

# **Story Generator Documentation**

*Release 1.0.0*

**FB Persona**

July 25, 2018



<b>1</b>	<b>storygen</b>	<b>1</b>
1.1	SimpleNLG module . . . . .	1
1.2	driver module . . . . .	1
1.3	categories_dbinitialization module . . . . .	1
1.4	StoryOverview module . . . . .	1
1.5	OverviewObject module . . . . .	2
1.6	StoryLikedPagesAndEvents module . . . . .	3
1.7	LikedPagesEventsObject module . . . . .	4
1.8	StoryFangirlFanboy module . . . . .	4
1.9	StoryFoodie module . . . . .	5
1.10	StoryGamer module . . . . .	7
1.11	StorySportsFanatic module . . . . .	8
1.12	UserPreferenceObject module . . . . .	9
1.13	DatabaseConnection module . . . . .	11
1.14	AssertionModel module . . . . .	11
1.15	CategoryModel module . . . . .	11
1.16	ProfileModel module . . . . .	12
1.17	AssertionType module . . . . .	12
1.18	DirectAssertion module . . . . .	12
1.19	DirectKnowledge module . . . . .	12
1.20	LikedPagesAndEvents module . . . . .	12
1.21	Assertion module . . . . .	12
1.22	Birth module . . . . .	13
1.23	Category module . . . . .	13
1.24	Gender module . . . . .	13
1.25	LivingIn module . . . . .	13
1.26	Person module . . . . .	14
1.27	UserPreference module . . . . .	14
1.28	JSONParser module . . . . .	14
1.29	JSONReader module . . . . .	14
1.30	FileWriter module . . . . .	14
1.31	render module . . . . .	15
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



## 1.1 SimpleNLG module

```
class SimpleNLG.SimpleNLG
    Bases: object

    importsimplenlg ( )
        imports the SimpleNLG through the use of Py4j (enables Python programs to dynamically
        access Java objects) and initializes the SimpleNLG classes and objects
```

## 1.2 driver module

```
class driver.Driver
    Bases: object

    init ( )
        initializes the objects needed for the story generation

    run ( )
        runs the story generator
```

## 1.3 categories\_dbinitialization module

```
categories_dbinitialization.populate_categories ( self )
```

## 1.4 StoryOverview module

```
class StoryOverview.StoryOverview
    Bases: object

    agephrase ( birth, name )
        returns a DocumentElement with the information of the user's age

    Parameters
        • Birth object containing the birth date of the user (birth) –
        • whole name of the user (name) –
```

Returns DocumentElement extension of NLG comprises of a sentence with the user's age

**birthphrase** ( *birth* )

returns a DocumentElement with the information of the user's birth date

Parameters **Birth** object containing the birth date of the user (*birth*) –  
Returns DocumentElement extension of NLG comprises of a sentence with the user's birth information

**generateoverview** ( *overviewobject* )

calls the methods needed to form the Overview and realizes the elements into one paragraph

Parameters **contains different objects needed by this method**  
(*overviewobject*) –

Returns story an overview paragraph containing the basic information about the user

**getage** ( *birthday* )

computes the user's age

Parameters **birth date of the user in Date format** (*birthday*) –  
Returns current age of the user

**hometownphrase** ( *livingin* )

returns a DocumentElement with the information of the user's hometown

Parameters **LivingIn** object containing the the hometown of the user  
(*livingin*) –

Returns DocumentElement extension of NLG comprises of a sentence with the user's hometown

**locationphrase** ( *livingin* )

returns a DocumentElement with the information of the user's location

Parameters **LivingIn** object containing the location and address where  
the user lives (*livingin*) –

Returns DocumentElement extension of NLG comprises of a sentence with the user's location

## 1.5 OverviewObject module

**class** OverviewObject.**OverviewObject**

Bases: `object`

**birthObj**

alias of `storygenappv2.storygen.objects.Birth.Birth`

**genderObj**

alias of `storygenappv2.storygen.objects.Gender.Gender`

**initializeBirthday** ( )

initializes the birth date of the user

**initializeGender** ( )

initializes the gender of the user

**initializeLivingIn** ( )

initializes the location, hometown and current living of the user

**initializePerson** ( )

initializes the name of the person

**initializeobject ( )**  
initializes all the objects

**livinginObj**  
alias of `storygenappv2.storygen.objects.LivingIn.LivingIn`

**personObj**  
alias of `storygenappv2.storygen.objects.Person.Person`

## 1.6 StoryLikedPagesAndEvents module

**class StoryLikedPagesAndEvents.StoryLikedPagesAndEvents**

Bases: `object`

**attended\_eventsphrase ( *events\_assertions* )**  
returns a DocumentElement with the gathered information of the attended events of the user  
includes listing the name of the attended events that the user

**Parameters** **assertions** containing all the attended events of the user  
(*events\_assertions*) –

**Returns** DocumentElement extension of NLG comprises of the paragraph listing the names of the events in each assertion

**generatesummary ( *likeseventsobj, pronoun, user\_pref* )**  
calls the methods needed to form the paragraph for Liked Pages and Events and realizes the elements into one paragraph.

**Parameters**

- **LikedPagesEventsObject** containing the objects needed by this method (*likeseventsobj*) –
- **contains the referring expression, pronoun** (*pronoun*) –
- **user preference of the user** (*user\_pref*) –

**Returns** story contains a paragraph of the user's liked pages and events

**interested\_eventsphrase ( *events\_assertions* )**  
returns a DocumentElement with the gathered information of the events that the user is interested in  
includes listing the name of the events that the user

**Parameters** **assertions** containing all the events that the user is interested in (*events\_assertions*) –

**Returns** DocumentElement extension of NLG comprises of the paragraph listing the names of the events in each assertion

**likesphrase ( *likes\_assertions* )**  
returns a DocumentElement with the gathered information of the user's liked pages on a specific user preference  
includes listing the name of the liked pages in the given assertions

**Parameters** **assertions** containing all the liked pages of the user  
(*likes\_assertions*) –

**Returns** DocumentElement extension of NLG comprises of the paragraph listing the names of the liked pages in each assertion

## 1.7 LikedPagesEventsObject module

`class LikedPagesEventsObject.LikedPagesEventsObject`

Bases: `object`

**getAssertionsBySpecificParam** (*param\_name*, *param\_value*, *assertions*)

returns list of specific assertions based on the specific parameter-value pair requirement given a list of assertions

Parameters

- **name of the parameter** (*param\_name*) –
- **value of the parameter** (*param\_value*) –
- **assertions** –

Return specific assertions that have the parameter-value pair

**getSpecificAssertions** (*type*, *user\_pref*)

returns the specific assertions based on the user preference of the user and the assertion type

Parameters

- **assertion type of the assertions to be returned** (*type*) –
- **user preference of the user** (*user\_pref*) –

Return assertions

**initializeAsseertions** (*user\_pref*)

initializes all the assertions for the liked pages and events of the user based on his/her user preference

Parameters **UserPreference object containing the user preference of user and the list of assertions** (*user\_pref*) –

## 1.8 StoryFangirlFanboy module

`class StoryFangirlFanboy.StoryFangirlFanboy`

Bases: `object`

**describephrase** (*assertions*, *artists*)

generates a DocumentElement with a summary sentence of the user's descriptions about the specific artists that he/she is a fan of

Parameters

- **contains the specific assertions that includes the user's description about the artists** (*assertions*) –
- **list of artists that the user describes** (*artists*) –

**fandomphrase** (*fandoms*, *artist*)

returns a DocumentElement with a summary sentence about the fandoms that the user joins for a specific artist

Parameters

- **list of fandom groups that the user joins** (*fandoms*) –
- **persona/group that the user is a fan of** (*artist*) –

Returns a summary sentence which includes listing of the fandoms that the user joins

**fanofphrase** (*category*, *assertions*, *artists*)

generates a DocumentElement with a summary sentence about the artists that the user



supports on a specific category  
calls the describephrase method

**Parameters**

- **type of artist** (*category*) –
- **contains the specific assertions about the artists** (*assertions*) –
- **list of artists that the user supports** (*artists*) –

**generateFangirlFanboy** (*fangirlobj, pronoun*)

calls the methods needed to form the story for The Fangirl/Fanboy and realizes the elements into one paragraph.

**Parameters**

- **contains the different object needed by this method** (*fangirlobj*) –
- **contains the referring expression, pronoun** (*pronoun*) –

Returns story a fangirl/fanboy paragraph containing the story for “The Fangirl/Fanboy”

**genrephrase** (*category, genre*)

returns a DocumentElement with a summary sentence about the user’s favorite genre on a specific category.

**Parameters**

- **main category which the user is a fan of** (*category*) –
- **favorite genre category of the user** (*genre*) –

Returns a summary sentence about the user’s favorite genre

**introphrase** (*category*)

returns a DocumentElement with the sentence introductory about the user’s fan of.

**Parameters** **main category which the user is a fan of** (*category*) –  
Returns a sentence introductory about the category that the user is a fan of

**specific\_fanphrase** (*category, assertions, objects*)

generates a DocumentElement with a summary sentence about the objects that the user loves about a specific category

**Parameters**

- **main category of the objects that the user is a fan of** (*category*) –
- **contains the specific assertions about the objects** (*assertions*) –
- **list of objects that the user is a fan of** (*objects*) –

## 1.9 StoryFoodie module

**class StoryFoodie.StoryFoodie**

Bases: `object`

**cuisinephrase** (*cuisine, assertions, foods*)

returns a DocumentElement with a summary sentence about the foods of a specific cuisine

**Parameters**

- **specific cuisine** (*cuisine*) –
- **containing the specific cuisine as its category** (*assertions*) –
- **list of foods that the user likes** (*foods*) –

Returns a summary sentence about the foods that the user likes which belong to the specified cuisine

**describephrase** (*assertions*)

returns a DocumentElement with a summary about the user's description about a specific food

Parameters containing those that the user have described a specific food (*assertions*) –

Returns a summary sentence about the food's description made by the user

**dessertphrase** (*assertions, foods*)

returns a DocumentElement with a summary sentence about the specific desserts that the user usually eats

Parameters • containing "desserts and baking" as its category (*assertions*) –

• list of desserts (*foods*) –

Returns a summary sentence about the favorite desserts of the user

**drinkphrase** (*category, assertions, drinks*)

returns a DocumentElement with a summary sentence about the specific beverages that the user usually drinks based on the specific category

Parameters • specific category of the beverages (*category*) –

• assertions –

• list of beverages that the user usually drinks (*drinks*) –

Returns a summary sentence about the beverages that the user usually drinks according to its category

**foodphrase** (*category, assertions, foods*)

returns a DocumentElement with a summary sentence about the foods of a specific category that the user likes

Parameters • specific category that the user likes (*category*) –

• containing the specific food as its category (*assertions*) –

• list of foods that the user likes (*foods*) –

Returns a summary sentence about the foods that the user likes which belong to the specified category

**generatebycategory** (*foodieobj*)

calls the methods to generate and to form the story by category

Parameters contains the different object needed by this method (*foodieobj*) –

**generatefoodie** (*foodieobj, pronoun*)

calls the methods needed to form the story for The Foodie and realizes the elements into one paragraph.

Parameters • contains the different object needed by this method (*fangirlobj*) –

• contains the referring expression, pronoun (*pronoun*) –

Returns story a foodie paragraph containing the story for "The Foodie"

**introphrase** ( *main\_categ, categories* )

returns a DocumentElement with the sentence introductory about the food categories that the user usually eats

Parameters    • **main category which the user likes** (*main\_categ*) –  
                  • **containing the subcategories** (*categories*) –

Returns a sentence introductory about the main food category that the user usually eats

**placevisitedphrase** ( *places* )

returns a DocumentElement with a summary sentence about where did the user usually go for the food places

Parameters **list of food places that the user went to** (*places*) –  
Returns a summary sentence about the food places that the user have visited

## 1.10 StoryGamer module

**class** StoryGamer.**StoryGamer**

Bases: `object`

**activityphrase** ( *assertions, activity, gametype* )

returns a DocumentElement with a summary sentence about the activities that the user do for a specific game type.

Parameters    • **specifically activity assertions** (*assertions*) –  
                  • **specific activity that the user does** (*activity*) –  
                  • **gametype** –

Returns a summmary sentence about the user's activities for a given type of game

**generategamer** ( *gamerobj, pronoun* )

calls the methods needed to form the story for The Gamer and realizes the elements into one paragraph.

Parameters    • **contains the different object needed by this method** (*gamerobj*) –  
                  • **contains the referring expression, pronoun** (*pronoun*) –

Returns story a gamer paragraph containing the story for "The Gamer"

**introphrase** ( *gametypes* )

returns a DocumentElement with the sentence introductory about the types of games that the user usually play

Parameters **main category of the games** (*gametypes*) –

Returns a sentence introductory about the types of games that the user usually play

**playphrase** ( *games, gametype* )

returns a DocumentElement with a summary sentence about the specific games that the user usually play for a specific type of game

Parameters    • **list of games that the user play** (*games*) –  
                  • **type of game** (*gametype*) –

Returns a summary sentence about the games that the user usually play for the given game type

**teamphrase** ( *assertions, countries* )

returns a DocumentElement with a summary sentence about the teams per country that the user supports in general

Parameters    • **that contains the 'Support' assertions** (*assertions*) –  
                   • **countries** –

Returns a summary sentence about the teams that the user supports for each country

## 1.11 StorySportsFanatic module

**class** StorySportsFanatic.StorySportsFanatic

Bases: *object*

**fanphrase** ( *athlete\_assertions, team\_assertions, sport* )

returns a DocumentElement with a summary sentence about the sports team and athletes that the user is supporting

Parameters    • **assertions containing the specific athletes that the user is a fan of** (*athlete\_assertions*) –  
                   • **assertions containing the specific teams that the user is a fan of** (*team\_assertions*) –  
                   • **sport** –

Returns a summary sentence about the teams and athletes that the user support

**generatesportsfanatic** ( *sportsfanaticobj, pronoun* )

calls the methods needed to form the story for The Sports Fanatic and realizes the elements into one paragraph.

Parameters    • **contains the different object needed by this method** (*sportsfanaticobj*) –  
                   • **contains the referring expression, pronoun** (*pronoun*) –

Returns story a sports fanatic paragraph containing the story for “The Sports Fanatic”

**introphrase** ( *sport* )

returns a DocumentElement with a sentence introductory about the specific sports that the user likes to play

Parameters **sport** –

Returns a sentence introductory about the sports that the user plays

**leaguephrase** ( *leagues, sport* )

returns a DocumentElement with a summary sentence about the sport leagues that the user is following

Parameters    • **list of sport leagues** (*leagues*) –  
                   • **sport** –

Returns a summary sentence about the sport leagues that the user follow

**othersphrase** ( *sports* )

returns a DocumentElement with a summary sentence about the other sports that the user is interested in

Parameters **list of sports** (*sports*) –

Returns a summary sentence about other sports the user is interested in

**sharedaboutphrase** ( *topics*, *sport* )

returns a DocumentElement with a summary sentence about the topics that the user shared about

Parameters    • **list of topics about a specific sports** (*topics*) –  
                   • **sport** –

Returns a summary sentence about the topics of a specific sport

**sportphrase** ( *sport* )

returns a DocumentElement with a summary sentence about the specific sports that the user is interested in

Parameters **sport** –  
 Returns a summary sentence about the sports that the user is interested in

**userpref** = 'The Sports Fanatic'

## 1.12 UserPreferenceObject module

`class UserPreferenceObject.UserPreferenceObject`

Bases: `object`

**getAssertionBySpecificParamInAssertion** ( *param\_name*, *param\_value*, *assertions* )

returns the assertions that contains the given parameter-value pair requirements on a specific list of assertions

Parameters    • **name of the parameter** (*param\_name*) –  
                   • **value of the parameter** (*param\_value*) –  
                   • **assertions** –

Returns list of assertions containing all that satisfy the parameter-value pair requirement

**getAssertionsByCategory** ( *assertions*, *category*, *categ\_level* )

returns the assertions of a category in the specific list of assertions given the category and its level

Parameters    • **assertions** –  
                   • **category** –  
                   • **categ\_level** –

Returns list of assertions containing all the assertions of a specific category

**getAssertionsBySpecificParam** ( *param\_name*, *param\_value* )

return assertions given a specific parameter-value pair requirement

Parameters    • **name of the parameter** (*param\_name*) –  
                   • **expected value of the parameter** (*param\_value*) –

Returns list of assertions containing all the assertions that have the parameter-value pair requirement

**getAssertionsBySpecificType** ( *assertions*, *assertiontype* )

returns the assertions that belongs to the given assertion type on a specific list of assertions

Parameters    • **assertions** –  
                   • **assertiontype** –

Returns the list of assertions that belongs to the given assertion type

**getAssertionsByType** ( *assertiontype* )

returns the assertions that belongs to the user's specified assertion type

:param assertiontype : :returns list of assertions containing those that have the specified assertion type:

**getAssertionsWithCategory** ( *user\_pref* )

returns the assertions of a given user preference of the user that contains a category

**Parameters** **user preferences** (*user\_pref*) –

Returns list of assertions containing all that belongs to the user preference

**getCategoryOfAssertions** ( *categ\_level, assertions* )

returns all the categories found in the assertions given a specific category level

**Parameters** • **category level** (*categ\_level*) –

• **assertions** –

Returns list of categories that are found in the assertions

**getParametersList** ( *parameters* )

converts and returns the name of the parameters from the given String format to list format

**Parameters** **parameters** –

Returns list of all parameter names

**getSpecificParamType** ( *param\_name* )

returns the parameter values of a given parameter name in all assertions of the user preference of the user

**Parameters** **parameter name** (*param\_name*) –

Return list of values for a parameter name in all assertions

**getSpecificParamTypeWithAssertion** ( *param\_name, assertions* )

returns the parameter values of a given parameter name in a set of assertions

**Parameters** • **name of the parameter** (*param\_name*) –

• **assertions** –

Returns list of parameter values

**initializeAssertions** ( *user\_pref* )

initializes the all the assertions related to the user preference of the user

**Parameters** **user preference** (*user\_pref*) –

**removeSpecificAssertions** ( *param\_name, param\_value, assertions* )

returns the updated list of assertions after removing specific assertions based on their parameter-value pairs

**Parameters** • **name of the parameter** (*param\_name*) –

• **value of the parameter** (*param\_value*) –

• **assertions** –

Returns list of assertions excluding the specific assertions to be removed

## 1.13 DatabaseConnection module

```
class DatabaseConnection.DatabaseConnection
```

```
    Bases: object
```

```
    database = 'story_db'
```

```
    getconnection ( )  
        Returns the database connection
```

```
    getdatabasename ( )  
        Returns the database name
```

```
    host = 'localhost'
```

```
    password = 'root'
```

```
    username = 'root'
```

## 1.14 AssertionModel module

```
class AssertionModel.AssertionModel
```

```
    Bases: object
```

```
    getAssertionByUserPreference ( user_pref )  
        Parameters user_pref –  
        Returns the assertions by user preference
```

```
    getAssertionType ( assertiontype, user_pref )  
        Parameters    • assertiontype –  
                      • user_pref –  
  
        Returns the assertion types of a specific user preference
```

```
    getAssertionTypeID ( assertion_type, user_pref )  
        Parameters    • assertion_type –  
                      • user_pref –  
  
        Returns the id of the assertion type of a specific user preference
```

```
    getLikedPagesAndEvents ( )  
        Returns list of LikedPagesAndEvents object
```

## 1.15 CategoryModel module

```
class CategoryModel.CategoryModel
```

```
    Bases: object
```

```
    checkIfContainsCategory ( pref_type, user_pref )  
        checks if the type parameter of the assertion contains one of the categories  
        Parameters    • type parameter of the assertion (pref_type) –
```

- **user\_pref** –

Returns if it contains the category (true or false) and Category object containing the three-level category

**getCategoryByUserPref** ( *user\_pref* )

Parameters **user\_pref** –

Returns the categories by specific user preference

## 1.16 ProfileModel module

**class ProfileModel.ProfileModel**

Bases: `object`

**getprofile** ( )

Returns a list containing the profile information of the user

**getspecificdirectknowledge** ( *type* )

Parameters **information needed in the user's profile** (*type*) –

Returns a specific information in the user's profile

## 1.17 AssertionType module

**class AssertionType.AssertionType** ( *user\_pref, type, parameters* )

Bases: `object`

## 1.18 DirectAssertion module

**class DirectAssertion.DirectAssertion** ( *user\_pref, assertion\_type, parameters* )

Bases: `object`

## 1.19 DirectKnowledge module

**class DirectKnowledge.DirectKnowledge** ( *data, type* )

Bases: `object`

## 1.20 LikedPagesAndEvents module

**class LikedPagesAndEvents.LikedPagesAndEvents** ( *type, parameters* )

Bases: `object`

## 1.21 Assertion module

**class Assertion.Assertion** ( *assertion\_type, parameterlist* )

Bases: `object`



**category** = None

**containsspecificparam** ( *param\_name*, *param\_value* )

checks if the parameter list of the assertion contains a specific parameter-value pair

Parameters • **name of the parameter** (*param\_name*) –  
• **value of the parameter** (*param\_value*) –

Returns ifContains (True or False)

**getspecificcategory** ( *categ\_level* )

Parameters **category level** (*categ\_level*) –

Returns the specific category in a certain category level

**getspecificparameter** ( *param\_name* )

Parameters **param\_name** –

Returns the value of a specific parameter

**setcategory** ( *category* )

sets the category of an assertion

Parameters **category** –

**setspecificparam** ( *param\_name*, *param\_value* )

sets the value of a specific parameter in the assertion

Parameters • **name of the parameter** (*param\_name*) –  
• **value of the parameter** (*param\_value*) –

## 1.22 Birth module

**class** Birth.**Birth** ( *birthday*, *place* )

Bases: `object`

## 1.23 Category module

**class** Category.**Category** ( *level1*, *level2*, *level3* )

Bases: `object`

## 1.24 Gender module

**class** Gender.**Gender** ( *gender*, *pronoun* )

Bases: `object`

## 1.25 LivingIn module

**class** LivingIn.**LivingIn** ( *address*, *location*, *hometown* )

Bases: `object`

```
address = "  
  
hometown = "  
  
location = "
```

## 1.26 Person module

```
class Person.Person ( name )  
    Bases: object
```

## 1.27 UserPreference module

```
class UserPreference.UserPreference ( name, assertionlist )  
    Bases: object
```

## 1.28 JSONParser module

```
class JSONParser.JSONParser  
    Bases: object  
  
    getParsedParameters ( parameters, assertionparameter )  
        Parameters    • parameters in JSON format (parameters) –  
                      • AssertionParameter object (assertionparameter) –  
  
        Returns the list of parameters
```

## 1.29 JSONReader module

```
class JSONReader.JSONReader  
    Bases: object  
  
    readjson ( filename )  
        reads the JSON file  
  
        Parameters filename –  
        Returns the assertions in JSON format
```

## 1.30 FileWriter module

```
class FileWriter.FileWriter ( overview, userpreferences, userpref_story )  
    Bases: object  
  
    save ( )  
        saves the story output in a text file
```

## 1.31 render module

- *Index*
- *Module Index*
- *Search Page*



## a

Assertion, 12  
AssertionModel, 11  
AssertionType, 12

## b

Birth, 13

## c

categories\_dbinitialization, 1  
Category, 13  
CategoryModel, 11

## d

DatabaseConnection, 11  
DirectAssertion, 12  
DirectKnowledge, 12  
driver, 1

## f

FileWriter, 14

## g

Gender, 13

## j

JSONParser, 14  
JSONReader, 14

## l

LikedPagesAndEvents, 12  
LikedPagesEventsObject, 4  
LivingIn, 13

## o

OverviewObject, 2

## p

Person, 14  
ProfileModel, 12

## s

SimpleNLG, 1  
StoryFangirlFanboy, 4  
StoryFoodie, 5  
StoryGamer, 7  
StoryLikedPagesAndEvents, 3  
StoryOverview, 1  
StorySportsFanatic, 8

## u

UserPreference, 14  
UserPreferenceObject, 9



## A

activityphrase() (StoryGamer.StoryGamer method), 7  
address (LivingIn.LivingIn attribute), 14  
agephrase() (StoryOverview.StoryOverview method), 1  
Assertion (class in Assertion), 12  
Assertion (module), 12  
AssertionModel (class in AssertionModel), 11  
AssertionModel (module), 11  
AssertionType (class in AssertionType), 12  
AssertionType (module), 12  
attended\_eventsphrase() (StoryLikedPagesAndEvents.StoryLikedPagesAndEvents method), 3

## B

Birth (class in Birth), 13  
Birth (module), 13  
birthObj (OverviewObject.OverviewObject attribute), 2  
birthphrase() (StoryOverview.StoryOverview method), 2

## C

categories\_dbinitialization (module), 1  
category (Assertion.Assertion attribute), 13  
Category (class in Category), 13  
Category (module), 13  
CategoryModel (class in CategoryModel), 11  
CategoryModel (module), 11  
checkIfContainsCategory() (CategoryModel.CategoryModel method), 11  
containspecificparam() (Assertion.Assertion method), 13  
cuisinephrase() (StoryFoodie.StoryFoodie method), 5

## D

database (DatabaseConnection.DatabaseCon-

nection attribute), 11

DatabaseConnection (class in DatabaseConnection), 11

DatabaseConnection (module), 11

describephrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 4

describephrase() (StoryFoodie.StoryFoodie method), 6

dessertphrase() (StoryFoodie.StoryFoodie method), 6

DirectAssertion (class in DirectAssertion), 12

DirectAssertion (module), 12

DirectKnowledge (class in DirectKnowledge), 12

DirectKnowledge (module), 12

drinkphrase() (StoryFoodie.StoryFoodie method), 6

Driver (class in driver), 1

driver (module), 1

## F

fandomphrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 4

fanofphrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 4

fanphrase() (StorySportsFanatic.StorySportsFanatic method), 8

FileWriter (class in FileWriter), 14

FileWriter (module), 14

foodphrase() (StoryFoodie.StoryFoodie method), 6

## G

Gender (class in Gender), 13

Gender (module), 13

genderObj (OverviewObject.OverviewObject attribute), 2

generatebycategory() (StoryFoodie.StoryFoodie method), 6

generateFangirlFanboy() (StoryFangirlFanboy.StoryFangirlFanboy method), 5

generatefoodie() (StoryFoodie.StoryFoodie

method), 6  
generategamer() (StoryGamer.StoryGamer method), 7  
generateoverview() (StoryOverview.StoryOverview method), 2  
generatesportsfanatic() (StorySportsFanatic.StorySportsFanatic method), 8  
generatesummary() (StoryLikedPagesAndEvents.StoryLikedPagesAndEvents method), 3  
genrephrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 5  
getage() (StoryOverview.StoryOverview method), 2  
getAssertionBySpecificParamInAssertion() (UserPreferenceObject.UserPreferenceObject method), 9  
getAssertionByUserPreference() (AssertionModel.AssertionModel method), 11  
getAssertionsByCategory() (UserPreferenceObject.UserPreferenceObject method), 9  
getAssertionsBySpecificParam() (LikedPagesEventsObject.LikedPagesEventsObject method), 4  
getAssertionsBySpecificParam() (UserPreferenceObject.UserPreferenceObject method), 9  
getAssertionsBySpecificType() (UserPreferenceObject.UserPreferenceObject method), 9  
getAssertionsByType() (UserPreferenceObject.UserPreferenceObject method), 10  
getAssertionsWithCategory() (UserPreferenceObject.UserPreferenceObject method), 10  
getAssertionType() (AssertionModel.AssertionModel method), 11  
getAssertionTypeID() (AssertionModel.AssertionModel method), 11  
getCategoryByUserPref() (CategoryModel.CategoryModel method), 12  
getCategoryOfAssertions() (UserPreferenceObject.UserPreferenceObject method), 10  
getConnection() (DatabaseConnection.DatabaseConnection method), 11  
getdatabasename() (DatabaseConnection.DatabaseConnection method), 11  
getLikedPagesAndEvents() (AssertionModel.AssertionModel method), 11  
getParametersList() (UserPreferenceObject.UserPreferenceObject method), 10  
getParsedParameters() (JSONParser.JSONParser method), 14  
getprofile() (ProfileModel.ProfileModel method), 12  
getSpecificAssertions() (LikedPagesEventsObject.LikedPagesEventsObject method), 4

getspecificcategory() (Assertion.Assertion method), 13  
getspecificdirectknowledge() (ProfileModel.ProfileModel method), 12  
getspecificparameter() (Assertion.Assertion method), 13  
getSpecificParamType() (UserPreferenceObject.UserPreferenceObject method), 10  
getSpecificParamTypeWithAssertion() (UserPreferenceObject.UserPreferenceObject method), 10

## H

hometown (LivingIn.LivingIn attribute), 14  
hometownphrase() (StoryOverview.StoryOverview method), 2  
host (DatabaseConnection.DatabaseConnection attribute), 11

## I

importsimplenlg() (SimpleNLG.SimpleNLG method), 1  
init() (driver.Driver method), 1  
initializeAsseertions() (LikedPagesEventsObject.LikedPagesEventsObject method), 4  
initializeAssertions() (UserPreferenceObject.UserPreferenceObject method), 10  
initializeBirthday() (OverviewObject.OverviewObject method), 2  
initializeGender() (OverviewObject.OverviewObject method), 2  
initializeLivingIn() (OverviewObject.OverviewObject method), 2  
initializeobject() (OverviewObject.OverviewObject method), 3  
initializePerson() (OverviewObject.OverviewObject method), 2  
interested\_eventsphrase() (StoryLikedPagesAndEvents.StoryLikedPagesAndEvents method), 3  
introphrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 5  
introphrase() (StoryFoodie.StoryFoodie method), 7  
introphrase() (StoryGamer.StoryGamer method), 7  
introphrase() (StorySportsFanatic.StorySportsFanatic method), 8

## J

JSONParser (class in JSONParser), 14  
JSONParser (module), 14  
JSONReader (class in JSONReader), 14  
JSONReader (module), 14



## L

leaguephrase() (StorySportsFanatic.StorySportsFanatic method), 8  
 LikedPagesAndEvents (class in LikedPagesAndEvents), 12  
 LikedPagesAndEvents (module), 12  
 LikedPagesEventsObject (class in LikedPagesEventsObject), 4  
 LikedPagesEventsObject (module), 4  
 likesphrase() (StoryLikedPagesAndEvents.StoryLikedPagesAndEvents method), 3  
 LivingIn (class in LivingIn), 13  
 LivingIn (module), 13  
 livinginObj (OverviewObject.OverviewObject attribute), 3  
 location (LivingIn.LivingIn attribute), 14  
 locationphrase() (StoryOverview.StoryOverview method), 2

## O

othersphrase() (StorySportsFanatic.StorySportsFanatic method), 8  
 OverviewObject (class in OverviewObject), 2  
 OverviewObject (module), 2

## P

password (DatabaseConnection.DatabaseConnection attribute), 11  
 Person (class in Person), 14  
 Person (module), 14  
 personObj (OverviewObject.OverviewObject attribute), 3  
 placevisitedphrase() (StoryFoodie.StoryFoodie method), 7  
 playphrase() (StoryGamer.StoryGamer method), 7  
 populate\_categories() (in module categories\_dbinitialization), 1  
 ProfileModel (class in ProfileModel), 12  
 ProfileModel (module), 12

## R

readjson() (JSONReader.JSONReader method), 14  
 removeSpecificAssertions() (UserPreferenceObject.UserPreferenceObject method), 10  
 run() (driver.Driver method), 1

## S

save() (FileWriter.FileWriter method), 14  
 setcategory() (Assertion.Assertion method), 13  
 setspecificparam() (Assertion.Assertion method), 13  
 sharedaboutphrase() (StorySportsFanatic.StorySportsFanatic method), 9

SimpleNLG (class in SimpleNLG), 1  
 SimpleNLG (module), 1  
 specific\_fanphrase() (StoryFangirlFanboy.StoryFangirlFanboy method), 5  
 sportphrase() (StorySportsFanatic.StorySportsFanatic method), 9  
 StoryFangirlFanboy (class in StoryFangirlFanboy), 4  
 StoryFangirlFanboy (module), 4  
 StoryFoodie (class in StoryFoodie), 5  
 StoryFoodie (module), 5  
 StoryGamer (class in StoryGamer), 7  
 StoryGamer (module), 7  
 StoryLikedPagesAndEvents (class in StoryLikedPagesAndEvents), 3  
 StoryLikedPagesAndEvents (module), 3  
 StoryOverview (class in StoryOverview), 1  
 StoryOverview (module), 1  
 StorySportsFanatic (class in StorySportsFanatic), 8  
 StorySportsFanatic (module), 8

## T

teamphrase() (StoryGamer.StoryGamer method), 8

## U

username (DatabaseConnection.DatabaseConnection attribute), 11  
 userpref (StorySportsFanatic.StorySportsFanatic attribute), 9  
 UserPreference (class in UserPreference), 14  
 UserPreference (module), 14  
 UserPreferenceObject (class in UserPreferenceObject), 9  
 UserPreferenceObject (module), 9

