Workshop.
**MS Bot Framework**
**Face API**

infopulse

# MS Bot Framework

**Валерий Поляков,** аналитик, Cognitive отдел, EVRY Ukraine

*September 27, 2018*

## Intro

- Your previous experience with:
    - Programming languages
    - VCS
    - Bot frameworks
    - Azure or other cloud
- Your expectations
    - What do you want to know?
    - What makes you boring?

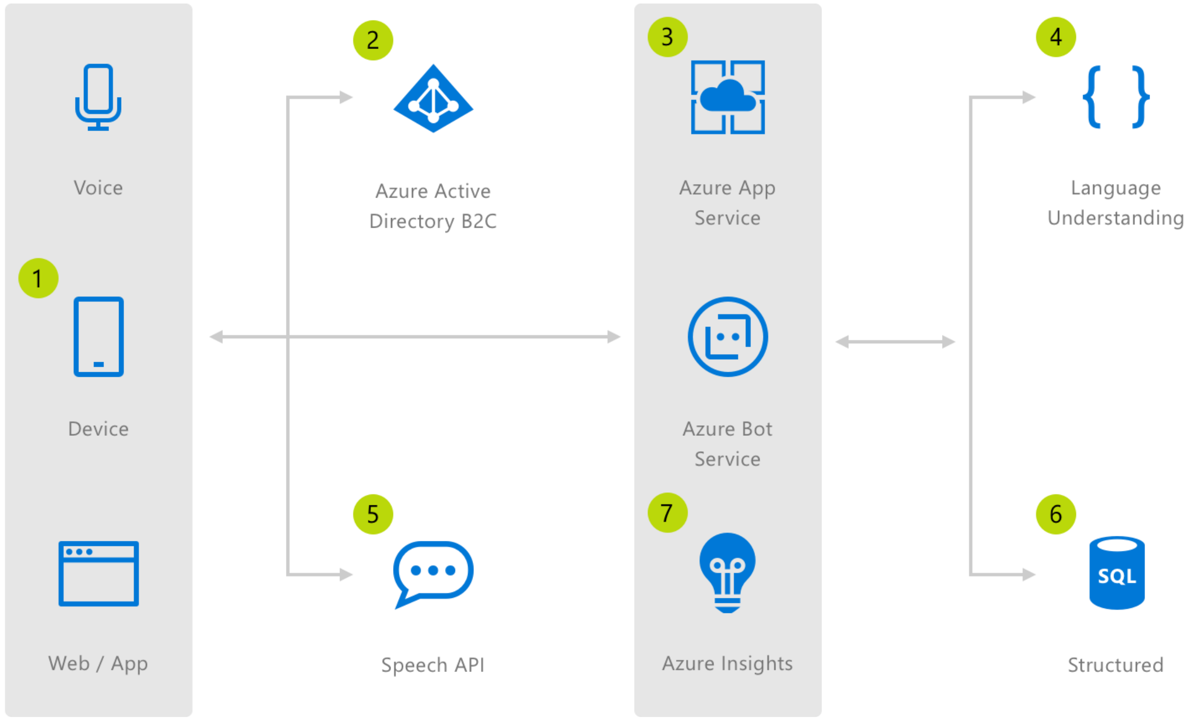
## Prerequisites

- Git
  - Setup git: https://git-scm.com/downloads
  - Register/Sigin GitHub: https://github.com
- Node.js 8.12
  - https://nodejs.org
- Bot Framework Emulator
  - https://github.com/Microsoft/BotFramework-Emulator/releases
- Visual Studio Code
  - https://code.visualstudio.com/download
- Mongodb
  - https://www.mongodb.com/
- Azure
  - https://portal.azure.com

## Agenda

- Bot architecture
- Console Bot
- Echo Bot
- Deploy to Azure

## Bot architecture

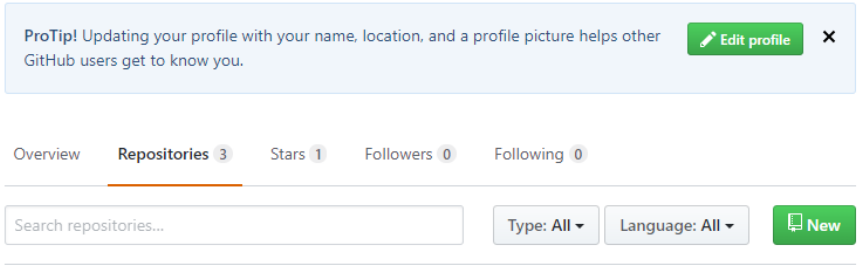


infopulse

A part of Nordic IT group EVRY

## Bot: init

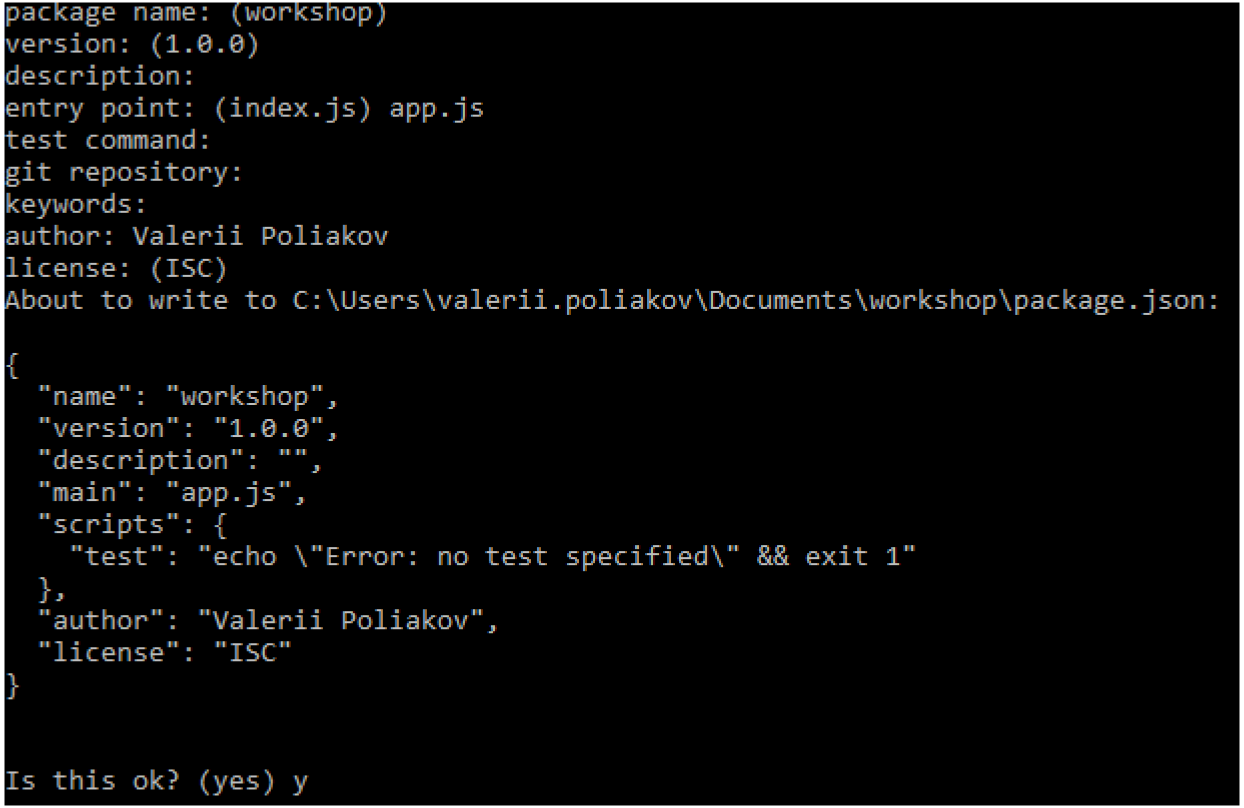
- Create git repo Botlab
  - Click “Repositories” then “New”
- Create folder
  - C:\Users\valerii.poliakov\Documents\workshop
- Run “cmd” and navigate to created folder
- Clone from git:
  - git clone https://github.com/Valerii-Poliakov/BotLab.git
- Navigate to BotLab  folder
- Create .gitignore file and add to the GitHub
  - git add .gitignore
  - git commit -m "gitignore added"
  - git push
- Run following npm command
  - npm init


ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

## .gitignore

```
# Dependency directories
node_modules/
jspm_packages/

# Optional npm cache directory
.npm

# dotenv environment variables file
.env

# Other
.DS_Store
.vscode/
```

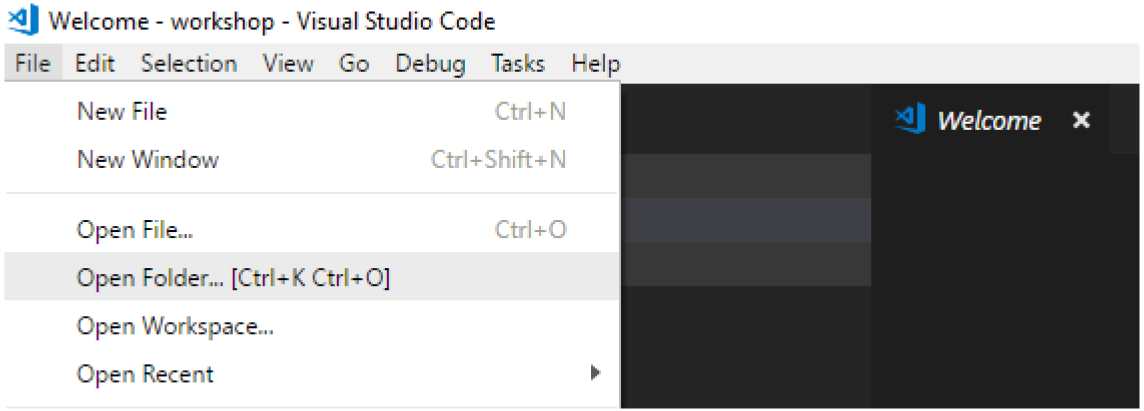
## npm init

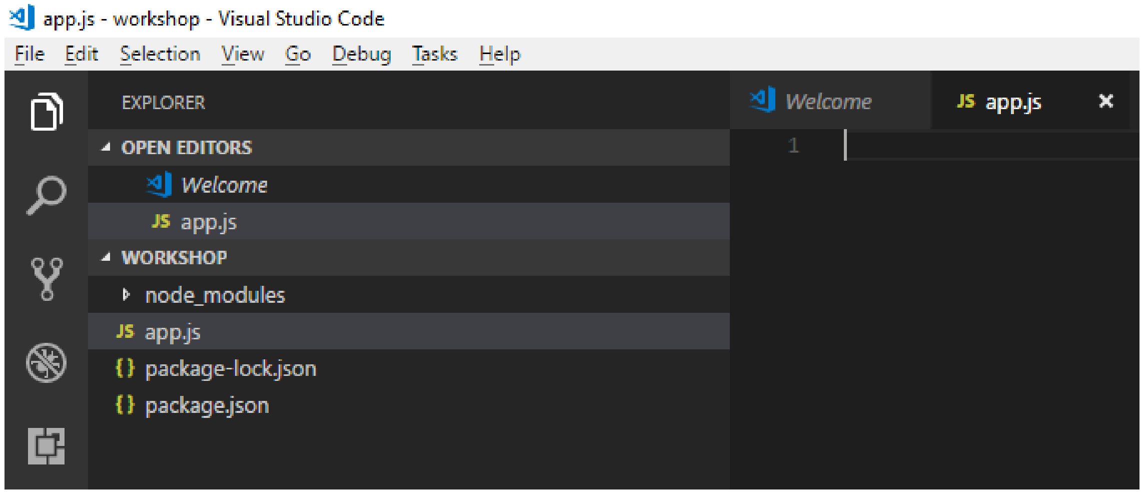


```
package name: (workshop)
version: (1.0.0)
description:
entry point: (index.js) app.js
test command:
git repository:
keywords:
author: Valerii Poliakov
license: (ISC)
About to write to C:\Users\valerii.poliakov\Documents\workshop\package.json:

{
  "name": "workshop",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Valerii Poliakov",
  "license": "ISC"
}


Is this ok? (yes) y
```

# Bot: init

- Install bot builder SDK
  - npm i --save botbuilder@3.15.0
- Run Visual Studio Code
  - Open folder “BotLab”
- Create a new file named app.js.


Welcome - workshop - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
New File Ctrl+N
New Window Ctrl+Shift+N
Open File... Ctrl+O
Open Folder... [Ctrl+K Ctrl+O]
Open Workspace...
Open Recent
Welcome



infopulse
A part of Nordic IT group EVRY

## Console Bot: App.js

```
var builder = require('botbuilder');

var connector = new builder.ConsoleConnector().listen();
var bot = new builder.UniversalBot(connector, function (session) {
session.send("You said: %s", session.message.text);
});
```

infopulse

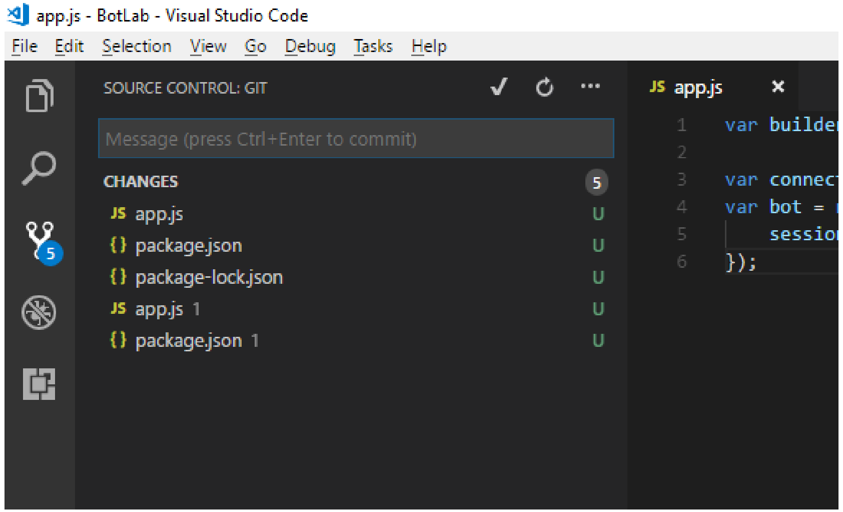
A part of Nordic IT group EVRY

## Console Bot: run

- Run bot
  - node app.js
- Test bot
  - Type any message and press enter

# Push changes

- From command line:
  - git add *
  - git commit -m “commit message“
  - git push
- From UI


app.js - BotLab - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
SOURCE CONTROL: GIT
Message (press Ctrl+Enter to commit)
CHANGES 5
app.js U
package.json U
package-lock.json U
app.js 1 U
package.json 1 U
app.js

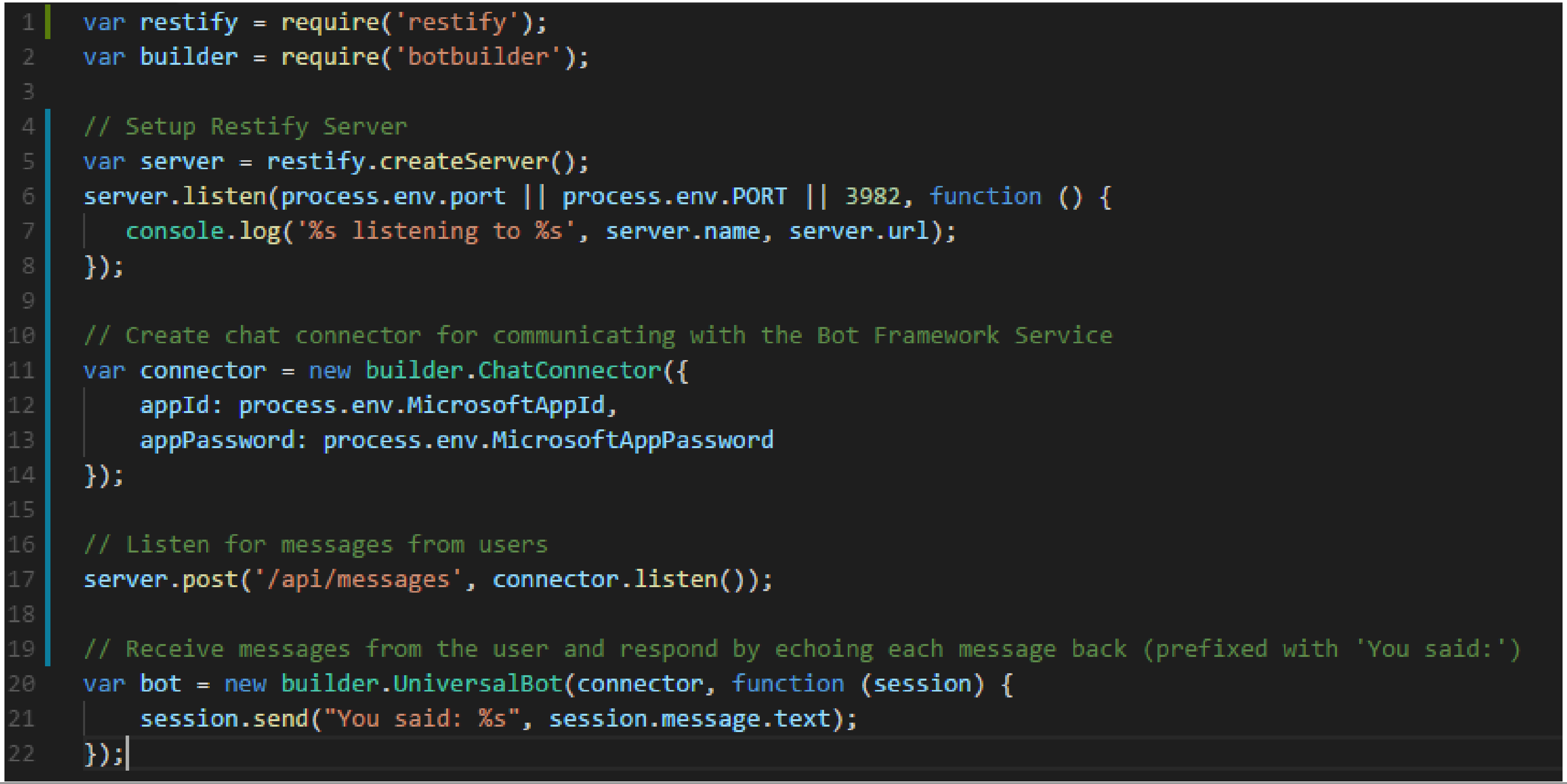
```
var builde
 
var connec
var bot = 
    sessio
});
```

infopulse
A part of Nordic IT group EVRY

## Echo bot

- Install Restify
  - npm i --save restify
- Modify app.js:
  - Require ‘restify’ module
  - Change ‘ConsoleConnector’ to ChatConnector
  - Add Microsoft App ID and App Passwrd.
  - Have the connector listen on an API endpoint.

# Echo Bot: app.js

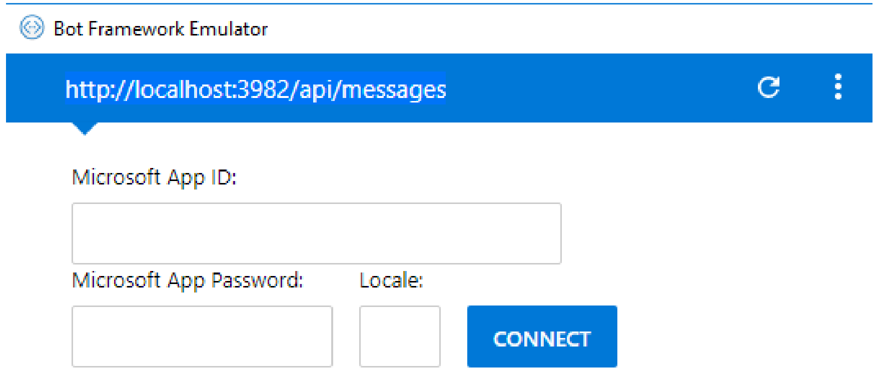


```javascript
var restify = require('restify');
var builder = require('botbuilder');

// Setup Restify Server
var server = restify.createServer();
server.listen(process.env.port || process.env.PORT || 3982, function () {
    console.log('%s listening to %s', server.name, server.url);
});

// Create chat connector for communicating with the Bot Framework Service
var connector = new builder.ChatConnector({
    appId: process.env.MicrosoftAppId,
    appPassword: process.env.MicrosoftAppPassword
});

// Listen for messages from users
server.post('/api/messages', connector.listen());

// Receive messages from the user and respond by echoing each message back (prefixed with 'You said:')
var bot = new builder.UniversalBot(connector, function (session) {
    session.send("You said: %s", session.message.text);
});
```

A part of Nordic IT group EVRY

## Echo Bot: run

- Run bot
    - node app.js
- Run botframework-emulator
- Connect to the bot
    - http://localhost:port-number/api/messages
- Chat with the bot


Bot Framework Emulator

http://localhost:3982/api/messages

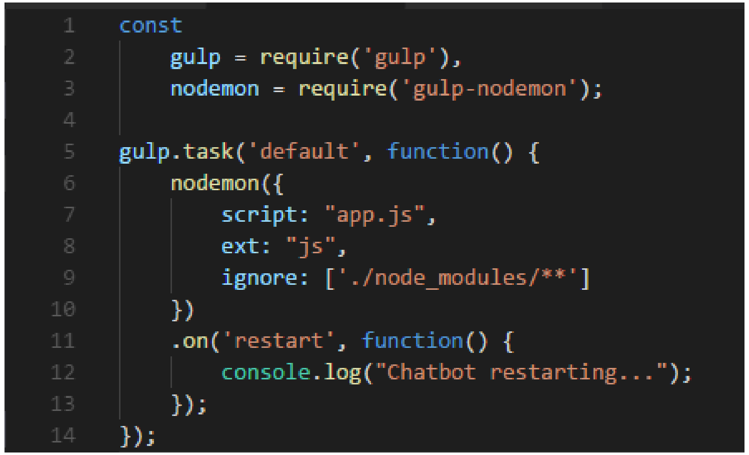
Microsoft App ID:

Microsoft App Password:

Locale:

# In Memory Storage

- Add storage for sessions.
  - const inMemoryStorage = new builder.MemoryBotStorage();
  - …
  - var bot = new builder.UniversalBot(connector, function (session) {
  -     session.send("You said: %s", session.message.text);
  - }).set('storage', inMemoryStorage);
- Gulp
  - npm i –save gulp
  - npm i --save gulp-nodemon
  - add gulpfile.js



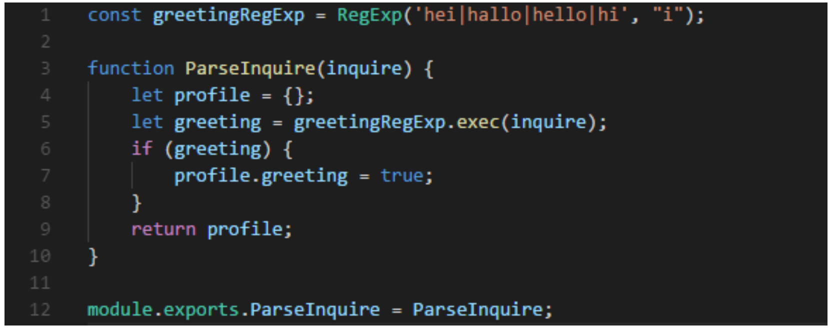
```
const
    gulp = require('gulp'),
    nodemon = require('gulp-nodemon');

gulp.task('default', function() {
    nodemon({
        script: "app.js",
        ext: "js",
        ignore: ['./node_modules/**']
    })
    .on('restart', function() {
        console.log("Chatbot restarting...");
    });
});
```

infopulse
A part of Nordic IT group EVRY

## i18n

- i18n
  - npm i --save i18n
  - App.js
    - var i18n = require('i18n');
    - i18n.configure({
    -     defaultLocale: process.env.DEFAULT_LOCALE ? process.env.DEFAULT_LOCALE : 'en',
    -     directory: __dirname + '/locales'
    - });
  - locales/en.json
  - en.json
    - {
    -               "greeting": "Hi! I'm your chatbot."
    - }

## Greeting



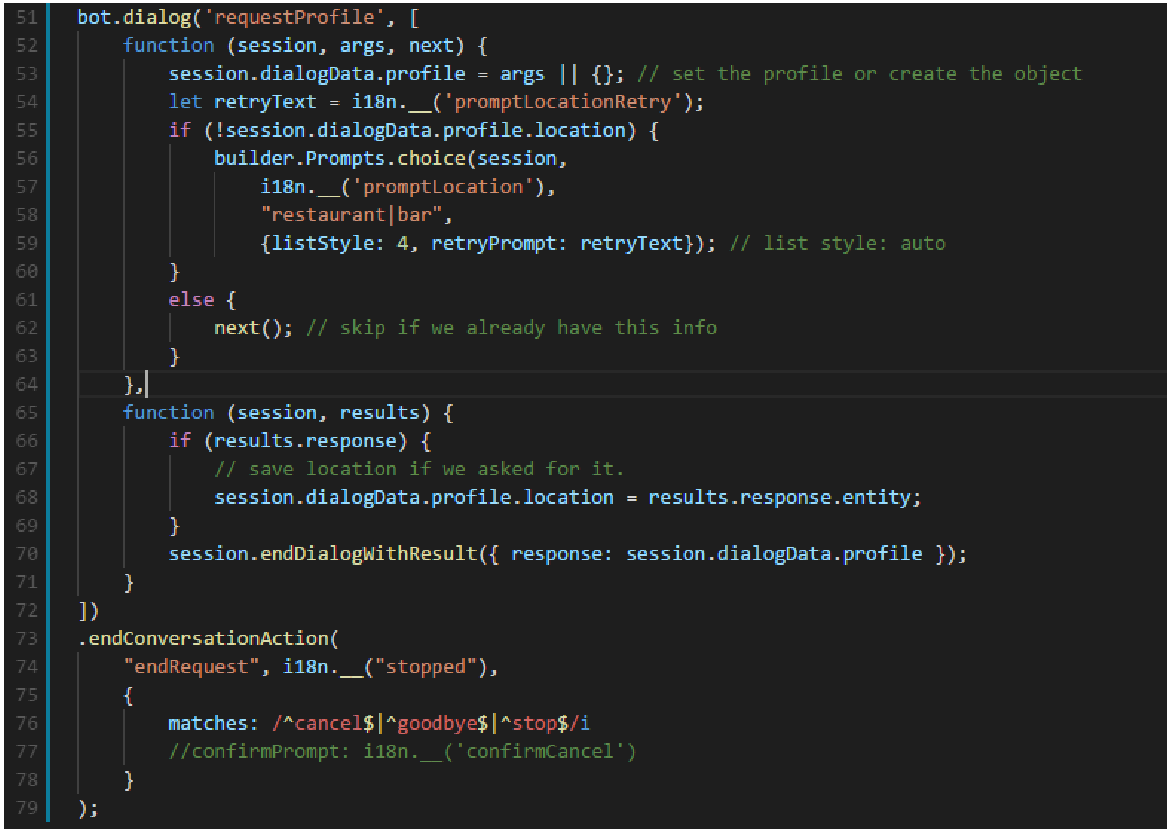
```
const greetingRegExp = RegExp('hei|hallo|hello|hi', "i");

function ParseInquire(inquire) {
    let profile = {};
    let greeting = greetingRegExp.exec(inquire);
    if (greeting) {
        profile.greeting = true;
    }
    return profile;
}

module.exports.ParseInquire = ParseInquire;
```

- Add parse-inquire.js
- Modify app.js
  - var parser = require('./parse-inquire');
  - …
  - var bot = new builder.UniversalBot(connector, function (session) {
  - session.userData.profile = parser.ParseInquire(session.message.text);
  - if (session.userData.profile.greeting) {
  - session.send(i18n.__("greeting"));
  - }
  - else {
  - session.send("You said: %s", session.message.text);
  - }
  - }).set('storage', inMemoryStorage);

infopulse

A part of Nordic IT group EVRY

# Prompt user input

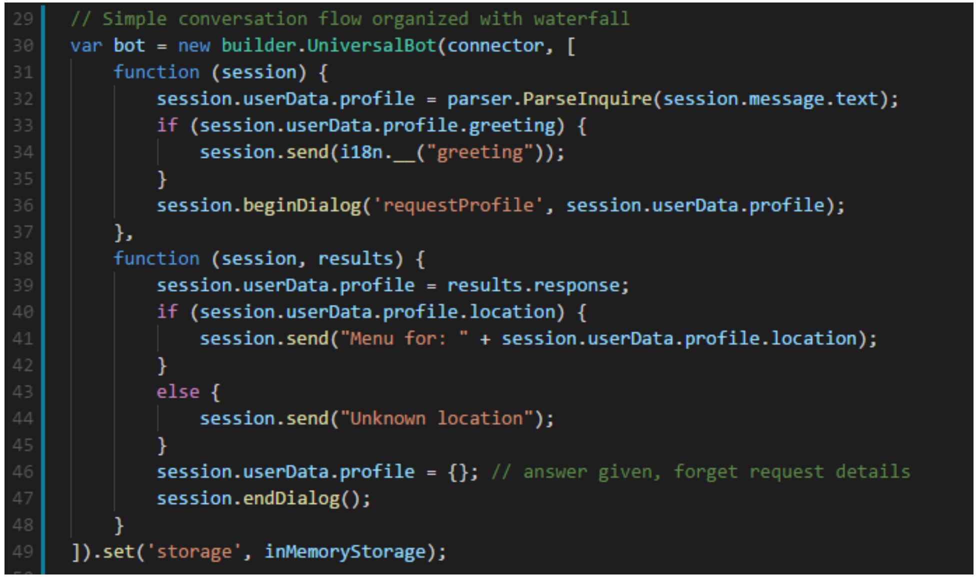


```
bot.dialog('requestProfile', [
    function (session, args, next) {
        session.dialogData.profile = args || {}; // set the profile or create the object
        let retryText = i18n.__('promptLocationRetry');
        if (!session.dialogData.profile.location) {
            builder.Prompts.choice(session,
                i18n.__('promptLocation'),
                "restaurant|bar",
                {listStyle: 4, retryPrompt: retryText}); // list style: auto
        }
        else {
            next(); // skip if we already have this info
        }
    },
    function (session, results) {
        if (results.response) {
            // save location if we asked for it.
            session.dialogData.profile.location = results.response.entity;
        }
        session.endDialogWithResult({ response: session.dialogData.profile });
    }
])
.endConversationAction(
    "endRequest", i18n.__("stopped"),
    {
        matches: /^cancel$|^goodbye$|^stop$/i
        //confirmPrompt: i18n.__('confirmCancel')
    }
);
```

infopulse
A part of Nordic IT group EVRY

## Conversation flow

- Manage conversation flow:
  - https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-dialog-manage-conversation-flow?view=azure-bot-service-3.0

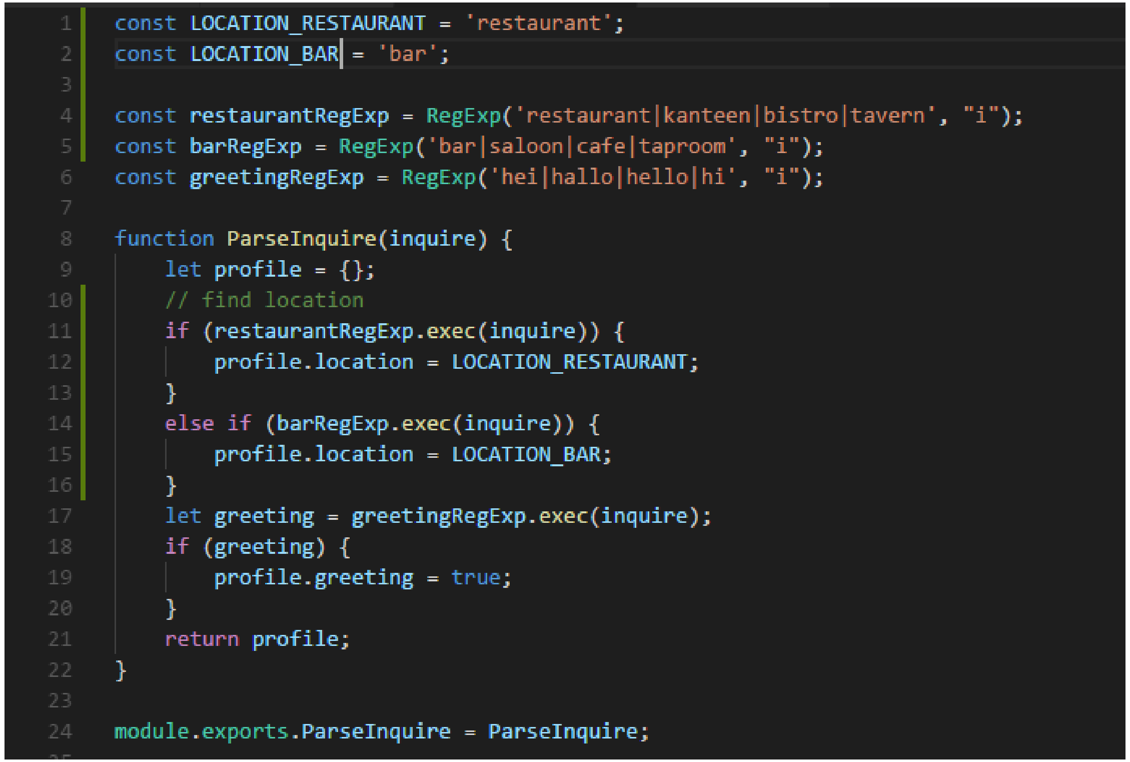


```
// Simple conversation flow organized with waterfall
var bot = new builder.UniversalBot(connector, [
    function (session) {
        session.userData.profile = parser.ParseInquire(session.message.text);
        if (session.userData.profile.greeting) {
            session.send(i18n.__("greeting"));
        }
        session.beginDialog('requestProfile', session.userData.profile);
    },
    function (session, results) {
        session.userData.profile = results.response;
        if (session.userData.profile.location) {
            session.send("Menu for: " + session.userData.profile.location);
        }
        else {
            session.send("Unknown location");
        }
        session.userData.profile = {}; // answer given, forget request details
        session.endDialog();
    }
]).set('storage', inMemoryStorage);
```

infopulse
A part of Nordic IT group EVRY

## Improved perception

- Modify parse-inquire.js



```
const LOCATION_RESTAURANT = 'restaurant';
const LOCATION_BAR = 'bar';

const restaurantRegExp = RegExp('restaurant|kanteen|bistro|tavern', "i");
const barRegExp = RegExp('bar|saloon|cafe|taproom', "i");
const greetingRegExp = RegExp('hei|hallo|hello|hi', "i");

function ParseInquire(inquire) {
    let profile = {};
    // find location
    if (restaurantRegExp.exec(inquire)) {
        profile.location = LOCATION_RESTAURANT;
    }
    else if (barRegExp.exec(inquire)) {
        profile.location = LOCATION_BAR;
    }
    let greeting = greetingRegExp.exec(inquire);
    if (greeting) {
        profile.greeting = true;
    }
    return profile;
}

module.exports.ParseInquire = ParseInquire;
```

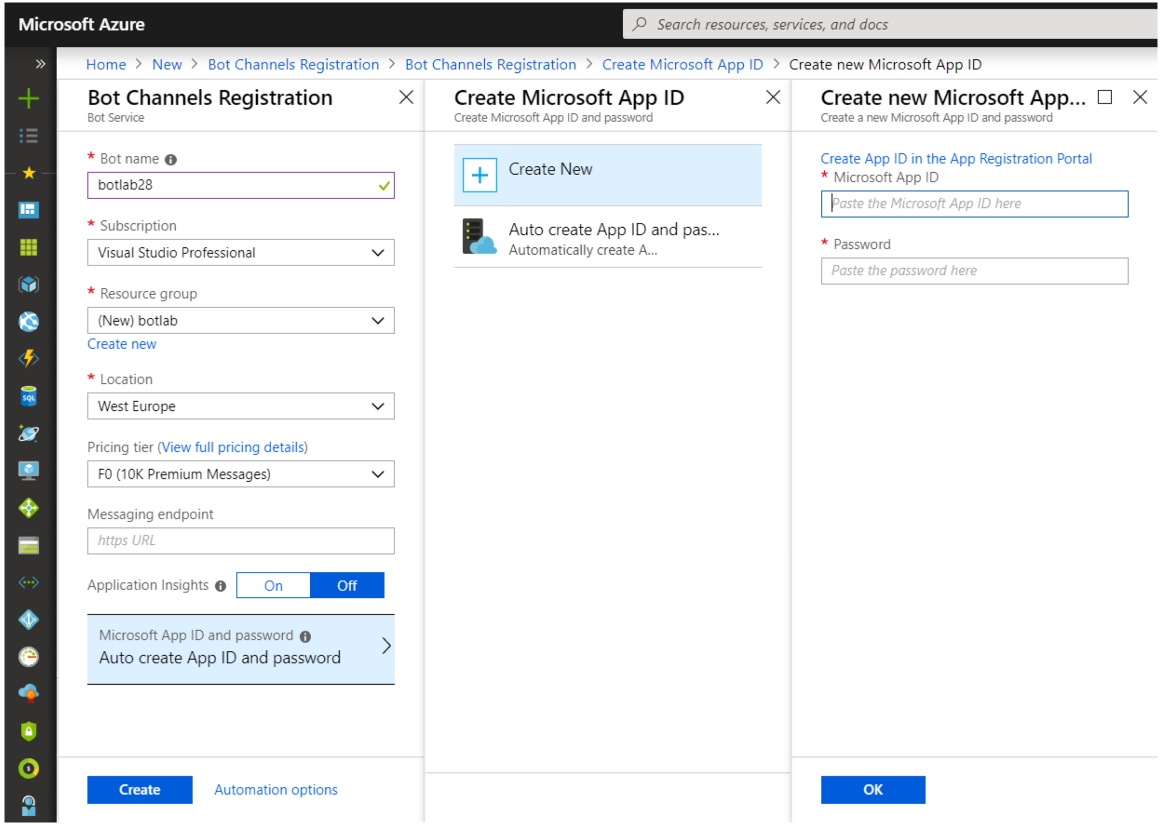
infopulse
A part of Nordic IT group EVRY

## Deploy

infopulse

# Publish Bot: Step 1

- Create .env file in your project
- Go to portal.azure.com
- Add Bot Channel Registration
  - Enter bot name
  - Create resource group
  - Choose location
  - Pricing: F0
  - Application Insights: off
  - Create App ID in the App Registration Portal
  - Copy App ID/Password to .env
  - Copy to App Id and Password fields and click OK
  - Click Create
- Wait for deployment completion


Microsoft Azure

Home > New > Bot Channels Registration > Bot Channels Registration > Create Microsoft App ID > Create new Microsoft App ID

**Bot Channels Registration**
Bot Service

- * Bot name: botlab28
- * Subscription: Visual Studio Professional
- * Resource group: (New) botlab — Create new
- * Location: West Europe
- Pricing tier (View full pricing details): F0 (10K Premium Messages)
- Messaging endpoint: https URL
- Application Insights: On [✓ Off]
- Microsoft App ID and password: Auto create App ID and password

Create | Automation options

**Create Microsoft App ID**
Create Microsoft App ID and password

- Create New
- Auto create App ID and pas... Automatically create A...

**Create new Microsoft App...**
Create a new Microsoft App ID and password

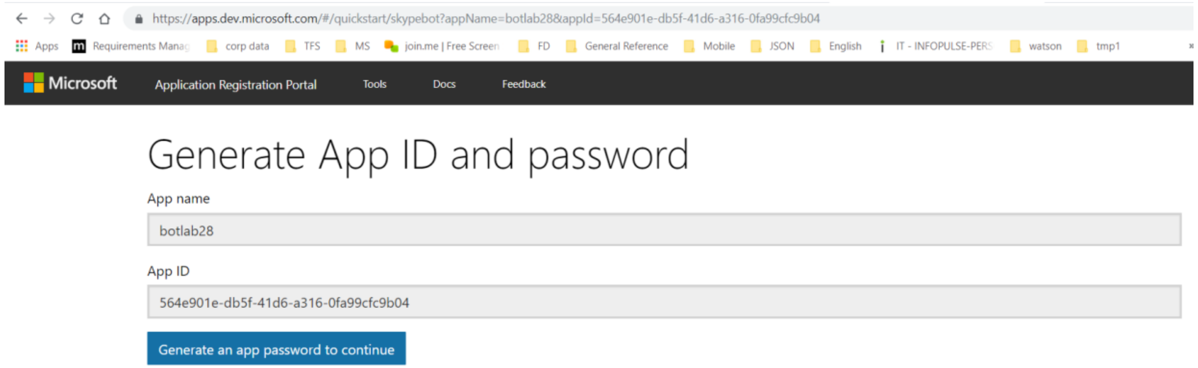
Create App ID in the App Registration Portal

- * Microsoft App ID: Paste the Microsoft App ID here
- * Password: Paste the password here

OK

infopulse
A part of Nordic IT group EVRY

## Application Registration


https://apps.dev.microsoft.com/#/quickstart/skypebot?appName=botlab28&appId=564e901e-db5f-41d6-a316-0fa99cfc9b04
Apps
Requirements Manag
corp data
TFS
MS
join.me | Free Screen
FD
General Reference
Mobile
JSON
English
IT - INFOPULSE-PERS
watson
tmp1
Microsoft
Application Registration Portal
Tools
Docs
Feedback
Generate an app password to continue

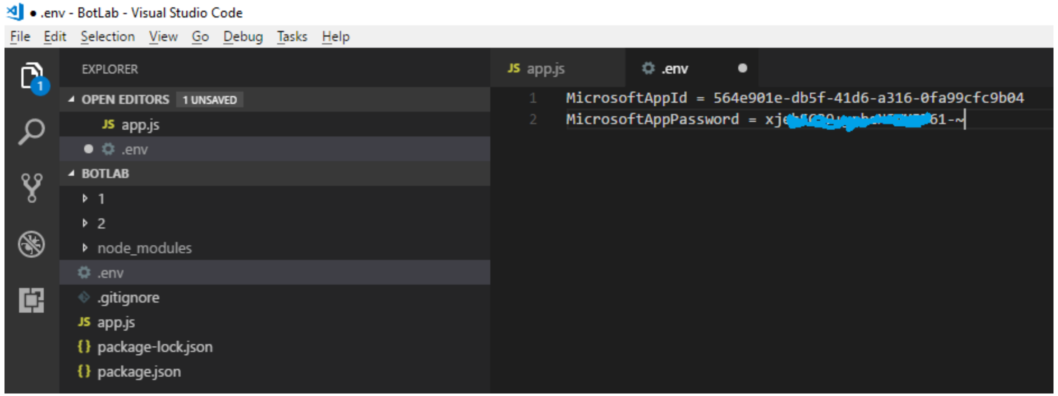
Generate App ID and password

App name

botlab28

App ID

564e901e-db5f-41d6-a316-0fa99cfc9b04



```
MicrosoftAppId = 564e901e-db5f-41d6-a316-0fa99cfc9b04
MicrosoftAppPassword = xj[redacted]61-~
```

infopulse

A part of Nordic IT group EVRY

## Publish Bot: Step 2

- Create application service (Web App)
    - App name – same as bot channel registration name (not necessary but convenient)
    - Use existing resource group – same as bot channel registration
    - OS: Linux
    - Runtime Stack: Node.js 8.11
    - Click ‘Create’ and wait for deployment…
- Configure web app settings
    - Deployment option -> GitHub
        - Choose project
        - Choose branch (master by default)
        - Click OK
    - Set App ID and Password
        - Click ‘Application Settings’
        - Startup File: app.js
        - Add new setting: MicrosoftAppId, MicrosoftAppPassword
        - Save and Restart

# Publish Bot: Step 3

- Test endpoint: https://botlab28.azurewebsites.net/api/messages
- Go to our bot channel registration
- Click ‘Settings’
- Set messaging endpoint:
  - https://botlab28.azurewebsites.net/api/messages
- Save
- Test in Web Chat

## Bonus

infopulse

## Setup mongodb

- Install mongodb instance
- Install “mongoose”
  - npm i --save mongoose
- Modify app.js
  - const mongoose = require('mongoose');
  - …
  - // mongodb connection
  - mongoose.connect(process.env.MONGO_URL || 'mongodb://localhost/menu');
  - let db = mongoose.connection;
  - db.on('error', console.error.bind(console, 'connection error:'));
  - db.once('open', function() {
  -     console.log("Menu DB connected.");
  - });
  -

## Model and Schema

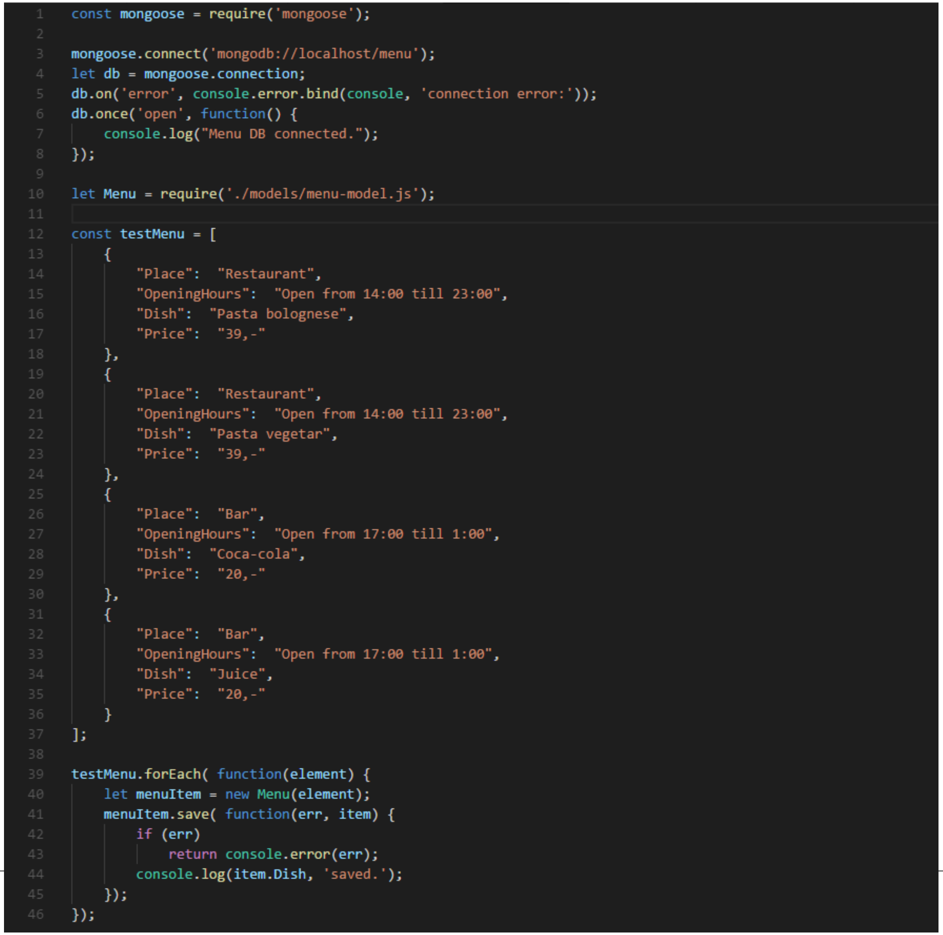
- Create models\menu-model.js
- menu-model.js
    - const mongoose = require('mongoose'),
    -     Schema = mongoose.Schema;

    - let menuModel = new Schema({
    -     Place: { type: String },
    -     OpeningHours: { type: String },
    -     Dish: { type: String },
    -     Price: { type: String }
    - });

    - module.exports = mongoose.model('Menu', menuModel);

## Init DB

- Init-db.js

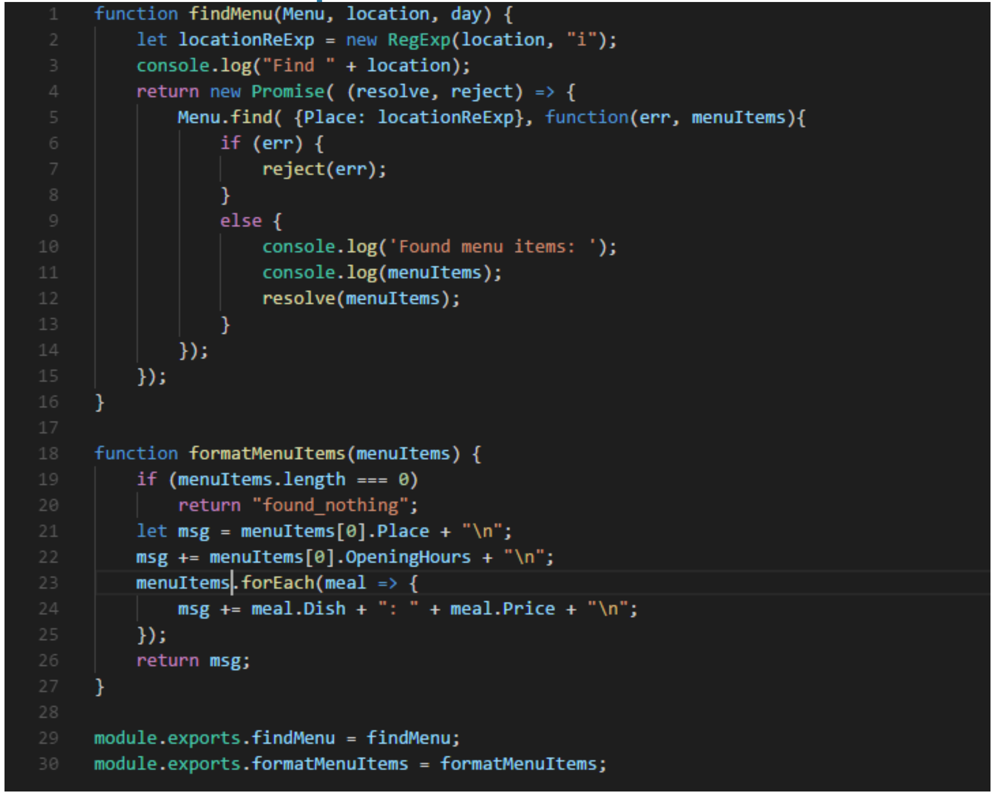


```
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost/menu');
let db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function() {
    console.log("Menu DB connected.");
});

let Menu = require('./models/menu-model.js');

const testMenu = [
    {
        "Place":  "Restaurant",
        "OpeningHours":  "Open from 14:00 till 23:00",
        "Dish":  "Pasta bolognese",
        "Price":  "39,-"
    },
    {
        "Place":  "Restaurant",
        "OpeningHours":  "Open from 14:00 till 23:00",
        "Dish":  "Pasta vegetar",
        "Price":  "39,-"
    },
    {
        "Place":  "Bar",
        "OpeningHours":  "Open from 17:00 till 1:00",
        "Dish":  "Coca-cola",
        "Price":  "20,-"
    },
    {
        "Place":  "Bar",
        "OpeningHours":  "Open from 17:00 till 1:00",
        "Dish":  "Juice",
        "Price":  "20,-"
    }
];

testMenu.forEach( function(element) {
    let menuItem = new Menu(element);
    menuItem.save( function(err, item) {
        if (err)
            return console.error(err);
        console.log(item.Dish, 'saved.');
    });
});
```

infopulse
A part of Nordic IT group EVRY

## Query DB and format output

- menu-queries.js



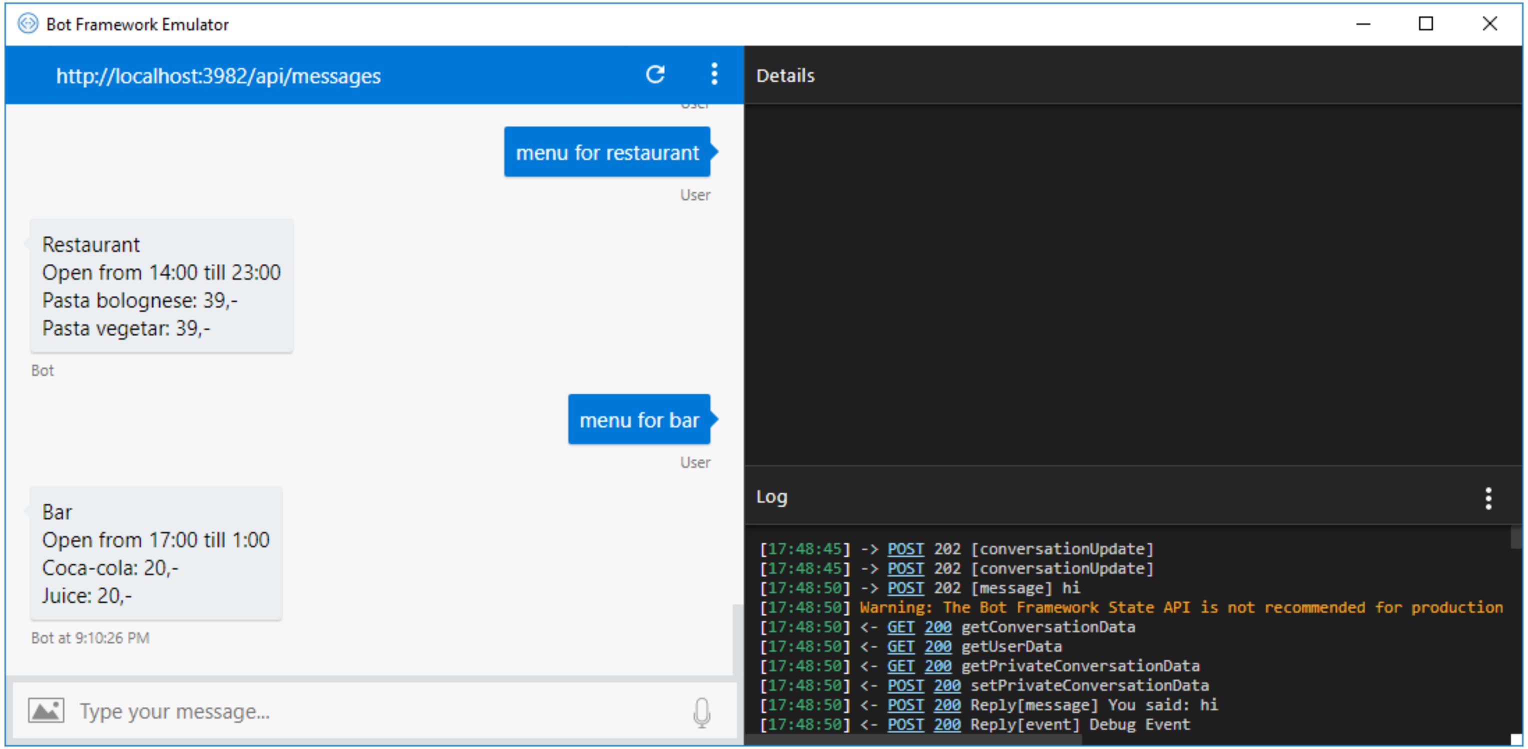
```
function findMenu(Menu, location, day) {
    let locationReExp = new RegExp(location, "i");
    console.log("Find " + location);
    return new Promise( (resolve, reject) => {
        Menu.find( {Place: locationReExp}, function(err, menuItems){
            if (err) {
                reject(err);
            }
            else {
                console.log('Found menu items: ');
                console.log(menuItems);
                resolve(menuItems);
            }
        });
    });
}

function formatMenuItems(menuItems) {
    if (menuItems.length === 0)
        return "found_nothing";
    let msg = menuItems[0].Place + "\n";
    msg += menuItems[0].OpeningHours + "\n";
    menuItems.forEach(meal => {
        msg += meal.Dish + ": " + meal.Price + "\n";
    });
    return msg;
}

module.exports.findMenu = findMenu;
module.exports.formatMenuItems = formatMenuItems;
```

infopulse
A part of Nordic IT group EVRY

## Query menu information

- Modify app.js
  - const Menu = require('./models/menu-model.js');
  - const queries = require('./queries/menu-queries');
  - …
  - if (session.userData.profile.location) {
  -         //session.send("Menu for: " + session.userData.profile.location);
  -         queries.findMenu(Menu, session.userData.profile.location)
  -         .then( (result) => {
  -            session.send(queries.formatMenuItems(result));
  -         })
  -         .catch( (err) => {
  -            session.send(err)
  -         });
  -      }

## Demo



menu for restaurant

User

Restaurant
Open from 14:00 till 23:00
Pasta bolognese: 39,-
Pasta vegetar: 39,-

Bot

menu for bar

User

Bar
Open from 17:00 till 1:00
Coca-cola: 20,-
Juice: 20,-

Bot at 9:10:26 PM

```
[17:48:45] -> POST 202 [conversationUpdate]
[17:48:45] -> POST 202 [conversationUpdate]
[17:48:50] -> POST 202 [message] hi
[17:48:50] Warning: The Bot Framework State API is not recommended for production
[17:48:50] <- GET 200 getConversationData
[17:48:50] <- GET 200 getUserData
[17:48:50] <- GET 200 getPrivateConversationData
[17:48:50] <- POST 200 setPrivateConversationData
[17:48:50] <- POST 200 Reply[message] You said: hi
[17:48:50] <- POST 200 Reply[event] Debug Event
```

infopulse

Дякуємо за увагу!

infopulse

## Наші контакти:

+38 044 585-25-00

www.infopulse.com

Вул. Польова, 24, Київ, Україна, 03056

info@infopulse.com

/InfopulseGlobal

/InfopulseGlobal

/company/infopulse