

AMELIA V3

Day Four



IPsoft Proprietary – For Training Purposes Only
© 2018 IPsoft Inc.



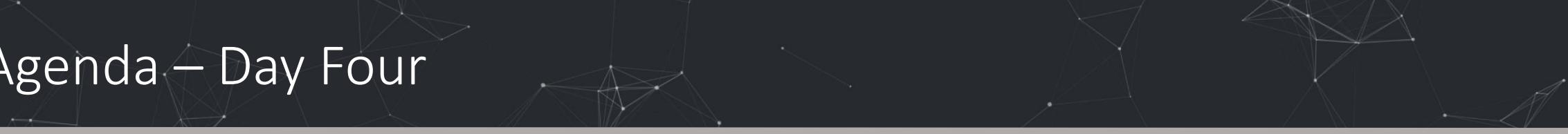
Presented by
IPsoft Global Training and Development

Recap – Day Three

- BPN Version Control
- Amelia Trainer: Additional Tools
- Role Entity
- Spanless Entity
- Intelligent Arbitration
- Semantic Memory: Intent FAQs, Semantic Memory FAQs, Manual
- BPN Script Tasks and Libraries
- Additional Scripting and Services



Agenda – Day Four



- Formatting Entities with quantityService and qty:
- Clarifying Question Asker (CQA)
- Tabular Data
- Custom Features
- Custom Datum Entity Normalization
- Grammars
- Integration Framework
- Putting It Together: BPN and Integration
- Introduction to Amelia Custom UI
- Custom UI Theme Builder
- URL Launch
- Stochastic BPN
- Affective Memory
- Admin and System Settings Overview



Learning Objectives

By the end of this course, you will be able to:

Train Amelia to conduct a business process and answer organization-specific questions using:

- Process design (BPNs)
- Workflow decision-making based on end user responses (prefixes)
- End user intent recognition (intents)
- Customer answer capture (entities)
- Integration with API (integration framework)
- Custom UI elements such as widgets and chat notes (custom UI/script tasks)

Formatting Entities with quantityService and qty:

quantityService format and normalizeAndFormat

Exposed as **quantityService**

- Can normalize/format values from script tasks and libraries

format(AbstractDatum datum)

format(AbstractDatum datum, Formatter formatter)

format(AbstractDatum datum, Formatter formatter, >, timeZones<timezone>)

```
def checkInDateCap = contextService.latestSlot('checkInDate')
def checkInDateVal = quantityService.format(checkInDateCap, Formatters.DATE)
execution.setVariable('checkInDate', checkInDateVal)
```

With String, normalize to AbstractDatum object and format with Formatter:

normalize(String text, DatumType.<datum type>)

normalizeAndFormat(String text, DatumType.<datum type>, Formatters.<formatter>)

normalizeAndFormat(String text, DatumType.<datum type>, Formatters.<formatter>, timeZones<timezone>)

```
def formattedValue = quantityService.normalizeAndFormat("12/01/2018", DatumType.DATE, Formatters.DATE_IN_WORDS)
Returns: Saturday, December 01 2018
```

When would you like to check-in?

Amelia | 8:11:33 AM

tomorrow

8:11:38 AM

So far so good.

Amelia | 8:11:39 AM

2018-04-26 is converted to 04/26/2018.

Amelia | 8:11:39 AM

All done.

Amelia | 8:11:39 AM

Formatters

Simple Formatters	Location Formatters	Unit Formatters	Simple Range Formatters	Unit Range Formatters
<ul style="list-style-type: none"> • BOOLEAN • DATE • DATE_IN_WORDS • DATE_LONG • DATE_MEDIUM • DATE_SHORT • DATE_TIME_12_HR • DATE_TIME_24_HR • DATE_TIME_IN_WORDS_12_HR • DATE_TIME_IN_WORDS_24_HR • DATE_TIME_LONG_12_HR • DATE_TIME_LONG_24_HR • DATE_TIME_MEDIUM_12_HR • DATE_TIME_MEDIUM_24_HR • DATE_TIME_SHORT_12_HR • DATE_TIME_SHORT_24_HR • DATE_TIME_WITH_TIME_ZONE_12_HR • DATE_TIME_WITH_TIME_ZONE_24_HR • DECIMAL • DECIMAL_IN_WORDS • EMAIL • INTEGER • INTEGER_IN_WORDS • ORDINAL • ORDINAL_IN_WORDS • ORGANIZATION • PERSON_FULL_NAME • PERSON_FIRST_NAME • PERSON_LAST_NAME • PHONE_NUMBER • TEXT • TIME_12_HR • TIME_24_HR 	<ul style="list-style-type: none"> • AIRPORT_NAME • AIRPORT_CODE • CAPITAL • COUNTRY_NAME • COUNTRY_CODE • CITY • US_COUNTY • US_STATE_NAME • US_STATE_CODE • STREET_ADDRESS • POSTAL_CODE 	<ul style="list-style-type: none"> • AGE • AGE_IN_WORDS • AREA_METRIC • AREA_IMPERIAL • AREA_METRIC_IN_WORDS • AREA_IMPERIAL_IN_WORDS • CURRENCY • CURRENCY_IN_WORDS • DURATION • DURATION_IN_WORDS • LENGTH_METRIC • LENGTH_IMPERIAL • LENGTH_METRIC_IN_WORDS • LENGTH_IMPERIAL_IN_WORDS • PERCENTAGE • PERCENTAGE_IN_WORDS • SPEED_METRIC • SPEED_METRIC_IN_WORDS • SPEED_IMPERIAL • SPEED_IMPERIAL_IN_WORDS • TEMPERATURE_METRIC • TEMPERATURE_METRIC_IN_WORDS • TEMPERATURE_IMPERIAL • TEMPERATURE_IMPERIAL_IN_WORDS • VOLUME_METRIC • VOLUME_METRIC_IN_WORDS • VOLUME_IMPERIAL • VOLUME_IMPERIAL_IN_WORDS • WEIGHT_METRIC • WEIGHT_METRIC_IN_WORDS • WEIGHT_IMPERIAL • WEIGHT_IMPERIAL_IN_WORDS 	<ul style="list-style-type: none"> • TIME_RANGE_12_HR • TIME_RANGE_24_HR • DATE_RANGE • DATE_RANGE_IN_WORDS • DATE_RANGE_LONG • DATE_RANGE_MEDIUM • DATE_RANGE_SHORT • DATE_TIME_RANGE_12_HR • DATE_TIME_RANGE_24_HR • DATE_TIME_RANGE_IN_WORDS_12_HR • DATE_TIME_RANGE_IN_WORDS_24_HR • DATE_TIME_RANGE_LONG_12_HR • DATE_TIME_RANGE_LONG_24_HR • DATE_TIME_RANGE_MEDIUM_12_HR • DATE_TIME_RANGE_MEDIUM_24_HR • DATE_TIME_RANGE_SHORT_12_HR • DATE_TIME_RANGE_SHORT_24_HR • INTEGER_RANGE • INTEGER_RANGE_IN_WORDS • DECIMAL_RANGE • DECIMAL_RANGE_IN_WORDS • ORDINAL_RANGE • ORDINAL_RANGE_IN_WORDS 	<ul style="list-style-type: none"> • AGE_RANGE • AGE_RANGE_IN_WORDS • AREA_RANGE_METRIC • AREA_RANGE_METRIC_IN_WORDS • CURRENCY_RANGE • CURRENCY_RANGE_IN_WORDS • DURATION_RANGE • DURATION_RANGE_IN_WORDS • LENGTH_RANGE_METRIC • LENGTH_RANGE_METRIC_IN_WORDS • PERCENTAGE_RANGE • PERCENTAGE_RANGE_IN_WORDS • SPEED_RANGE_METRIC • SPEED_RANGE_METRIC_IN_WORDS • TEMPERATURE_RANGE_METRIC • TEMPERATURE_RANGE_METRIC_IN_WORDS • VOLUME_RANGE_METRIC • VOLUME_RANGE_METRIC_IN_WORDS • WEIGHT_RANGE_METRIC • WEIGHT_RANGE_METRIC_IN_WORDS • AREA_RANGE_IMPERIAL • AREA_RANGE_IMPERIAL_IN_WORDS • LENGTH_RANGE_IMPERIAL • LENGTH_RANGE_IMPERIAL_IN_WORDS • SPEED_RANGE_IMPERIAL • SPEED_RANGE_IMPERIAL_IN_WORDS • TEMPERATURE_RANGE_IMPERIAL • TEMPERATURE_RANGE_IMPERIAL_IN_WORDS • VOLUME_RANGE_IMPERIAL • VOLUME_RANGE_IMPERIAL_IN_WORDS • WEIGHT_RANGE_IMPERIAL • WEIGHT_RANGE_IMPERIAL_IN_WORDS

Normalize – Datum Types

Simple Datum

- DatumType.TEXT
- DatumType.BOOLEAN
- DatumType.INTEGER
- DatumType.DECIMAL
- DatumType.ORDINAL
- DatumType.AGE
- DatumType.EMAIL
- DatumType.PERSON
- DatumType.ORGANIZATION
- DatumType.DATE
- DatumType.TIME

Location Datum

- DatumType.AIRPORT
- DatumType.PHONE_NUMBER
- DatumType.POSTAL_CODE
- DatumType.CAPITAL
- DatumType.COUNTRY

- DatumType.US_CITY
- DatumType.US_STATE
- DatumType.US_COUNTY
- DatumType.US_STREET_ADDRESS

Range Datum

- DatumType.DATE_TIME
- DatumType.DATE_RANGE
- DatumType.DATE_TIME_RANGE
- DatumType.DECIMAL_RANGE
- DatumType.INTEGER_RANGE
- DatumType.ORDINAL_RANGE
- DatumType.TIME_RANGE
- DatumType.AGE_RANGE
- DatumType.AREA_RANGE
- DatumType.CURRENCY_RANGE
- DatumType.DURATION_RANGE
- DatumType.LENGTH_RANGE
- DatumType.PERCENTAGE_RANGE

- DatumType.SPEED_RANGE
- DatumType.TEMPERATURE_RANGE
- DatumType.VOLUME_RANGE
- DatumType.WEIGHT_RANGE

Unit Datum

- DatumType.CURRENCY
- DatumType.DURATION
- DatumType.AGE
- DatumType.AREA
- DatumType.LENGTH
- DatumType.VOLUME
- DatumType.WEIGHT
- DatumType.SPEED
- DatumType.PERCENTAGE
- DatumType.TEMPERATURE



Time Zone Examples

TimeZones.ETC_GMT_PLUS_12
TimeZones.ETC_GMT_PLUS_11
TimeZones.MIT
TimeZones.PACIFIC_APIA
TimeZones.PACIFIC_MIDWAY
TimeZones.PACIFIC_NIUE
TimeZones.PACIFIC_PAGO_PAGO
TimeZones.PACIFIC_SAMOA
TimeZones.US_SAMOA
TimeZones.AMERICA_ADAK
TimeZones.AMERICA_ATKA
TimeZones.ETC_GMT_PLUS_10
TimeZones.HST
TimeZones.PACIFIC_FAKAOFO
TimeZones.PACIFIC_HONOLULU
TimeZones.PACIFIC_JOHNSTON
TimeZones.PACIFIC_RAROTONGA
TimeZones.PACIFIC_TAHITI
TimeZones.SYSTEMV_HST10

TimeZones.US_ALEUTIAN
TimeZones.US_HAWAII
TimeZones.PACIFIC_MARQUESAS
TimeZones.AST
TimeZones.AMERICA_ANCHORAGE
TimeZones.AMERICA_JUNEAU
TimeZones.AMERICA_NOME
TimeZones.AMERICA_YAKUTAT
TimeZones.ETC_GMT_PLUS_9
TimeZones.PACIFIC_GAMBIER
TimeZones.SYSTEMVYST9
TimeZones.SYSTEMVYST9YDT
TimeZones.US_ALASKA
TimeZones.AMERICA_DAWSON
TimeZones.AMERICA_ENSENADA
TimeZones.AMERICA_LOS_ANGELES
TimeZones.AMERICA_TIJUANA
TimeZones.AMERICA_VANCOUVER
TimeZones.AMERICA_WHITEHORSE

TimeZones.CANADA_PACIFIC
TimeZones.CANADA_YUKON
TimeZones.ETC_GMT_PLUS_8
TimeZones.MEXICO_BAJANORTE
TimeZones.PST
TimeZones.PST8PDT
TimeZones.PACIFIC_PITCAIRN
TimeZones.SYSTEMV_PST8
TimeZones.SYSTEMV_PST8PDT
TimeZones.US_PACIFIC
TimeZones.US_PACIFIC_NEW
TimeZones.AMERICA_BOISE
TimeZones.AMERICA_CAMBRIDGE_BAY
TimeZones.AMERICA_CHIHUAHUA
TimeZones.AMERICA_DAWSON_CREEK
TimeZones.AMERICA_DENVER
TimeZones.AMERICA_EDMONTON
TimeZones.AMERICA_HERMOSILLO
TimeZones.AMERICA_INUVIK

qty: Prefix

Service prefix for formatting entities in tasks

```
format(String slotCode)  
format(String slotCode, Formatters.<FormatterType>)  
format(String slotCode, Formatters.<FormatterType>, TimeZones.<timeZone>)
```

- Formats latest slot instance with default formatter for datum type of object

Example:

```
say ${qty:format('decimal',Formatters.DECIMAL_IN_WORDS)}
```



What's your favorite decimal?

Amelia | 8:25:59 AM

3.14159265359

8:26:09 AM

3.14.

Amelia | 8:26:10 AM

Three point one four.

Amelia | 8:26:10 AM

Activity



Timing: 20 minutes

Validation: Mind View Testing

quantityService & qty:

- Add a formatter either in a script task using quantityService or in a task using qty: prefix.
- Test your revised BPN in Process Memory in mind view.



Clarifying Question Asker (CQA)

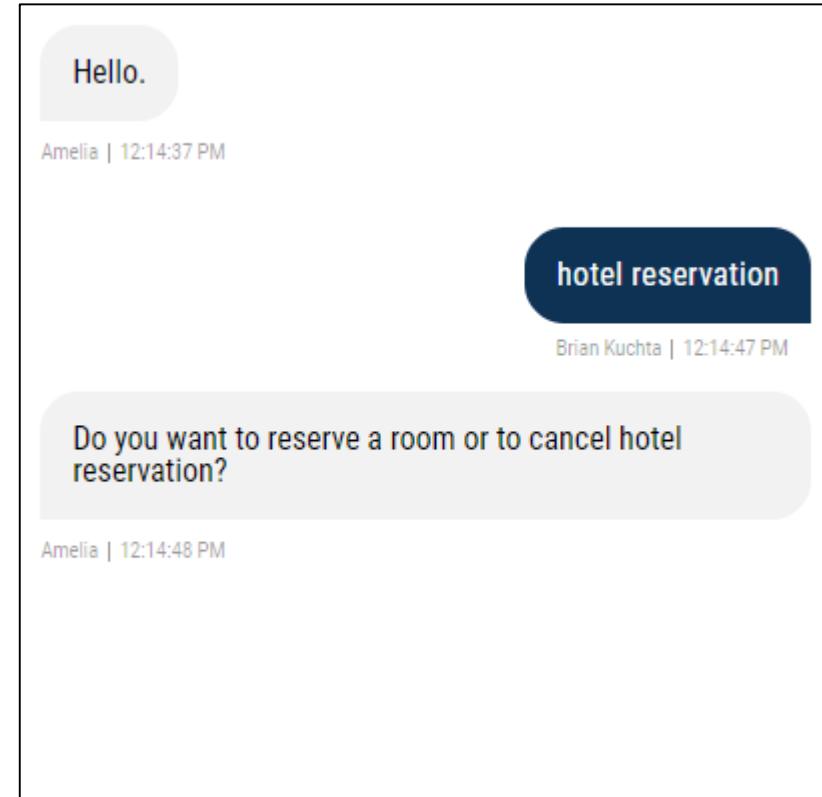


Clarifying Question Asker (CQA)

Generates a clarifying question when possible ambiguity between two similar intents

Example intents with potential for ambiguity:

- Cancel a hotel reservation
- Book a hotel reservation
- Open checking account
- Open saving account
- Order room service
- Order pay per view movie



Clarifying Question Asker (CQA)

Create at least two intents in which there could be ambiguity

- Make a hotel reservation
- Cancel a hotel reservation

Include the relevant example utterances for the intents in the intents or as a separate training corpus uploaded into annotation framework

Intents		
Intent Name	Description	Actions
autoInsurance	Used to get a quote for auto insurance	<input type="checkbox"/>
billDiscrepancy	Used to test Grammar goal	<input type="checkbox"/>
bookLocalAttractions	Used to book local attraction near the hotel	<input type="checkbox"/>
cancelModifyReservation	Used to cancel or change a hotel reservation (stochastic demo)	<input type="checkbox"/>
checkInTimeIntentFAQ	FAQ answer with check in time info	<input type="checkbox"/>
CQACancelModifyReservation	This is a test of the CQA functionality	<input type="checkbox"/>
CQACancelModifyReservationTwo	This is a test of the CQA functionality and include action phrase - The user wants to...	<input type="checkbox"/>
CQAOderPayPerViewMovie	This is a test of the CQA functionality	<input type="checkbox"/>
CQAOderRoomService	This is a test of the CQA functionality	<input type="checkbox"/>
CQAReserveARoom	This is a test of the CQA functionality	<input type="checkbox"/>
<input type="button" value="<"/> <input type="button" value=">"/>	Page 1 of 3 -- 22 total Intents	<input type="button" value="Bulk Actions"/> <input type="checkbox"/> Select All

Clarifying Question Asker (CQA)

Upload in-domain and out-of-domain negative utterances into annotation framework

Annotation Framework

Intents Entities Predict **Annotate**

1. Load / Learn ▾ 2. Annotate ▾ 3. Train ▾ 4. Export ▾

103 total utterances Auto Annotate Train Save As CQANegativeReserveARoo

Intents Entities

A blind date is a date with someone you don't know

Always remember that the night drop is here, and we really appreciate you returning your books for all to use

any way to check the rate on a suite

Are you going to vote in the mayor's race

are you on an exercise program now

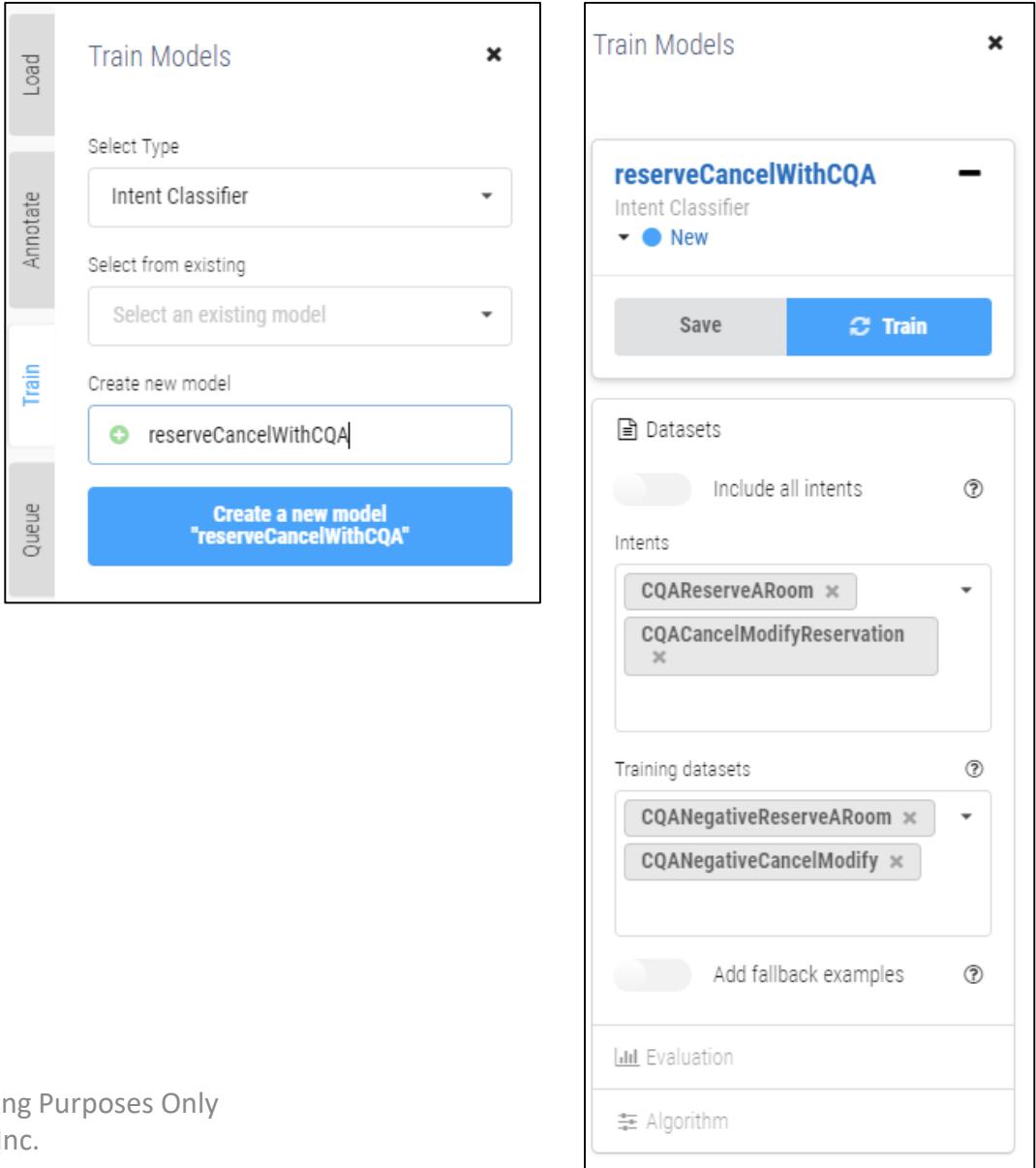
Be careful about the time limits on the streets

Because of work , I can only go look at houses on the weekend

Clarifying Question Asker (CQA)

Train the intent classifier model

- Select the appropriate datasets – example utterances within the intent and/or intent utterances uploaded into annotation framework
- Include the appropriate negative dataset(s)



The screenshot shows two panels of a software application for training machine learning models. The left panel is titled 'Train Models' and has tabs for Load, Annotate, Train (which is selected), and Queue. It contains fields for 'Select Type' (set to 'Intent Classifier'), 'Select from existing' (dropdown menu), and 'Create new model' (text input field containing 'reserveCancelWithCQA'). A large blue button at the bottom right of this panel says 'Create a new model "reserveCancelWithCQA"'.

The right panel also has a title 'Train Models' and includes sections for 'Datasets' (with a toggle for 'Include all intents'), 'Intents' (listing 'CQAReserveARoom' and 'CQACancelModifyReservation'), 'Training datasets' (listing 'CQANegativeReserveARoom' and 'CQANegativeCancelModify'), 'Evaluation' (with a toggle for 'Add fallback examples'), and 'Algorithm' (which is currently empty).

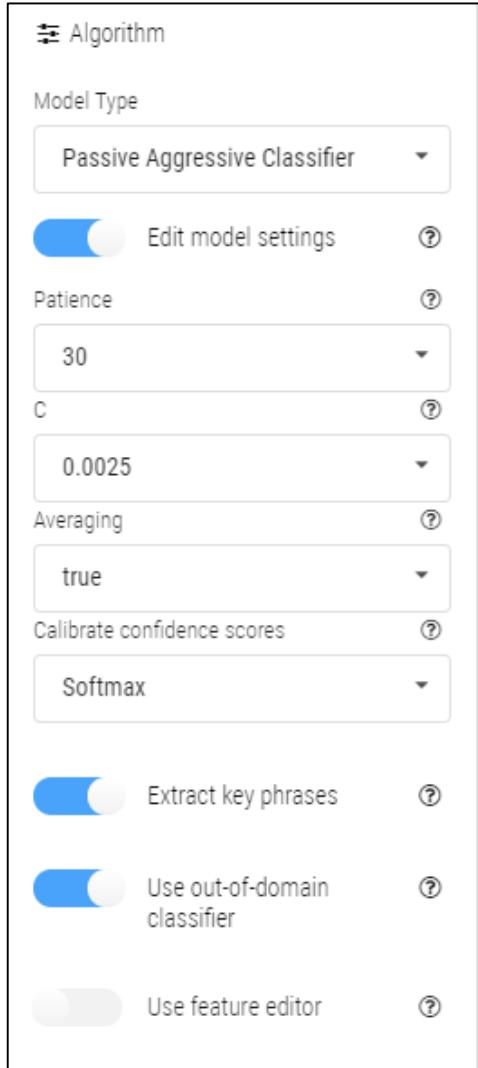
Clarifying Question Asker (CQA)

In the Algorithm panel

- Select your chosen algorithm
 - If using Passive Aggressive, LSVM, or Logistic Regression, set Calibrate Confidence Score to Softmax (recommended)
- Turn on Extract key phrases
- Turn on Use out-of-domain classifier

Ambiguity detection uses intent classifier scores

- Scores need to be calibrated (preferably using SoftMax) – allows for output of probabilities
- Key phrases extracted to distinguish intents/as intent descriptors
- Out-of-domain classifier distinguishes between all in-domain utterances and out-of-domain (negative) to support ambiguity detection



Clarifying Question Asker (CQA)

After training completes, review Dashboard

- Two models created – intent model (distinguishes the two intents) and out-of-domain model (distinguishes between in-domain and out-of-domain)
- Both models need to be deployed to take advantage of CQA

Dashboard

Intents Entities Predict Annotate

Model Name

Model Name Revision Created By Utterances Algorithm Actions Logs Stats Status

> reserveCancelWithCQA v.1.0 Brian Kuchta 390 PAC

> reserveCancelWithCQA.OOD v.1.0 Brian Kuchta 390 PAC

reserveCancelWithCQA_5-fold_cv: Statistics

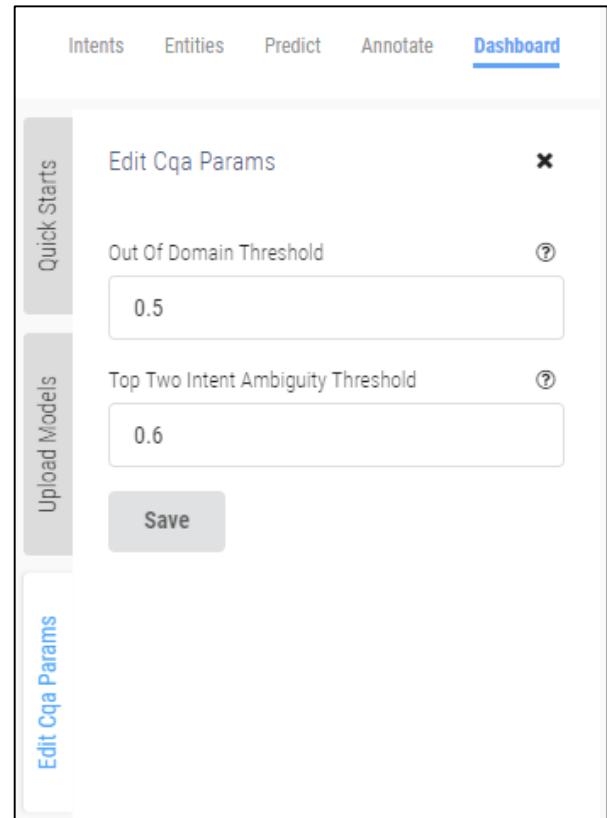
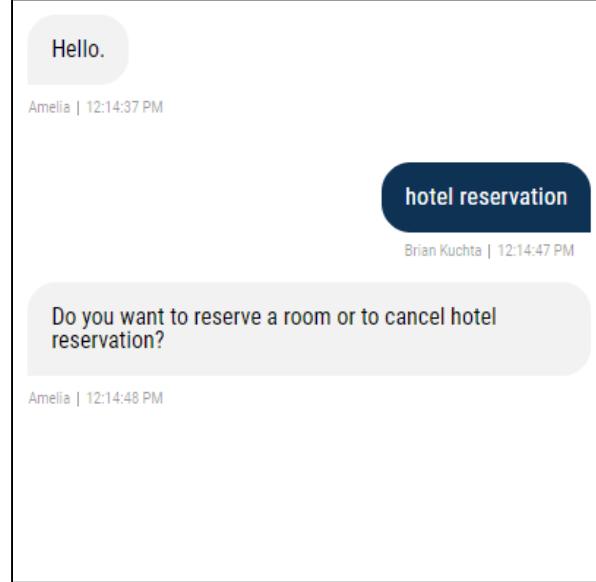
	Correct	Gold	System	Precision	Recall	F1
CQACancelModifyReservation	76	91	83	91.57	83.52	87.36
CQAReserveARoom	84	91	99	84.85	92.31	88.42
Overall	160	182	182	87.91	87.91	87.91

reserveCancelWithCQA.OOD_5-fold_cv: Statistics

	Correct	Gold	System	Precision	Recall	F1
None	199	208	211	94.31	95.67	94.99
in_domain	170	182	179	94.97	93.41	94.18
Overall	369	390	390	94.62	94.62	94.62

How CQA Works

- End user utterance analyzed by Out-Of-Domain model to determine if In-Domain given the Out-Of-Domain threshold
 - If out of domain, negative class is triggered
 - If in domain, determines if there is ambiguity given the confidence score of top two intents
 - If difference between top two intent scores is **not less than Top-Two-Intent-Ambiguity-Threshold**, there is NO ambiguity and winning intent triggers
 - If difference between top two intent scores is **less than Top-Two-Intent-Ambiguity-Threshold**, there is ambiguity and CQA triggers
- Out-Of-Domain Threshold and Top-Two-Intent-Ambiguity threshold set in Dashboard -> Edit Cqa Params



Intents Entities Predict Annotate **Dashboard**

Edit Cqa Params

Out Of Domain Threshold

0.5

Top Two Intent Ambiguity Threshold

0.6

Save

Quick Starts

Upload Models

Edit Cqa Params

Testing CQA in Predict

- Deploy both CQA models in Dashboard
- Enter test utterance in Predict
- Select CQA tab
- If in-domain and difference between top two intents is less than the **Top-Two-Intent-Ambiguity-Threshold** => AMBIGUITY exists

Predict Intents and Entities

Intents Entities Predict Annotate Dashboard

hotel reservation Predict

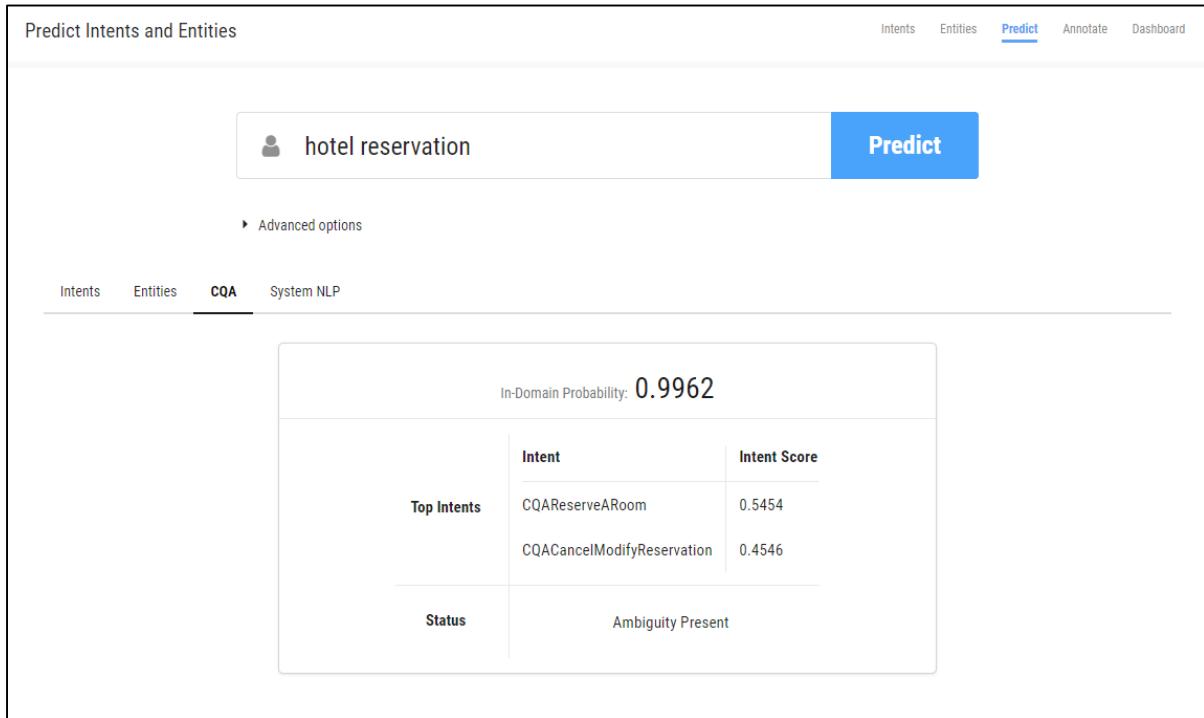
Advanced options

Intents Entities CQA System NLP

In-Domain Probability: 0.9962

Top Intent	Intent	Intent Score
CQAReserveARoom	CQAReserveARoom	0.5454
CQACancelModifyReservation	CQACancelModifyReservation	0.4546

Status Ambiguity Present



Hello.

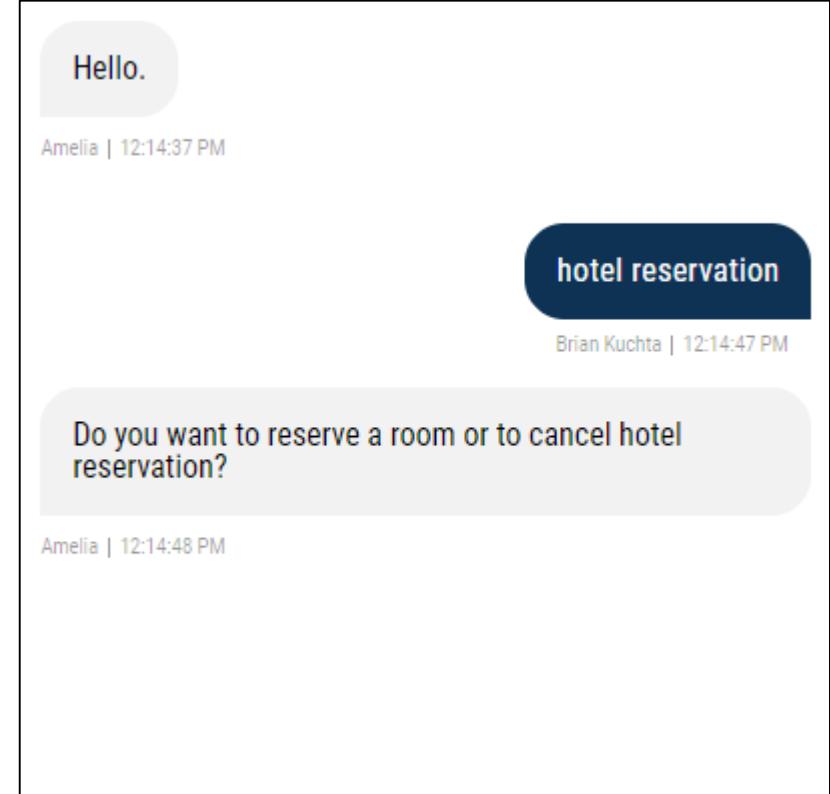
Amelia | 12:14:37 PM

hotel reservation

Brian Kuchta | 12:14:47 PM

Do you want to reserve a room or to cancel hotel reservation?

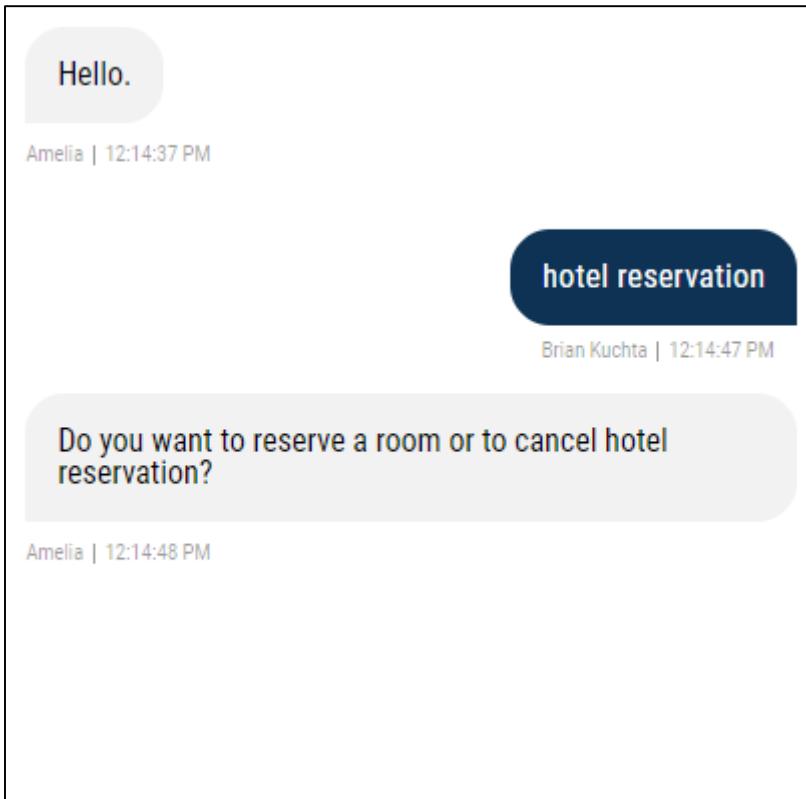
Amelia | 12:14:48 PM



- In mind view, Amelia responds with a clarifying question

Modifying Clarifying Questions

- If ambiguity is detected, Amelia automatically responds with a clarifying question (i.e., a discriminative question)
 - Built from Intent keywords and phrases extracted during model training
 - Can be overwritten by specifying word/phrase in intent -> “The user wants to...”



The screenshot shows the "Intent Settings" page for the intent "CQAReserveARoom". The page includes tabs for Intents, Entities, Predict, Annotate, and Dashboard. The "Intents" tab is selected. The intent name is "CQAReserveARoom". Below it is a note: "This is a test of the CQA functionality and include action phrase - The user wants to...". There are dropdown menus for "When intent is identified..." (set to "Execute Process") and "Execute the process..." (set to "cqaReserveARoom"). A red box highlights the "The user wants to..." field, which contains "reserve a room". Another red box highlights the "Intent keywords and phrases..." section, which lists "book", "reserve", "bed", "a hotel reservation", and "to book". The "USER SAYS" section contains the user input "Because I am there for a conference, need a hotel reservation.". At the bottom right, there are buttons for "Save changes", "Revert changes", "Disable intent", and "Delete". The status bar at the bottom right shows "92 Utterances | See all >".

Tabular Data



Tabular Data

- Allows script tasks to query tabular data tables
 - Static data
- Comma-delimited format recommended
 - < 20 MB
 - Encoding must be UTF-8
 - File must have unique name
 - Column headers mandatory
 - Needs at least one row besides headers
 - Best practice to include ID as first column
 - Note: lowercase Strings are easier to work with

The screenshot shows the IPsoft Process Memory interface. On the left, there's a sidebar with icons for Process Memory, Process Knowledge, Script Library, Tabular Data (which is selected and highlighted in blue), and Content Manager. The main area is titled "Tabular Service". At the top right of this area are tabs for Process Knowledge, Script Library, Tabular Data (selected), and Content Manager. Below the tabs is a button labeled "Import new file". A table lists three tables: "colorsLowercaseTest", "hotelIntentFAQstestHTML", and "movieCostv2". Each table has columns for Table name, Owner, Row count, File size, Domain, Created timestamp, and Actions. Below the table, it says "3 file(s), Page 1 of 1". To the right of the table, there's a detailed view of the "movieCostv2" table, which has columns id, month, hotel_type, room_type, and cost_night. The data rows are:

id	month	hotel_type	room_type	cost_night
1	april	economy	double	100
2	august	economy	single	80
3	august	economy	double	90
4	december	economy	single	75
5	february	economy	double	110
6	january	economy	single	95
7	january	economy	double	115
8	july	economy	single	85
9	june	economy	double	110
10	march	economy	single	83
11	march	economy	double	112
12	may	economy	single	89
13	november	economy	double	104
14	october	economy	single	67
15	october	economy	double	92
16	september	economy	single	84

Tabular Data

Import the CSV file

1 **Import File**
Select and import a new file.

2 **Delimiter and Text Qualifier**
Select the delimiter and text qualifier.

3 **Validate Column Types**
Selected the expected column type for each column.

4 **Summary**

File Upload

Drag and Drop your file in this area
or [Browse](#) your computer

[← Tabular Service Home](#)

Tabular Data

Preview delimiter and text qualifier

1 **Import File**
Select and import a new file.

2 **Delimiter and Text Qualifier**
Select the delimiter and text qualifier.

3 **Validate Column Types**
Selected the expected column type for each column.

4 **Summary**
monthsTest.csv

Metadata Selection
Select the delimiter and text qualifier

Metadata	Options
Delimiter	Comma
Text Qualifier	None

Preview
A preview of the first ten rows is displayed.

id	Raw	Normalized
1	january	JAN
2	february	FEB
3	march	MAR
4	april	APR
5	may	MAY
6	june	JUN
7	july	JUL
8	august	AUG
9	september	SEP

Raw file preview

```
id,Raw,Normalized
1,january,JAN
2,february,FEB
3,march,MAR
4,april,APR
5,may,MAY
```

Tabular Data

Validate Column Types

The interface consists of four numbered steps:

- 1 Import File**: Select and import a new file.
- 2 Delimiter and Text Qualifier**: Select the delimiter and text qualifier.
- 3 Validate Column Types**: Selected the expected column type for each column.
- 4 Summary**: monthsTest.csv

Column type selection
Select the data type for the columns

Column Name	Column Type
id	Integer
Normalized	String
Raw	String

Preview
A preview of the first ten rows is displayed.

id	Raw	Normalized
1	january	JAN
2	february	FEB
3	march	MAR
4	april	APR
5	may	MAY
6	june	JUN
7	july	JUL
8	august	AUG
9	september	SEP

[← Back](#) [Next →](#)



Tabular Data

Review Summary

1 **Import File**
Select and import a new file.

2 **Delimiter and Text Qualifier**
Select the delimiter and text qualifier.

3 **Validate Column Types**
Selected the expected column type for each column.

4 **Summary**
monthsTest.csv

File was validated and imported successfully.

File name: monthsTest.csv

Table name: monthsTest

Row count: 12

Number of columns: 3

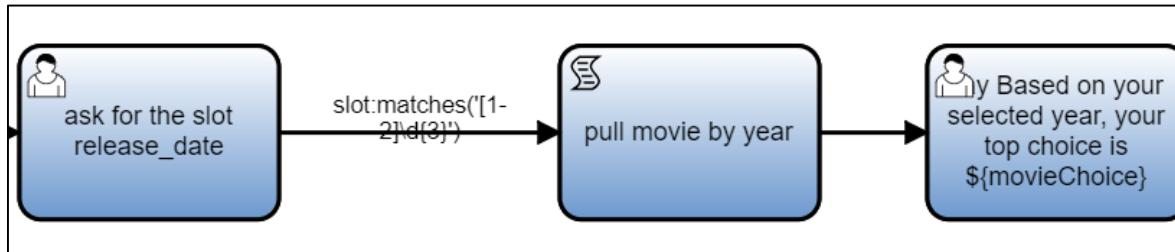
File size: 192 Bytes

[← Back](#) [Okay →](#)



Tabular Data Syntax

- `tabularDataService.query('tableName', 'columnName1', 'columnName2', ...)`
 - Empty parameter(s) throw error
 - Array notation used to access values provided by queries
 - Table name = file name (unless explicitly changed)
 - Must include columns from which values will be extracted as parameter



```
def releaseDateOne = contextService.latestSlot('release_date')

def formattedValueReleaseDate = quantityService.format(releaseDateOne,
    Formatters.INTEGER)

def movieChoice = tabularDataService.query('movieCostv2', 'release_date',
    'title', 'cost').is('release_date', formattedValueReleaseDate).first()
    ()['title'];

execution.setVariable("movieChoice", movieChoice)
```

Tabular Data – Operators

is (columnName, value)

- Like equals operation
- Example: is('cost',formattedValueCost)

isNot (columnName, value)

- Like not equals operation
- Example: isNot('beds',2)

in (columnName, anyAdditionalParameters)

- Like IN operation in SQL -- minimum of two parameters
- All rows that satisfy at least one parameter returned
- Example: in('baths',1,1.5,2)

notIn (columnName, anyAdditionalParameters)

- Like NOT IN operation in SQL -- minimum of two parameters
- Example: notIn('release_date',2015,2016,2017)

isGt (columnName, value)

Example: isGt('baths',0)

isGte (columnName, value)

Example: isGte('baths',1)

isLt (columnName, value)

Example: isLt('cost',formattedValueCost)

isLte (columnName, value)

Example: isLte('area',1200)

like (columnName, value)

likeAny (columnName, values)

- ? -> single character
- %->multiple characters

Example: like('title','wond?r woman')

Example: like('title','wonder wom%')

sortAsc(columnName)

sortDesc(columnName)

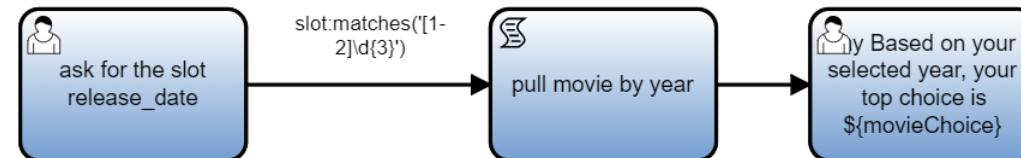
Example: sortDesc('cost')

```
def movieCost = contextService.latestSlot('cost')

def formattedValueCost = quantityService.format(movieCost, Formatters
    .INTEGER)

def movieChoice = tabularDataService.query('movieCostv1.2','release_date'
    , 'title', 'cost').is('cost',formattedValueCost).first()['title'];

execution.setVariable("movieChoice", movieChoice)
```





Tabular Data – Operators

first()['columnName']

last()['columnName']

- Returns a map representing the first/last row that satisfies query criteria
- Example: `first()['title']`
- Example: `last()['release_date']`

list operator

- Returns map of maps representing all rows that satisfy the query criteria
- Each inner map = row

firstN(n) ['columnName']

lastN(n) ['columnName']

- Returns a map of maps representing the first/last N rows that satisfy the query criteria
- Each inner map = row
- N = integer

index(N)

- Returns a map representing a row at index N
- N = integer

first(), last() and index(N) return one row:

...`first()['column_name']`

...`last()['column_name']`

...`index(10)['column_name']`

list(), firstN(n) and lastN(n) return a map of map
`<index_number, <column_name, column_value>:`

...`list()[index]`

...`firstN(10)[index]`

...`lastN(10)[index]`

Value of a particular row:

...`list()[index]['column_name']`

...`firstN(10)[index]['column_name']`

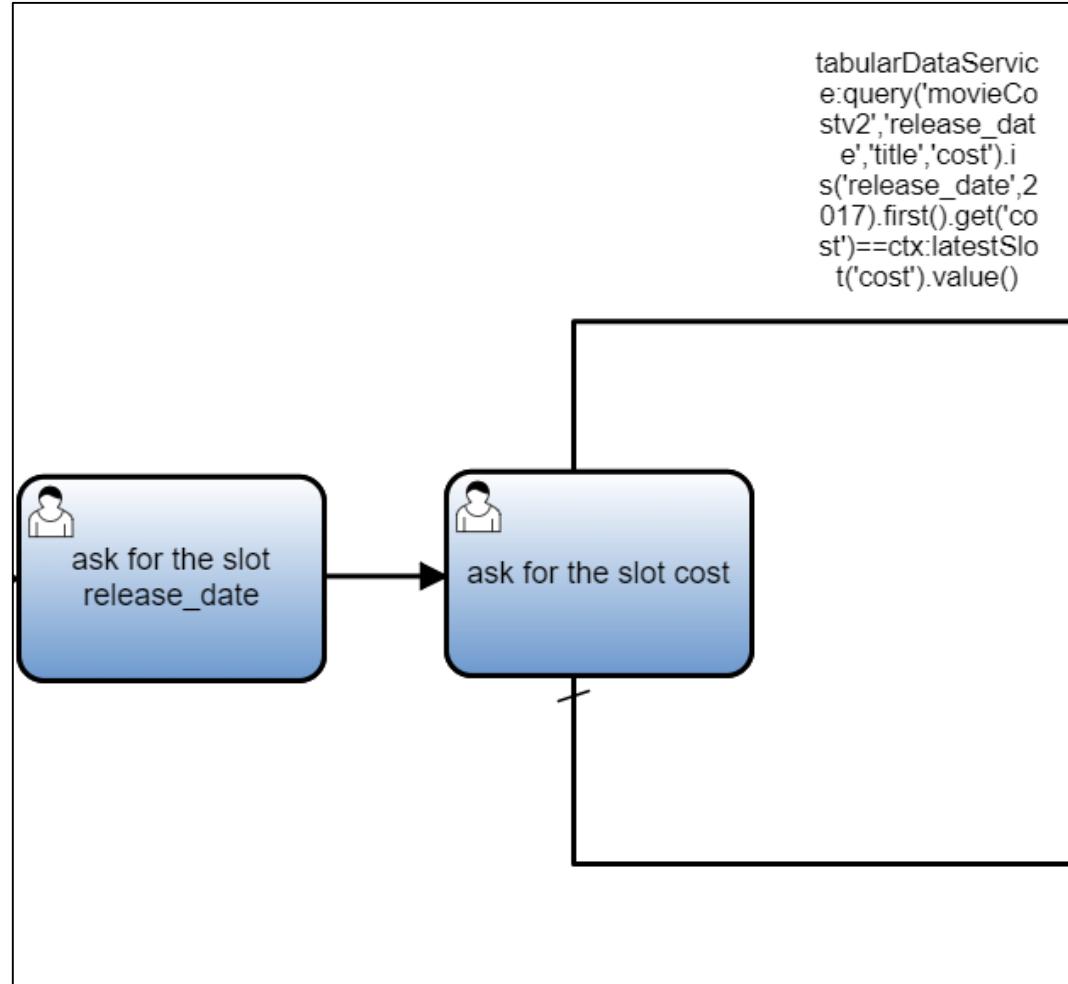
...`lastN(10)[index]['column_name']`

tabularDataService Prefix

- Used for testing data from query of tabular data and responses from end user

Syntax:

```
tabularDataService:query('tabularDataSetName','column1','release_date','cost').is('release_date',2017).first().get('cost') == ctx:latestSlot('cost').value()
```



Custom Features



Feature Extraction Framework

Supports Amelia's ability to select the best features from end user responses

- Allows referencing of domain-specific knowledge from tabular data set
 - Behind the scenes, XML descriptors specify complex feature extraction pipelines for sequence tagging and classification

The screenshot displays the 'Model Feature Editor' interface, which includes three main sections:

- Features**: A list of existing features:
 - unigrams (Unigrams/Lemma)
 - bigrams (Bigrams/Lemma)
 - gameApplication (SyntacticBigrams/Gazetteer)A green '+' button labeled 'Add Feature' is located at the bottom.
- Interactive Tester**: A section titled 'view extracted values for current features' with a text input field 'Enter a test utterance' containing 'enter|a bigrams'. Below it, several blue rectangular boxes show extracted tokens: 'enter|unigrams', 'a|unigrams', 'test|unigrams', and 'utterance|unigrams'.
- Preprocessors**: A list of existing preprocessors:
 - Replace Mentions (ReplaceMentions)
 - Punctuation Filter (PunctuationFilter)A green '+' button labeled 'Add Processor' is located at the bottom.



Resource - Gazetteer

Import tabular data with domain-specific knowledge into Amelia
- Include ID as first column

id	key	color
2	color	absolute zero
3	color	acid green
4	color	aero
5	color	aero blue
6	color	african violet
7	color	air force blue
8	color	air superiority blue
9	color	alabama crimson
10	color	alabaster
11	color	alice blue
12	color	alien armpit
13	color	alizarin crimson
14	color	alloy orange
15	color	almond
16	color	amaranth
17	color	amaranth deep purple

Three types of Gazetteers:

Exact:

- Only matches when annotation text exactly equals gazetteer entry

Regex:

Gazetteer entries=> regex – compiled/used to match against input text
Example dictionary accepted by regex gazetteer:

- PHONE ((\+?\d)?[- .]?(\\([]?\d{3}[]?)|\\d{3})[- .]?)?\\d{3} *[- .]? *\d{4}
- 7DIGIT \d{7}
- 9DIGIT \d{9}
- 10DIGIT \d{10}
- EMAIL [^\s@]+@[^\s@]+\.[^\s@]+

Approximate Matching (Levenshtein distance):

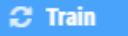
- Approximately matches gazetteer entries with target annotations
- Match=> distance between gazetteer and target annotations below manually set threshold
- Distance = 1

Threshold set at maximum-accepted Levenshtein distance:
"New York" matches gazetteer entry key "New York"

Custom Features – Entities

Train Models

colors
Entity Tagger
New

Save 

Datasets

Evaluation

Algorithm

Model Type
Passive Aggressive Tagger

Edit model settings

Use feature editor



Model Feature Editor

Features modify or create new features

- lemma_-2_-1 Collocation/Lemma
- lemma_-2 Collocation/Lemma
- lemma_-1 Collocation/Lemma
- lemma_0 Collocation/Lemma
- shape_0 Collocation/Form
- lemma_1 Collocation/Lemma
- lemma_2 Collocation/Lemma
- prediction_-2 Collocation/Prediction
- prediction_-1 Collocation/Prediction

Add Feature

Interactive Tester view extracted values for current features

Enter a test utterance

enter lemma_0 a lemma_1 test lemma_2

xxxxx shape_0

enter lemma_-1 enter lemma_1 utterance lemma_0 <OOB> prediction_-2

x shape_0

a lemma_-1 enter lemma_1 test lemma_0 utterance lemma_0 <OOB> prediction_-2

Model Feature Editor

Feature Name
testCustomFeature

Context Type
Collocation

Collocation Offsets
0 ×

Extractor
Lookup

Lookup Type
Exact Match

Check for Negation

Table
colorsLowercase

Input Column
color

Output Column
key

String Preprocessors
Lowercase × Normalize Digits × Normalize Unicode ×

Delete Cancel or 

Using custom features allows for the support of entity extraction including preprocessors and lookup against a tabular data set

Use Custom Features – Intents

Train Models

reserveATable
Intent Classifier
New

Save  Train

Datasets

Evaluation

Algorithm

Model Type
Passive Aggressive Classifier

Edit model settings 

Extract key phrases 

Use out-of-domain classifier 

Use feature editor 

 Open feature editor

Model Feature Editor

Features modify or create new features

- unigrams Unigrams/Lemma
- bigrams Bigrams/Lemma
- + Add Feature

Interactive Tester view extracted values for current features

Enter a test utterance 

utterance bigrams a|test bigrams
test|utterance bigrams enter unigrams
a unigrams test unigrams
utterance unigrams

Model Feature Editor

Name
Punctuation Filter

Preprocessor Type
Punctuation Remover

Punctuation regex
`[!"#$%&(')+,-./;<=@^_`{}|\~\[\]\\\]+`

 Delete  or  Submit

Using custom features can support intent models including specification of punctuation filter, stop words, and coreference resolution

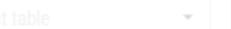
Model Feature Editor

Feature Name
Enter feature name

Context Type
Unigrams

Extractor
Lemma

Check for Negation 

Table  Input Column  Output Column 

String Preprocessors
Lowercase  Normalize Digits  Normalize Unicode 

 Delete  or 



Preprocessors

Intents:

- Punctuation Remover
- Stopword Filter
- Replace Mentions

Model Feature Editor

Name
Enter pre-processor name

Preprocessor Type
Punctuation Remover

Punctuation Remover Strip off punctuation matching the given regex.

Stopword Filter Remove stopwords from input text.

Replace Mentions Replace words like "it", "she", "they", "those", etc. with referring mentions using coreference resolution.

Entities:

- Lowercase
- Normalize Digits
- Normalize Unicode
- Ignore Digits
- Normalize Words
- Extract Prefix
- Extract Suffix

String Preprocessors

Lowercase × Normalize Digits × Normalize Unicode × |

Ignore Digits	Replace numbers with a single character
Normalize Words	Extract only shape of words based on capitalization/digits
Extract Prefix	Extract the first two characters of a word
Extract Suffix	Extract the last three characters of a word

Custom Datum Entity



Custom Datum Entity

- Supports extraction or normalizes the value stored in the entity
- Can be used in lieu of training entity tagger
 - Best for when there is a fixed number of values that can fill entity

1. Add entity
2. Select CUSTOM_DATUM for datum type
 - Enables lookup from tables or grammars

Add Entities

1

Name *

Code *

Datum Type *

Select Datum Type

Description

Description

Spanless Entity

Add

2

Datum Type *

AGE

CUSTOM_DATUM

DATE

Custom Entity Type

- Maps values extracted from end user responses to specific normalized values
- Used to group related words/phrases and map to useful values
- Specify lookup table, input column, output column
 - Include unique ID in first column

Custom Entity Settings

Custom entity type

Tabular Data Lookup

Extract/Normalize

Extract and Normalize

Fallback strategy

Original

Lookup strategy

Exact Match

Preprocessors

Lowercase × Normalize Unicode ×

Lookup table

Input column

Output column

applicationsLowercaseNormalized

applications

normalized applications

id	color	normalized color
1	absolute zero	blue
2	acid green	green
3	aero	blue
4	aero blue	blue
5	african violet	purple
6	air force blue	blue
7	air superiority blue	blue
8	alabama crimson	red
9	alice blue	blue
10	alien armpit	green
11	alizarin crimson	red

id	Email	EmailType
1	\S+@gmail.com	GmailAccount
2	\S+@google.com	GoogleAccount
3	\S+@yahoo.com	YahooAccount
4	\S+@msn.com	MSNAccount
5	\S+@me.com	AppleAccount

Extract/Normalize

- Extract and Normalize => returns the normalized value
- Extract Only => returns the original text entered by the end user
- Normalize Only => used with an entity tagger model
 - Must train model manually
 - Spans extracted used as input to normalizer

Custom Entity Settings

Custom entity type

Tabular Data Lookup

Extract/Normalize

Extract and Normalize

Fallback strategy

Original

Lookup strategy

Exact Match

Preprocessors

Lowercase × Normalize Unicode ×

Lookup table

applicationsLowercaseNormalized

Input column

applications

Output column

normalized applications

The screenshot shows the 'Custom Entity Settings' dialog box. Under 'Custom entity type', 'Tabular Data Lookup' is selected. In the 'Extract/Normalize' section, 'Extract and Normalize' is chosen. The 'Fallback strategy' is set to 'Original'. Under 'Lookup strategy', 'Exact Match' is selected. Preprocessors include 'Lowercase' and 'Normalize Unicode'. The 'Lookup table' is set to 'applicationsLowercaseNormalized'. The 'Input column' is 'applications' and the 'Output column' is 'normalized applications'.

Fallback Strategy

- Allows specification of what should happen if extracted value cannot be normalized
 - None => entity will NOT be extracted
 - Original => returns original extracted text
 - Provided => must specify default that is extracted when entity located but not successfully normalized

Custom Entity Settings

Custom entity type
Tabular Data Lookup

Extract/Normalize
Normalize

Fallback strategy
Provided

Fallback *
Magic Color

Lookup strategy
Exact Match

Preprocessors
Lowercase × Normalize Unicode ×

Lookup table
colorsNormalized

Input column
color

Output column
normalized color

Lookup Strategy – Gazetteer Lookup

- Maps values extracted from end user responses to specific normalized values
- Used to group related words/phrases and map to useful values
- Specify table, input column, output column
 - Be sure to include ID as first column

Normalizer Type
Extract and Normalize

Fallback Type
Original

Lookup Strategy
Gazetteer Lookup

Gazetteer Type
Exact Match

Table *
colorsNormalized

Input Column *
color

Preprocessors
Lowercase × Normalize Unicode ×

Output Column *
normalized color

id	color	normalized color
1	absolute zero	blue
2	acid green	green
3	aero	blue
4	aero blue	blue
5	african violet	purple
6	air force blue	blue
7	air superiority blue	blue
8	alabama crimson	red
9	alice blue	blue
10	alien armpit	green
11	alizarin crimson	red

id	Email	EmailType
1	\S+@gmail.com	GmailAccount
2	\S+@google.com	GoogleAccount
3	\S+@yahoo.com	YahooAccount
4	\S+@msn.com	MSNAccount
5	\S+@me.com	AppleAccount



Lookup Strategy

- Exact Match
- Regex Match
- Approximate Match - Uses Levenshtein Distance
 - Distance computed at token-level for phrase entries
 - Only use with small data sets – more computationally intensive

Custom Entity Settings

Custom entity type

Tabular Data Lookup

Extract/Normalize

Extract and Normalize

Fallback strategy

Original

Lookup strategy

Exact Match

Preprocessors

Lowercase × Normalize Unicode ×

Lookup table

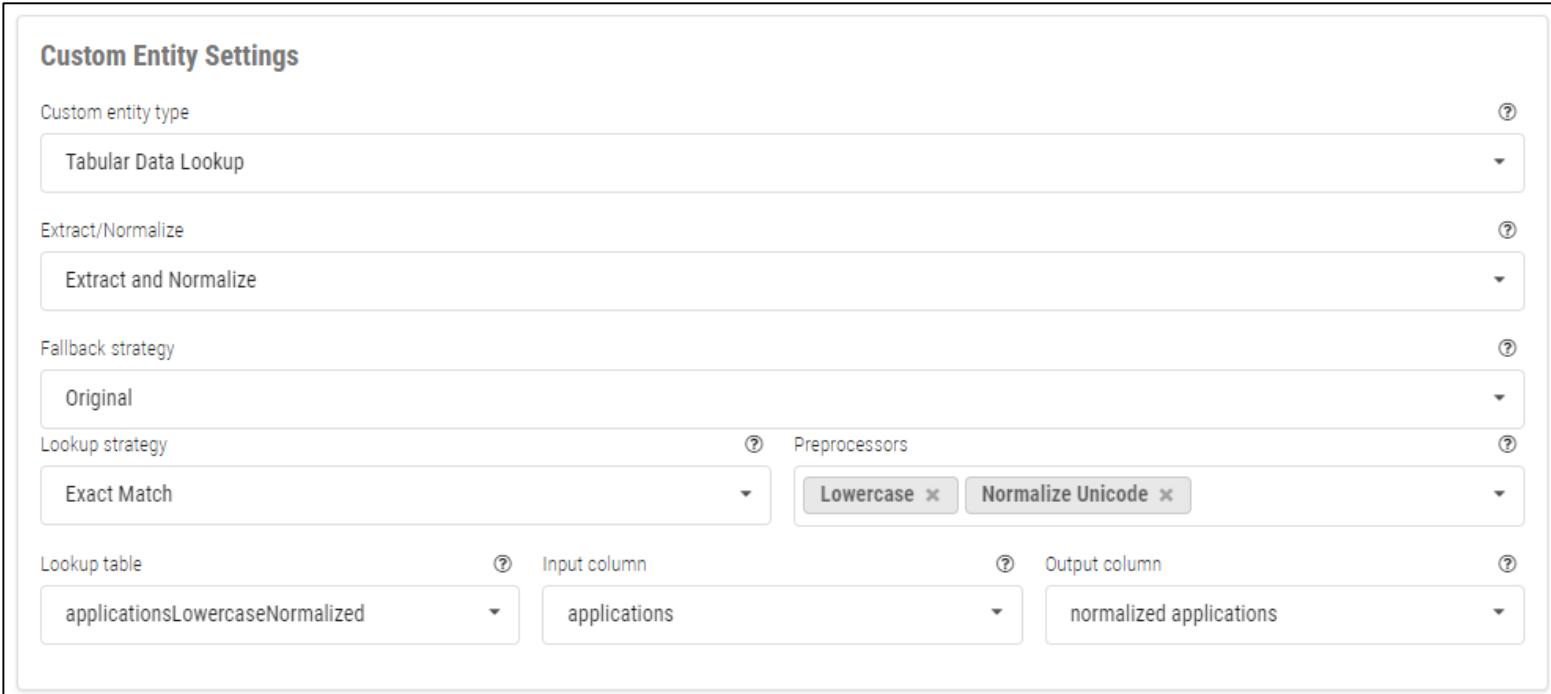
Input column

Output column

applicationsLowercaseNormalized

applications

normalized applications



Activity

Entity Settings

Intents Entities Predict Annotate Dashboard

Applications normalized

Spanless ⓘ Role ⓘ

Code

applications CUSTOM_DATUM

+ Add child

Custom Entity Settings

Custom entity type: Tabular Data Lookup

Extract/Normalize: Extract and Normalize

Fallback strategy: Original

Lookup strategy: Exact Match

Preprocessors: Lowercase, Normalize Unicode

Lookup table: applicationsLowercaseNormalized

Input column: applications

Output column: normalized applications

Timing: 20 minutes
Validation: Mind View Testing

Custom Datum

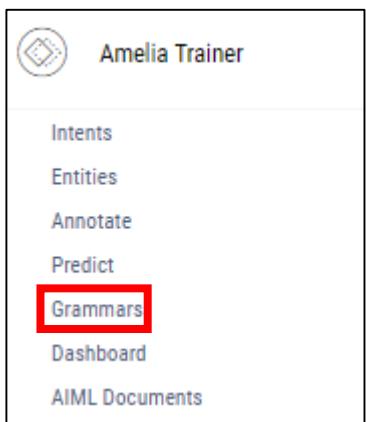
- Create a tabular data set with an ID column, value column, and normalized value column.
- Import into tabular data in process memory.
- Create a new entity, selecting the custom datum type.
- Reference the tabular data set as the gazetteer.
- Add to your BPN.

Grammars



Grammars

- Related to the goals and slots from V2
- Allows for usage of limited set of expected responses
- Must be imported into Amelia Trainer -> Grammars as XML file
- Some notation from V2 allowed



Grammars

Grammar Type	Grammar Name	Operations
GOAL	room_test	[Edit] [Delete]
SLOT	room_type_slot	[Edit] [Delete]
GOAL	hotel_billing_discrepancy	[Edit] [Delete]
SLOT	choice_of_food_type	[Edit] [Delete]
SLOT	new_york_tourist_spot	[Edit] [Delete]

5 Grammars, Page 1 of 1

Import Grammar File
Select Grammar file: Upload

Export Grammar File
Brian_grammar.xml Export File

Grammars – Goals

- Goal exported from V2 instance in XML
 - Could also be created in XML
- Can be used in tandem with intent model
 - If specific utterance being missed by model, can ensure BPN launch by including utterance in Grammar

```
<language-units>
  <lru type="GOAL" name="hotel billing discrepancy" principle="false" askType="DEFAULT">
    <grammar><! [CDATA[
      [_billDiscrepancy]
      (i have a billing discrepancy)
      (i need to discuss *an ISSUE with *a billing SITUATION)
      (billing issue)

      ISSUE
      (problem)
      (refund)
      (credit)

      SITUATION
      (discrepancy)
      (issue)
      (over charge)
    ;
  ]]></grammar>
  <customAskSentences/>
</lru>
</language-units>
```

Goal syntax:

[_reserveARoom]
(i would like to reserve a room)
(can you make a reservation)
;

Grammars - Slots

- Used for both extraction and normalization of entities
 - Matches value in parentheses
 - Pre-terminals - returns pre-terminal value
 - Grammar XML – Slot Name same as Slot Code

```
<language-units>
  <lu type="SLOT" name="choice of food type" principle="false" askType="DEFAULT">
    <grammar><! [CDATA[
      (pizza)
      (salad)
      (bread)
      (hamburger)
    ;
  ]]></grammar>
  <customAskSentences/>
</lu>
</language-units>
```

Slot syntax:

```
[typeOfRoom]
  (single)
  (double)
;
```

Slot with pre-terminal syntax:

```
[typeOfRoom]
  ([_single])
  ([_double})
;
[_single]
  (one bed)
  (king)
;
```

Import Grammar File

Grammar Type	Grammar Name	Operations
GOAL	room_test	[Edit] [Delete]
SLOT	room_type_slot	[Edit] [Delete]
GOAL	hotel_billing_discrepancy	[Edit] [Delete]
SLOT	choice_of_food_type	[Edit] [Delete]
SLOT	new_york_tourist_spot	[Edit] [Delete]

5 Grammars, Page 1 of 1

Import Grammar File

Select Grammar file:

Export Grammar File

Brian_grammar.xml Export File

All Revisions for Grammar: choice_of_food_type

Domain: Brian

Revision No.	Grammar String	Status	Actions
1.0	[typeOfFoodChoice] (pizza) (salad) (bread) (hamburger) ;	NEW	Current Revision



New Intent – Import from Grammars

Add Intents

Enter an intent name

Add a description

Add Intent

Upload Intents

Merge

Format: TSV, Max Size: 2Mb

Drag and Drop your file in this area or [Browse](#) your computer

Import from Grammars

billDiscrepancy

Used to test Grammar goal

When intent is identified...

When resuming the process, Amelia asks...

The user wants to...

Intent keywords and phrases...

USER SAYS

Absolutely sure that there is an issue with my hotel bill

Use intent grammar... hotel_billing_discrepancy

Save changes

Revert changes

Disable intent

Delete

AMELIA

I will process a refund for you.

It's a pleasure to meet you.

Amelia | 2:37:01 PM

I have a billing discrepancy

Brian Kuchta | 2:37:10 PM

I will process a refund for you.

Amelia | 2:37:11 PM

Process Memory

Process:
Version:

Context Group

```
graph LR; Start(( )) --> Node1[say I will process a refund for you]; Node1 --> End(( ))
```

Type message ...

Send

Debug



New Entity – Import from Grammars

Add Entities

Name * Code *

Datum Type *

Select Datum Type

Description

Spanless Entity

Add

Upload Entities

Merge ▾

Format: TSV. Max Size: 2Mb

Drag and Drop your file in this area or [Browse](#) your computer

[Import from Grammars](#)

typeOfFoodChoice

Grammar Test

Spanless ⓘ Role ⓘ

Code typeOfFoodChoice Datum type CUSTOM_DATUM

+ Add child

Custom Entity Settings

Custom entity type Grammars

Extract/Normalize Extract and Normalize

Fallback strategy None

Grammar choice_of_food_type

AMELIA ASKS

What kind of food do you want?

Save Undo Delete

All Revisions for Grammar: choice_of_food_type

Domain: Brian

Revision No.	Grammar String	Status
1.0	[typeOfFoodChoice] (pizza) (salad) (bread) (hamburger)	NEW

Current Revision

Hello, Brian. It's always a pleasure to see you.

Amelia | 9:30:51 PM

run the workflow foodChoiceSlotGrammar

9:30:58 PM

What kind of food do you want?

Amelia | 9:30:58 PM

I want a pizza

9:31:07 PM

I got pizza.

Amelia | 9:31:08 PM

Process Memory

Process: foodChoiceSlotGrammar Version: 2.0

Process Group

Process: foodChoiceSlotGrammar

```
graph LR; Start(( )) --> Ask[ask for the slot typeOfFoodChoice]; Ask --> Say[say i got ${ctx:latest Slot[typeOfFoodChoice].value()}]; Say --> End(( ));
```

IP SOFT

Integration Framework

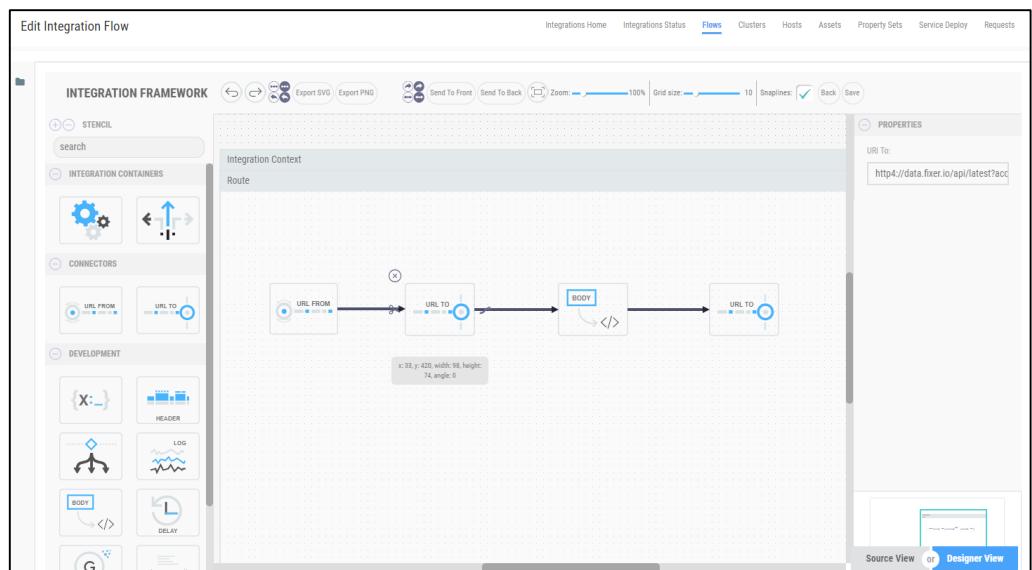


Introduction to V3 Integrations

- Amelia connects to APIs via Integration flows
 - Created in Amelia via Designer tool or Spring XML
 - Deployed to one or more hosts
- Integrations run separately from Amelia in Camel context
- Created in Amelia Admin-> Integrations
 - Deployed over gRPC
 - Integration request comes from BPN and calls deployed flow
 - Integration service unpacks bundle/deploys in Spring application context
 - Uses Spring to do Direct RPC-style call over RabbitMQ
 - Internally hands-off to Camel to execute request
- run the integration flow task triggers integration flow
 - Contains process-scope variables and conversation metadata
 - Can be set as exchange properties for use in integration flow
 - Example: \${exchangeProperty.firstName}
- XML can include property references
 - Encryption possible
- Separate class loader also available for Assets, shielding flows from each other

The screenshot shows the 'Integration Framework' dashboard. It includes sections for 'Services Status' (3 INTEGRATIONS RUNNING, 0 INTEGRATIONS PENDING, 1 ACTIVE HOSTS), 'Integration Flow' (described as enabling communication between heterogeneous applications), 'Cluster' (easy and faster way to deploy and un-deploy a group of Hosts at the same time), 'Hosts' (management connection to network nodes), and 'Property Sets' (manage properties shared across flows). A table titled 'Top 5 Integration Status' lists host names, domains, services, versions, and statuses.

Host Name	Domain	Service	Version	Status
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	triviaQuestion	5.0	RUNNING
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	internationalNameGenerator	1.0	RUNNING
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	exchangeRate	7.0	RUNNING





Apache Camel

Rule-based routing and mediation engine

- Selectively moves message around based on route's configuration
- Define routing rules
- Determine how to process and send messages to other destinations
- Can interact with APIs regardless of protocol/data type
 - Components provide specific implementation of API
- Based on Enterprise Integration Patterns (EIPs)



Camel: Components

Components

- Main extension point => Acts as a factory of endpoint instances
- Associated with name used in URI
 - Examples:
 - `https4://uri?authUsername={{http.user}}&authPassword={{http.password}}`
 - `https4://api.fixer.io/latest?base=USD&symbols=${exchangeProperty.currencyType}`
 - `jdbc:mysql://${mysql.host}:${mysql.port}/${mysql.name}`

Camel Integration Templates Designed Internally:

- HTTP
- MySQL
- MS SQL
- LDAP
- SMTP
- SOAP
- SMPP

Camel Integration Templates Designed Internally:

Currently available Camel Components in V3:

HTTP4: <http://camel.apache.org/http4.html>

Email: <http://camel.apache.org/mail.html>

Ldap: <http://camel.apache.org/ldap.html>

Jdbc: <http://camel.apache.org/jdbc.html>

Printer: <http://camel.apache.org/printer.html>

Spring-Ldap: <http://camel.apache.org/spring-ldap.html>

FTP: <http://camel.apache.org/ftp.html>

SMPP: <http://camel.apache.org/smpp.html>

ServiceNow: <http://camel.apache.org/servicenow.html>

Integration Framework

1

Integrations

Integrations Home

Integrations Status

2

Integration Framework

New Integration Flow →

Integrations Home Integrations Status Flows Clusters Hosts Assets Property Sets Service Deploy Requests

Services Status

3 INTEGRATIONS RUNNING

0 INTEGRATIONS PENDING

1 ACTIVE HOSTS

Integration Flow
Enabling communication between two heterogeneous applications or systems.

Cluster
Easy and faster way to deploy and un-deploy a group of Hosts at the same time.

Hosts
Management connection to a network nodes.

Property Sets
Manage properties that can be shared across flows.

Top 5 Integration Status

Host Name	Domain	Service	Version	Status
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	triviaQuestion	5.0	RUNNING
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	internationalNameGenerator	1.0	RUNNING
app01.playgroundus-ipsoft-v3.amelia.ipcenter.com	Brian	exchangeRate	7.0	RUNNING

3



Integration Flows

Integration Flows Integrations Home Integrations Status **Flows** Clusters Hosts Assets Property Sets Service Deploy Requests

3

Search
Name, Code

Name
exchangeRate
exchangeRate2
internationalNameGenerator
triviaQuestion

4

New Integration Flow

New Integration Service Revision Number: 1.0

Name *
Code *
Domain *
Assets
Description *

Properties ?
Edit Preview
Property Sets

INTEGRATION FRAMEWORK

- STENCIL
- search
- INTEGRATION CONTAINERS
- CONNECTORS
- URL FROM URL TO
- DEVELOPMENT
- {X: } HEADER
- BODY LOG
- DELAY

Integration Context
Route

5

URL FROM → URL TO

Source View or Designer View

The screenshot shows the IPsoft Integration Flows interface. On the left, there's a list of existing integration flows: exchangeRate, exchangeRate2, internationalNameGenerator, and triviaQuestion. A large central window is titled 'New Integration Flow' and contains a sub-section for creating a 'New Integration Service'. This section includes fields for 'Name', 'Code', and 'Domain', along with dropdowns for 'Assets' and 'Description'. Below this is a 'Properties' section with tabs for 'Edit' and 'Preview', and a 'Property Sets' button. To the right of the service creation is the 'INTEGRATION FRAMEWORK' panel, which contains various stencil components like 'URL FROM', 'URL TO', 'BODY', 'LOG', and 'DELAY'. A 'Route' section shows a connection between 'URL FROM' and 'URL TO' nodes. At the bottom right, there are 'Source View' and 'Designer View' buttons.



Name Panel

Name: Readable for flow

- Changeable after creation
- Autocomplete does not present correctly in BPN

Code: alternate flow key

- Immutable after creation
- Only one flow with specific code per Amelia domain
- Use code in run the integration flow

Domain: every flow belongs to only one

Assets: Binary elements included with flow

- In jar file assembled for deployment

Description

The screenshot shows a configuration dialog for a new integration service. At the top right, there is a blue button labeled 'Revision Number: 1.0'. Below it are fields for 'Name' (with a red asterisk), 'Code' (with a red asterisk), and 'Domain' (with a red asterisk). A 'Assets' section contains a dropdown menu and a '+' button. A 'Description' field with a red asterisk is also present. On the right side, there is a 'Properties' section with tabs for 'Edit' (selected) and 'Preview'. Under 'Property Sets', there is a list item 'Internal Properties Since New Flow 1.0'. Below that is a section titled 'Internal Properties for Flow Property Set' with a 'Import Properties into Grid' button and an 'Add Row' button at the bottom.

Properties

Properties: Name/value pairs referenced in XML as

Spring/Camel properties

- Optionally encrypted
- Decrypts only at flow runtime, not during deployment process

Internal Properties:

- Only for use with current flow

Shared Property Sets:

- Created in Property Sets for use in multiple flows

The screenshot shows the 'Properties' interface with the 'Edit' tab selected. It displays a list of 'Property Sets' including 'Internal Properties Since exchangeRate 3.0' and an option to 'Add Property Set to Integration Flow'. Below this is a section titled 'Internal Properties for Flow Property Set' with a 'Import Properties into Grid' button. A table grid is present with columns for 'Name', 'Value', and 'Encrypted'. A tooltip 'Add Row' is visible above the grid.

The screenshot shows the 'Edit Integration Property Set' interface. The top navigation bar includes 'Property Sets' which is underlined. The main area shows a property set named 'testSet' with fields for 'Name*' (set to 'testSet') and 'Domain*' (set to 'Two'). Below this is a 'Import Properties into Grid' section with a message 'Added row 0. Save changes' and a table grid. The table has columns for 'Name', 'Value', and 'Encrypted'. A single row is shown with the 'Value' column empty and the 'Encrypted' column set to 'false'. There is also an 'Add Row' button and a delete icon.

- If using multiple property sets, the last set containing a specific property name will provide the value



Integration Containers and Connectors

Integration Containers (provided in Designer View for new flow)

- Integration context
- Route

Connectors:

- URL FROM
 - Always set URI to direct:start
- URL TO
 - Set message channel
 - Designer view=> https4://api.fixer.io/latest?base=USD&symbols=JPY
 - XML=> <camel:toD uri="https4://api.fixer.io/latest?base=USD;symbols=JPY">
- URL TO (Default)
 - bean:varpop?method=moveToOutboundVariables('thing,bod')
 - Obtains any return from call



Expression Languages

- Groovy
- JavaScript
- JSONPath
- Camel:simple
 - Covers many common Amelia skills when a little expression based script needed in Camel routes
For more info, refer to
<http://camel.apache.org/simple.html>
- Camel:constant
 - Specify constant strings as type of expression
 - Message from test:a Endpoint has headerTest header set to constant value of constantValue

PROPERTIES

Expression:

Language:

camel:constant

Property Name:

Language:

camel:constant

camel:constant

camel:jsonpath

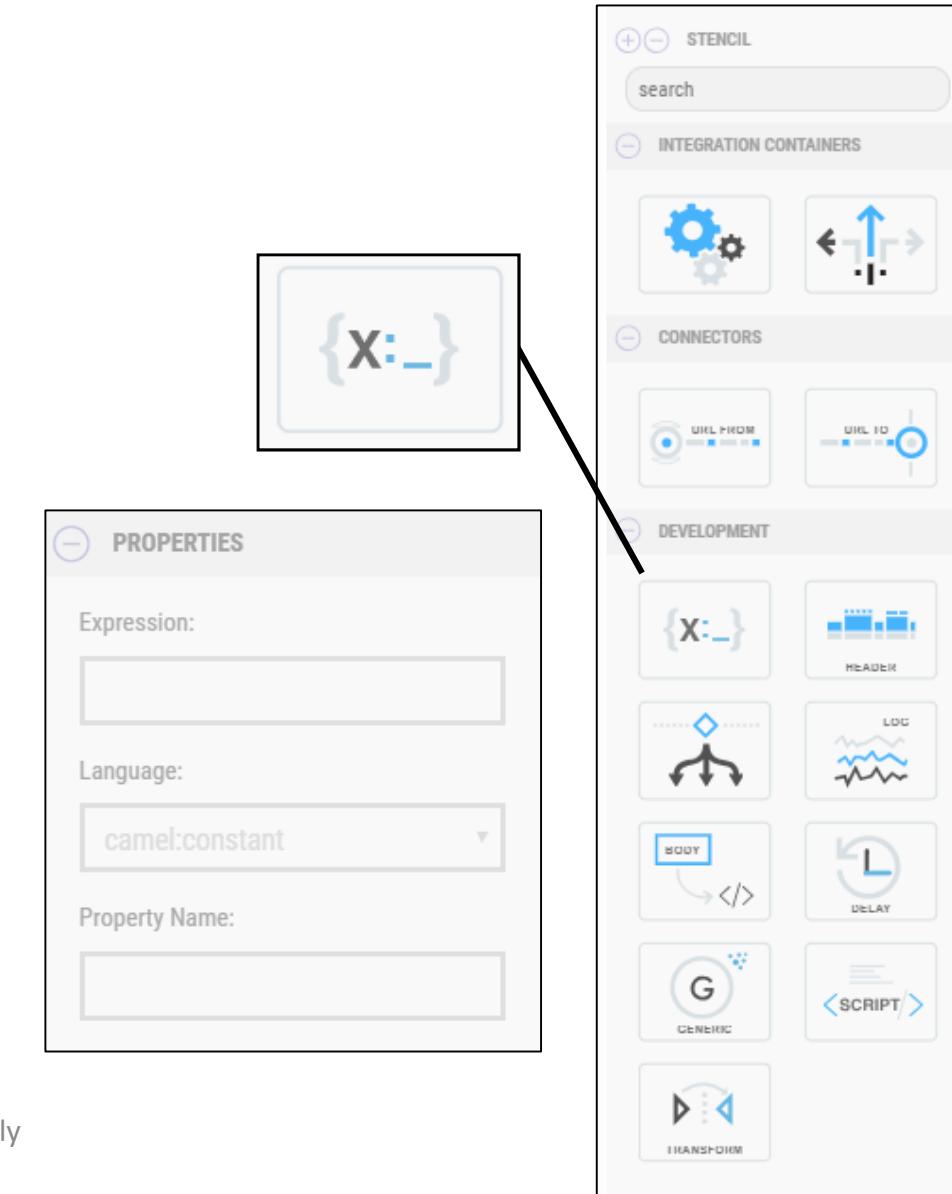
camel:simple

camel:groovy

camel:javaScript

Development – Set Properties

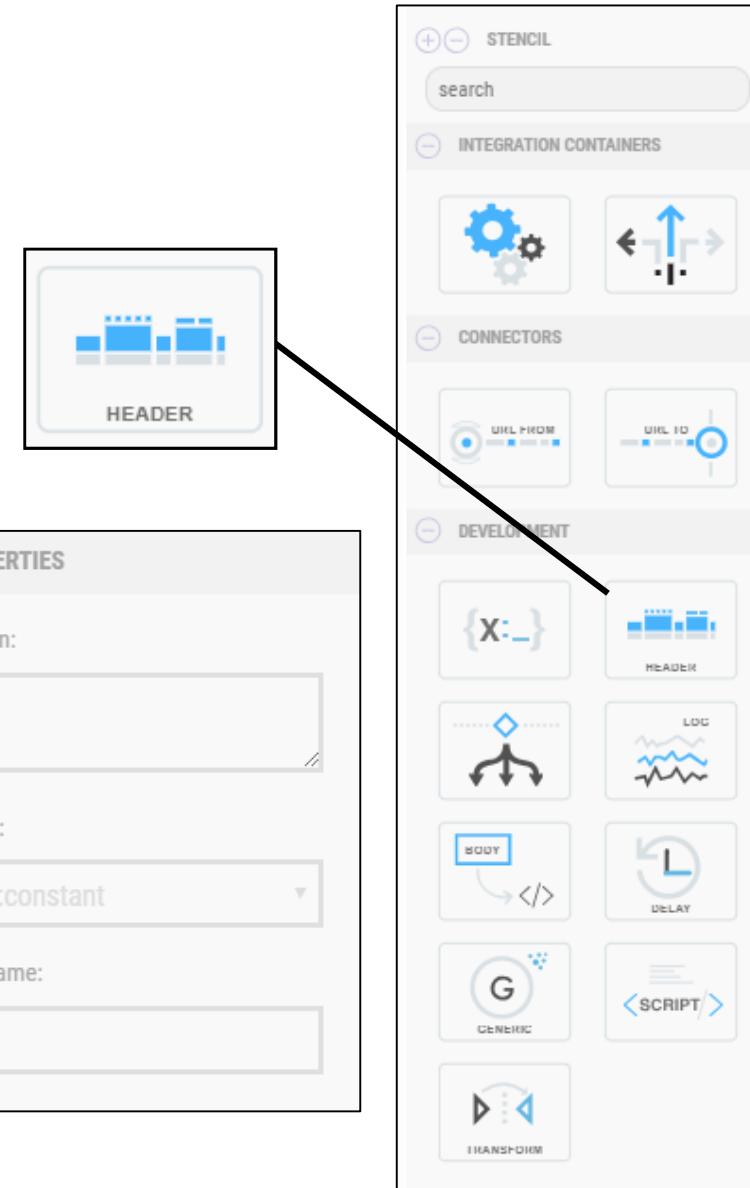
- Set properties for use in integration flow



Development – Set Headers

Headers => values associated with message

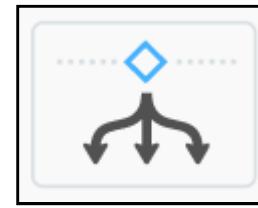
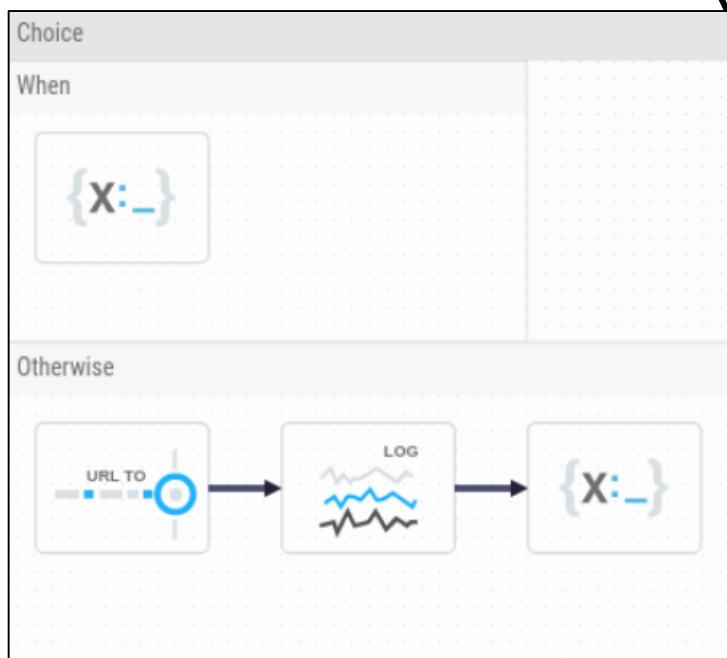
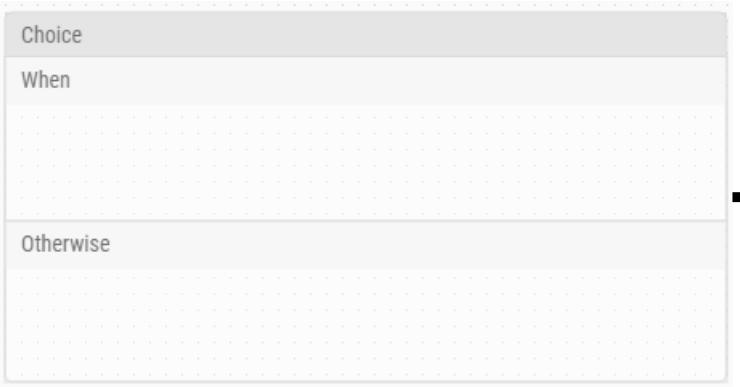
- Stored as name-value pairs
- Name = case insensitive String
- Value = java.lang.Object
- Stored as map within message



Development – Choice

Choice (acts like a message router)

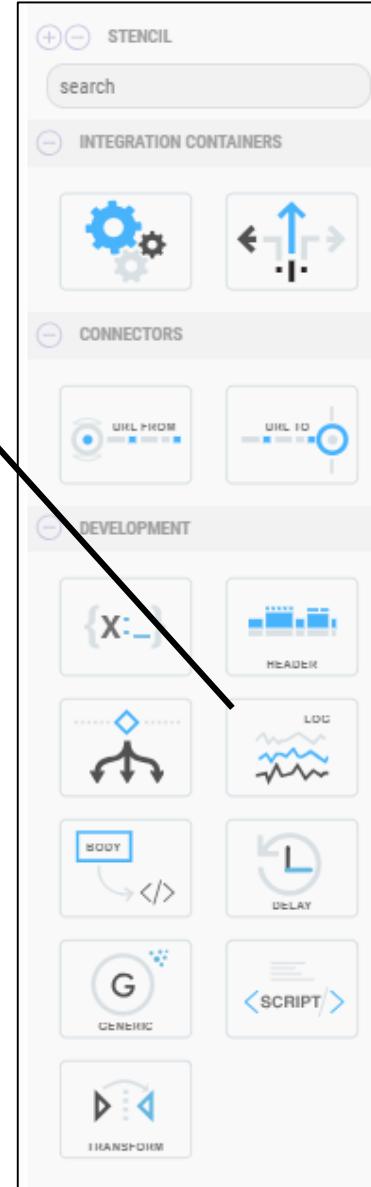
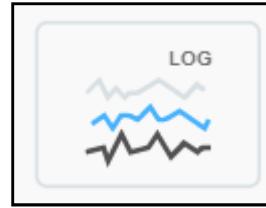
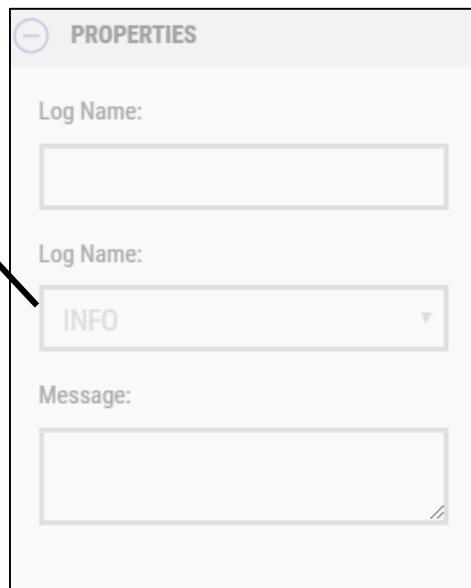
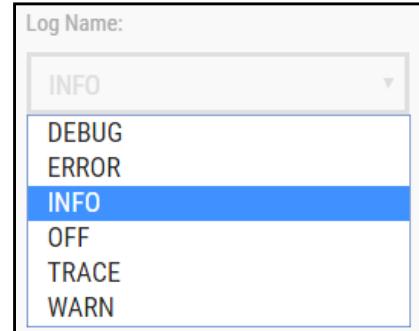
- Consumes from an input destination
- Evaluates some predicate then chooses right output destination
- Routes a request from an input queue:a endpoint to either queue:b, queue:c or queue:d depending on the evaluation of various Predicate expressions
- If no otherwise, any unmatched exchanges will be dropped by default



Development – Log

Message appears in integration server log - prefixed with logging service

```
<camel:log message="running  
$simple{property.sqlStatement}" loggingLevel="INFO">  
</camel:log>
```

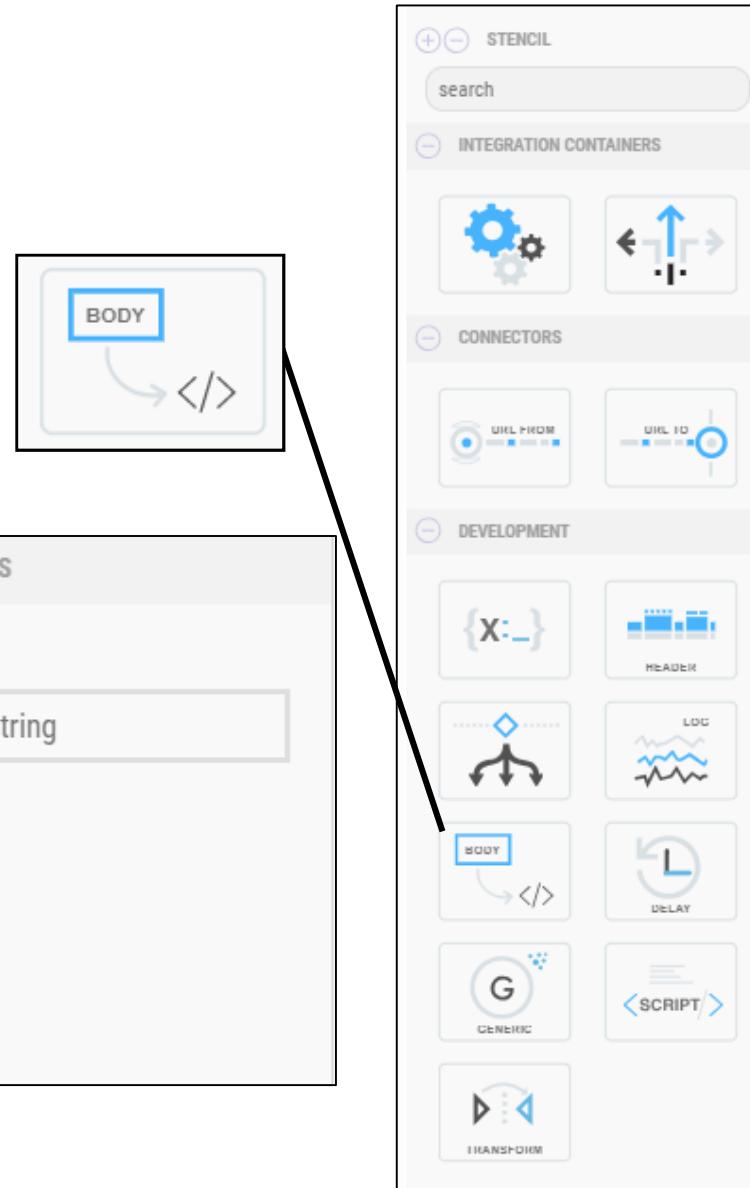


Development – Body (Converter)

Converts message body to given class type

- Uses a hierarchy of Type Converters

<http://camel.apache.org/convertbodyto.html>
<http://camel.apache.org/type-converter.html>



Development – Delay

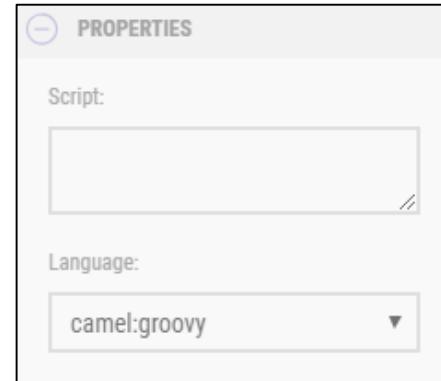
Delay delivery of message to destination

- Delay fixed amount of time from when delayer receives message
 - Use camel:constant => e.g., 10000 (in milliseconds)



Development – Generic, Script, and Transform

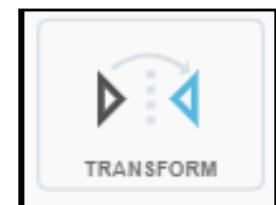
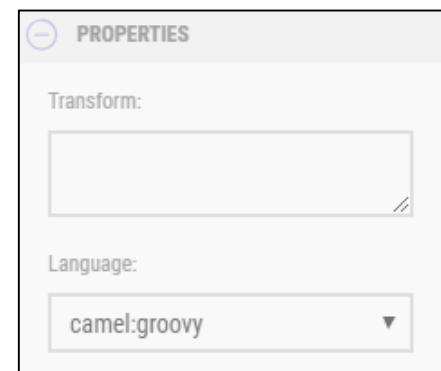
- Include Groovy script in flow



- Include/edit raw XML from this node



- Inline transformation of message content from one form to another using Groovy





Passing Return to Amelia

Main route typically ends with camel:to bean element

- Specify values to pass back to Amelia
 - Standard bean =>
 - bean:varpop?method=moveToOutboundVariables('thing,bod')
 - Takes comma delimited String argument
 - 'thing,bod' placeholders -> replace with return variables
 - Variables in either headers or exchange properties to be passed to Amelia
 - _body_
 - Special variable that takes message body and passes into Amelia context as the variable “body”
 - To be parsed in a script task
- Return elements should be serializable – else results not guaranteed



Camel: Spring

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:camel="http://camel.apache.org/schema/spring" xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-4.2.xsd
    http://camel.apache.org/schema/spring
    http://camel.apache.org/schema/spring/camel-spring.xsd
">
    <camel:camelContext id="mainContext">
        <camel:route id="route1">
            <camel:from id="765371cf-56b6-4727-b003-844cbd41a8d9" uri="direct:start">
            </camel:from>
            <camel:setProperty id="df57e35b-c7d2-4462-a75b-4ec75df6a30f" propertyName="currencyType">
                <camel:simple id="b6614538-5de2-41a0-952b-88660a3cedcc"><![CDATA[exchangeProperty.currencyType]]></camel:simple>
            </camel:setProperty>
            <camel:toD id="4eb5bf22-8ce1-4e64-a586-ae151cff336d" uri="https://api.fixer.io/latest?base=USD&symbols=${exchangeProperty.currencyType}">
            </camel:toD>
            <camel:convertBodyTo id="9731878e-fdcf-42d9-8d40-5c85e98c0a72" type="java.lang.String">
            </camel:convertBodyTo>
            <camel:to id="19d24f95-3e86-47a6-a772-4ba8bccfe471" uri="bean:varpop?method=moveToOutboundVariables(&'_body_&apos;)">
            </camel:to>
        </camel:route>
    </camel:camelContext>
</beans>
```

Hosts

- Typically Integration Service instance on same host as Amelia V3
- amelia-integration-service deployable on any server
- Multiple hosts can be added in Admin UI to deploy flows
- Administrative listen port = 4635
- Global permissions needed to deploy host

New Integration Host

Integrations Home Integrations Status Flows Clusters **Hosts** Assets Property Sets Service Deploy Requests

New Integration Service

Name*

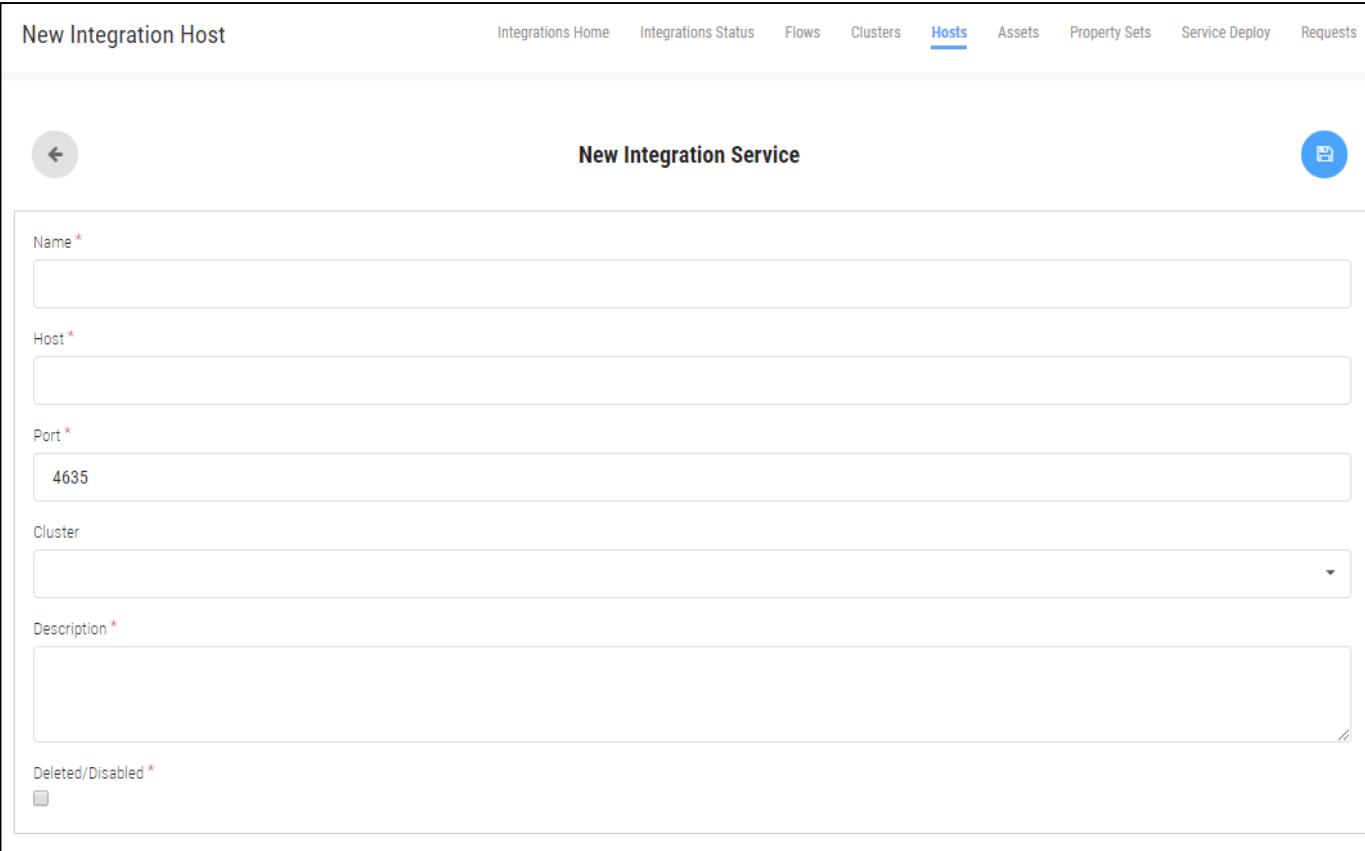
Host*

Port*

Cluster

Description*

Deleted/Disabled*



Name: readable name for host

Host: fully qualified domain name (FQDN) of host, reachable from Amelia instance

Port: where gRPC administrative service running

Cluster: for bulk operations – flow deployed/undeployed from multiple hosts at once

Description: info on Integration Host

Deleted/Disabled: deleted/disabled host still in system - not eligible for deployments

Clusters

- Group of hosts
 - Not a true cluster in infrastructure sense
- Allows deployment of flows to a group of hosts at once

New Cluster

Name *

Description *

[Close](#) [Submit](#)

Assets

- Binary elements included in jar file assembled for deployment
 - Anything not inside XML but can be referenced in XML
 - How you reference asset depends entirely on content of asset
- Assets can be included in any flow

New Assets

Name *

Domain *

ATTACH FILES

Drop a file here, or click to select the file to upload.

Close Submit

Service Deploy: Deploy & Undeploy

- If integration referenced in deployed BPN, cannot undeploy -> error
 - Can deploy a new version

Integration Service Deploy

Integrations Home Integrations Status Flows Clusters Hosts Assets Property Sets **Service Deploy** Requests

Flow

Hosts To Deploy *

Clusters To Deploy *

Hosts To Deploy

Clusters To Deploy

Run

Integration Service Deploy

Integrations Home Integrations Status Flows Clusters Hosts Assets Property Sets **Service Deploy** Requests

Flow

exchangeRate - Brian

Hosts To Deploy *

Clusters To Deploy *

Hosts to Undeploy *

Clusters To Undeploy *

app01.playgroundus-ipsoft-v3.amelia.ipcenter.com

Clusters To Undeploy

Undeploying could affect the following BPNs: exchangerate (EU Training domain).

Undeploying will break the following BPNs, which must have their references to this flow removed: integrationPurchaseCurrencyRate-NW (Brian domain), integrationExchangeRate (Brian domain).

Confirm Action

Run



Requests

Integration Requests

Integrations Home Integrations Status Flows Clusters Hosts Assets Property Sets Service Deploy Requests

Search

Host Name, Flow Code, or Flow Name

Status

Domain



Flow Domain	Flow Name	Flow Revision	Request Type(s)	Status	Created	Last Modified
Brian	exchangeRate2	1.2	UNDEPLOY	PENDING	7/25/2018, 4:15:31 PM	7/25/2018, 4:15:31 PM
Brian	exchangeRate	7.0	DEPLOY	SUCCESS	6/18/2018, 9:35:41 PM	6/18/2018, 9:35:50 PM
Brian	exchangeRate	6.0	DEPLOY	SUCCESS	6/14/2018, 9:14:55 AM	6/14/2018, 9:15:03 AM
Brian	exchangeRate	5.0	DEPLOY	SUCCESS	6/14/2018, 9:07:51 AM	6/14/2018, 9:08:00 AM
Brian	exchangeRate	5.0	DEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM
Brian	exchangeRate	4.0	DEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM
Brian	triviaQuestion	5.0	DEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM
Brian	exchangeRate2	1.0	UNDEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM
Brian	exchangeRate	3.0	DEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM
Brian	exchangeRate	3.0	UNDEPLOY	SUCCESS	4/26/2018, 12:29:32 PM	4/26/2018, 12:29:40 PM

20 Integration Requests, Page 1 of 2

exchangeRate2

Flow Domain	Brian
Flow Name	exchangeRate2
Flow Version	1.2
Result Status	PENDING
Created	7/25/2018, 4:15:31 PM
Last Updated	7/25/2018, 4:15:31 PM
Request Type(s)	UNDEPLOY

app01.playgroundus-ipsoft-v3.amelia.ipcenter.com

Request Type	UNDEPLOY
Request Status	PENDING

Delete Integration Flows

Integration Flows

Integrations Home Integrations Status Flows Clusters Hosts Assets Property Sets Service Deploy Requests

+ Search Status Domain

Name	Code	Version	Domain
exchangeRate	exchangeRate	7.0	Brian
exchangeRate2	exchangeRate2	1.2	Brian
internationalNameGenerator	internationalNameGenerator	1.0	Brian
triviaQuestion	triviaQuestion		

4 Integration Flows, Page 1 of 1

Integration Flows Failed to delete integration flow: Integration flow is deployed.

+ Search Status Domain

Name	Code	Version	Domain
exchangeRate	exchangeRate	7.0	Brian
internationalNameGenerator	internationalNameGenerator	1.0	Brian
triviaQuestion	triviaQuestion	5.0	Brian

3 Integration Flows, Page 1 of 1

Group Activity: Integration Step-by-Step

1. Create a new integration flow:

Name: trivia_YourInitials

Code: trivia_YourInitials

Domain: Your Domain

Description: Gets Trivia Question

2. Change the URL TO properties to

`https://opentdb.com/api.php?amount=1&category=18&type=multiple`

3. Add a Body

4. Add a URL TO with properties set to

`bean:varpop?method=moveToOutboundVariables('_body_')`

5. Save the integration

6. Go to Service Deploy, click + and deploy the integration on the localhost

7. Check Service Deploy Results for Success

Putting It Together: BPN and Integration



Run the Integration Flow Task

run the integration flow camelWorkflowCode

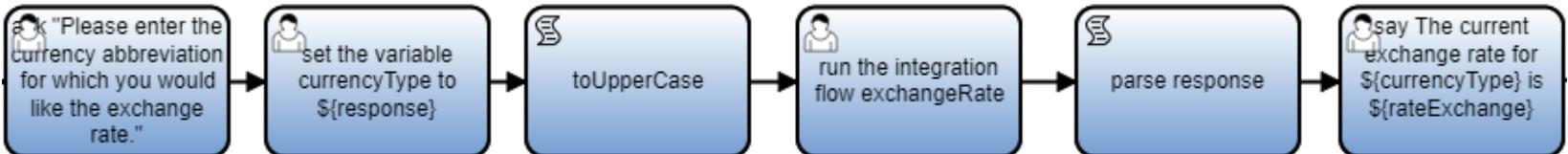
- Integrations created in Amelia with Apache Camel standard
 - Communicate with external systems via APIs and Web services
- Integration workflow
 - Create integration flow in Amelia Admin-> Integrations
 - Call integration flow code in BPN
 - Parse output with Script task
 - Use output in BPN



Executing Integrations

Integration Flows: executed by code and domain

- BPN executes Integration Flow => client attempts to find integration flow by specific code in conversation domain
- Falls back up the domain hierarchy until match found



Properties:

- Variable propagation – bidirectional, push only, pull only, none
- Integration flow timeout – execution timeout (default 30 seconds)
- Integration flow log level – error, debug, info, warn
- Integration completion style – SYNC, ASYNC (Beta in 3.5)
 - ASYNC allows the conversation to continue prior to the completion of the integration flow
 - Requires the following code in integration flow to continue the conversation:
`<camel:toD uri="bean:asyncMessage?method=sendFinal(${property.conversationId}, '') />`

Properties Custom Properties

Main Properties

Name
run the integration flow exchangeRate

Variable Propagation
Bidirectional

Type of variable propagation to and from the child process.

Integration flow timeout
30 seconds

Integration flow execution timeout.

Integration flow log level
ERROR

The level of flow log statements to return to Amelia.

Integration completion style
SYNC

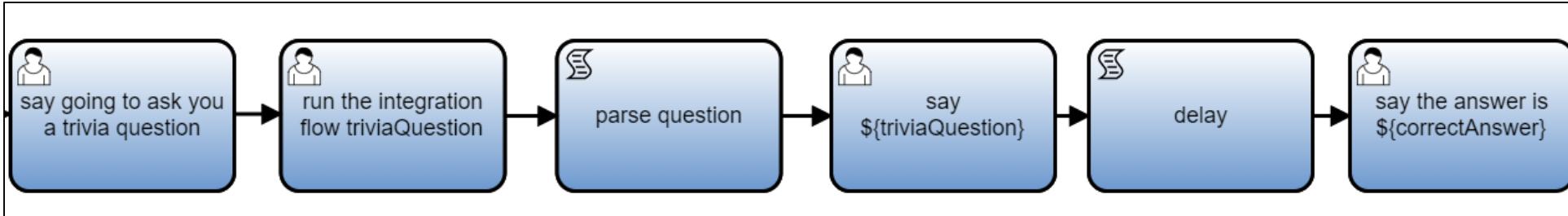
How the integration flow indicates completion.

Outgoing variables +

Incoming variables +

Group Activity: Integration Step-by-Step

8. Go to Process Knowledge and import BPN



Parse question script:

```
import groovy.json.JsonSlurper;

def triviaQuestion = "No question"
def correctAnswer = "No answer"

def bodyAsString = execution.getVariable("body")
if (bodyAsString != null) {
    def parser = new JsonSlurper()
    def output = parser.parseText(bodyAsString)
    triviaQuestion = output.results.question[0]
    correctAnswer = output.results.correct_answer[0]

}
execution.setVariable("triviaQuestion", triviaQuestion)
execution.setVariable("correctAnswer", correctAnswer)
```

Delay Script:

```
sleep(8000)
```

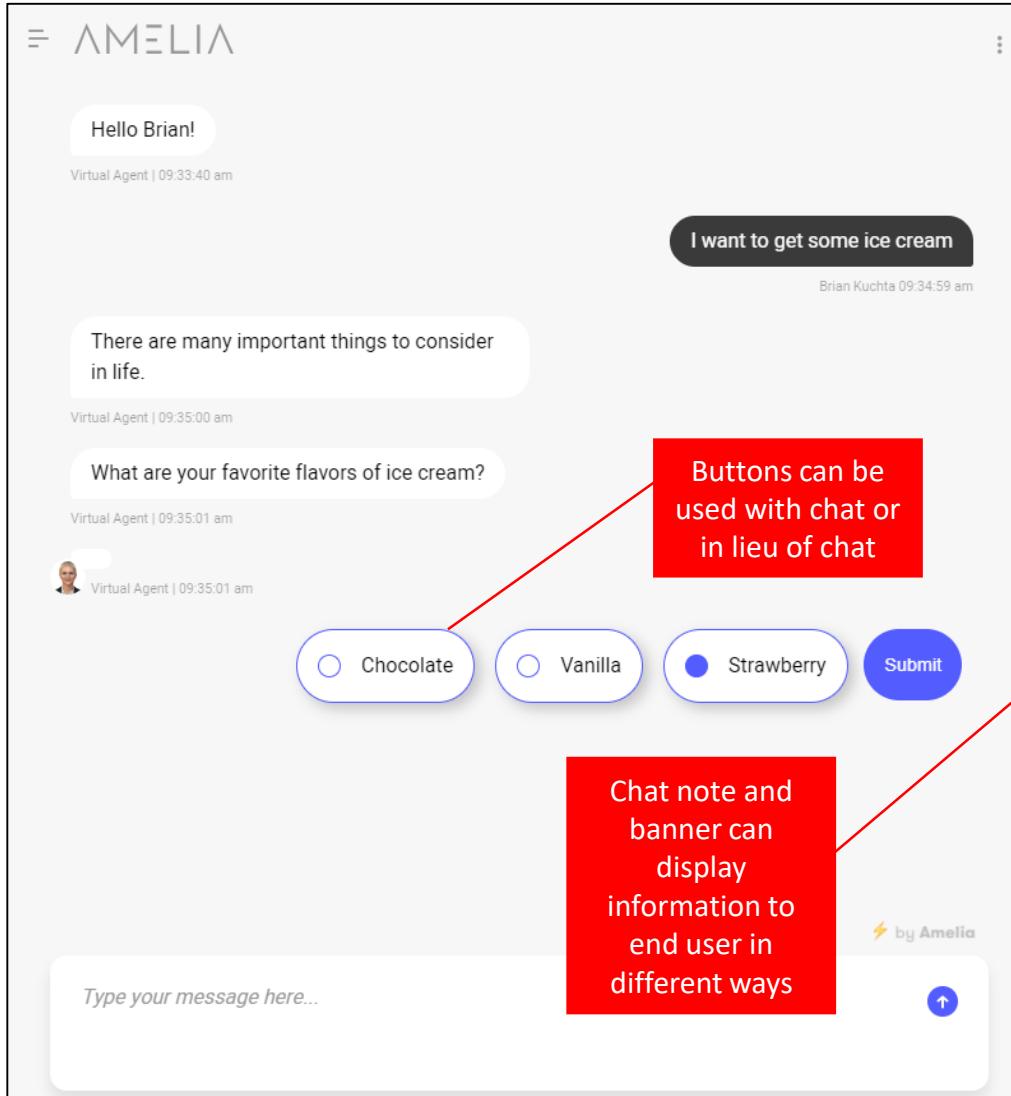
9. Deploy and run the workflow

Introduction to Amelia Custom UI



Custom UI

- Installed with all new Amelia instances by default
- Buttons, chat notes, and other features can make the end user journey more effective and interactive
- trainingus instance custom UI URL:
<https://ipsoft-amelia-trainingus-v3.ipsoft.com/Amelia/ui/AmeliaCust>



Congratulations!
You have chosen an excellent hotel.
You are about to have an amazing experience.
It's going to be spectacular.

Book a Room Now

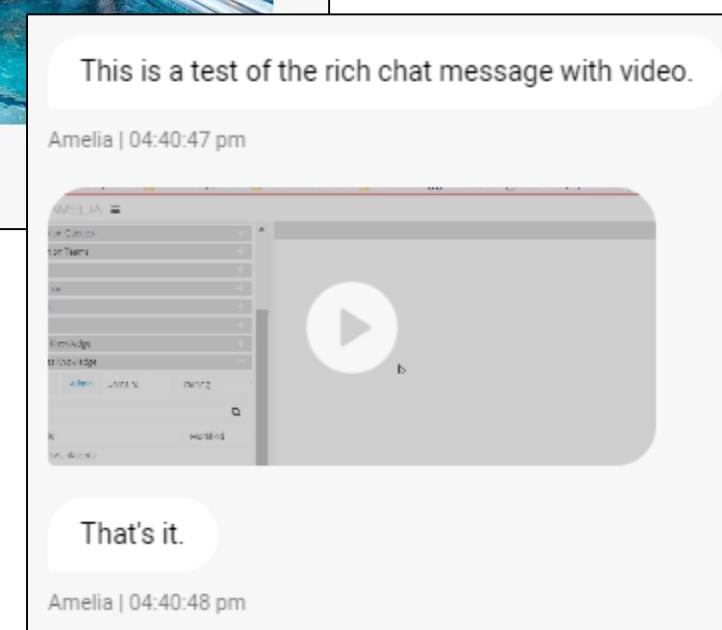
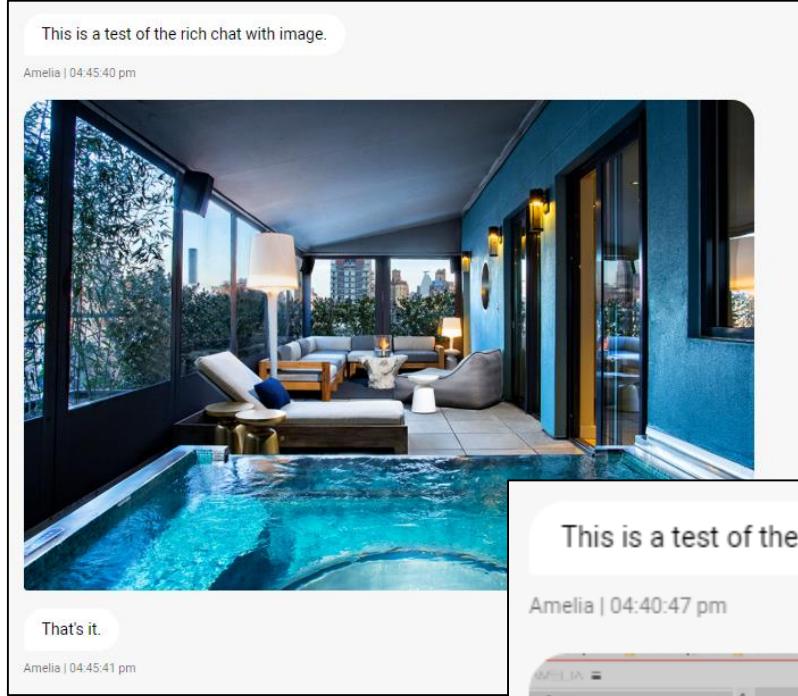
AMELIA ASKS QUESTIONS

Table

Name	Food	Color	Animal
John	Pizza	Red	Dog
Sue	Shrimp	Blue	Cat
Sally	Taco	Green	Fish
Tom	Steak	Orange	Bird
Wilma	French Fries	Purple	Hamster

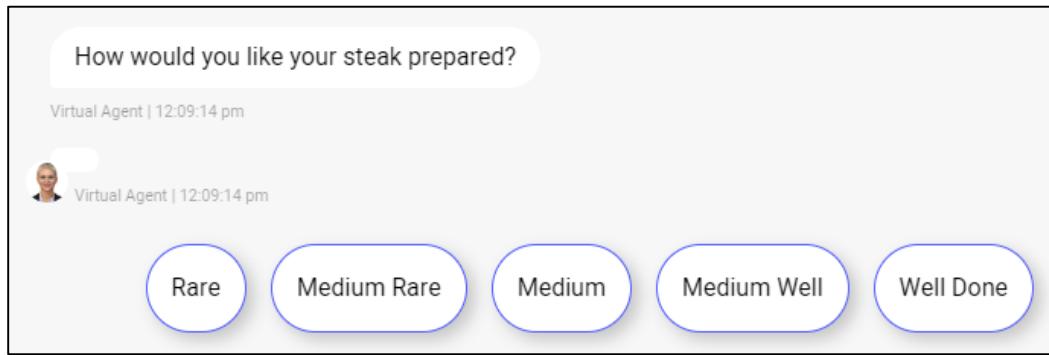
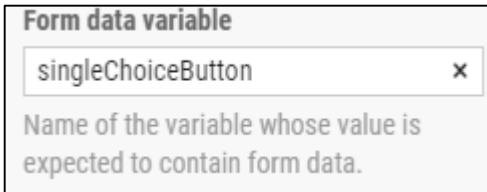
Rich Chat

- Displays multimedia messages inline
 - Documents
 - Videos
 - Links
 - Images
- Sizes:
 - Small (URLs without a preview, documents)
 - Medium (images)
 - Large Rich (URLs with thumbnails, documents with thumbnails, videos)



Buttons

- Payload for buttons created in script task
- Payload sent to custom UI by setting Form data variable property for ask task to payload variable name



```
def fieldBuilder = formInputDataBuilder.create()  
    .name("singleChoiceButton")  
    .nameForDisplay("Single Choice Button")  
    .formType("sampleForm")  
    .staticSelectionUtterance("")  
    .allowedUserInputs("form_only")  
    .addField()  
        .name("someChoice")  
        .fieldType("singleSelection")  
        .postfixedSelectionUtterance("", "")  
        .addOption()  
            .name("Rare")  
            .value("Rare")  
            .backToField()  
        .addOption()  
            .name("Medium Rare")  
            .value("Medium Rare")  
            .backToField()  
        .addOption()  
            .name("Medium")  
            .value("Medium")  
            .backToField()  
    .backToForm()  
    .build()  
execution.setVariable('singleChoiceButton', fieldBuilder.toString())
```

Single Selection Multiple Choice Buttons

Form data variable
singleChoiceButton x

Name of the variable whose value is expected to contain form data.

```
graph LR; A[say The steak is an excellent choice] --> B[single selection button]; B --> C[ask "How would you like your steak prepared?"]
```

How would you like your steak prepared?

Virtual Agent | 12:09:14 pm

Virtual Agent | 12:09:14 pm

Rare Medium Rare Medium Medium Well Well Done

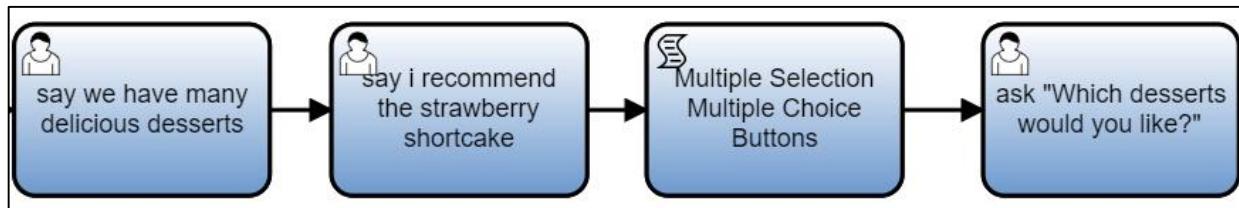
Multiple Selection Multiple Choice

Form data variable

multiChoiceButton

x

Name of the variable whose value is
expected to contain form data.



We have many delicious desserts.
Virtual Agent | 12:13:56 pm

I recommend the strawberry shortcake.
Virtual Agent | 12:13:56 pm

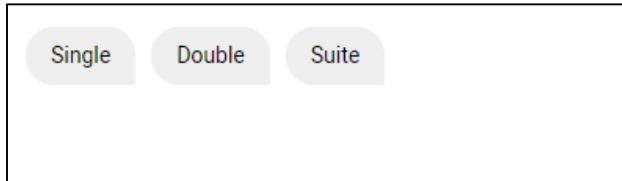
Which desserts would you like?
Virtual Agent | 12:13:57 pm

Virtual Agent | 12:13:57 pm

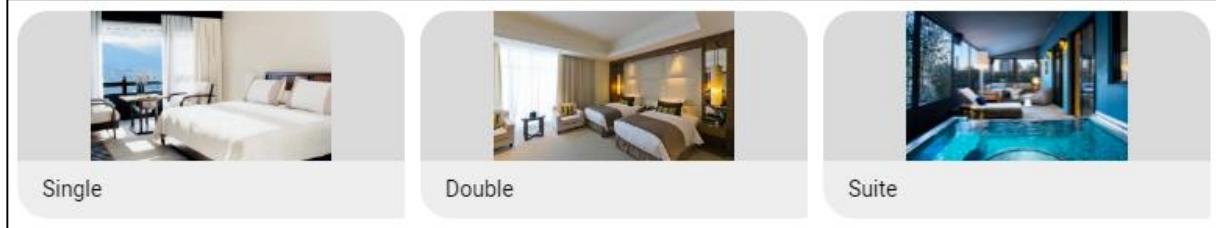
Chocolate Pie Vanilla Bean Cake Strawberry Shortcake

Submit

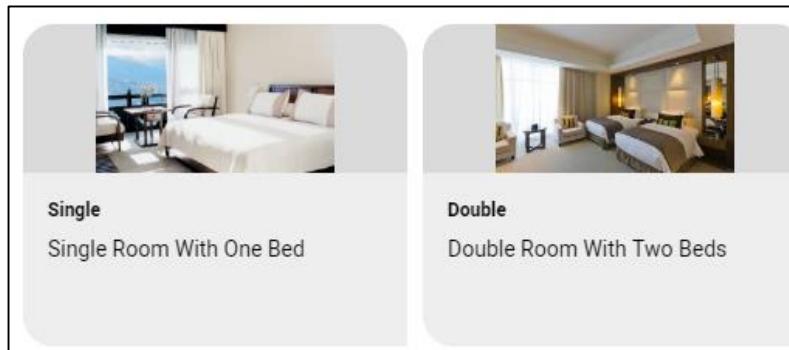
Button Sizes Examples



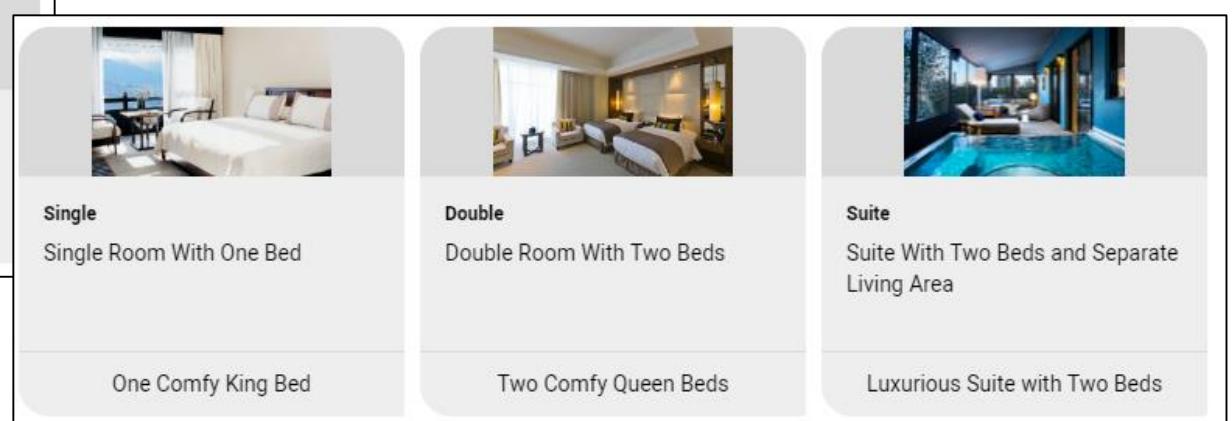
Small



Medium



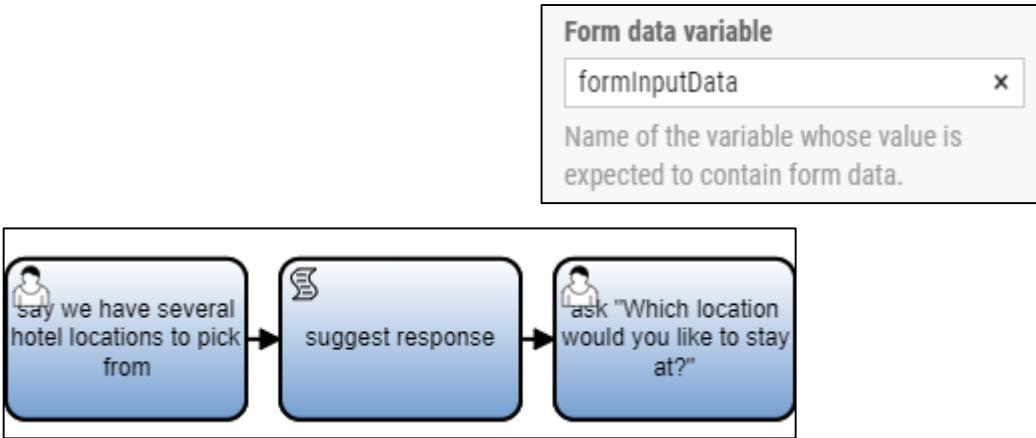
Large



X-Large

Suggested Response

- Payload for suggested response created in script task
- Payload sent to custom UI by setting Form data variable property for ask task to payload variable name



Form data variable

formInputData

Name of the variable whose value is expected to contain form data.

```
def formInputData = formInputDataBuilder.create()
    .name("locationSelectionForm")
    .allowedUserInputs("both")
        .instructions("strict")
    .addField()
        .name("locationSelectionField")
        .fieldType("AutoComplete")
        .postfixedSelectionUtterance("I select the location", "I
select the location")
    .addOption()
        .name("New York Midtown")
        .nameForDisplay("NY Midtown")
        .value("nyMid")
        .backToField()
    .addOption()
        .name("New York Downtown")
        .nameForDisplay("NY Downtown")
        .value("nyDowntown")
        .backToField()
    .addOption()
        .name("Brooklyn Downtown")
        .nameForDisplay("Brooklyn Downtown")
        .value("bDowntown")
        .backToField()
    .backToForm()
.build();

execution.setVariable('formInputData', formInputData.toString())
```

Chat Notes

action:

- addChatNotesIntegration
- deleteChatNoteIntegration

payload -> contains components/subcomponents

- **component** -> ChatNote or Banner
- **header** -> for entire Chat Notes area
 - Pass header with each new integration message
 - Remove header -> pass empty string
 - If adding to chat note and no header provided, previous one used
- **title** -> unique, used for chat note changes -> adding, modifying, deleting
- **actionProcess** -> BPN triggered from within Chat Note via Edit button
- **subcomponents** -> array of subcomponents

```
import groovy.json.JsonOutput

def payload = [
    action: 'addChatNotesIntegration',
    componentType: 'ChatNote',

    payload   : [
        [
            component: 'ChatNote',
            header: 'Welcome to Amelia',
            title: 'Amelia Asks Questions',
            actionProcess: 'actionProcess',
            subcomponents: [
                [
                    subcomponent: 'Markup',
                    markupType : 'Text',
                    label     : 'Timing',
                    value     : '<b>Timing is Everything</b> as there are ' +
                                'times in our lives that we all need <i>more time</i>' +
                                ' to accomplish all that we seek to do. Time is a ' +
                                'precious gift, <u>one that should NOT be ' +
                                'squandered</u>. As such, one might question
                                whether ' + 'wearing a watch is a waste of time as one
                                may not ' + 'want to check one\'s watch to save time.'
                ],
                ...
            ]
        ]
    ]
]
```

Amelia Asks Questions

Timing

Timing is Everything as there are times in our lives that we all need *more time* to accomplish all that we seek to do. Time is a precious gift, one that should NOT be squandered. As such, one might question whether wearing a watch is a waste of time as one may not want to check one's watch to save time.

Components integrated into Chat Note:

- Types: Chat Note and Banner
- Multiple subcomponents of different types possible
- Payload passed as JSON string using **send the integration message** in BPN

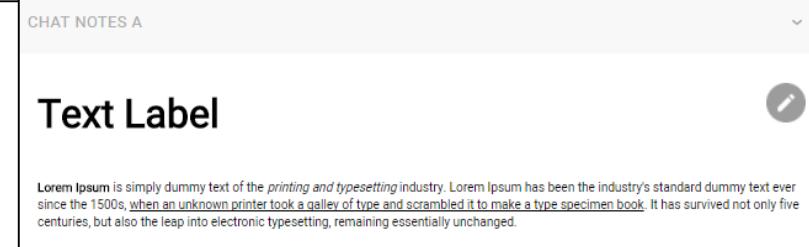
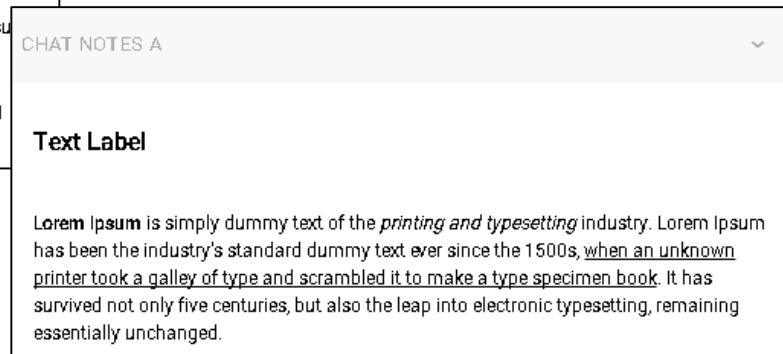
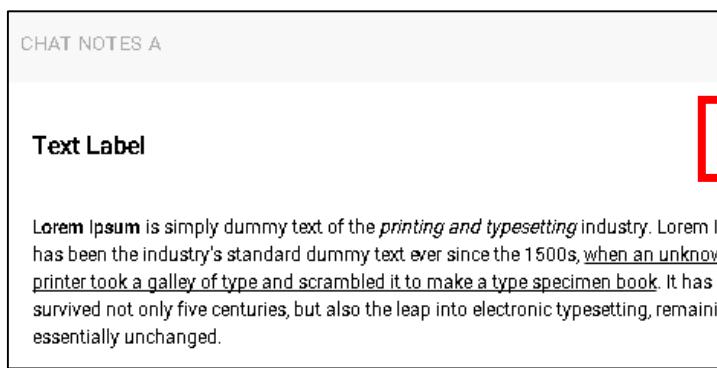
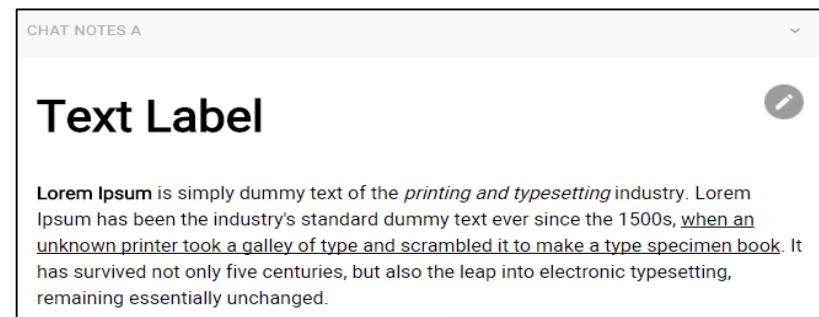


send the integration message jsonString

Chat Notes actionProcess and Styling

- Can only be triggered via edit button
- Only one edit button per chat note
 - If no actionProcess, edit button hidden
- **actionName** -> name of BPN to trigger
- **actionLabel** -> name on button
- **actionArguments** -> additional data to pass to target BPN
 - processScopeVariable: value
- **actionMessage** -> custom message output as end user's utterance
 - Default: "I have invoked an action"

- Allows custom styling for labels, text and description elements
 - Chat note level – applies to all labels and text in chat note
 - Preset font size and font weight options
 - labelSize -> subhead, title, headline and display
 - labelWeight -> regular and bold
 - bodySize -> small, caption, body



Chat Notes – ImageSmall and ImageLarge

```
subcomponents: [
```

```
  [
    subcomponent: 'Markup',
    markupType: 'ImageSmall',
    imageBase64: 'https://resourcelocation.com/picture',
    label: 'Hotel Picture',
    description: 'This is an image of a hotel.'
  ],
```

CHAT NOTES A

Hotel Picture

This is an image of a hotel.



```
subcomponents: [
```

```
  [
    subcomponent: 'Markup',
    markupType: 'ImageLarge',
    label: 'Hotel',
    imageBase64: 'https://resourcelocation.com/picture',
    label: 'Training Domain Hotel',
    description: 'A glorious picture of our establishment!'
  ],
```

CHAT NOTES A

Yummy Food

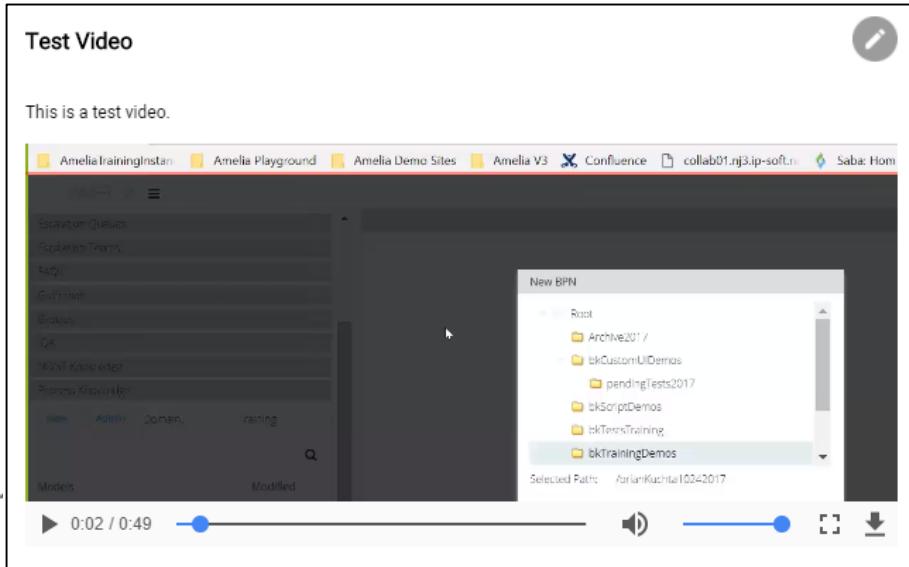
This is an image of delicious food.



Chat Notes – Video and Embedded Video

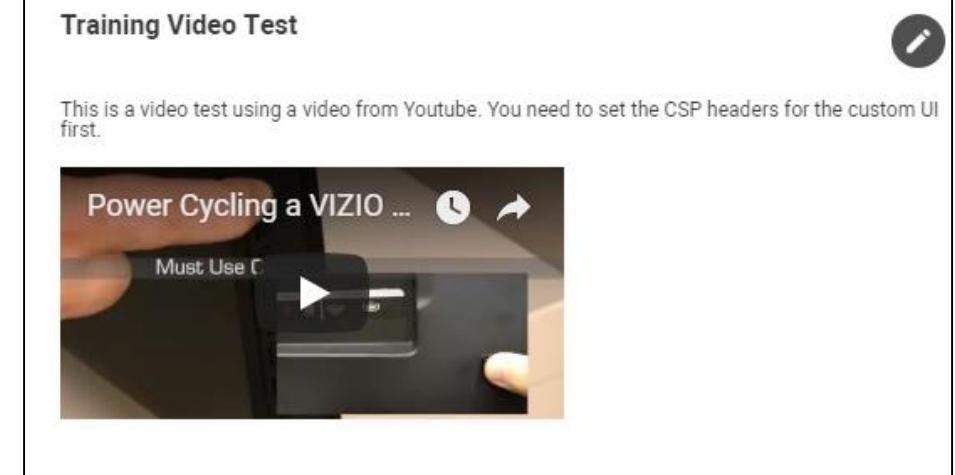
subcomponents: [

```
[  
    subcomponent: 'Markup',  
    markupType: 'Video',  
    url: 'https://app01.playgroundus-ipsoft-  
v3.amelia.ipcenter.com/Amelia/api/cm/download/LBKNATHNQAIA  
A-1/ce2df31c-e488-4a8f-a566-047e1043389e',  
    label: 'Training Video Test',  
    description: 'This is a video test using a video from content  
manager.'  
],
```



subcomponents: [

```
[  
    subcomponent: 'Markup',  
    markupType : 'VideoEmbed',  
    url: "https://www.youtube.com/embed/nLsGBp0bJS0",  
    label : 'Training Video Test',  
    description : 'This is a video test using a video from Youtube.  
You need to set the CSP headers for the custom UI first.'  
],
```



Chat Notes – Table

subcomponents: [

```
[  
  subcomponent: 'Markup',  
  markupType: 'Table',  
  label: 'Table',  
  headers: ["Name", "Food", "Color", "Animal"],  
  rows: [  
    ["John", "Pizza", "Red", "Dog"],  
    ["Sue", "Shrimp", "Blue", "Cat"],  
    ["Sally", "Taco", "Green", "Fish"],  
    ["Tom", "Steak", "Orange", "Bird"]  
  ],
```

Table

Name	Food	Color	Animal
John	Pizza	Red	Dog
Sue	Shrimp	Blue	Cat
Sally	Taco	Green	Fish
Tom	Steak	Orange	Bird
Wilma	French Fries	Purple	Hamster

Chat Notes – Text Block

- Useful when limited space in layout
 - Amelia embedded into iframe

Line options:

- line - simple line of text with no decoration
- lineActive - line of text with green check
- lineDisabled - greyed out line of text with X
- lineBreak - line break

Training Insurance Providers

Coverages

Bodily Injury & Property Damage
Uninsured / Underinsured
Comprehensive

Collision
Glass Damage
100,000 per person
\$300,000 per accident



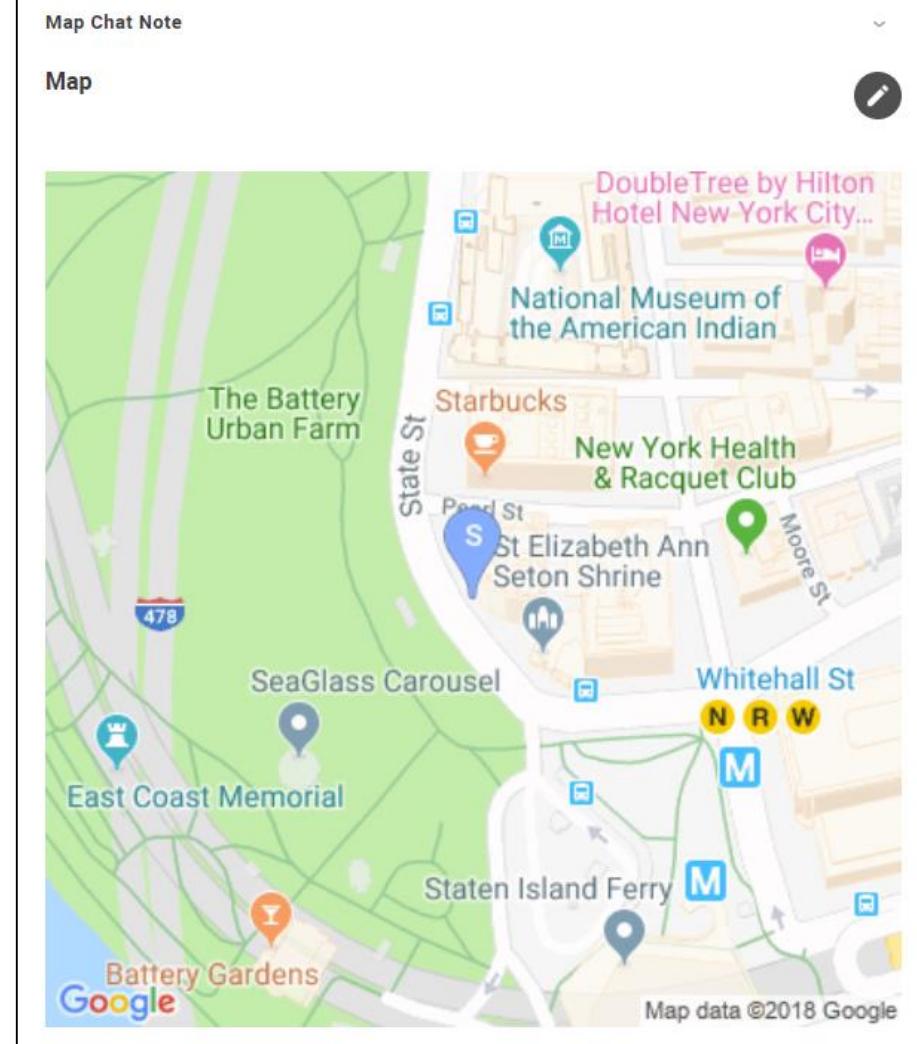
```
subcomponents: [
  [
    subcomponent: 'Markup',
    markupType: 'TextBlock',
    label: 'Coverages',
    lines: [
      lineActive: 'Bodily Injury & Property Damage',
      lineActive: 'Uninsured / Underinsured',
      lineActive: 'Comprehensive',
      [lineBreak: ''],
      lineDisabled: 'Collision',
      lineDisabled: 'Glass Damage',
      line: '100,000 per person',
      line: '$300,000 per accident'
    ]
  ]
]]
```

Chat Notes – Google Map Image

- Based on an end user response to a location question
- Obtain image via Google Maps
- Requires API Key
- Requires modification to CSP headers in custom UI bundle to access Google Maps

subcomponents: [

```
[  
  subcomponent: "Markup",  
  markupType: "ImageLarge",  
  label: "Map",  
  streetAddress: "...",  
  addressCoordinates: [...],  
  linkUrl: "https://www.google.com/maps/search/" +  
    URLEncoder.encode(address, 'UTF-8') + '/',  
  imageBase64: "..."  
,
```



Banner – Groovy

Standalone Chat Note Component

- Position fixed at top of Chat Note
- Subcomponents:
Text, Default, Button

Text/Default Subcomponent Styles:

- fontSize
 - Small, Caption, Body, Subhead, Title, Headline, Display
- fontWeight
 - Regular, Bold

Button Subcomponent

- Include label, value, actionProcess /link objects
- actionProcess functionality -> triggers BPN
- link -> redirects to URL

```
...
payload: [
  [
    component: 'Banner',
    header: 'Training Domain Hotels',
    title: 'Welcomes YOU',
    actionProcess: [
      actionName: 'buttonSubProcess',
    ],
    subcomponents: [
      [
        subcomponent: 'Markup',
        markupType: 'Default',
        value: 'Congratulations!',
        fontSize: 'title',
        fontWeight: 'bold'
      ],
      [
        subcomponent: 'Markup',
        markupType: 'Default',
        value: 'You have chosen an excellent hotel.',
        fontSize: 'title',
        fontWeight: 'normal'
      ],
      [
        subcomponent: 'Markup',
        markupType: 'Buttons',
        buttons: [
          //this is where a square bracket was missing
          [label: 'Book a Room Now',
          value: 'Reservation',
          actionProcess:[
            actionName: 'reserveARoom',
            actionMessage: 'Making a Hotel Reservation'
          ]
        ...
      ]
    ]
  ]
]
```

Congratulations!

You have chosen an excellent hotel.

You're about to have an amazing experience.

It's going to be spectacular.

Book a Room
Now

Congratulations!

You have chosen an excellent hotel.

You're about to have an amazing experience.

It's going to be spectacular.

Book a Room
Now

Banner – JSON String

Standalone Chat Note Component

- Position fixed at top of Chat Note
- Subcomponents:
Text, Default, Button

Text/Default Subcomponent Styles:

- fontSize
 - Small, Caption, Body, Subhead, Title, Headline, Display
- fontWeight
 - Regular, Bold

Button Subcomponent

- Include label, value, actionProcess /link objects
- actionProcess functionality -> triggers BPN
- link -> redirects to URL

```
def jsonString = ""  
{  
    "action": "addChatNotesIntegration",  
    "componentType": "ChatNote",  
    "payload": [{  
        "component": "Banner",  
        "header": "Training Domain Hotels",  
        "title": "Welcomes YOU",  
        "styles": {  
            "height": "100%",  
            "color": "#fff",  
            "backgroundColor": "#4d5aff"  
        },  
        "actionProcess": {  
            "actionName": "bannerSubFlowTester",  
            "actionMessage": "You have clicked edit"  
        },  
        "subcomponents": [{  
            "subcomponent": "Markup",  
            "markupType": "Text",  
            "value": "You will love our hotel!",  
            "fontSize": "title",  
            "fontWeight": "regular"  
        },  
        {  
            "subcomponent": "Markup",  
            "markupType": "Buttons",  
            "buttons": [{  
                "label": "Book a Room Now",  
                "value": "Reservation",  
                "actionProcess": {  
                    "actionName": "reserveARoom",  
                    "actionMessage": "Making a Hotel Reservation"  
                }  
            }]  
        }]  
    }  
}  
""  
  
execution.setVariable('jsonString',jsonString)
```

Congratulations!

You have chosen an excellent hotel.

You're about to have an amazing experience.

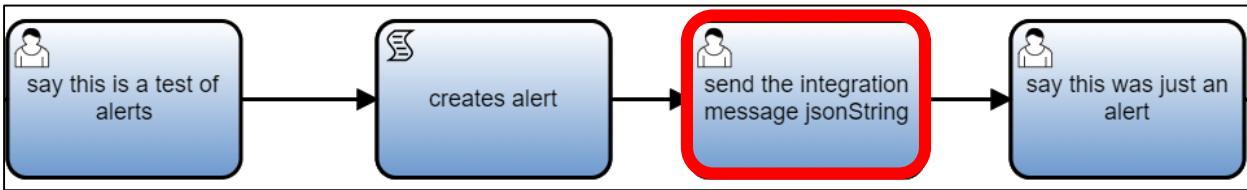
It's going to be spectacular.

Book a Room
Now

Check Out Our
Website

[Search](#)

Alerts



This is the content for my info banner

Type your message here...



This allows us to provide a more extensive message to the end user should the message itself require more information and material
than a standard informational alert.

[Click here for more information.](#)

This is the content for my warning banner

Type your message here...

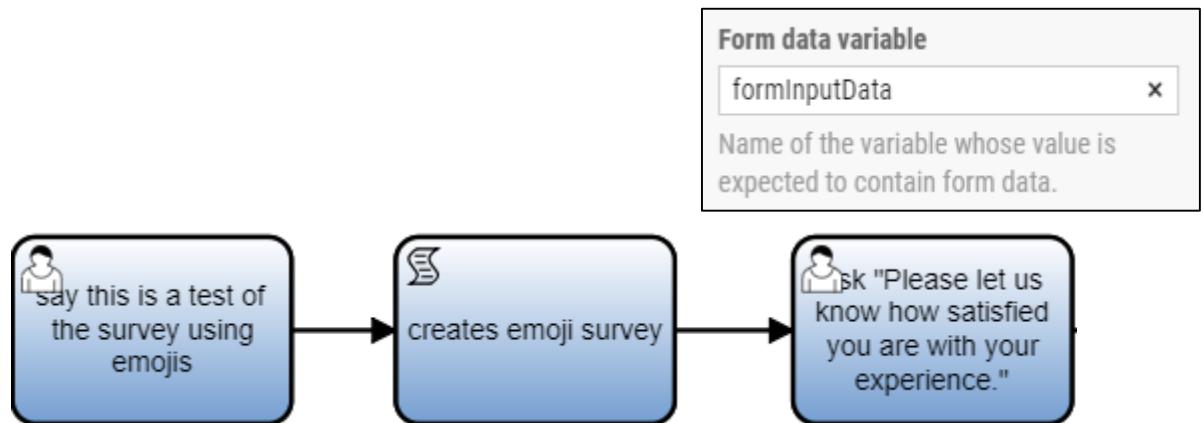


This is the content for my error banner
[click this for help](#)

Type your message here...

Survey

- Options: Stars, Emoji, Numbers
 - Emoji have 5 expressions only
 - Not currently supported in IE11



Please let us know how satisfied you are with your experience.

Amelia | 11:24:12 am

⌚

😊 😐 😐 😊 😊

⭐ ⭐ ⭐ ⭐ ⭐

⌚

BAD

1 2 3 4 5

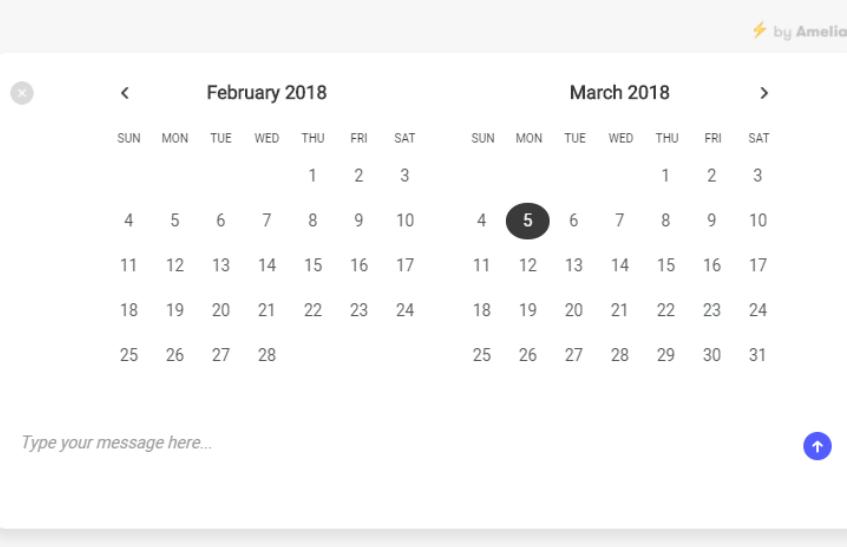
GREAT

⌚

Calendar, Metric Graph, and Table Graph

When would you like to attend?

Virtual Agent | 01:56:46 pm



Income
\$5,222.00

That's a metric graph.

Virtual Agent | 03:45:16 pm

Account Balance
\$19,020.22

+2,000 (10.5%) PAST MO

That's a large metric graph.

Virtual Agent | 03:57:41 pm

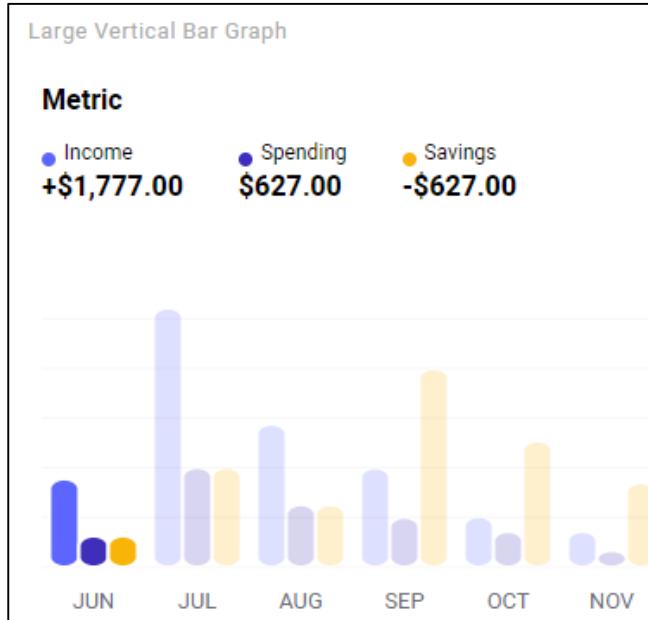
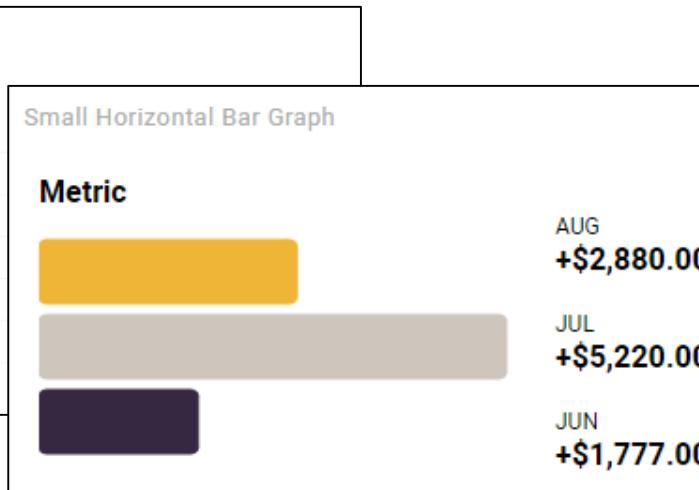
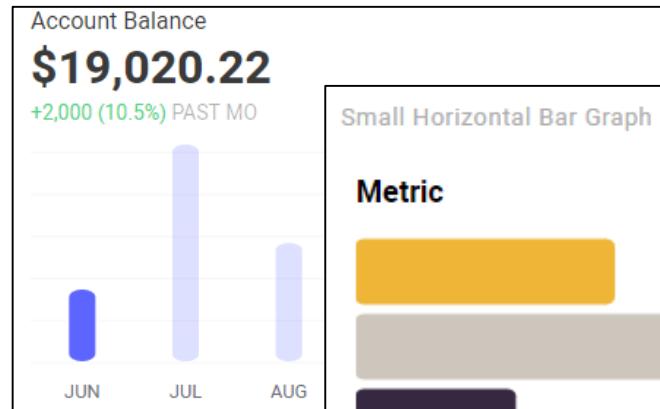
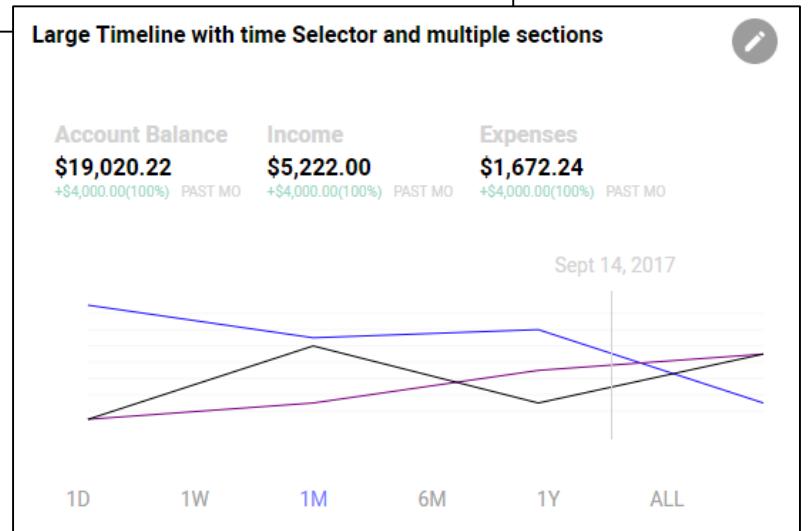
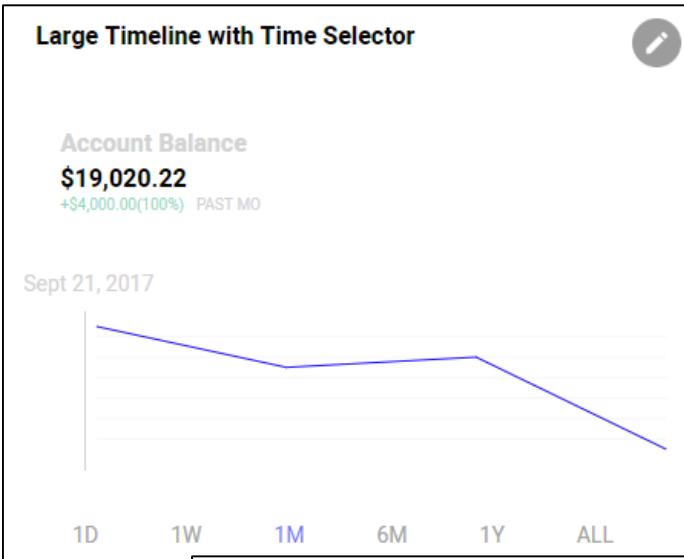
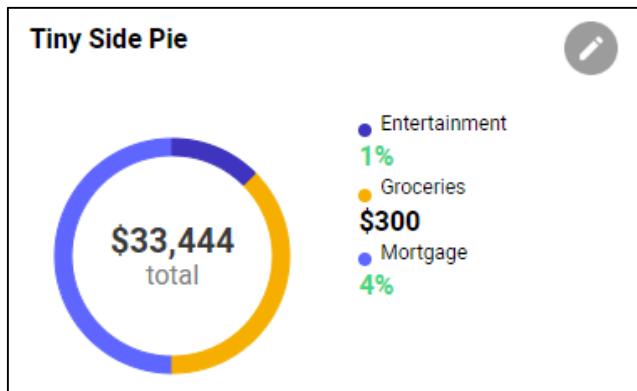
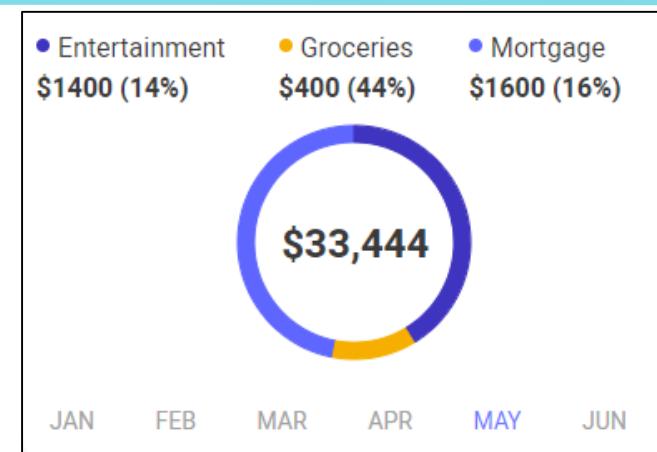
This is a test of the table in chat in custom UI.

Virtual Agent | 10:27:06 am

Account Balance

MONTH ▾	INCOME	SPENDING	SAVING
Feb	\$3,123	\$889	\$402
Mar	\$5,423	\$489	\$932
Apr	\$6,423	\$789	\$32
May	\$6,423	\$789	\$902
Jun	\$6,423	\$789	\$902

Pie, Timeline, and Bar Graphs



Custom UI – Theme Builder



Theme Builder Configuration Tool

The screenshot shows a live chat interface on the left and the 'Custom UI Builder' tool on the right.

Live Chat Interface:

- Header: AMELIA
- Message: Hi Raul. Great to see you!
- Timestamp: Amelia | 12:39:21 pm
- Text input field: Type your message here...
- Message: "Opening Configurator".
- Timestamp: Amelia | 12:39:25 pm
- Text input field: Chat with our Virtual Assistant by typing your responses in the text box.
- Message: "Configurator Open".
- Timestamp: Amelia | 12:39:25 pm

Custom UI Builder:

- Header: Custom UI Builder
- Section: UI Themes

 - Test for Leads not published
 - Test for Leads not published
 - Test for I not publ

- Section: Theme Name

 - Test for Leads

- Section: Header

 - Header Color: #FF
 - Header Menu:
 - Hamburger (selected)
 - Exposed Links

The screenshot shows several configuration panels for theme settings.

Nav Menu Items:

- Reset Conversation
- Select Domain
- Agent Help
- Logout

Logo:

- Name: Amelia
- Image: logo.png
- Dimensions: 20px height (40px @2x)
- Type: PNG
- Size: 150 kb

Color Theme:

- Light UI
- Dark UI

Primary Color & Associated Font Color:

- Color: #002663
- Associated Components:
 - Buttons
 - Tooltip
 - Menu

Secondary Color & Associated Font Color:

- Color: #ECAC00
- Associated Components:
 - User chat bubbles
 - Selection Option Buttons
 - Pickers (List, Calendar etc)

Selected Components Preview:

- Large Button
- Tooltip

Font:

- Font: Noto Serif
- Preview URL: <https://fonts.google.com/specimen/Noto+Serif>
- Font Preview:
AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQuRrSsTtUuVvWwXxYyZz
- Font Preview:
AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQuRrSsTtUuVvWwXxYyZz

Changes the look of Custom UI at a global level for a specific bundle

- Change the logo and avatar
- Modify the theme
- Change primary and secondary font colors
- Include/exclude select menu items
- Requires Global permissions

URL Launch



IPsoft Proprietary – For Training Purposes Only
© 2018 IPsoft Inc.

URL Launch – Custom UI 5.7.0+

Trigger a BPN directly:

- https://.../Amelia/ui/UI_BUNDLE_NAME/?bpn_processName=BPN_Name_Here
- https://.../Amelia/ui/UI_BUNDLE_NAME/?bpn_processName=reserveARoom

Pass BPN variable names/values:

- https://.../Amelia/ui/UI_BUNDLE_NAME/?bpn_processName=BPN_Name_Here&bpn_varName=varValue
- https://.../Amelia/ui/UI_BUNDLE_NAME/?bpn_processName=conversationServiceURLVar2&bpn_username=Fred&bpn_color=red&bpn_number=5
- Passed variable process scope - can be used in tasks (e.g., say \${variableName})

Pass initialConversationAttributes:

- https://.../Amelia/ui/UI_BUNDLE_NAME/?attrib_voice=true
 - Can also be set in the config.son for the bundle

Specify domain in Custom UI:

- https://.../Amelia/ui/UI_BUNDLE_NAME/?domainCode=brian
- Specify domain with language pack in Custom UI:
https://.../Amelia/ui/UI_BUNDLE_NAME/?bpn_processName=reserveARoom&language=en

Stochastic BPN





Introduction to Stochastic BPNs

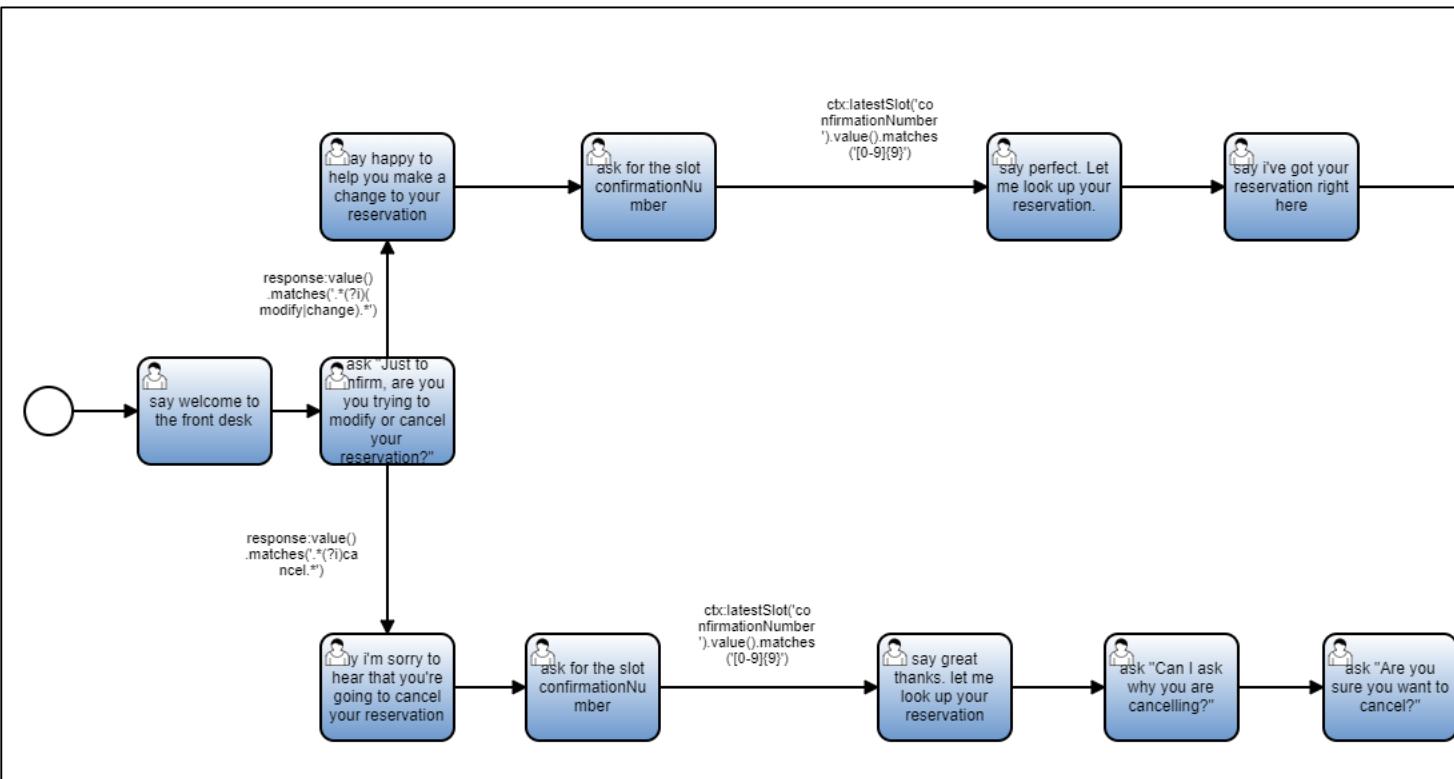
- Makes Intra-BPN navigation possible
 - Nodes given potential value based on an end user utterance
 - Uses spreading activation algorithm to determine correct node to go to next
 - Can navigate/jump to nodes with highest potential
 - User provides needed info, allowing Amelia to skip answered questions to next unanswered node
 - Relies on recognizing/filling slots in different paths in the BPN
 - As slot filled, user is redirected to correct path
 - Algorithm seeks path from current node to destination node (i.e., one with max potential)
 - More than one possible, takes first node
 - Path found, returns destination node or first preconditioned node (between current and destination nodes)
 - No path found = Least Common Ancestor of both returned as next node

Main Properties	
Name	cancelModifyReservation
Type	Stochastic
BPN navigation types	
Documentation	
Default escalation queue	
Default escalation queue for this and children BPNs.	
Time Decay Factor	0.4

Not a context switch as path redirected and never returns to the original path unless another stochastic switch occurs

Stochastic BPN

- No advantages when used with script task or set task in BPN
 - Both tasks require execution before forward movement
 - Focus should be on ask and say tasks
- Use with “wrapper” BPNs
 - Jump around to different paths based on end user utterance - decisions



- Task set to precondition
 - Stochastic algorithm executes task even if high potential nodes farther down flow
 - Algorithm gets path from current node to destination node
 - Checks if any node preconditioned
 - If yes, precondition node next

Question selection policy

Fixed

Question selection policy for slots with multiple questions for a particular BPN.

Precondition

No

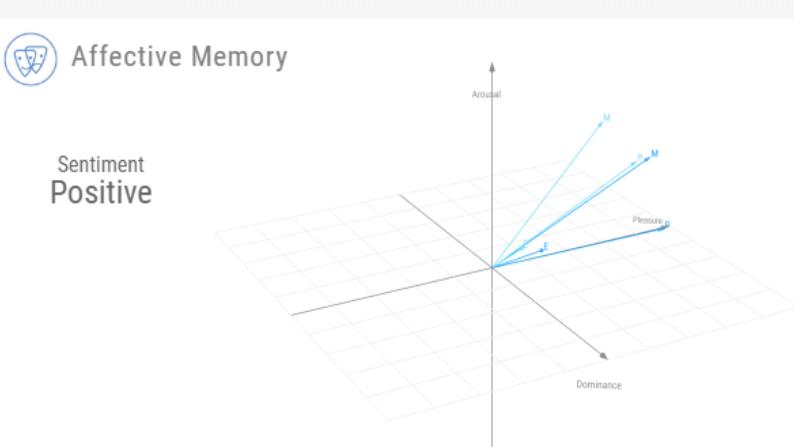
Determines whether this slot is a goal precondition or not.

Affective Memory



Mind View: Affective Memory

AMELIA Brian Kuchta ●



Affective Memory

Sentiment Positive

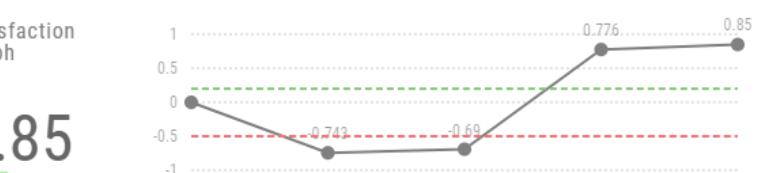
User Mind 3D CONVERSATIONS ADMIN Demos

User Amelia

	Pleasure	Arousal	Dominance
Emotion	0.915 0.564	-0.341 -0.151	0.563 0.325
Mood	0.169 0.26	0.095 0.202	0.29 0.298
Personality	0.233 0.402	0.128 0.189	0.252 0.217

Satisfaction Graph

0.85 +0.07



High ↑

Low ↓

My pleasure.

What do you appreciate?

Amelia | 10:01:52 AM

You are very helpful Thank you

Brian Kuchta | 10:02:57 AM

My pleasure.

Amelia | 10:02:59 AM

Type message ...

C + 📱

Send

Microphone icon

Pleasure-Arousal-Dominance (PAD)

Pleasure Arousal Dominance (PAD) graph

- Based on deep learning models/linguistic rules
 - Pleasure: positive/negative attitude
 - Arousal: energy level
 - Dominance: control

Emotion

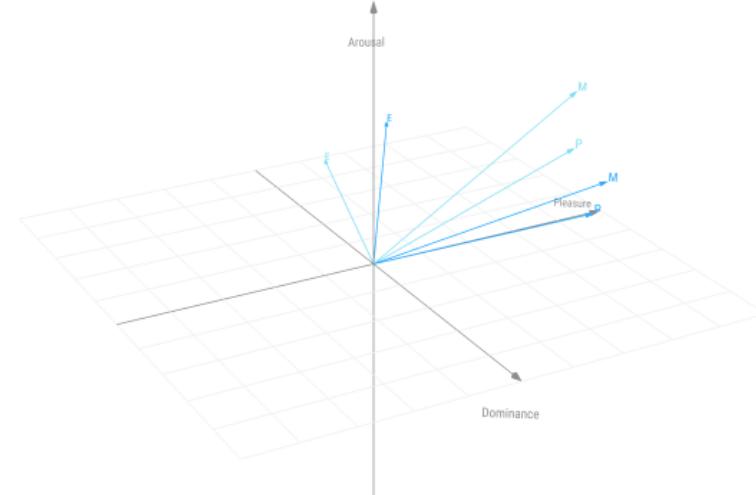
- Short-lived and triggered in the moment
- Can change from turn to turn

Mood

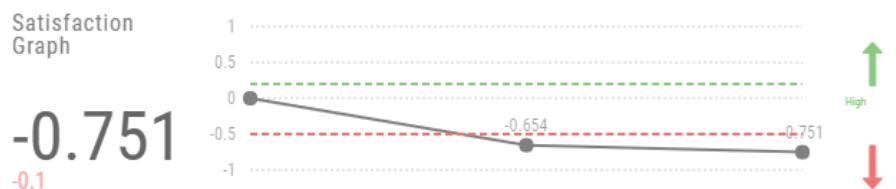
- General feeling that extends from emotion
- Longer lasting
- Takes more to change

Personality

- More stable across period of time
- Not caused by event/event appraisal



	Pleasure	Arousal	Dominance
Emotion	-0.803 -0.609	0.246 0.377	0.004 0.125
Mood	0.153 0.365	0.21 0.263	0.259 0.252
Personality	0.249 0.402	0.143 0.189	0.228 0.217

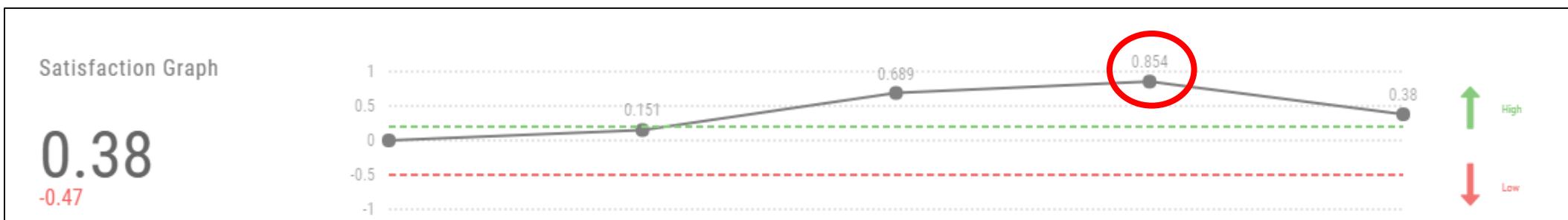




Satisfaction Scoring

- PAD and OCC scores aggregated to determine SAT score
- Computed at each dialog turn, currently after Amelia's response
- Satisfaction graph
 - Range: -1.0 to 1.0
- Emotional quotient for user and Amelia
 - Shows Amelia's mood over time: gray line
 - Satisfaction threshold: green line
 - Negative threshold: red line
- Takes into account Amelia's response

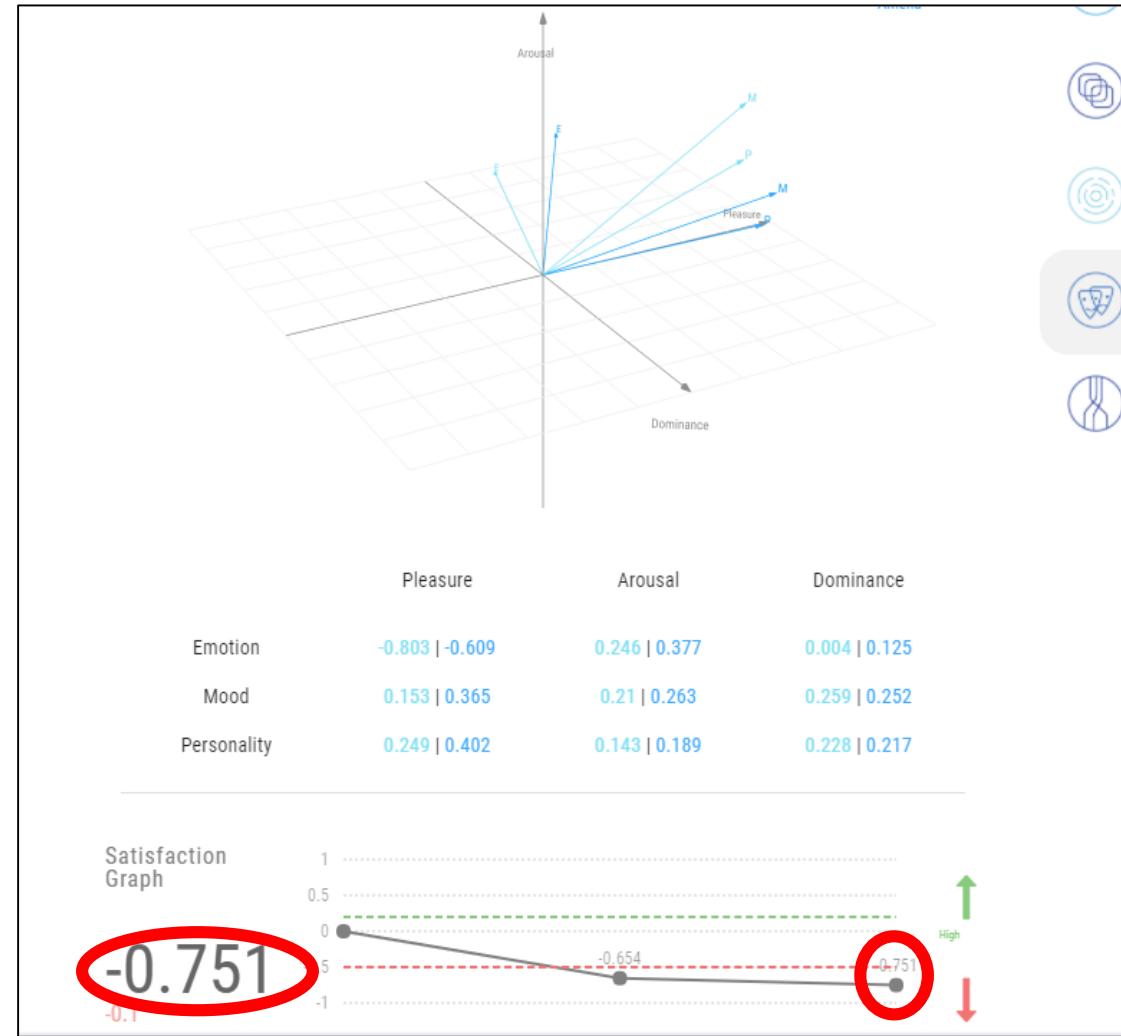
```
def score = conversationService.getEmotionalSatisfactionScore()  
  
execution.setVariable("userSatisfactionScore", score)
```





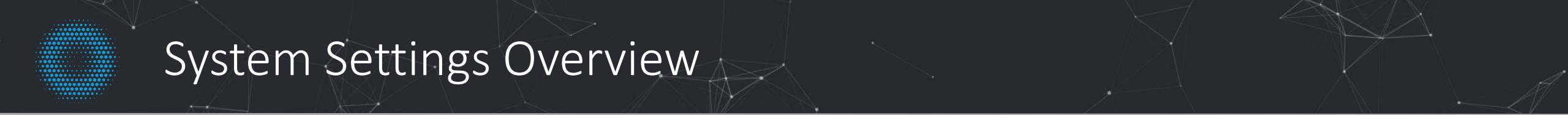
Amelia Auto-Escalate – Property Setting

- Amelia can auto-escalate based on affective memory
- Set engine application properties:
 - `amelia.humanization.escalation.enabled=true`
 - `amelia.humanization.escalation.low-satisfaction-threshold= -0.60` (default)



Admin and System Settings Overview





System Settings Overview

- Authentication Systems
- Authentication Policies
- Escalation Teams
- Escalation Queues
- Domains
- Groups
- Response Pools
- Users
- Roles
- UI Bundles
- Audit Log

Domains: Greeting, Pre-Escalation, Pre-Close Process

System Settings -> Domains

- Can set specific BPNs to run during all other processes at specific times:
 - Greeting
 - Pre-Escalation
 - Pre-Close

Conversation Event Processes

Greeting Process	<input type="text"/>	
Pre-Escalation Process	<input type="text"/>	
Pre-Close Process	<input type="text"/>	

Domains: Transcripts

Replace Personal Information (PI) with mask

Anonymize User = masks user data such as name and email address

- Match (Key): Replace (Replacement Value)
- Regex: Mask Value

The screenshot shows a software interface titled "Transcripts". At the top left, there is a checkbox labeled "Anonymize User". Below it are two buttons: "Match" and "Replace". To the right of these buttons is a green "+" sign. Further down, there is a section labeled "Auto-Cleanup Days Retained" with a corresponding input field.

Questions





Thank You

Brian Kuchta

Senior Amelia Trainer, Global Training and Development

Brian.Kuchta@ipsoft.com



AMELIA V3

Thank You



IPsoft Proprietary – For Training Purposes Only
© 2018 IPsoft Inc.



Presented by
IPsoft Global Training and Development