Privacy Manager

System Documentation

---

0. Brief Overview of Alexa Concepts

Feel free to skip this section if you are already familiar with “intents”, “utterances”, “slots”, and “AWS S3”.

0.1 Intents and Utterances

The user voice commands in any skill map to appropriately named “intents”, which are an Alexa concept that amounts to being just a bag of “utterances” (user input phrases) that the model is trained to classify. So if the utterances “find aliens” and “search for unidentified flying objects” were both listed under the “FindAlien” intent, then the model is trained to return “FindAlien” whenever the user says those phrases. In the main loop of our code, we catch the intent name passed and match it to a branch of code which will then respond appropriately.

0.2 Slots

Slots are simply an Alexa concept representing special variables saved within different scopes.

- ***Persistent slots*** are saved across sessions.
- ***Session slots*** are kept throughout the dialog of a current session.
- ***Intent slots*** are values which Alexa fills in from user input
    - If an utterance looks like “sign me in as {user_name}” then Alexa is configured to look for a value to fill into the “user_name” slot of the Login intent.
    - The developer must specify the data type of intent slots, but this is not necessary for other kinds of slots. Amazon provides default slot types, like “AMAZON.FirstName”.
    - Custom slot types can be made. They are just a list of potential words expected for that slot.

0.3 Alexa Persistent Storage

Alexa can be configured to use two different kinds of persistent storage: RDBMS, or S3.
Since the main developers working on this skill were not familiar with database management systems yet, this skill was configured to use Alexa’s pre-assigned AWS S3 bucket. This is essentially a folder to and from which can have files uploaded and downloaded.

0.4 Privacy Manager’s Intents

Below are the list of intents by name currently included in the Privacy Manager skill, excluding the default ones Alexa comes with:

1. Login
2. CreateNewUser
3. Logout
4. MakeRequest
5. ListRequests
6. HasAccess
7. AllAccess
8. RevokeAccess
9. ListFiles
10. ListPreferences
11. DenyRequest
12. AcceptRequest

1. Login

Records the name of the current user to the persistent slot “current_user”.

1.1 Accepted Utterances

- log on to me as {user_name}
- sign me as {user_name}
- log in with {user_name}
- log in as {user_name}
- sign me in as {user_name}
- log onto {user_name}
- sign onto {user_name}
- sign into {user_name}
- log into {user_name}

1.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Can you repeat the user name you said?
        - What was the name of the user?
    - Slot filling expected replies:
        - Sign onto {user_name}
        - Log into {user_name}
        - {user_name}

1.3 Pseudocode Logic

```
login_username = intent_slots['user_name']
if does_user_exist(login_username):
      sign_in(login_username)
```

2. CreateNewUser

Constructs a profile for a new user by making a folder named after the user in the S3 bucket "/users/" subdirectory. Then adds a blank template privacy_preferences.json file to said folder.

2.1 Accepted Utterances

- create a new user {user_name}
- make a new user {user_name}
- create user {user_name}
- create new user {user_name}
- make a new user named {user_name}
- sign up as {user_name}
- create a new user named {user_name}

2.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Would you please restate your desired username?
        - Sorry, what did you want your username to be?
    - Slot filling expected replies:
        - {user_name}
        - I want my username to be {user_name}
        - You may call me {user_name}
        - You can call me {user_name}
        - Call me {user_name}

2.3 Pseudocode Logic

```
new_user = intent_slots['user_name']
if does_user_exist(new_user):
      # cannot make a new user, name already exists
else:
      create_new_user(new_user)
      sign_in(new_user)
```

3. Logout

Removes the ‘current_user’ slot from the persistent attributes.

3.1 Accepted Utterances

- sign me out
- log me out
- log out
- sign out

3.2 Pseudocode Logic

```
if is_logged_in():
      sign_out()
else:
      # oo user is currently signed in
```

4. MakeRequest

Creates a request object containing the requesting user's name, desired access type, and reason for the request. Adds this request object to the "pending_requests" list inside the privacy_preferences.json file within the requestee's folder.

4.1 Accepted Utterances

- three
- request access from a user
- ask {user_name} for permission to {request_type} with their recordings for {reason} purposes
- ask {user_name} for permission to {request_type} with their recordings
- ask permission to {request_type} using recordings from {user_name} for {reason} purposes
- request permission to {request_type} using {user_name} recordings for {reason} purposes
- request access to recordings from {user_name} for {reason} purposes
- ask {user_name} for permission to use their recordings
- request access to recordings from user {user_name}
- demand access to recordings from another user
- demand access rights to someone's recordings
- request access rights to a user's recordings
- make an access request for a user's recordings
- ask for permission to access another user's recordings
- request access to recordings from another user

4.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Who do you want to ask permission from?
        - What's the name of the user from who you'd like to request access?
        - To which user's recordings are you requesting access?
        - Who are you requesting permission from?
    - Slot filling expected replies:
        - {user_name}
        - Request permission from {user_name}
        - I'd like to get access from {user_name}
        - I want to ask {user_name} for permission
        - I want to ask {user_name} for access
- request_type - RequestType
    - Slot filling prompts:
        - Would you want to copy, listen, transcribe, or train artificial intelligence using their recordings?
    - Slot filling expected replies:
        - {request_type}
        - Access to {request_type} from their recordings
        - I'd like to {request_type} with their recordings
- reason - RequestReason

- Slot filling prompts:
  - Is the work you intend to use these recordings for personal, educational, research-related, or commercial?
  - Would you best describe yourself as a private entity, student, teacher, researcher, or commercial entity?
- Slot filling expected replies:
  - I am on a {reason} team
  - For a {reason} project
  - For {reason} applications
  - {reason} purposes
  - I am a {reason} entity
  - I am a {reason}
  - {reason}

4.3 Pseudocode Logic

```
if not is_logged_in():
      # cannot do anything - no one is signed in
else:
      requester = get_current_user()
      requestee = intent_slots['user_name']
      access_type = intent_slots['request_type']
      reason = intent_slots['reason']

      if does_user_exist(requestee):
            response = make_request(requester, requestee, access_type, reason)
            if response:
                  # request was put in
            else:
                  # the requester has already requested from this requestee
      else:
            # requester requested from someone who doesn't exist
```

5. ListRequests

List the currently pending access requests for the current user's data.

5.1 Accepted Utterances

- one
- list pending requests
- list my requests
- what are my current requests

5.2 Pseudocode Logic

```
if is_logged_in():
            requests = list_requests(get_current_user())
            if len(requests) > 0:
                # we have a list of requests to tell the user
            else:
                # no requests have been made to this user
        else:
            # not signed in
```

6. HasAccess

Lists the permissions granted to the current user by a given user.

6.1 Accepted Utterances

- seven
- list access permissions from a specific user
- what can i access from {user_name}
- did {user_name} give me access
- can i use recordings from {user_name}
- has {user_name} granted me access
- does {user_name} allow me to use any of their recordings
- do i have access to anything from {user_name}

6.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Which user do you want me to check?
        - Whose recordings are you wanted to check for access to?
        - You want me to see if you have access to whose files?
    - Slot filling expected replies:
        - {user_name}
        - Recordings from {user_name}
        - Check for access to recordings from {user_name}

6.3 Pseudocode Logic

```
if not is_logged_in():
	# not signed in
else:
	requester = get_current_user()
	requestee = intent_slots['user_name'].value
	if does_user_exist(requestee):
		response = list_access_from(requester, requestee)
		if len(response) > 0:
			# we have a list of access objects to parse
		else:
			# no one has granted access to this user yet
	else:
		# requester is checking for access from a nonexistent user
```

7. AllAccess

Lists all access permissions granted to the current user by other users.

7.1 Accepted Utterances

- two
- list the files i have access to
- can you list me all of the permissions
- review my access permissions
- list all files i can access
- what files do i have access to
- who do I have access to
- list all access permissions

7.2 Pseudocode Logic

```
if not is_logged_in():
      # not signed in
else:
      requester = get_current_user()
      requestee = intent_slots['user_name'].value
      if does_user_exist(requestee):
            response = list_all_access(requester)
            if len(response) > 0:
                  # we have a list of access objects to parse
            else:
                  # no one has granted access to this user yet
      else:
            # requester is checking for access from a nonexistent user
```

8. RevokeAccess

Revokes all access the current user has given to another user.

8.1 Accepted Utterances

- six
- revoke access from a user
- revoke {user_name} access
- remove {user_name} access
- revoke access from {user_name} for {reason} projects
- forbid {user_name} from {request_type} access for {reason} purposes
- forbid {user_name} from access for {reason} projects
- revoke {request_type} access from {user_name}
- forbid {user_name} from {request_type} access
- revoke access from {user_name}

8.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Revoke access from which user?
        - Forbid who from access?
        - Who do you want to forbid accessing your data?
        - From whom would you like to revoke access?
    - Slot filling expected replies:
        - {user_name}
        - Revoke access to {user_name}
- request_type - RequestType
- reason - RequestReason

8.3 Pseudocode Logic

```
if not is_logged_in():
      # not signed in
else:
      requestee = get_current_user()
      requester = intent_slots['user_name'].value
      if does_user_exist(requester):
            response = revoke_access(requestee, requester)
            if len(response) > 0:
                  # we have revoked permissions
            else:
                  # no permissions were revoked - couldn’t find any
      else:
            # requester is revoking access from a nonexistent user
```

9. ListFiles

Lists the files within the "/recordings/" subdirectory of the current user's folder.

9.1 Accepted Utterances

- view recordings
- list recordings
- list files
- view files
- view my files
- List my files

9.2 Pseudocode Logic

```
if is_logged_in():
	current_user = get_current_user()
	recordings_key = f"Media/users/{current_user}/recordings/"
	recordings = list_file_names(recordings_key)

	if len(recordings) > 0:
		# there are recordings to list
	else:
		# there were no recordings found
else:
	# not signed in
```

10. ListPreferences

Lists who can access each of the current user's files.

10.1 Accepted Utterances

- eight
- list all access given
- list all access granted
- list all permissions given
- list all permissions granted

10.2 Pseudocode Logic

```
if not is_logged_in():
      # not signed in
else:
      user = get_current_user()
      response = list_preferences(user)
      if len(response) > 0:
            # we have a response to give the user
      else:
            # the user has yet to grant access to anybody
```

11. DenyRequest

Modifies the current user's privacy_preferences.json by removing the request object associated with a given user from the "pending_requests" list to the "denied_requests" list.

11.1 Accepted Utterances

- reject request from {user_name}
- deny request from {user_name}
- deny request from a user
- five

11.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Whose request would you like to reject?
    - Slot filling expected replies:
        - {user_name}
        - Reject {user_name}
        - Deny {user_name}

11.3 Pseudocode Logic

```
if not is_logged_in():
	# not signed in
else:
	requestee = get_current_user()
	requester = intent_slots['user_name']
	if does_user_exist(requester):
		response = deny_request(requestee, requester)
		if response:
			# relay response to user
		else:
			# no requests from this user to deny
	else:
		# tried to deny access to a user that doesn't exist
```

12. AcceptRequest

Modifies the current user's privacy_preferences.json by removing the request object associated with a given user from the "pending_requests" list to the list within "preferences[file_name]".

12.1 Accepted Utterances

- grant access to {user_name}
- four
- accept request from {user_name}
- accept request from {user_name} for {file_name}

12.2 Intent Slots

- user_name - AMAZON.FirstName
    - Slot filling prompts:
        - Whose request would you like to accept?
    - Slot filling expected replies:
        - {user_name}
- file_name - FileName
    - Slot filling prompts:
        - To which file would you like to grant them access, otherwise say 'all files'?
    - Slot filling expected replies:
        - {file_name}

12.3 Pseudocode Logic

```
if not is_logged_in():
      # not signed in
else:
      requestee = get_current_user()
      requester = intent_slots['user_name']
      file_name = intent_slots['file_name']
      if does_user_exist(requester):
            response = accept_request(requestee, requester, file_name)

            if response:
                  # accepted request
            else:
                  # no request to accept exists
      else:
            # accepting request from user which doesn't exist
```