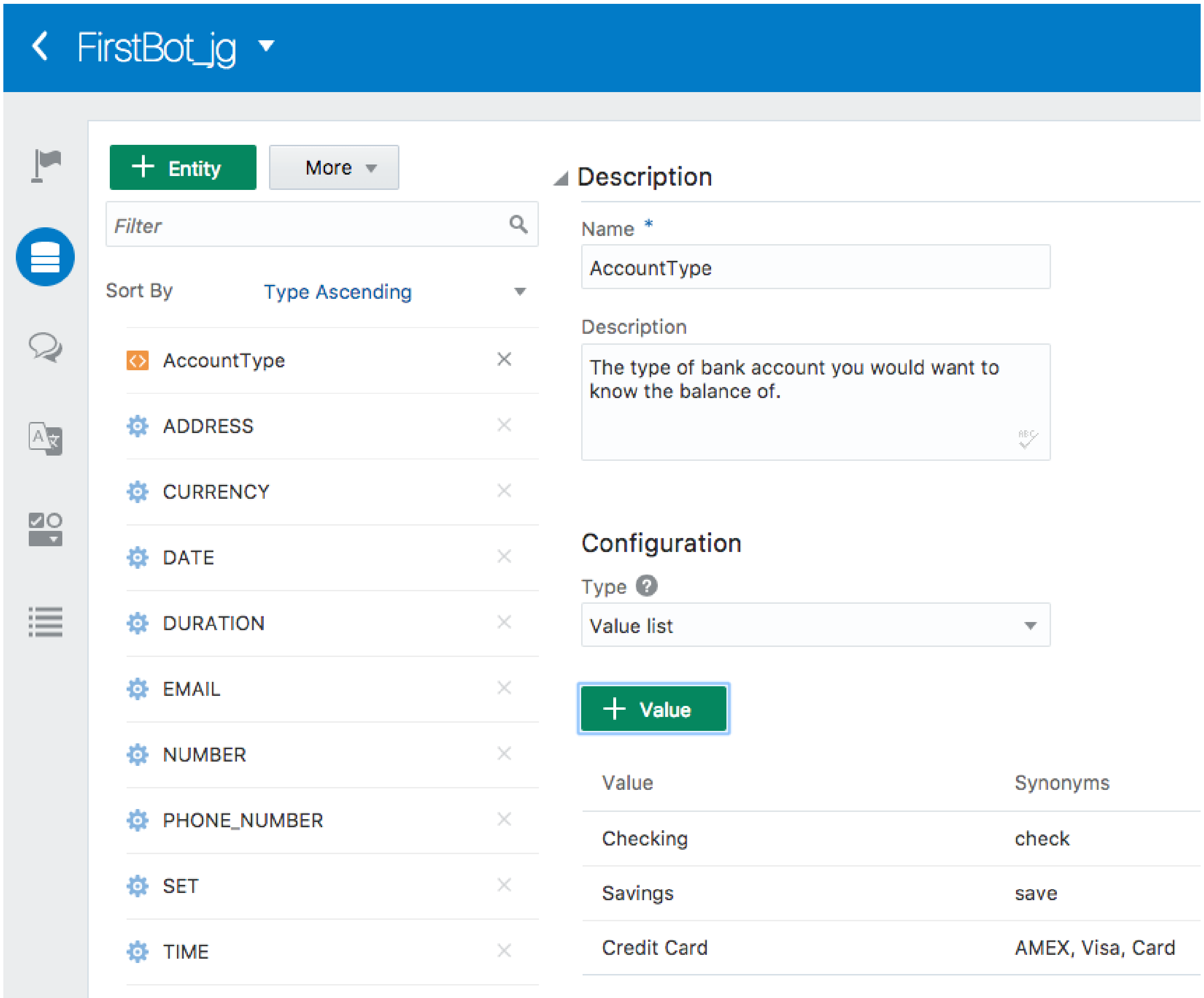
Value *

Credit Card

Synonyms

AMEX ✕ Visa ✕ Card ✕

When you’ve finished, your entity definition should look like the image below.



Sort By Type Ascending

- AccountType ✕
- ADDRESS ✕
- CURRENCY ✕
- DATE ✕
- DURATION ✕
- EMAIL ✕
- NUMBER ✕
- PHONE_NUMBER ✕
- SET ✕
- TIME ✕

Description

Name *

AccountType

Description

The type of bank account you would want to know the balance of.

Configuration

Type

Value list

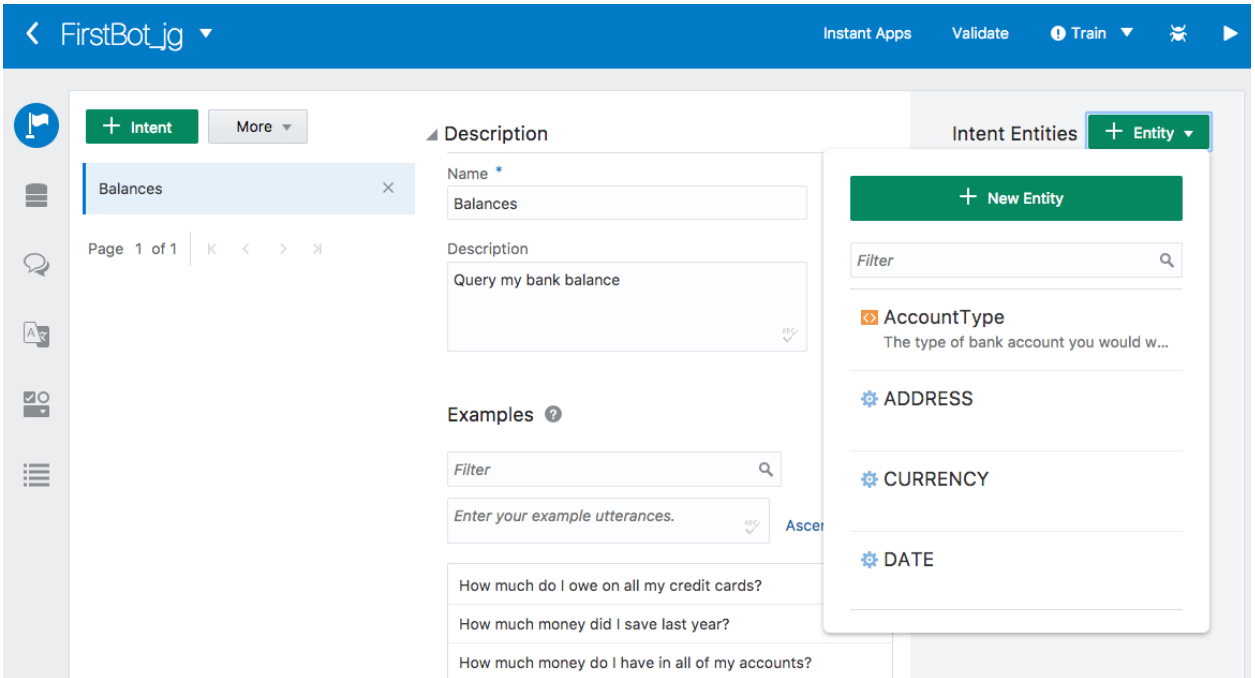
Value	Synonyms
Checking	check
Savings	save
Credit Card	AMEX, Visa, Card

Oracle MOOC: Oracle Intelligent Bots

14. Now that you've got an intent and the entity it works with, you need to associate the balance intent and account type entity. Don't worry – it's easy!

    a. Click the **Intents** icon in the left navbar. Locate the Intent Entities area at the right of the page. (If needed, dismiss the Tester by clicking the **Play** button.)

    b. Click the green **Add Entity** button and then select **AccountType** from the list.



Balances

Name * Balances

Description: Query my bank balance

Enter your example utterances.

How much do I owe on all my credit cards?

How much money did I save last year?

How much money do I have in all of my accounts?

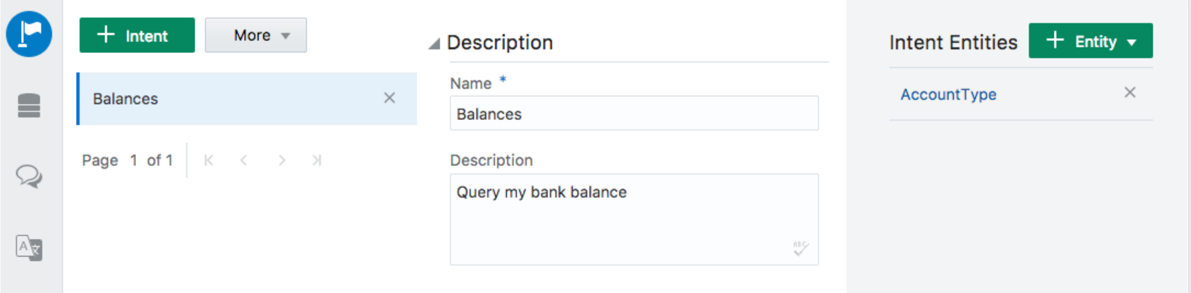
AccountType
The type of bank account you would w...

ADDRESS

CURRENCY

DATE

Tip: You can remove the entity by clicking the ***x*** to the right of the entity name, but don't do that here.



Balances

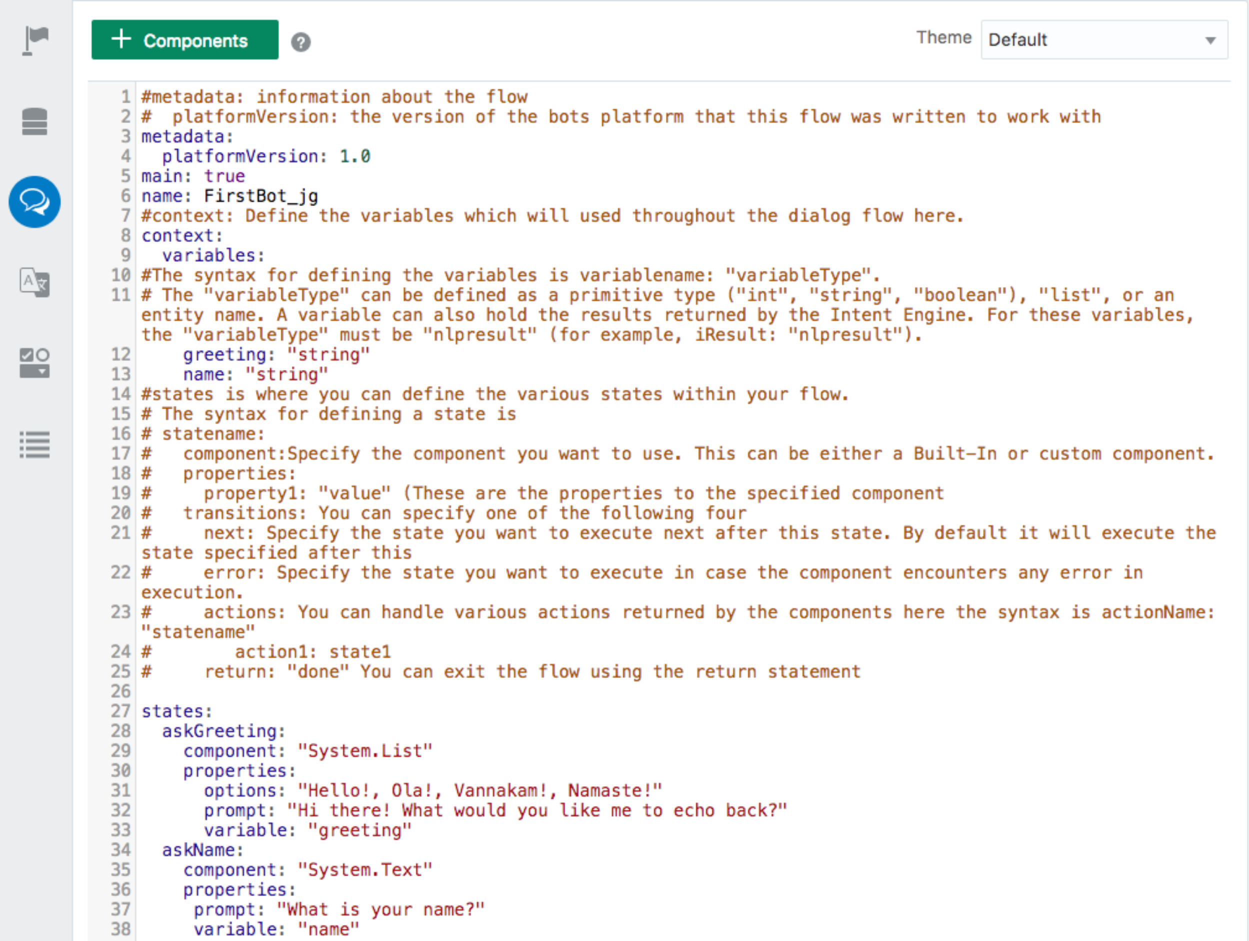
Name * Balances

Description: Query my bank balance

AccountType ×

15. With the intent and entity created, the next thing you need is a flow. Click the **Flows** icon on the left navbar.

16. There's BotML code in the editor that displays "hello" when you run the chatbot. However, we're not going to use this code. Instead we're going to add our own:



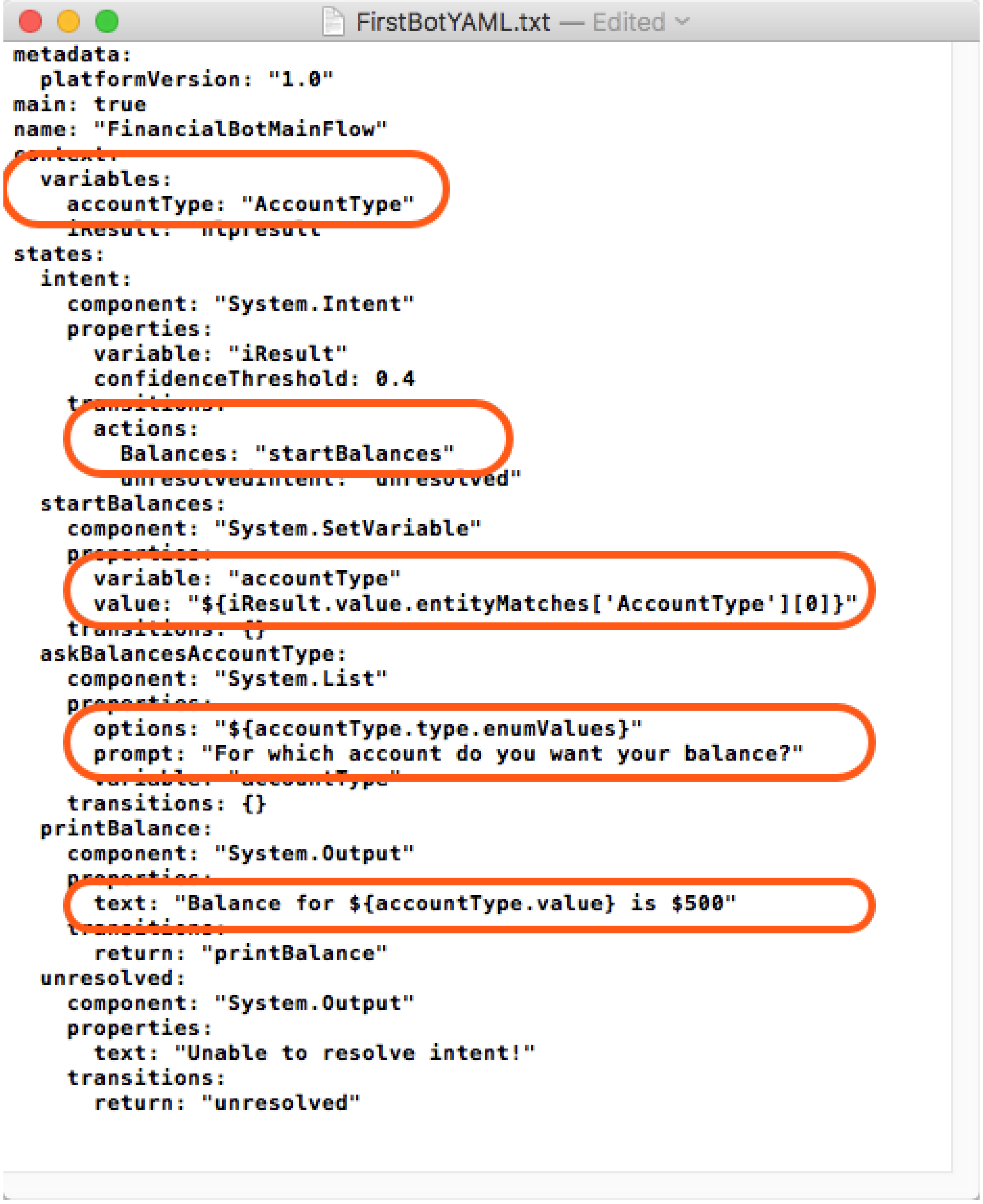
```
#metadata: information about the flow
#  platformVersion: the version of the bots platform that this flow was written to work with
metadata:
  platformVersion: 1.0
main: true
name: FirstBot_jg
#context: Define the variables which will used throughout the dialog flow here.
context:
  variables:
#The syntax for defining the variables is variablename: "variableType".
# The "variableType" can be defined as a primitive type ("int", "string", "boolean"), "list", or an entity name. A variable can also hold the results returned by the Intent Engine. For these variables, the "variableType" must be "nlpresult" (for example, iResult: "nlpresult").
    greeting: "string"
    name: "string"
#states is where you can define the various states within your flow.
# The syntax for defining a state is
# statename:
#   component:Specify the component you want to use. This can be either a Built-In or custom component.
#   properties:
#     property1: "value" (These are the properties to the specified component
#   transitions: You can specify one of the following four
#     next: Specify the state you want to execute next after this state. By default it will execute the state specified after this
#     error: Specify the state you want to execute in case the component encounters any error in execution.
#     actions: You can handle various actions returned by the components here the syntax is actionName: "statename"
#        action1: state1
#     return: "done" You can exit the flow using the return statement

states:
  askGreeting:
    component: "System.List"
    properties:
      options: "Hello!, Ola!, Vannakam!, Namaste!"
      prompt: "Hi there! What would you like me to echo back?"
      variable: "greeting"
  askName:
    component: "System.Text"
    properties:
     prompt: "What is your name?"
     variable: "name"
```

a. From the `/labfiles/code` directory on your system, open the `FirstBotYAML.txt` file in your text editor of choice.

b. Take a look at the BotML code. Under the `context` node near the top, notice that this flow definition names the AccountType entity as a variable (`accountType`) and further down in the intent state, names your intent (Balances) as one of the actions. Because flow definition includes the `accountType` variable in the `startBalances` state, the conversation flow proceeds to `askBalancesAccountType` and then finally onto the `printBalance` state, which displays the balance. When the `accountType` variable is not set, then the `askBalancesAccountType` state will prompt you for the account type using the value list values that belong to the AccountType entity. It then moves to the `printBalance` state.

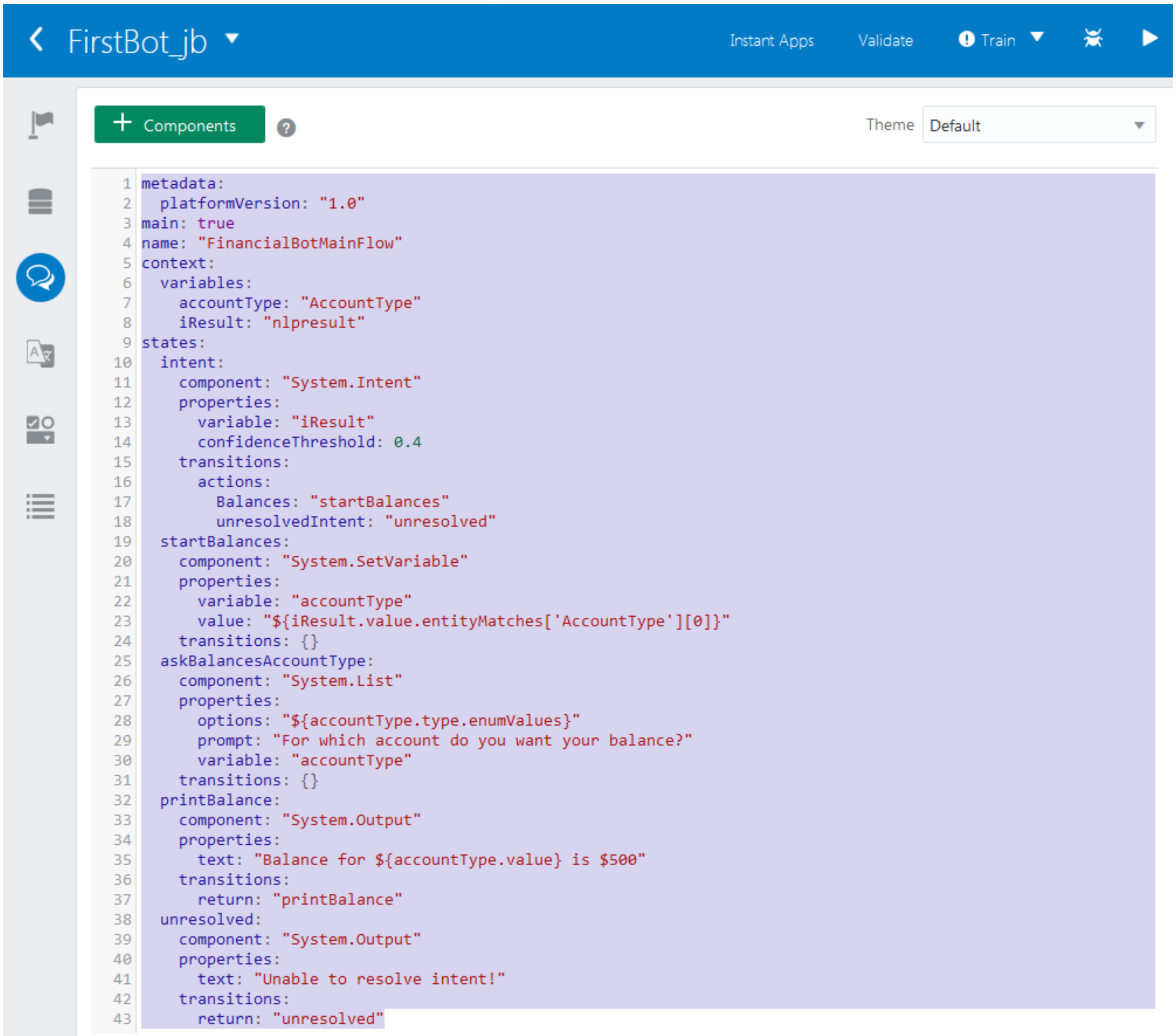
Oracle MOOC: Oracle Intelligent Bots



```
metadata:
  platformVersion: "1.0"
main: true
name: "FinancialBotMainFlow"
context:
  variables:
    accountType: "AccountType"
    iResult: "nlpresult"
states:
  intent:
    component: "System.Intent"
    properties:
      variable: "iResult"
      confidenceThreshold: 0.4
    transitions:
      actions:
        Balances: "startBalances"
        unresolvedIntent: "unresolved"
  startBalances:
    component: "System.SetVariable"
    properties:
      variable: "accountType"
      value: "${iResult.value.entityMatches['AccountType'][0]}"
    transitions: {}
  askBalancesAccountType:
    component: "System.List"
    properties:
      options: "${accountType.type.enumValues}"
      prompt: "For which account do you want your balance?"
      variable: "accountType"
    transitions: {}
  printBalance:
    component: "System.Output"
    properties:
      text: "Balance for ${accountType.value} is $500"
    transitions:
      return: "printBalance"
  unresolved:
    component: "System.Output"
    properties:
      text: "Unable to resolve intent!"
    transitions:
      return: "unresolved"
```

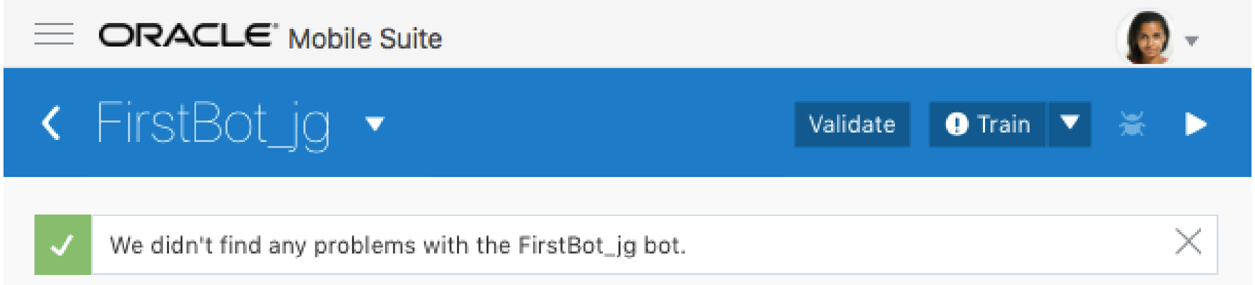
c. Copy the contents from the `FirstBotYAML.txt` file into the editor, replacing all of the code that's currently there. To prevent YAML formatting issues, paste any code from the beginning of Column 1.

Oracle MOOC: Oracle Intelligent Bots



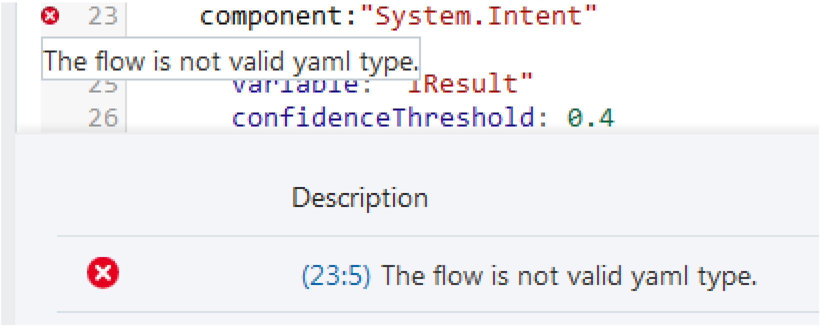
```
metadata:
  platformVersion: "1.0"
main: true
name: "FinancialBotMainFlow"
context:
  variables:
    accountType: "AccountType"
    iResult: "nlpresult"
states:
  intent:
    component: "System.Intent"
    properties:
      variable: "iResult"
      confidenceThreshold: 0.4
    transitions:
      actions:
        Balances: "startBalances"
        unresolvedIntent: "unresolved"
  startBalances:
    component: "System.SetVariable"
    properties:
      variable: "accountType"
      value: "${iResult.value.entityMatches['AccountType'][0]}"
    transitions: {}
  askBalancesAccountType:
    component: "System.List"
    properties:
      options: "${accountType.type.enumValues}"
      prompt: "For which account do you want your balance?"
      variable: "accountType"
    transitions: {}
  printBalance:
    component: "System.Output"
    properties:
      text: "Balance for ${accountType.value} is $500"
    transitions:
      return: "printBalance"
  unresolved:
    component: "System.Output"
    properties:
      text: "Unable to resolve intent!"
    transitions:
      return: "unresolved"
```

d. Finally, click the **Validate** button in the upper right. You should see a message that there were no problems found in your chatbot.



✓ We didn't find any problems with the FirstBot_jg bot.

Tip: You might get syntax errors as you update your dialog flow. You can troubleshoot them using the logger window, which reports errors on a line-by-line basis.



```
    component:"System.Intent"
The flow is not valid yaml type.
      variable: "iResult"
      confidenceThreshold: 0.4
```

Description

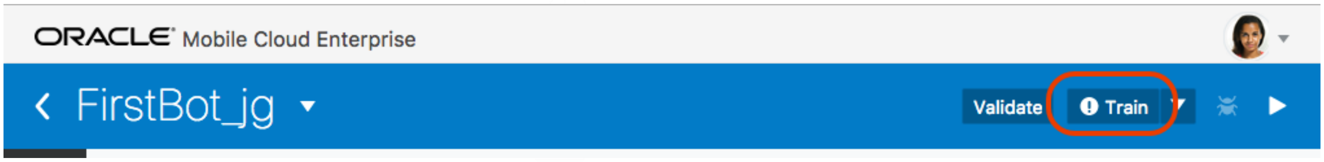
(23:5) The flow is not valid yaml type.

To open the logger window, click the debug icon ( ) in the upper-right of the screen.

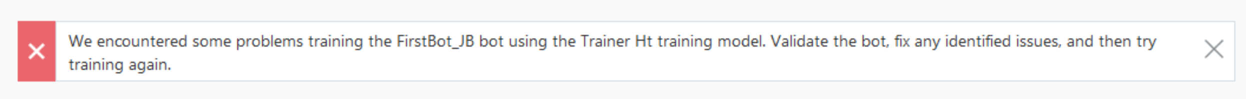
Step 3: Train and Test Your Chatbot

In this step, you will use the training tool on the chatbot. Training your chatbot enables it to understand phrases other than the utterances that you've defined for its intents. In other words, training allows your chatbot to understand user input that's similar to the utterances, but not exactly the same.

1. In the upper right, click the **Train** button. This will kick off a process that will run an algorithm that takes your example utterances and builds the model that will be used to ascertain the intents and entities. Whenever the Bots platform recognizes that your chatbot needs to be trained or re-trained, it will display an exclamation point in the train button. The training process may take a few moments, so be patient. Once the training is complete, the exclamation point is grayed out.

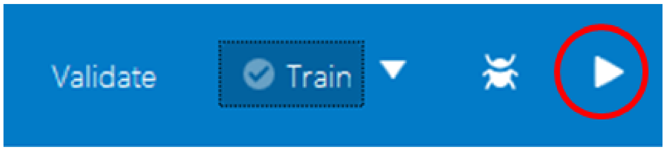


If you get the following error message, it's because this build requires at least two intents, which makes sense. To get around this error, simply add a new intent and add a few utterances to it. The name of this second intent doesn't matter; you simply need two intents to trian this bot, and both of these intents need their own utterances.

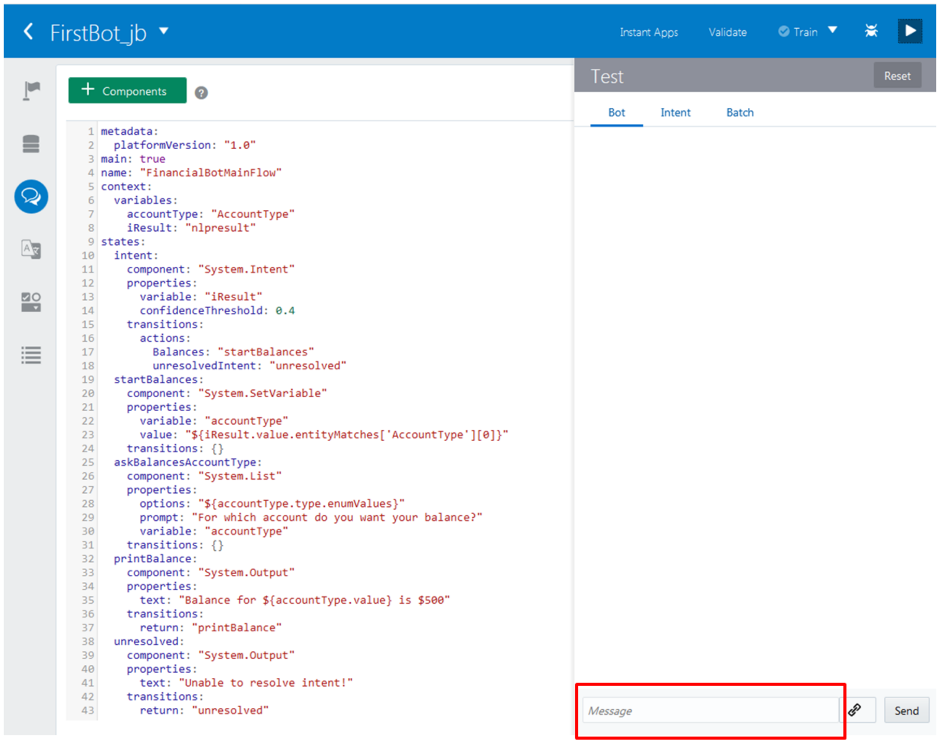


We encountered some problems training the FirstBot_JB bot using the Trainer Ht training model. Validate the bot, fix any identified issues, and then try training again.

2. To test the chatbot, click the **Play** icon in the upper right of the page. This opens the Tester where you can see two tabs: **Bot** and **Intent**.



3. Click the **Bot** tab in the Tester to test the chatbot. Remember that what you type into the Message area is what gets sent to the chatbot when you click the **Send** button.

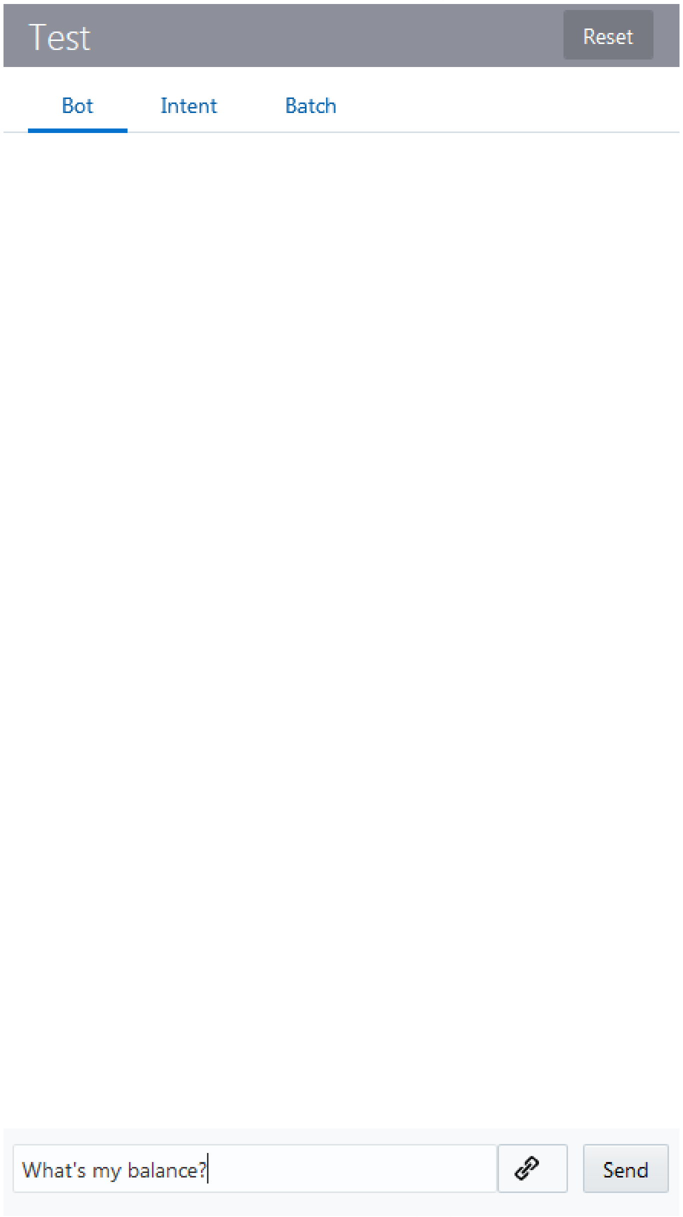


```
metadata:
  platformVersion: "1.0"
main: true
name: "FinancialBotMainFlow"
context:
  variables:
    accountType: "AccountType"
    iResult: "nlpresult"
states:
  intent:
    component: "System.Intent"
    properties:
      variable: "iResult"
      confidenceThreshold: 0.4
    transitions:
      actions:
        Balances: "startBalances"
        unresolvedIntent: "unresolved"
  startBalances:
    component: "System.SetVariable"
    properties:
      variable: "accountType"
      value: "${iResult.value.entityMatches['AccountType'][0]}"
    transitions: {}
  askBalancesAccountType:
    component: "System.List"
    properties:
      options: "${accountType.type.enumValues}"
      prompt: "For which account do you want your balance?"
      variable: "accountType"
    transitions: {}
  printBalance:
    component: "System.Output"
    properties:
      text: "Balance for ${accountType.value} is $500"
    transitions:
      return: "printBalance"
  unresolved:
    component: "System.Output"
    properties:
      text: "Unable to resolve intent!"
    transitions:
      return: "unresolved"
```

4. Let's start off with a simple test:

   a. In the Message area, type in *What's my balance?* and then click the **Send** button.

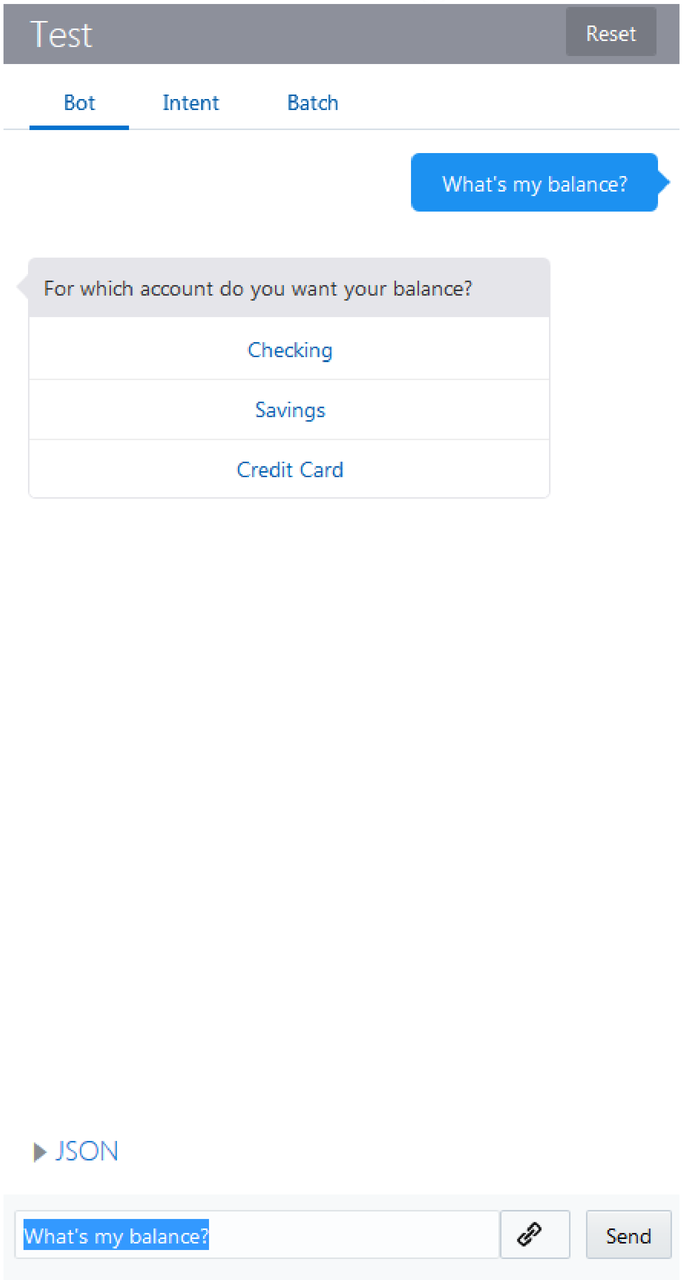
Oracle MOOC: Oracle Intelligent Bots



What's my balance?

b. Since the account type wasn't specified, the chatbot presents you with three options: Checking, Savings, Credit Card.

Oracle MOOC: Oracle Intelligent Bots



What's my balance?

For which account do you want your balance?

Checking

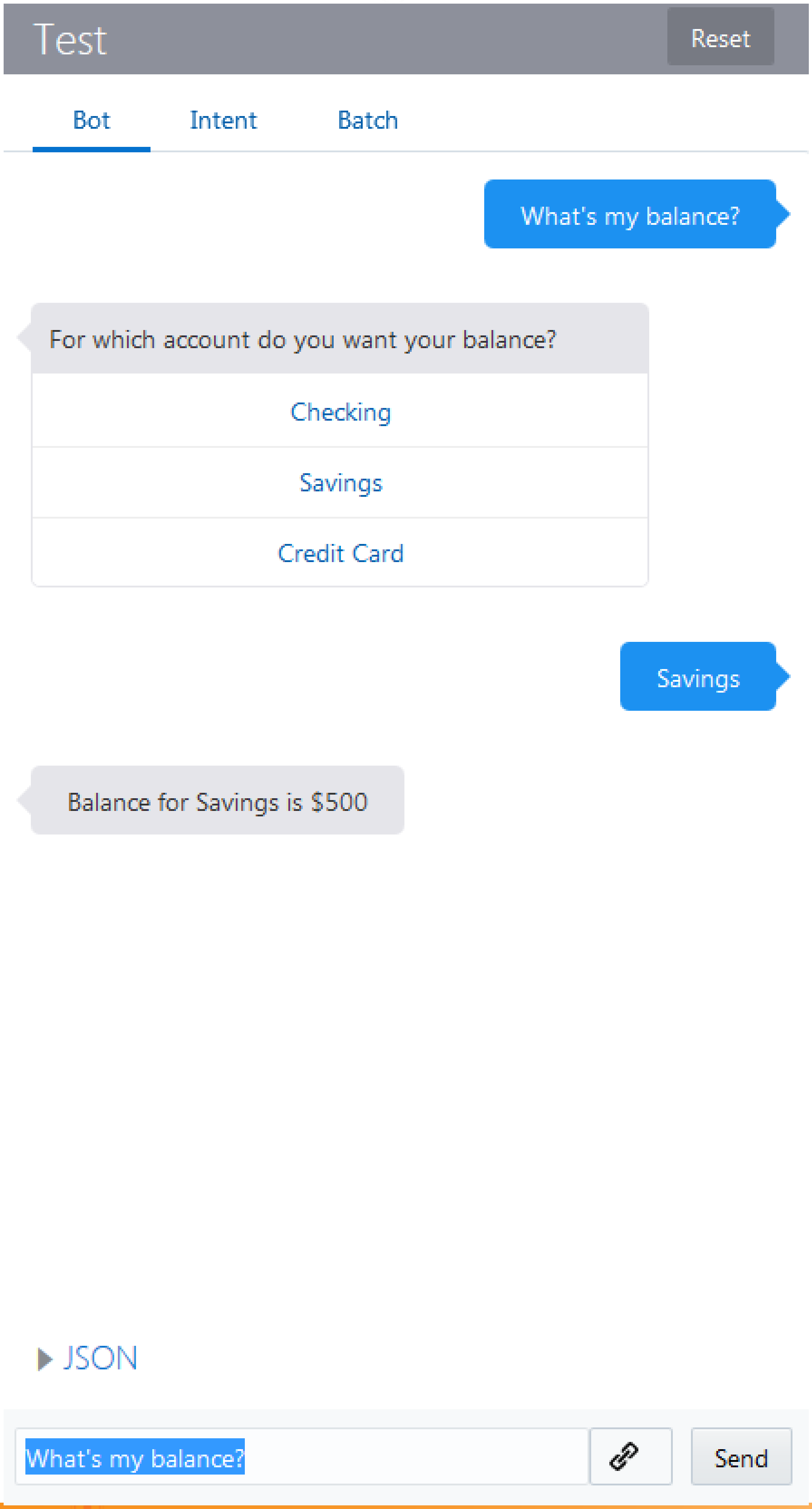
Savings

Credit Card

What's my balance?

c. Click one of the three options. The chatbot outputs text showing the chosen account and its balance.

Oracle MOOC: Oracle Intelligent Bots



What's my balance?

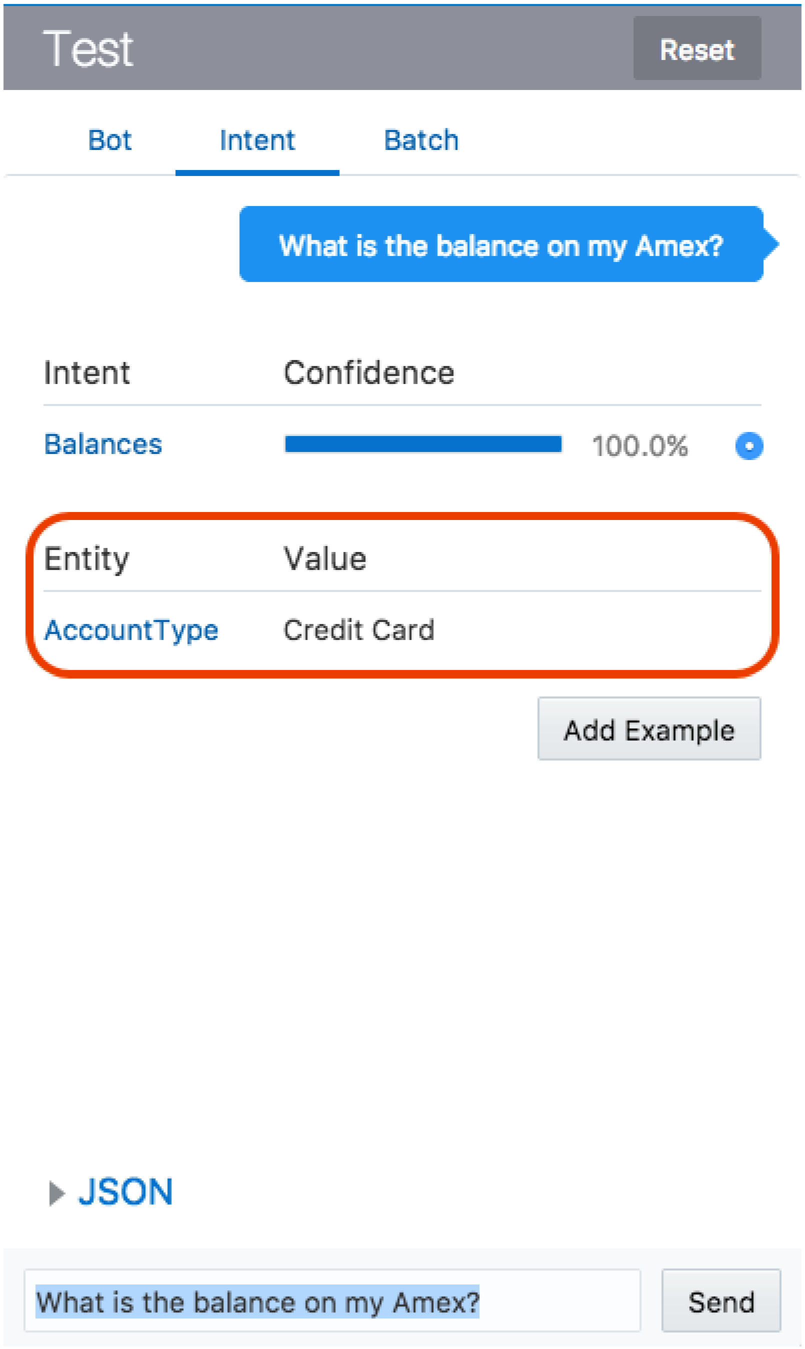
For which account do you want your balance?

Savings

Balance for Savings is $500

Oracle MOOC: Oracle Intelligent Bots

5. Now let's try a message that includes the account type. In the Tester, click the **Intent** tab.
6. In the message area, type in *What is the balance on my Amex?* and then click **Send**.



What is the balance on my Amex?

Intent	Confidence
Balances	100.0%

Entity	Value
AccountType	Credit Card

What is the balance on my Amex?

The Tester displays the level of confidence, expressed as a percentage, that the intent can resolve the user input. You also see that the AccountType entity is recognized as a credit card.

> **Important**: To avoid any confusion that can arise when the flow from a previous session remains incomplete, be sure to finish each round of requests and responses. Also, start a new session by clicking the **Reset** button.

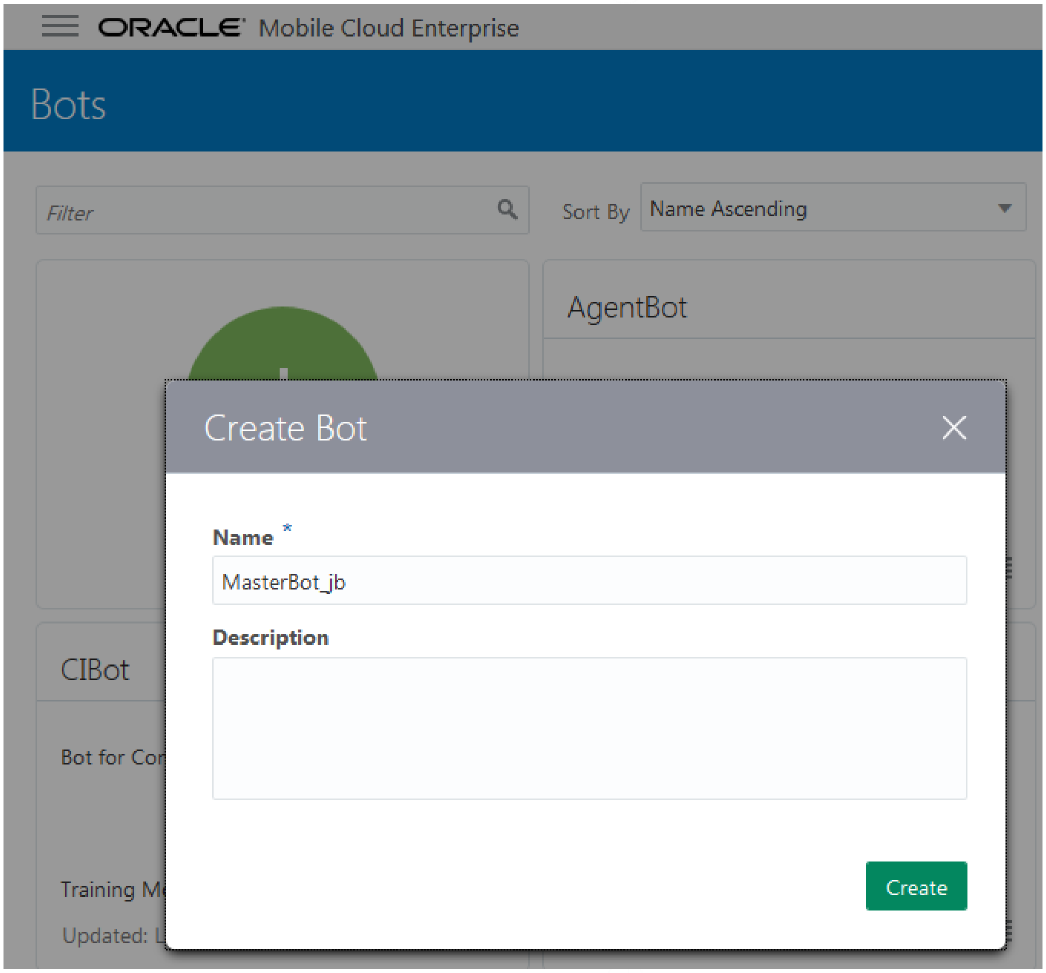
In the previous few steps, you've created a chatbot, added an intent, an entity (and its values) and validated the BotML code. You also trained your chatbot and tested it. In the next few steps, you'll create a more complex chatbot. You'll also take a look at the training options.

Step 4: Create a New Chatbot and Populate It with Intents and Entities

In this section, you will create a new chatbot, import some intents, and then create some entities. Next, you'll associate the entities with the intents. Don't worry about the BotML code in this section—you'll add it to the chatbot in a later lab.

1. Go back to the home page of your Intelligent Bots instance and click the **New Bot** button again. This time, name the chatbot *MasterBot_xx*, where *xx* are your initials. Next add a description and then click **Create**.

Oracle MOOC: Oracle Intelligent Bots



Create Bot

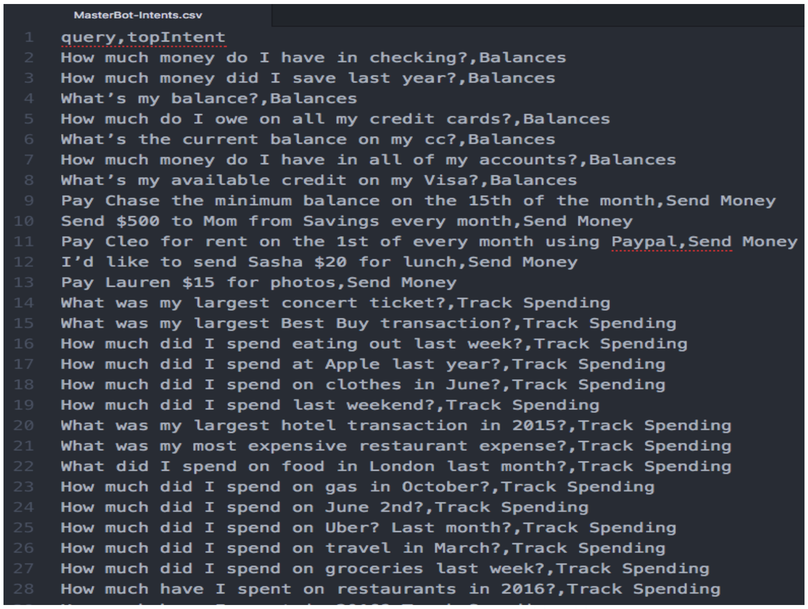
Name *

MasterBot_jb

Description

2. You could manually add the intents like you did in the previous chatbot, but you can also add intents quickly by importing them from a CSV file. If you’re interested, here’s how that file looks.

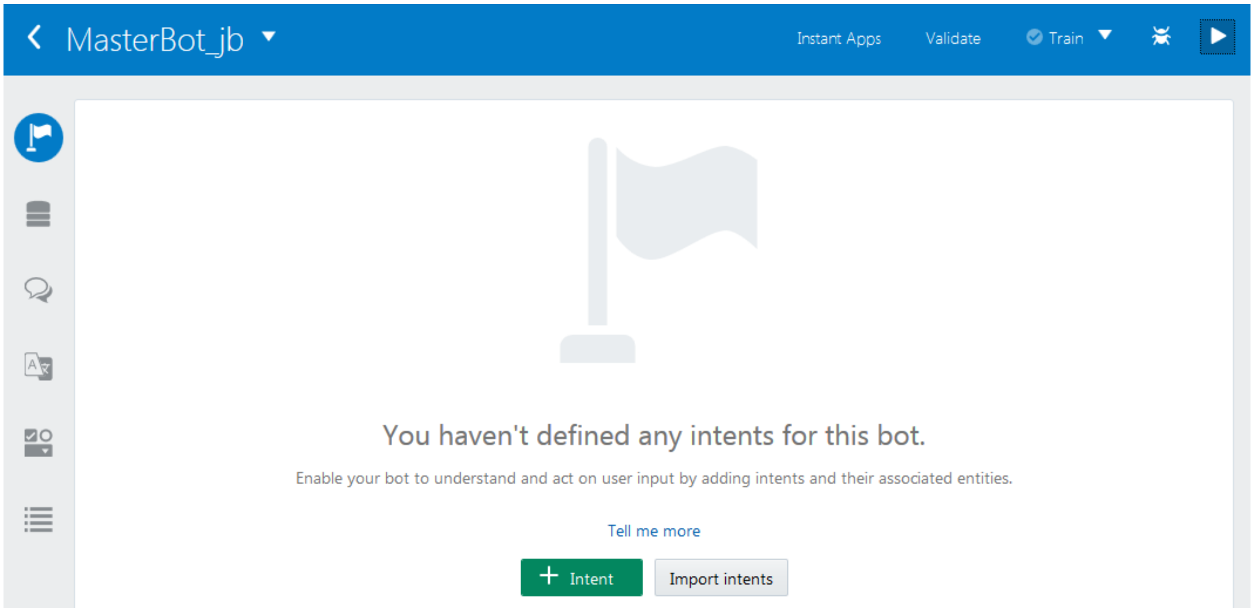
Oracle MOOC: Oracle Intelligent Bots



MasterBot-Intents.csv

```
query,topIntent
How much money do I have in checking?,Balances
How much money did I save last year?,Balances
What’s my balance?,Balances
How much do I owe on all my credit cards?,Balances
What’s the current balance on my cc?,Balances
How much money do I have in all of my accounts?,Balances
What’s my available credit on my Visa?,Balances
Pay Chase the minimum balance on the 15th of the month,Send Money
Send $500 to Mom from Savings every month,Send Money
Pay Cleo for rent on the 1st of every month using Paypal,Send Money
I’d like to send Sasha $20 for lunch,Send Money
Pay Lauren $15 for photos,Send Money
What was my largest concert ticket?,Track Spending
What was my largest Best Buy transaction?,Track Spending
How much did I spend eating out last week?,Track Spending
How much did I spend at Apple last year?,Track Spending
How much did I spend on clothes in June?,Track Spending
How much did I spend last weekend?,Track Spending
What was my largest hotel transaction in 2015?,Track Spending
What was my most expensive restaurant expense?,Track Spending
What did I spend on food in London last month?,Track Spending
How much did I spend on gas in October?,Track Spending
How much did I spend on June 2nd?,Track Spending
How much did I spend on Uber? Last month?,Track Spending
How much did I spend on travel in March?,Track Spending
How much did I spend on groceries last week?,Track Spending
How much have I spent on restaurants in 2016?,Track Spending
```

3. Click the **Intents** icon in the left navbar.



You haven't defined any intents for this bot.

Enable your bot to understand and act on user input by adding intents and their associated entities.

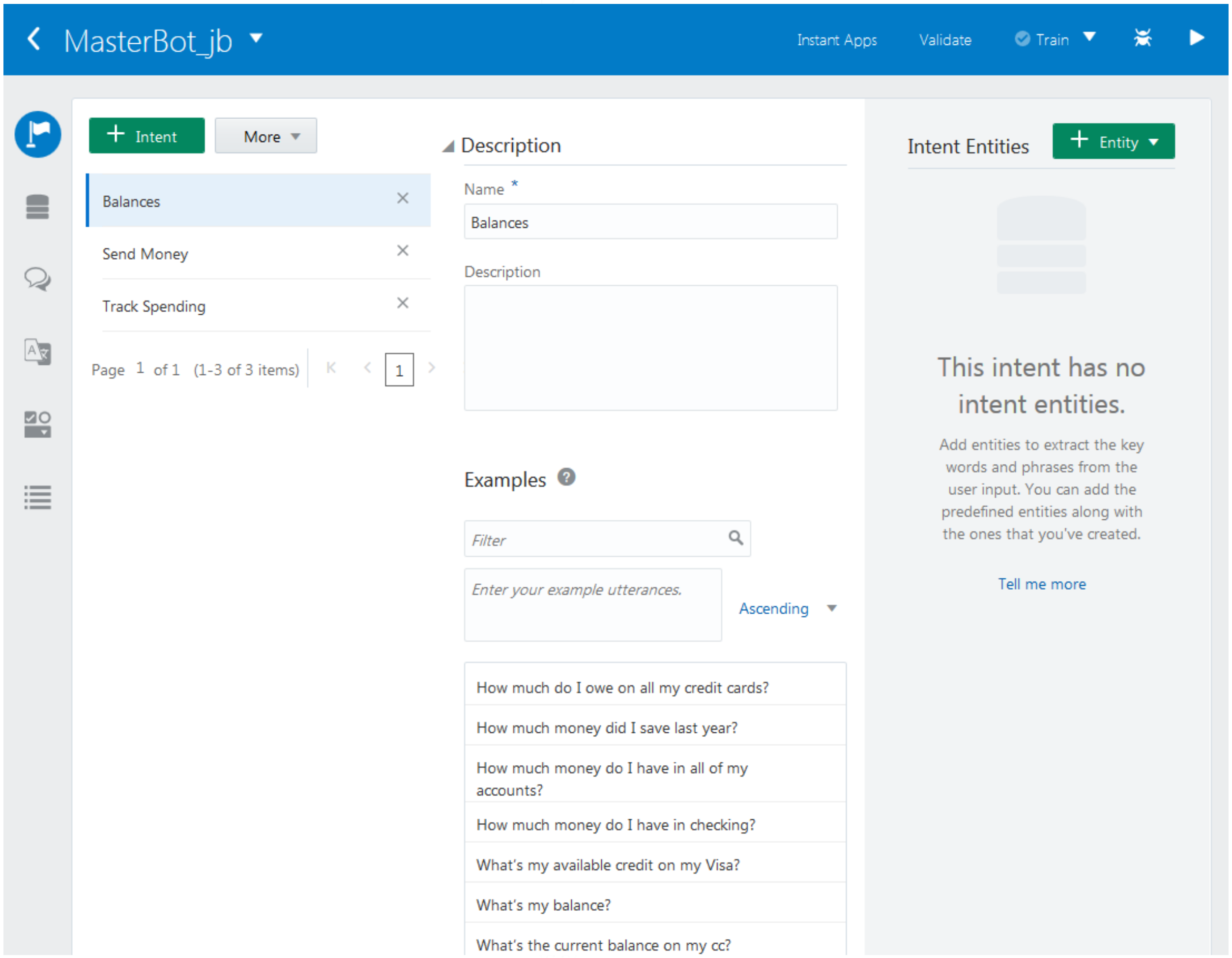
Tell me more

In the middle of the Intents page, click the **Import Intents** button and select the `MasterBot-Intents.csv` file found in the `/labfiles/code/` directory.

4. Next, click **Open**.

5. Three intents should be imported: Balances, Send Money, and Track Spending. Each intent has its own set of utterances.

6. To get a better idea of the how the language used in these utterances differentiates each of the intents, click each intent and then take a look at their respective example phrases.



Add entities to extract the key words and phrases from the user input. You can add the predefined entities along with the ones that you've created.

How much do I owe on all my credit cards?

How much money did I save last year?

How much money do I have in all of my accounts?

How much money do I have in checking?

What's my available credit on my Visa?

What's my balance?

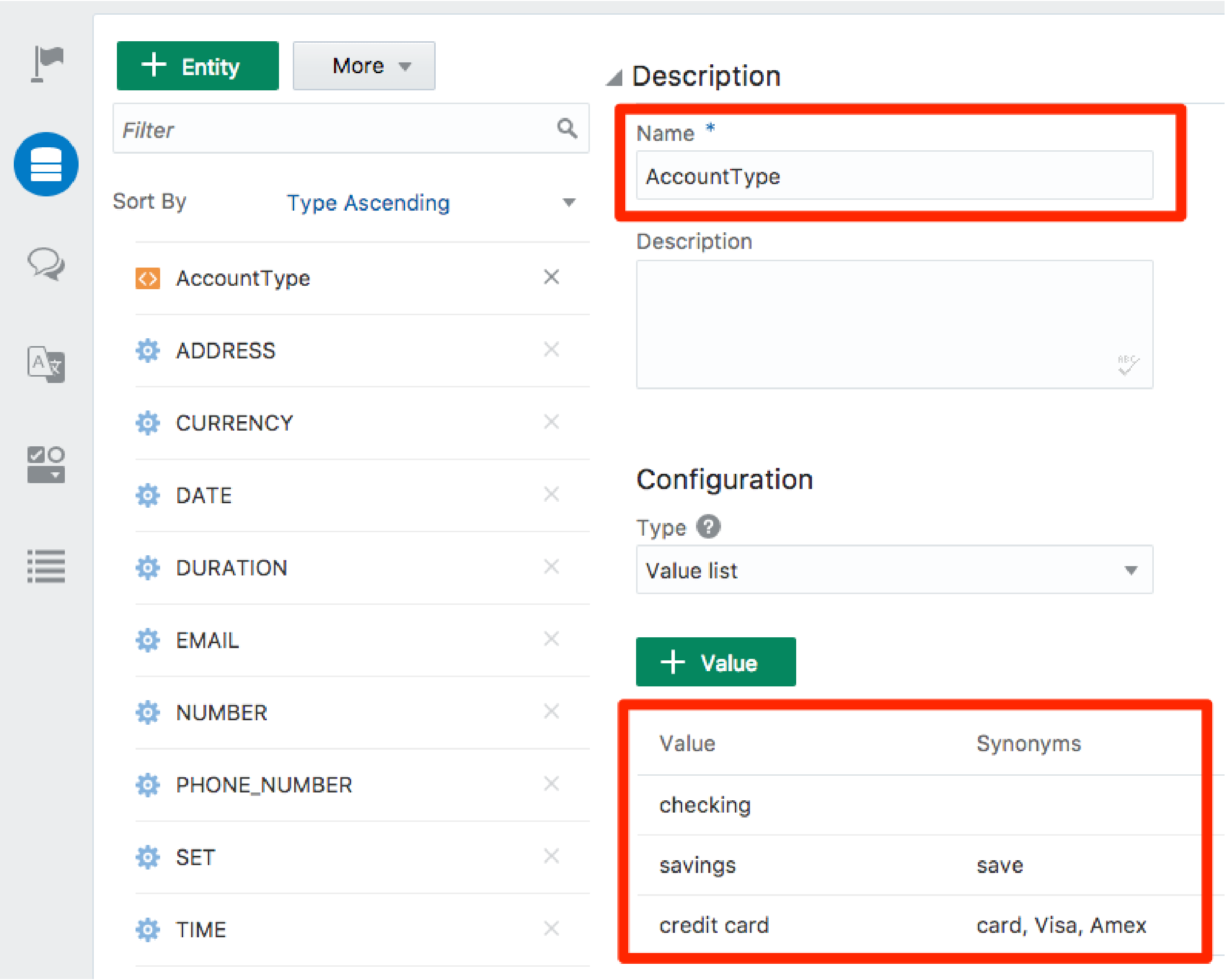
What's the current balance on my cc?

7. Now you're going to add some custom entities just like you did earlier. First, click the **Entities** icon in the left navbar and then add the AccountType entity as a value list entity.

8. Add these values and Synonyms and synonyms to the AccountType value list entity:

Entity Name	***Values***	***Synonyms***
AccountType	checking savings credit card	save Visa, Amex, card

Oracle MOOC: Oracle Intelligent Bots



Name: AccountType

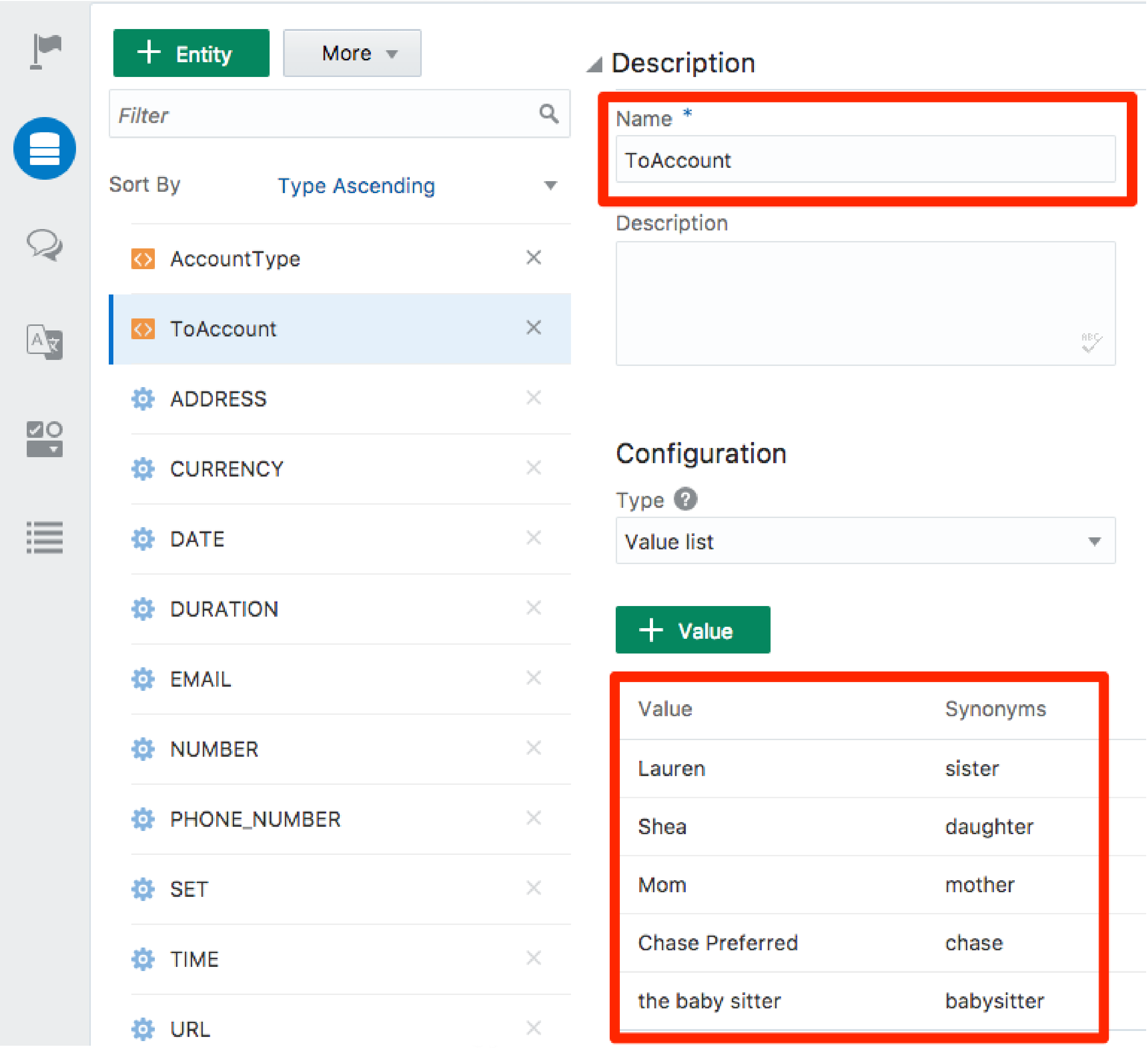
Type: Value list

Value	Synonyms
checking	
savings	save
credit card	card, Visa, Amex

9. Now, using the tables below, add a couple more custom entities.

   a. The first one, ToAccount, is for the recipients of money transfers.

Entity Name	***Values***	***Synonyms***
ToAccount	Lauren Shea Mom Chase Preferred the baby sitter	sister daughter mother Chase babysitter



Sort By Type Ascending

- AccountType
- ToAccount
- ADDRESS
- CURRENCY
- DATE
- DURATION
- EMAIL
- NUMBER
- PHONE_NUMBER
- SET
- TIME
- URL

Description

Name *: ToAccount

Description:

Configuration

Type: Value list

Value	Synonyms
Lauren	sister
Shea	daughter
Mom	mother
Chase Preferred	chase
the baby sitter	babysitter

Oracle MOOC: Oracle Intelligent Bots

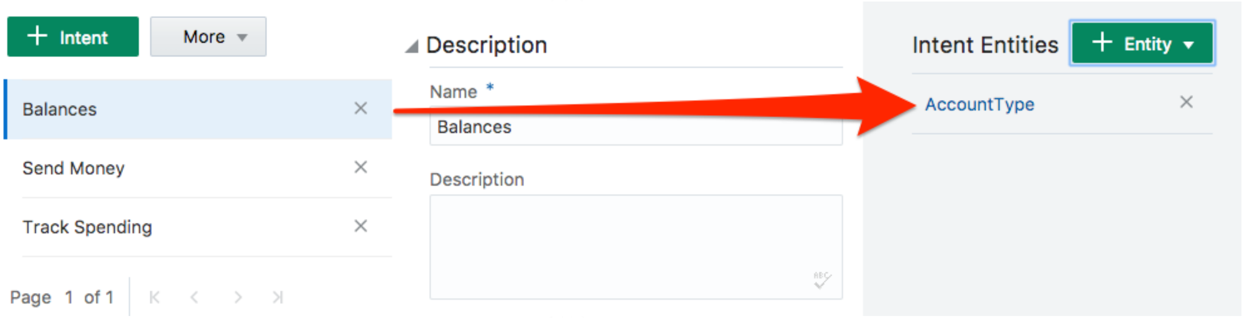
b. The TrackSpendingCategory custom entity defines the categories used to track spending. This entity has no synonyms.

Entity Name	***Values***
TrackSpendingCategory	gas retail travel uber restaurants coffee grocery

10. Now that you have intents and entities, go back to Intents page and assign the following associations like you did earlier:

a. Use the green Add Entity button to select the entity.

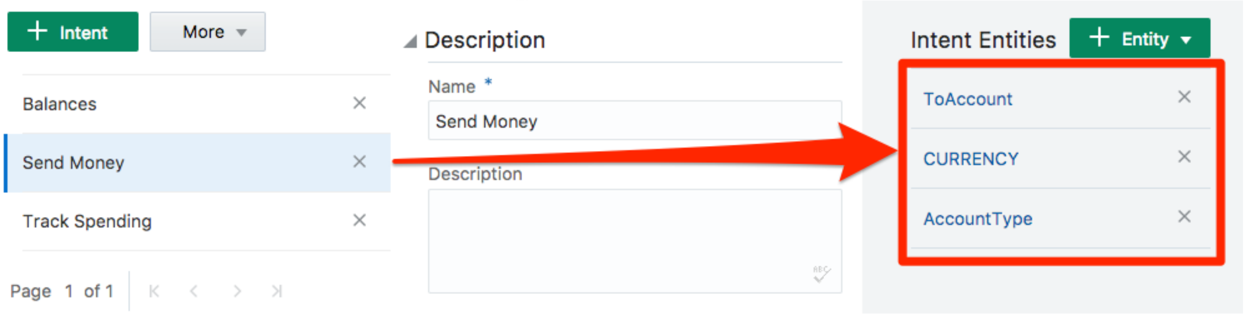
Intent	***Entity***
Balances	AccountType



b. Now associate the Send Money intent with its entities.

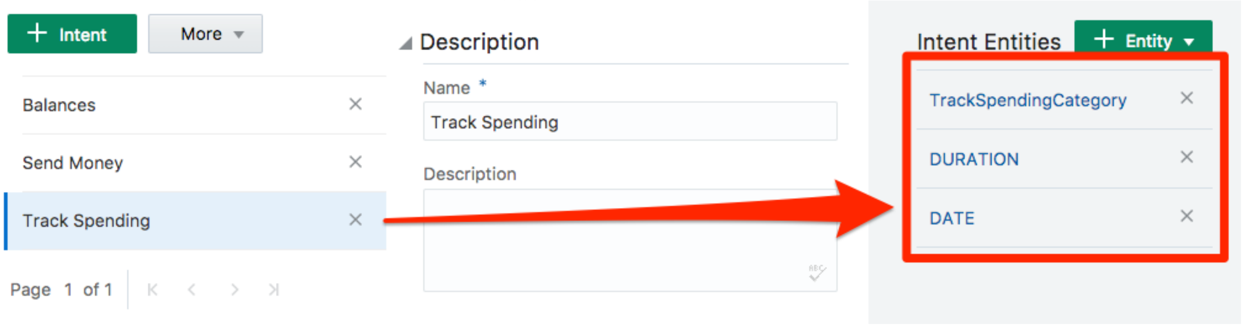
Oracle MOOC: Oracle Intelligent Bots

Intent	***Entities***
Send Money	AccountType CURRENCY (system entity) ToAccount



c. Finally, associate the Track Spending intent with its entities.

Intent	***Entities***
TrackSpending	DATE (system entity) DURATION (system entity) TrackSpendingCategory



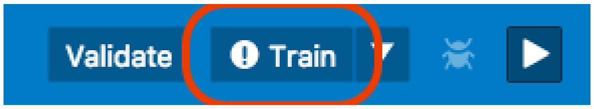
In the next section, you will train the chatbot and then test the intents.

Oracle MOOC: Oracle Intelligent Bots

Step 5: Test the Results

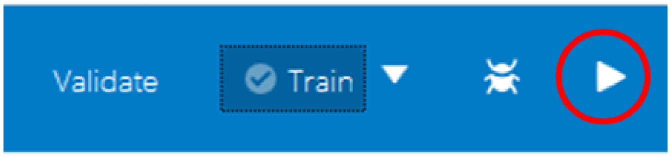
In this section, you will test the intents that you've just created.

1. In the upper-right side of the page, click the **Train** button to test the chatbot. Since you've added a few intents, the training process could take a couple of minutes. When the training is complete, the **Train** button will gray out a display a check mark.



2. Even though we've set up the Send Money and Track Spending intents, we are going to explore the Neural Net and Natural Language Processing (NLP) pipelines through the lens of the Balances intent instead.

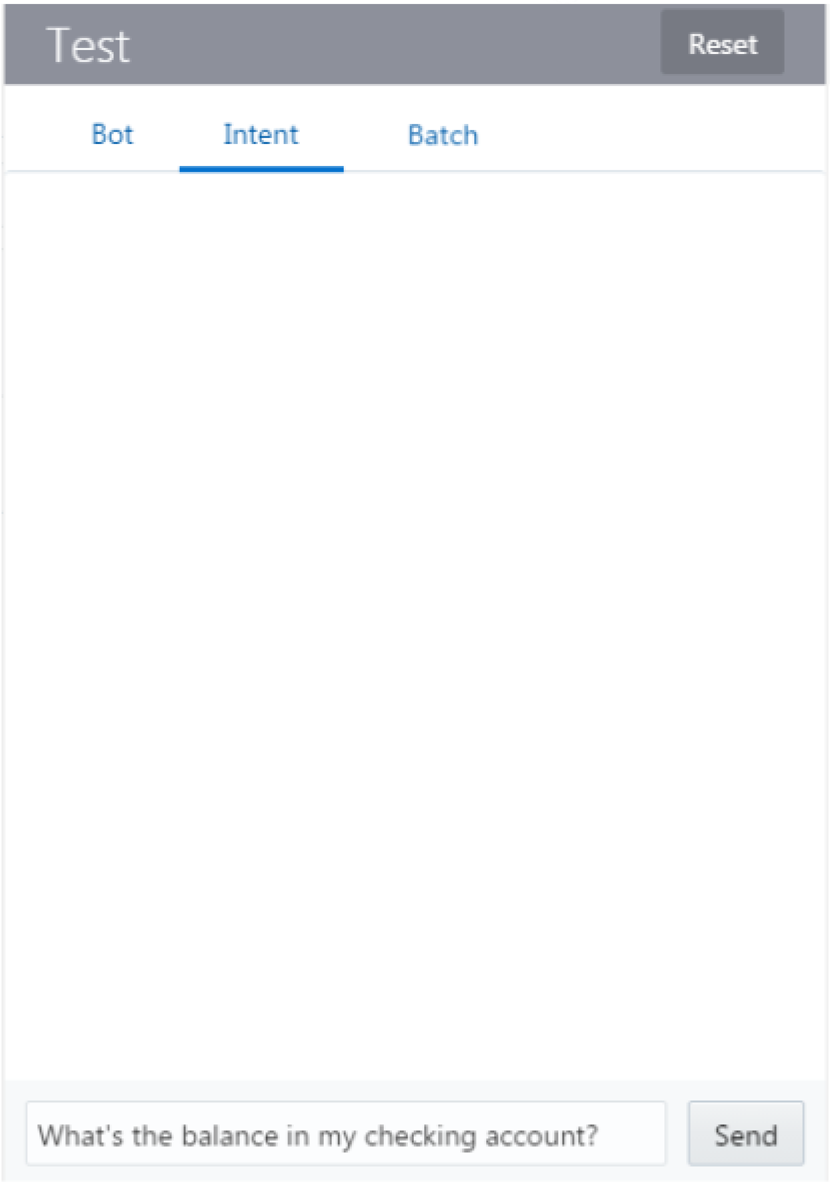
   Click the **Play** icon to open the Tester.



3. To examine how the training worked, click **Intent** tab in the Tester.

4. Then in the Message area, enter *What's the balance in my checking account?* and then click **Send**.

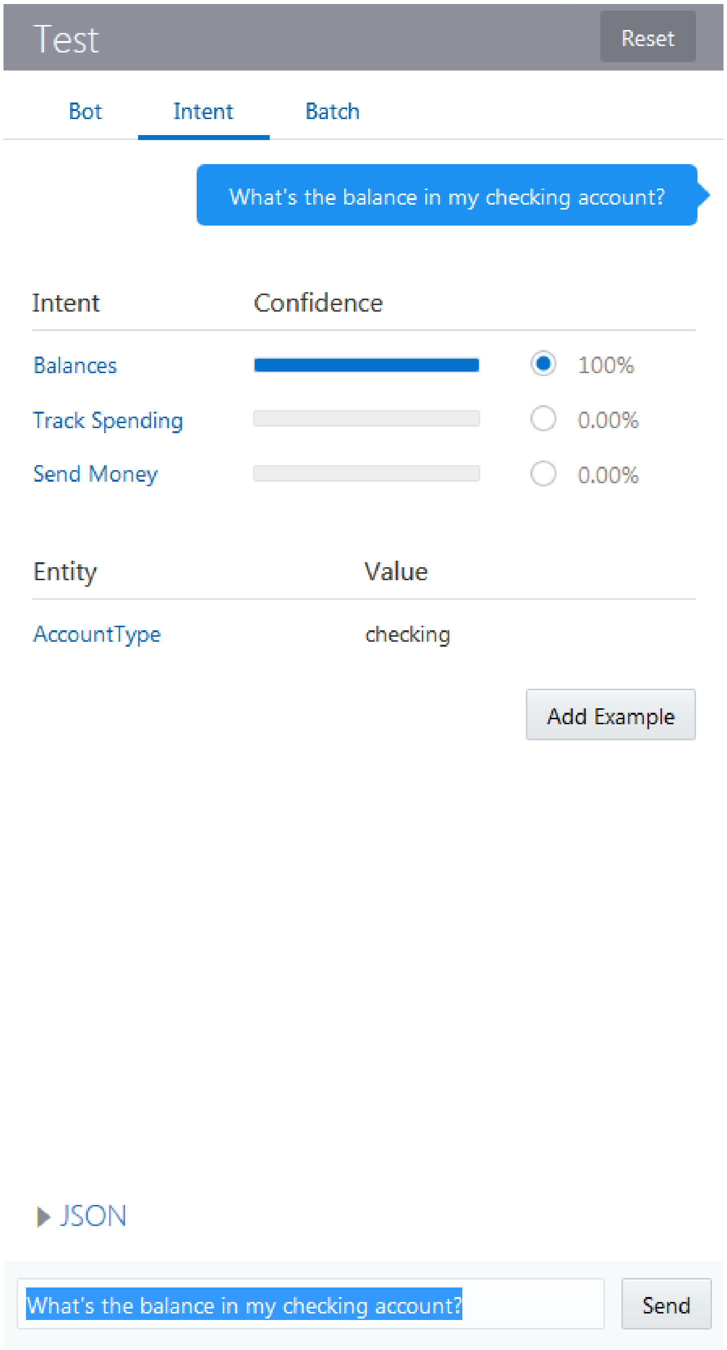
Oracle MOOC: Oracle Intelligent Bots



What's the balance in my checking account?

The Tester displays a list of all of the intents that you've added, each with a confidence percentage.

5. Notice how the Balances intent is first in the list of intents because the message that you just sent is specifically about balances.



What's the balance in my checking account?

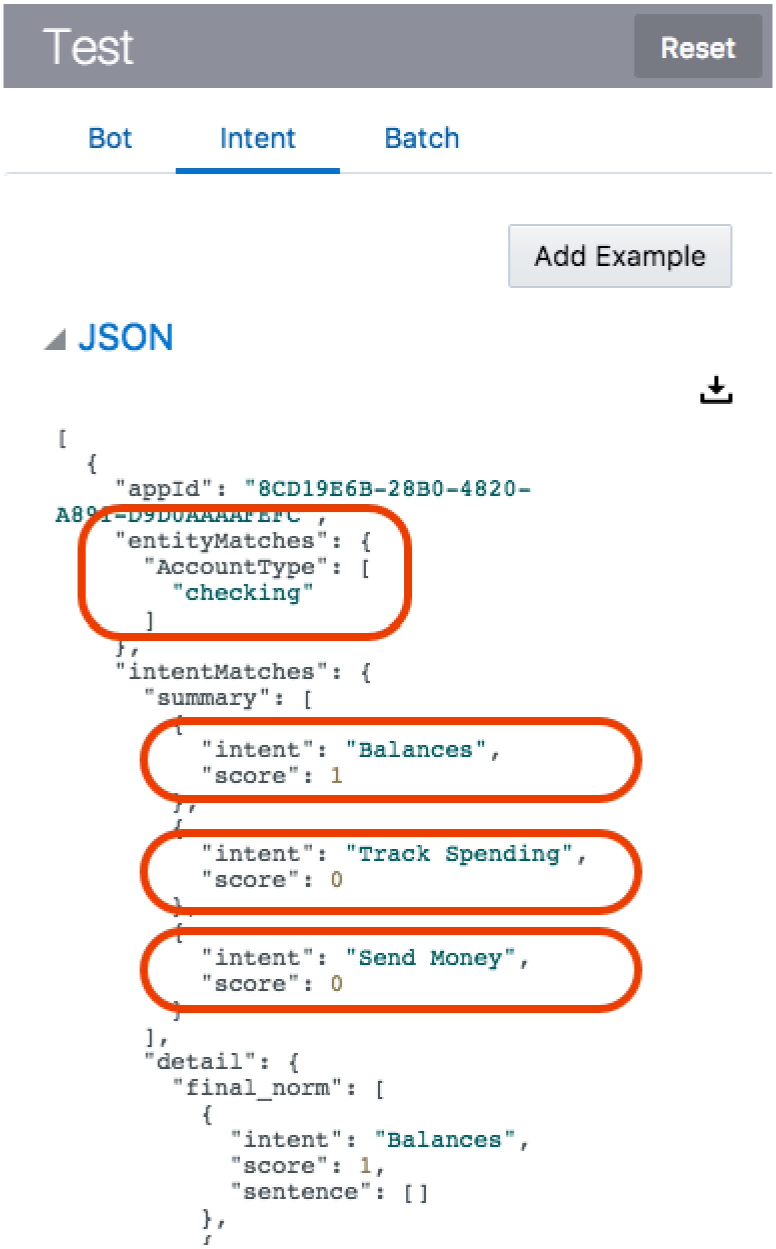
Intent	Confidence	
Balances		● 100%
Track Spending		○ 0.00%
Send Money		○ 0.00%

Entity	Value
AccountType	checking

What's the balance in my checking account?

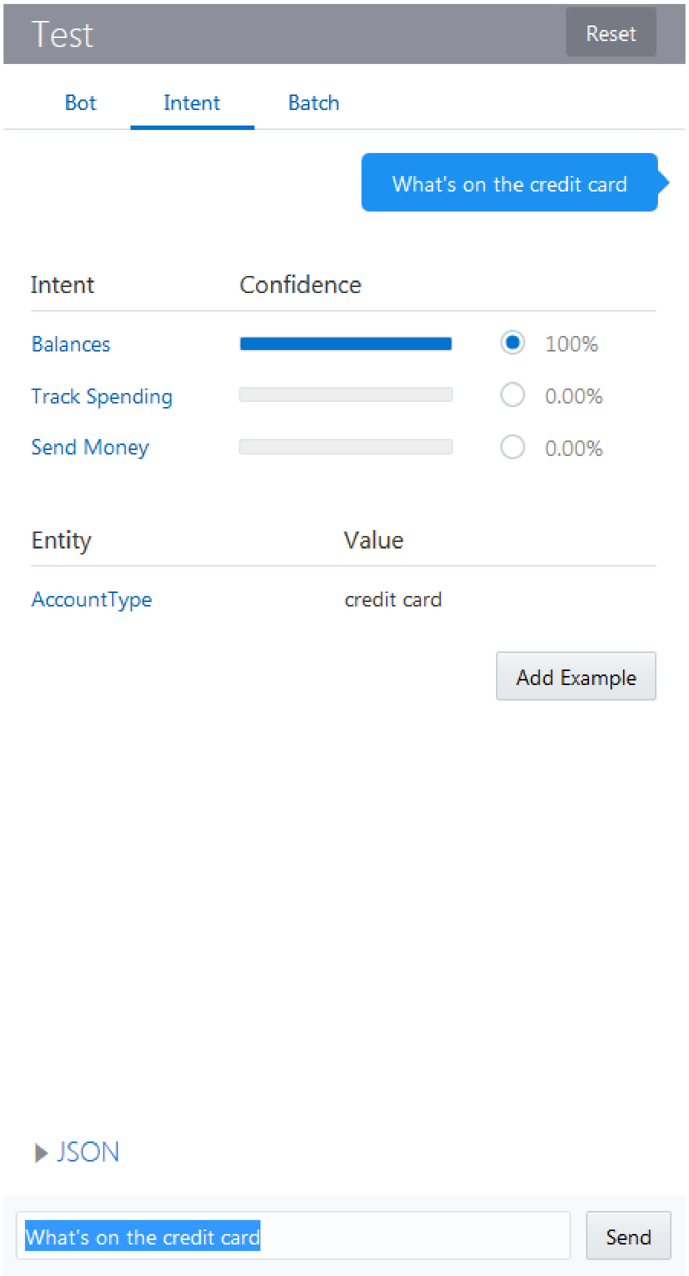
6. Now, click **JSON** (located above the message area) to see what has been returned by the algorithms. Using the slider bar to scroll down, you can see the account type and intent matches.

Oracle MOOC: Oracle Intelligent Bots



```
[
  {
    "appId": "8CD19E6B-28B0-4820-
A891-D9D0AAAAFEFC",
    "entityMatches": {
      "AccountType": [
        "checking"
      ]
    },
    "intentMatches": {
      "summary": [
        {
          "intent": "Balances",
          "score": 1
        },
        {
          "intent": "Track Spending",
          "score": 0
        },
        {
          "intent": "Send Money",
          "score": 0
        }
      ],
      "detail": {
        "final_norm": [
          {
            "intent": "Balances",
            "score": 1,
            "sentence": []
          },
          ,
```

7. Click the **Reset** button.
8. Now let's try a different message: *What's on the credit card*.
9. Now at this point, you may actually encounter an issue where the A.I. engine identifies an intent that you didn't expect as the more likely candidate to resolve the input. For example, in the following image, you can see that the Balances intent is rated higher than the Track Spending intent for the input, *What's on the credit card*.



What's on the credit card

Intent	Confidence	
Balances		● 100%
Track Spending		○ 0.00%
Send Money		○ 0.00%

Entity	Value
AccountType	credit card

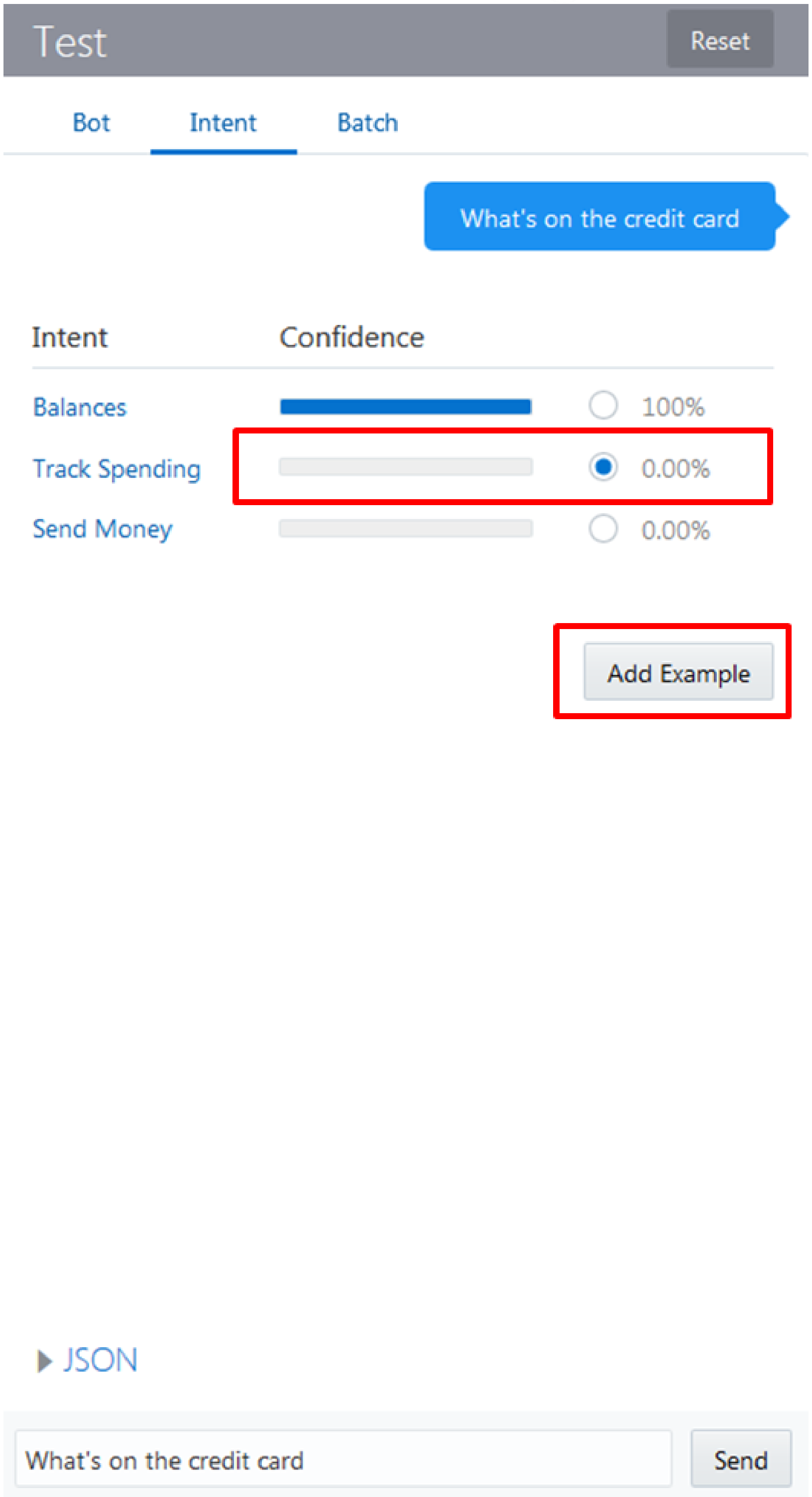
What's on the credit card

10. When this happens, you can increase the confidence level and the intent accuracy by first selecting the radio button by the correct intent and then by clicking the **Add Example** button. Doing this adds the text from the Message area as an utterance for the selected intent.

Be sure that the radio button by the **Track Spending** intent is selected and then click the **Add Example** button.

11. Next, train your chatbot again with this new example phrase.

Oracle MOOC: Oracle Intelligent Bots



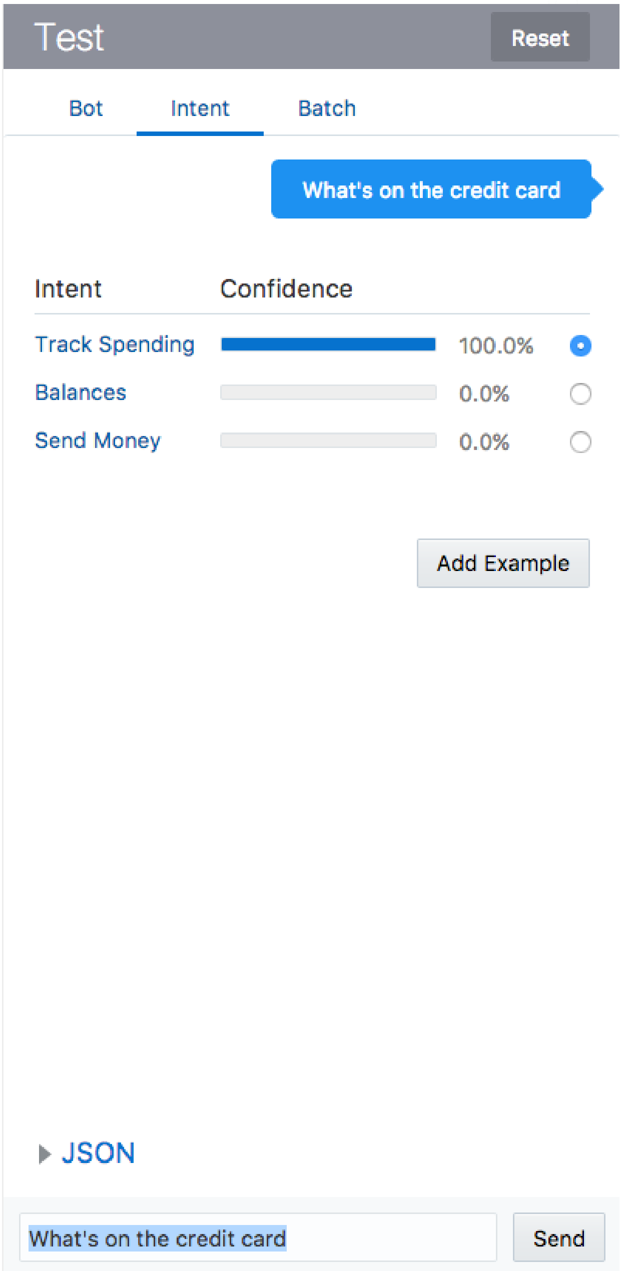
What's on the credit card

Intent	Confidence		
Balances	▬	○	100%
Track Spending		●	0.00%
Send Money		○	0.00%

What's on the credit card

12. Click **Reset** and then enter the same statement (*What's on the credit card*) again. Click **Send**.

The Track Spending intent should now be at the top of list because you added the new utterance and retrained the chatbot. By testing it with additional values, you can increase the pool of example utterances that your intent uses, making it more accurate.



What's on the credit card

Intent	Confidence		
Track Spending		100.0%	●
Balances		0.0%	○
Send Money		0.0%	○

What's on the credit card

You’re done with your second lab! Good job! In the next lab, you will learn more about the flow and BotML.