**Sayso – Technical Guide**

**Sayso Botservice**

**Location:**

* **TFS Root URL:**[*http://inrtnrepos:8080/tfs/Innoruption/iSurvey*](http://inrtnrepos:8080/tfs/Innoruption/iSurvey)
* **TFS Clone URL:** *http://*[*inrtnrepos*](http://inrtnrepos:8080/tfs/Innoruption/iSurvey)*:8080/tfs/Innoruption/\_git/iSurvey*
* **Server URL:** *http://msweb02:1112*

**Directory List:**

1. authentication
2. config
3. dialogs
4. doSurvey
5. error
6. feedback
7. formatSurveyQ&A
8. remindMeLater
9. responseCard
10. startup
11. surveyList
12. testData
13. webAPI

**Modules in Root Directory**:

1. server.js
2. .env
3. .gitignore
4. package.json
5. package-lock.json
6. Readme.md

* **Authentication:** This module is responsible to validate user against the **userid** passed as URL parameter. Accepting user id from URL, get user token by calling API, decrypt user token, etc. are the primary functionalities of this module.
  + **Files :**
    - identifyUserViaSharePointToken.js
* **Config:** This module is responsible to hold all API resource URI’s of different environment like development, staging, production, etc.
  + **Files :**
    - default.json
    - development.json
    - staging.json
    - production.json
* **Dialogs:** This module is responsible to hold all dialog functions used in the bot. This dialogs are getting triggered from different modules based on the application flow.
  + **Files :**
    - beginAllSurvey.js
    - botFeedback.js
    - gridChoiceTypeQuestions.js
    - startSurvey.js
    - root.js
* **DoSurvey:** This is module responsible to conduct the survey and submit response. Presenting survey questions to user, accept and store user response, send progress, submit response, etc. are various functionalities of this module.
  + **Files :**
    - gridChoiceTypeQuestion.js
    - infoToFrontEnd.js
    - partialSurveyQs.js
    - promptUserBasedOnQuestionType.js
    - surveyQ&A.js
    - submitSurveyResponse.js
    - takeAllSurvey.js
* **Error:** This module is responsible to **log** all **runtime errors** occurred during survey process.
  + **Files :**
    - logError.js
* **Feedback:** This module is responsible to ask user about his/her experience with the bot and send users rating to backend **(Currently this module is not in use, but it will get added in future releases)**.
  + **Files :**
    - botFeedback.js
* **FormatSurveyQ&A:** This module is acting as an interface between survey **question objects** (coming from API) and **adaptive cards** displayed to present query and accept user response. The primary responsibilities of this module are to format survey **question** and possible **answer options**, pass them to **adaptive card/suggested action** and send them to chat window.
  + **Files :**
    - choiceInputOption.js
    - dateTimeInputOption.js
    - dropDownListInputOption.js
    - multiValueInputOption.js
    - numberInputOption.js
    - textboxInputOption.js
* **RemindMeLater:** This module is responsible to check remind me options configured with the survey, calculate next reminder date and post remind me response to web API when user clicks on “**Remind Me Later**” button.
  + **Files :**
    - remindMeLaterResponse.js
* **ResponseCard:** This modules holds all the adaptive card schemas used to present survey query and accept response from user.
  + **Files :**
    - checkboxCard.js
    - dateTimeCard.js
    - dropDownCard.js
    - introCard.js
    - numberCard.js
    - textboxCard.js
* **StartUp:** This module is responsible to perform operations which needs to get executed at the beginning of conversation. Also this module is responsible to perform session management. In general, this module contains a bunch of useful **middleware** functions executed in request/response cycle.
  + **Files :**
    - auth.js
    - initiateConversationFromBotSide.js
    - serverConfiguration.js
    - sessionTimeout.js
    - typingIndicator.js
    - index.html
* **SurveyList:** This module is mainly responsible to present list of active surveys along with their expiry information and option to start surveys. The **survey information** and **options** are present in an adaptive card. Based on user’s choice, this module initiates corresponding flow.
  + **Files :**
    - activeSurveyList.js
    - displaySurveyAndDoOperations.js
* **TestData:** Contains a dummy **json** file having a list of survey query. This module is useful when there is need to **unit test** the bot without any API integration.
  + **Files :**
    - data.json
* **WebAPI:** This module holds parameters of all API’s. The resource URI of API’s are collected from **config** module based on environment (development/staging/production).
  + **Files :**
    - activeSurveySummaryAPI.js
    - allActiveSurveyQsAPI.js
    - authAPI.js
    - remindMeResponseAPI.js
    - submitSurveyResponseAPI.js
    - surveyDataAPI.js

**Other Useful Modules:**

* **server.js:** This module is the starting point of the application. All the dialogs and configuration related functions are getting triggered from this file only.
* **.env:** Environment file which holds information shared among different modules across the application. They are;
  + Authorization to get user id
  + Response submission interval
  + Reload interval
  + Completion check interval
  + Session timeout interval
* **.gitignore:** Includes files not to be considered while deployment. Ex: **node\_modules**.
* **Package.json**: Manifest file which holds some basic information about the application like name, version, author, etc. and holds the information about all the dependency node modules (**name and version**) in its attribute **dependencies**.

**Other Directories/Files:** There are few other directories/files included in the source code of **bot service**, but they are not to be directly handled by developers in order to develop bot service. Either they are auto generated during **publish/deployment** process or created during **installation** of third party modules. These directories are listed below;

1. Bin
2. Obj
3. Node\_modules
4. .pubxml files
5. Web.config & Web.Debug.config

**Directory saysoOtherComponents:** This directory holds all other components implementation code. The contents of these directory are;

1. saysoCustomizedChatUI
2. saysoOfflineConnector
3. saysoWebpart
4. saysoAnalytics

**Sayso Chat UI**

**Location:**

* **TFS URL:** [*http://inrtnrepos:8080/tfs/Innoruption/\_git/iSurvey?path=%2FI3L.iSurvey%2FI3L.iSurvey.Bot.Web%2FsaysoOtherComponents%2FsaysoCustomizedChatUI&version=GBmaster&\_a=contents*](http://inrtnrepos:8080/tfs/Innoruption/_git/iSurvey?path=%2FI3L.iSurvey%2FI3L.iSurvey.Bot.Web%2FsaysoOtherComponents%2FsaysoCustomizedChatUI&version=GBmaster&_a=contents)
* **Server URL:** *http://msweb02:1111*

**Directories:**

1. Public

**Modules in Root Directory:**

1. Server.js

**Public:** This directory contains all static assets (ex: images), styles and scripts. It also contains a host file which is served by express app. The contents of this directory are;

* **index.html** – Host file to be served by express app. This file should have the **address of offline connector**, to communicate with bot service.
* **botchat.css** – Styles related to chat
* **botchat.js** – Functionality related to rendering chat messages

If there is need to put some static assets like images, then we can have a new directory **assets** placed within **public**.

**Useful Modules:**

* **server.js:** Purpose of this module is very straightforward. This module **initialize** the express app and serve **index.html** on every request.
* **package.json**
* **.gitignore**

**Other Directory/Files**:

1. Bin
2. Node\_modules
3. Package-lock.json
4. Web.config and Web.Debug.config

**Sayso offline Connector**

**Location:**

* **TFS URL:** [*http://inrtnrepos:8080/tfs/Innoruption/\_git/iSurvey?path=%2FI3L.iSurvey%2FI3L.iSurvey.Bot.Web%2FsaysoOtherComponents%2FsaysoOfflineConnector&version=GBmaster&\_a=contents*](http://inrtnrepos:8080/tfs/Innoruption/_git/iSurvey?path=%2FI3L.iSurvey%2FI3L.iSurvey.Bot.Web%2FsaysoOtherComponents%2FsaysoOfflineConnector&version=GBmaster&_a=contents)
* **Server URL:** *http://msweb02:1113*

**Modules:**

1. server.js

**Server.js:** Purpose of this file to create an express instance and pass the instance to module **offline-directline**. This file contains the address (URL) of **botservice** to send/receive messages.

The **host address** (URL) of this app need to be passed to **offline-directline** package, so that the module can identify on which host to listen for **connection/requests**.

**Other Directory/Files:**

1. node\_modules
2. package.json & packagelock.json
3. Web.config