



# Интеллектуальные чат-боты и КОГНИТИВНЫЕ сервисы

**Дмитрий Сошников**

Технологический евангелист, Microsoft

[dmitri@soshnikov.com](mailto:dmitri@soshnikov.com)

[twitter.com/shwars](https://twitter.com/shwars)

DOT  
NEXT

**F# → AI**

Задача Microsoft – демократизация искусственного интеллекта...





LOG IN | SIGN UP

[TECH](#) [SCIENCE](#) [CULTURE](#) [CARS](#) [REVIEWS](#) [LONGFORM](#) [VIDEO](#) [CIRCUIT BREAKER](#) [TL;DR](#) [FORUMS](#)



PREVIOUS STORY

Just like dogs, goats will look you in the eye when they need your help

NEXT STORY

Bluetooth headphones are annoying

[LONGFORM](#) [MICROSOFT](#) [TECH](#) [EXCLUSIVE](#) [FEATURED STORY](#) [FEATURES](#)

114  
COMMENTS

# Exclusive: Why Microsoft is betting its future on AI

*Inside Satya Nadella's plan to outsmart Google*

By [Casey Newton](#) on July 7, 2016 08:00 am [Email](#) [@CaseyNewton](#)



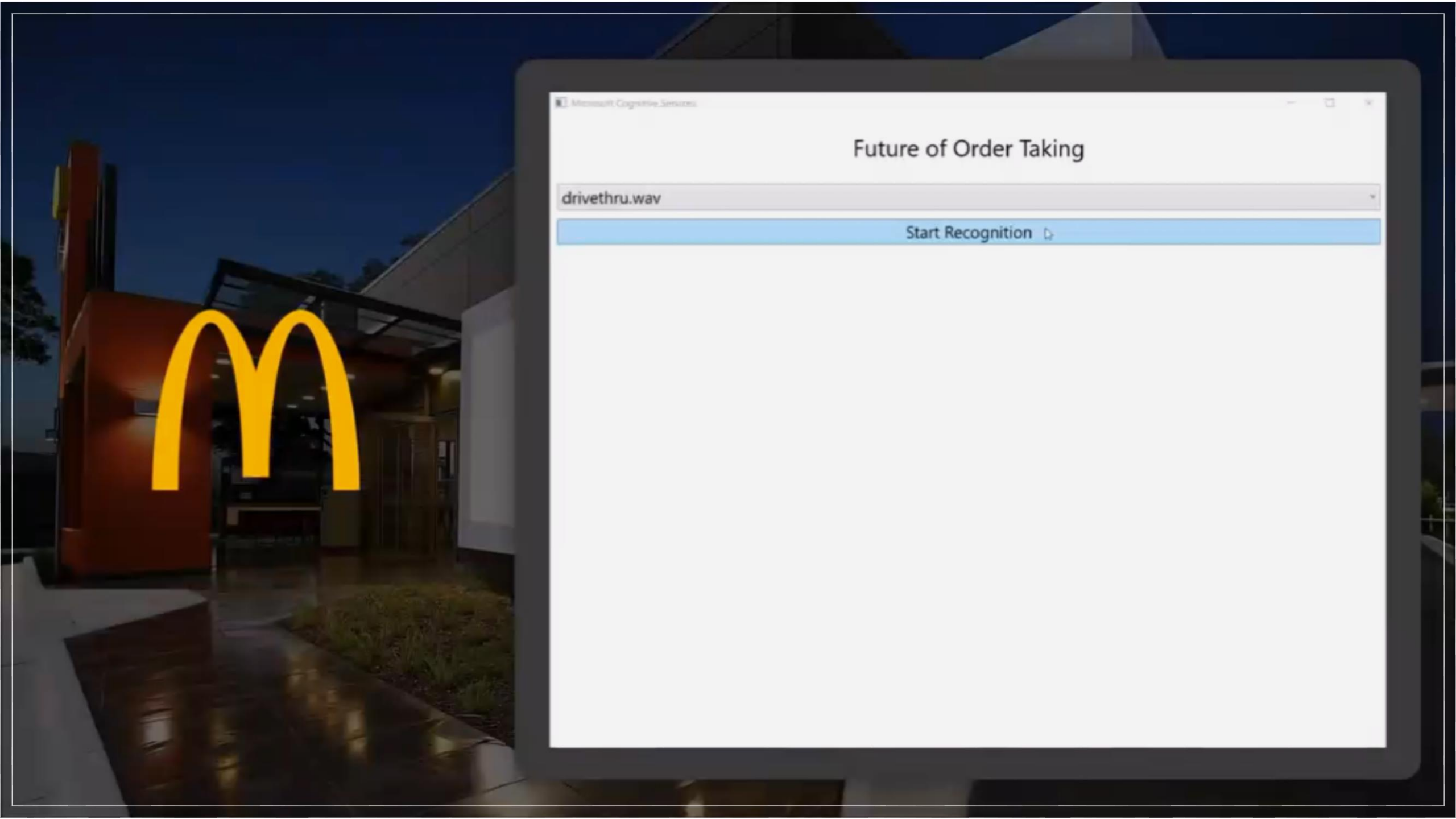
# Искусственный интеллект, факультет Прикладной математики МАИ



## Future of Order Taking

drivethru.wav

Start Recognition ▶

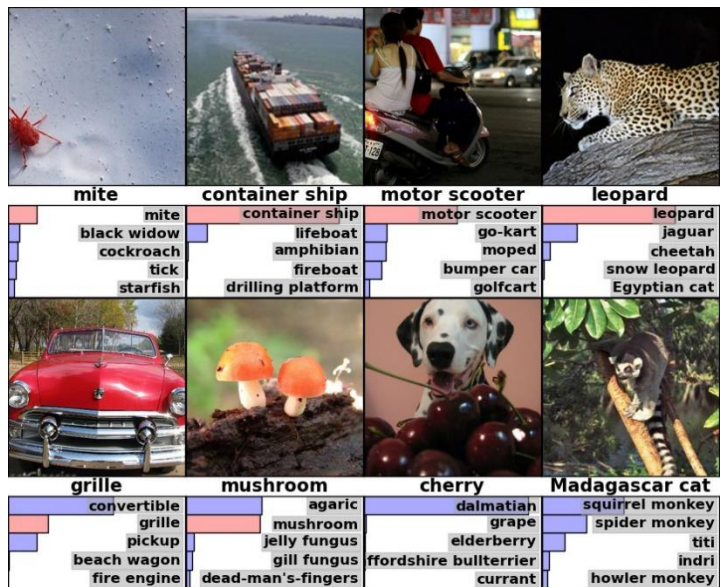




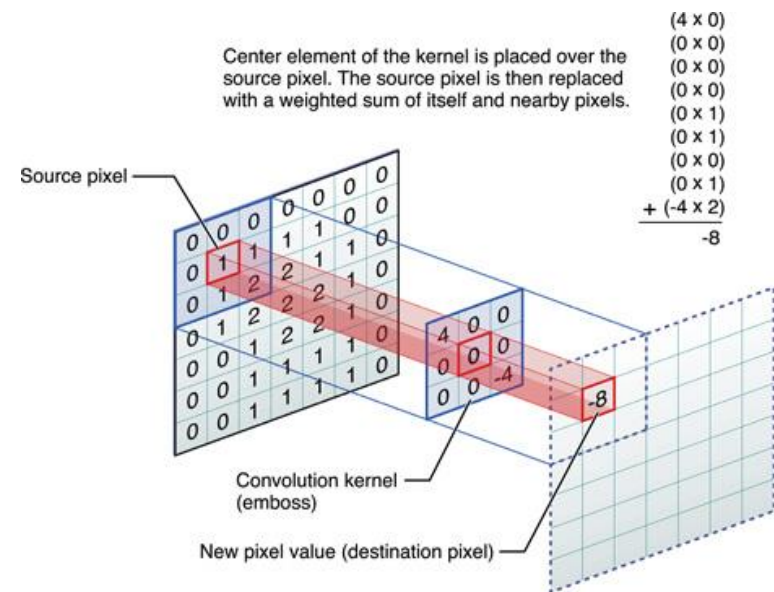
# Революция в области ИИ



Доступные  
вычислительные  
мощности

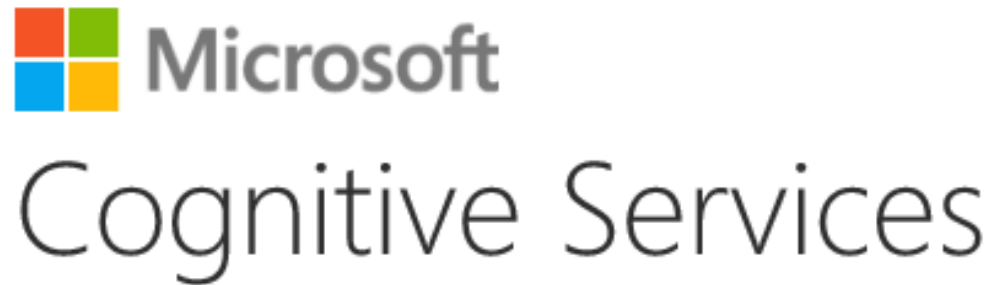


Открытые  
массивы  
данных



Глубинное  
обучение,  
сверточные сети

# Демократизация искусственного интеллекта



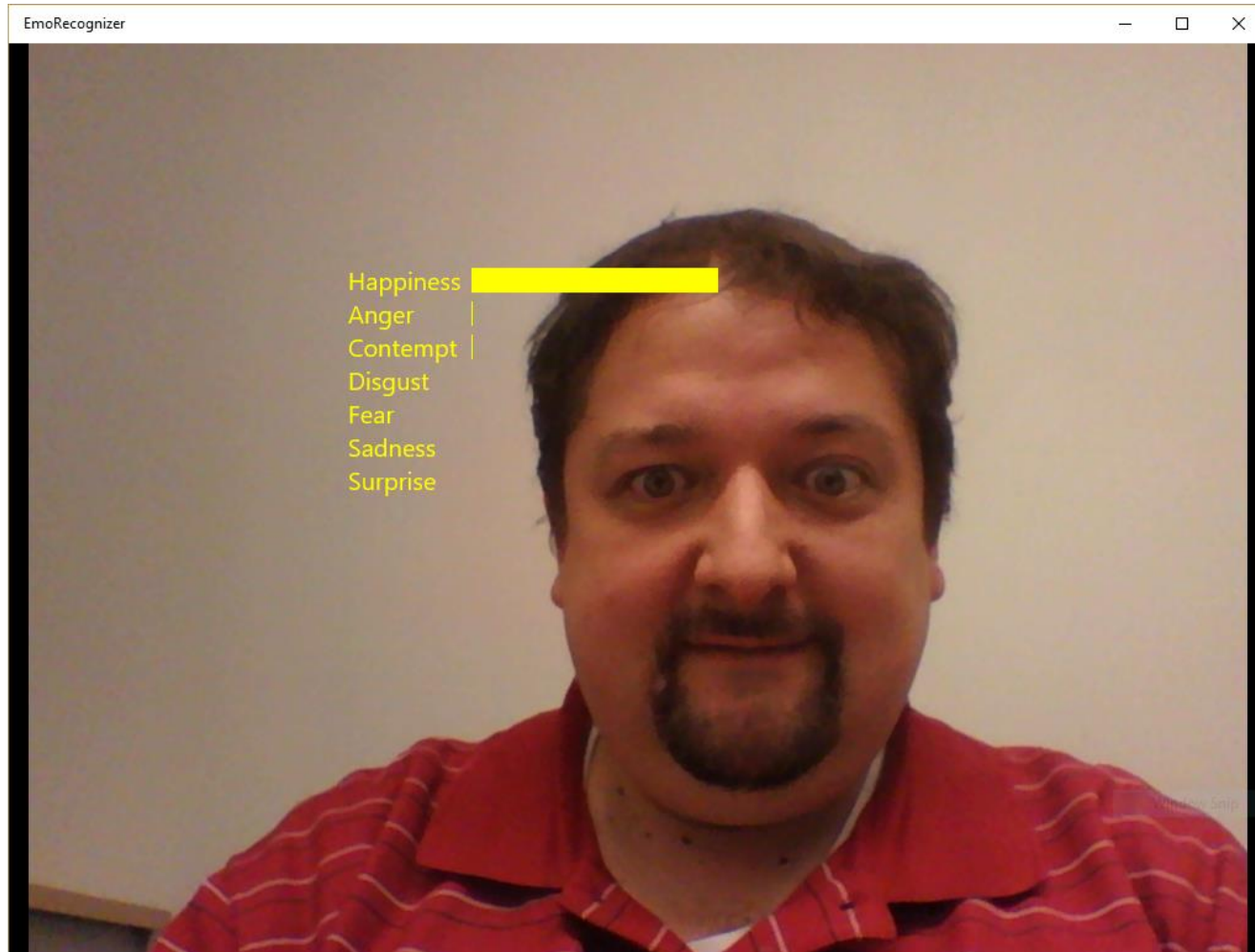
Bot Framework





# Microsoft Cognitive Services

<https://www.microsoft.com/cognitive-services/>



# Microsoft Cognitive Services

<http://microsoft.com/cognitive>

REST, C#, Node.js, ...

## Cognitive Services API



### Vision

Computer Vision | Emotion | Face | Video



### Speech

Bing Speech | Custom Recognition | Speaker Recognition



### Language

Bing Spell Check | Language Understanding  
Linguistic Analysis | Text Analytics | Translator |  
Web Language Model



### Knowledge

Academic Knowledge | Entity Linking | Knowledge  
Exploration | Recommendations



### Search

Bing Autosuggest | Bing Image Search | Bing  
News Search | Bing Video Search | Bing Web  
Search

# Где смотреть примеры и как учиться

- <http://microsoft.com/cognitive>
  - Прекрасные API Console и документация
- <http://github.com/evangelism/ModernAI>
  - Боты
  - Примеры на C#
  - Примеры на Python (Notebooks)
- Курс на Coursera – начало 2017 г.



# Vision API



a group of people sitting at a table

table, indoor, person, sitting, child, food, group, eating, laptop, computer, woman, people, front, small, boy, young, little, cake, man, desk, baby, room, plate, white



a man in a suit and tie

person, building, outdoor, man, cake, holding, carrying, standing, street, wearing, walking, young, food, woman, table, large, suit, city, people, riding

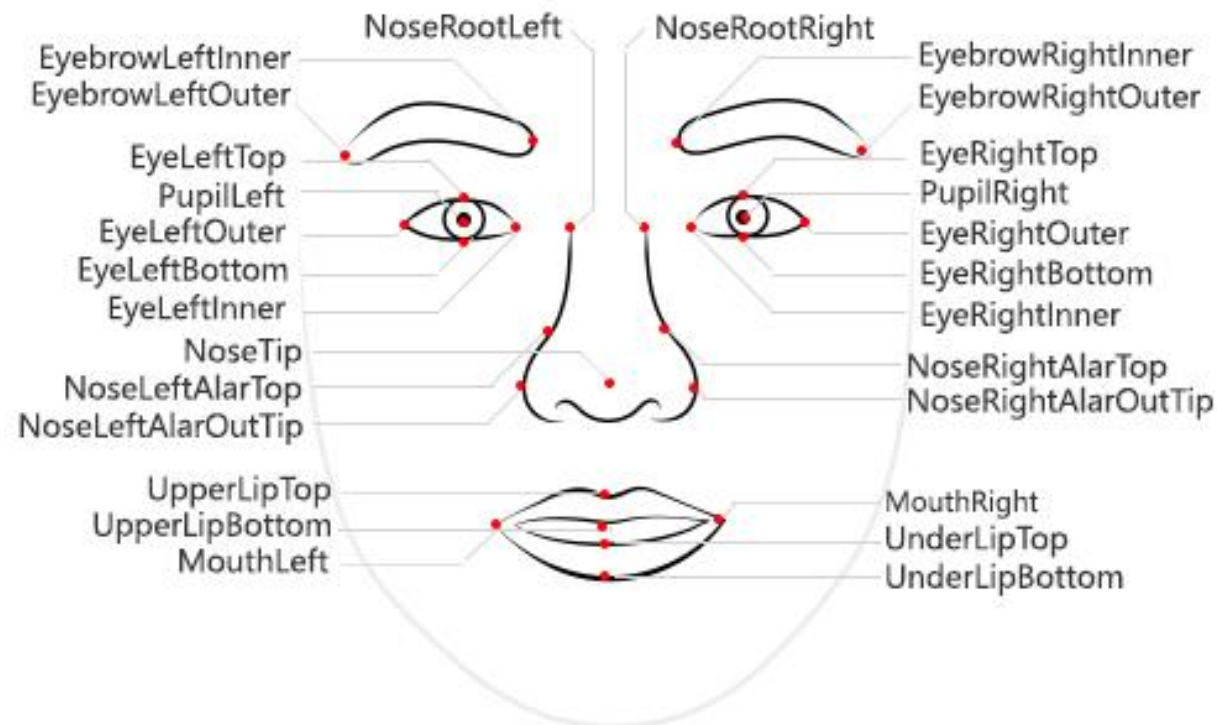
 Демонстрация

# Computer Vision API

<https://github.com/microsoft/cognitive-vision-windows>

# Face API

- Face Recognition (до 64 лиц, face rect + face landmarks)
  - Основной метод, распознаёт лицо и возвращает FaceID (Guid)
- Поиск похожих лиц (Find Similar)
  - Использует именованный (заранее созданный) список лиц или временный (до 24 часов)
  - Для создания списка лиц используются методы CreateFaceList
- Распознавание человека по фотографии (Identify)
  - Использует список групп лиц, каждая группа-на одного человека
  - Для работы с группами лиц существуют специальные методы
- Верификация (Verify)
- Группировка лиц (Group Faces)



Copyright (c) Microsoft. All rights reserved

**Microsoft.ProjectOxford.Face**



 Демонстрация

# Face API Console

<https://dev.projectoxford.ai/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395236>

# Emotion API

Обнаружение базовых эмоций на фотографии

- Anger, Contempt, Disgust, Fear, Happiness, Neutral, Sadness, Surprise
- Также может обнаружить FaceRect

Обработка видео

- Загружаем видео
- Через какое-то время получаем результат обработки
- Усредненный или по кадрам

**Microsoft.ProjectOxford.Emotion**

 Демонстрация

Angry Ballmer, Happy Gates...

# Microsoft Cognitive Services

<http://microsoft.com/cognitive>

REST, C#, Node.js, ...

## Cognitive Services API



### Vision

Computer Vision | Emotion | Face | Video



### Speech

Bing Speech | Custom Recognition | Speaker Recognition



### Language

Bing Spell Check | Language Understanding  
Linguistic Analysis | Text Analytics | Translator |  
Web Language Model



### Knowledge

Academic Knowledge | Entity Linking | Knowledge  
Exploration | Recommendations



### Search

Bing Autosuggest | Bing Image Search | Bing  
News Search | Bing Video Search | Bing Web  
Search

**Эпизод II:**

Боты и Microsoft Bot Framework

**Загадка:**

Почему боты популярны?



**Бот:** Здравствуйте! Вы можете заказать пиццу!

**Я:** Прекрасно, хочу!

**Бот:** Скажите «заказать» чтобы приступить к заказу.

**Я:** Заказать.

**Бот:** Спасибо! Вы хотите закрытую или открытую пиццу?

**Я:** Открытую.

**Бот:** Я вас не понял. Скажите «открытую» (или 1) или «закрытую» (2)

**Я:** 1

**Бот:** Отлично! На тонком или толстом тесте?

**Я:** На тонком.

**Бот:** Вы хотите пиццу пепперони, капрезе, кваттро стаджони? Напишите «состав пепперони», чтобы узнать подробнее состав пиццы.

**Я:** пепперони.

**Бот:** А не хотите ли вместо обычного теста сделать слоёное?

...



 <p>САЛЬМОН ДИ ГАРДЕН</p> <p>Цена за блюдо</p> <p><b>1 260</b> руб.</p> <p><a href="#">Заказать в 1 клик</a></p> <p> <b>Заказать</b></p>	 <p>СУПЕР ПЕППЕРОНИ</p> <p>Цена за блюдо</p> <p><b>950</b> руб.</p> <p><a href="#">Заказать в 1 клик</a></p> <p> <b>Заказать</b></p>	 <p>JACK'S СПЕШЛ</p> <p>Цена за блюдо</p> <p><b>1 050</b> руб.</p> <p><a href="#">Заказать в 1 клик</a></p> <p> <b>Заказать</b></p>
		

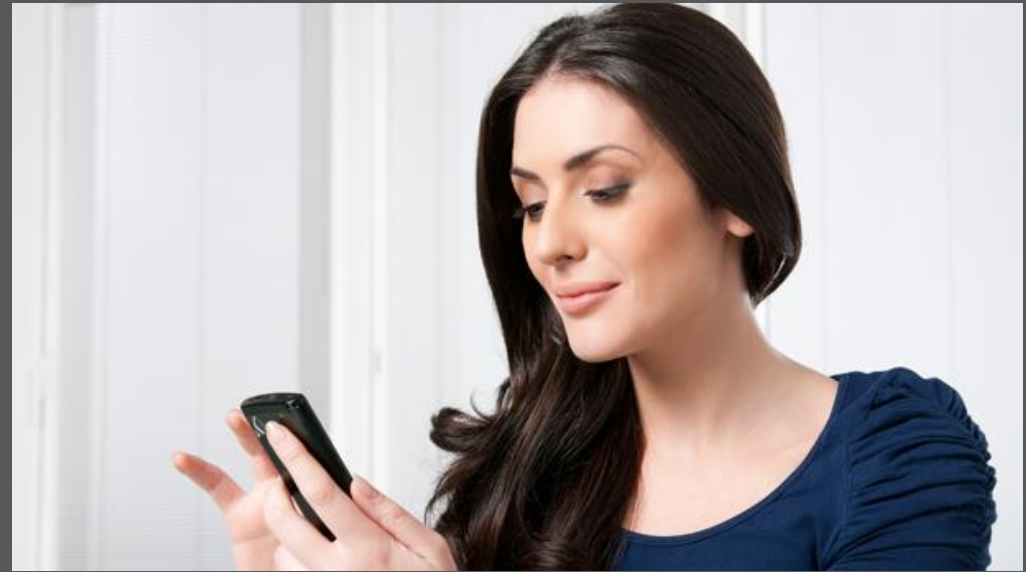
# Примеры хороших сценариев:

1. Групповое взаимодействие  
(аугментирование, человек-компьютер)
2. Голосовой диалог
3. Personal Assistant
4. Существующие бизнес-процессы
5. ... ? :(

# Боты как замена приложениям



Проще разрабатывать  
(нет дизайна)



Проще использовать  
(нет инсталляции)

# Microsoft Bot Framework

<http://botframework.com>

 Цели

Написал  
однажды —  
чатишься  
везде!

---

Бот создается как некоторый  
код в облаке, далее к нему  
подключаются разные  
каналы общения

Удобные  
абстракции

---

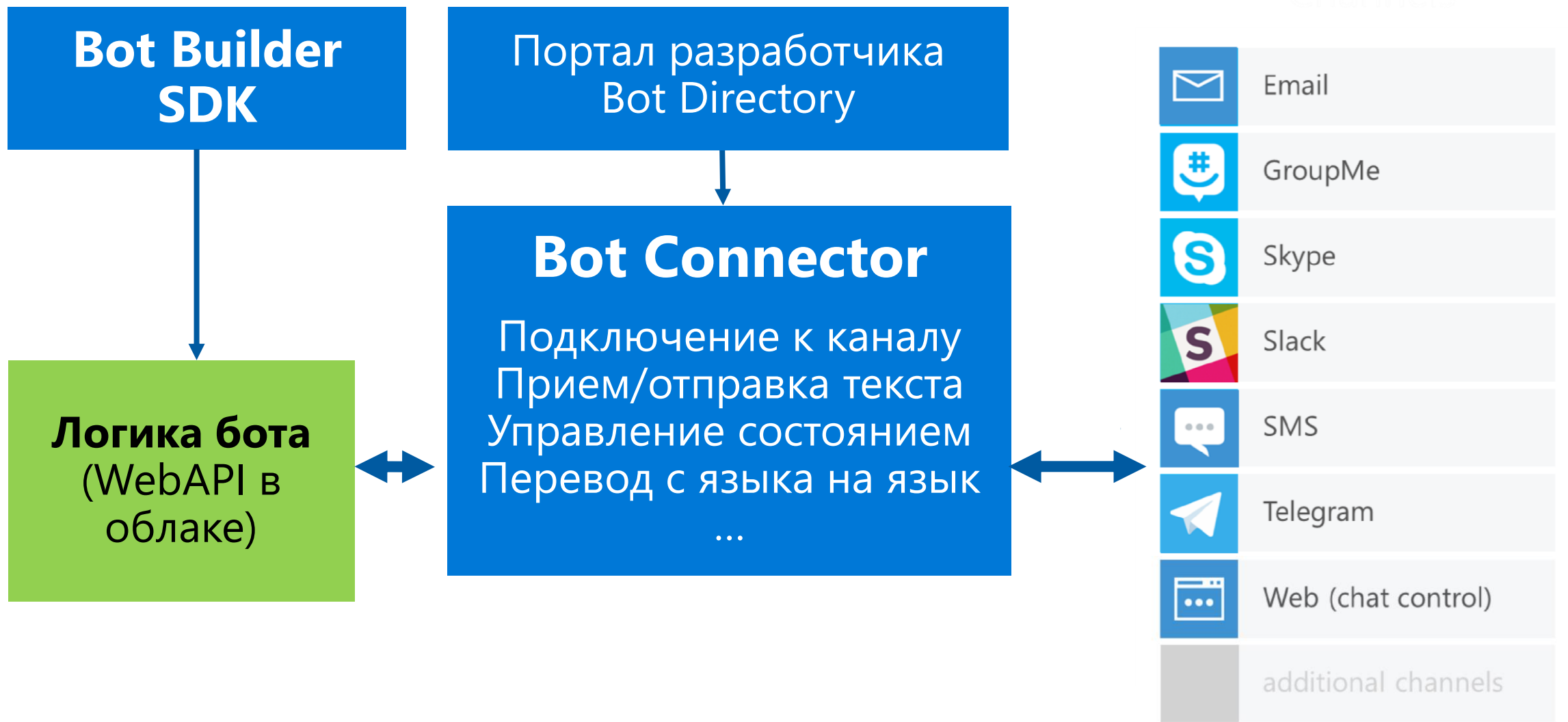
Диалоги, композиции  
диалогов, упрощение  
асинхронного  
взаимодействия

Мощные  
инструменты

---

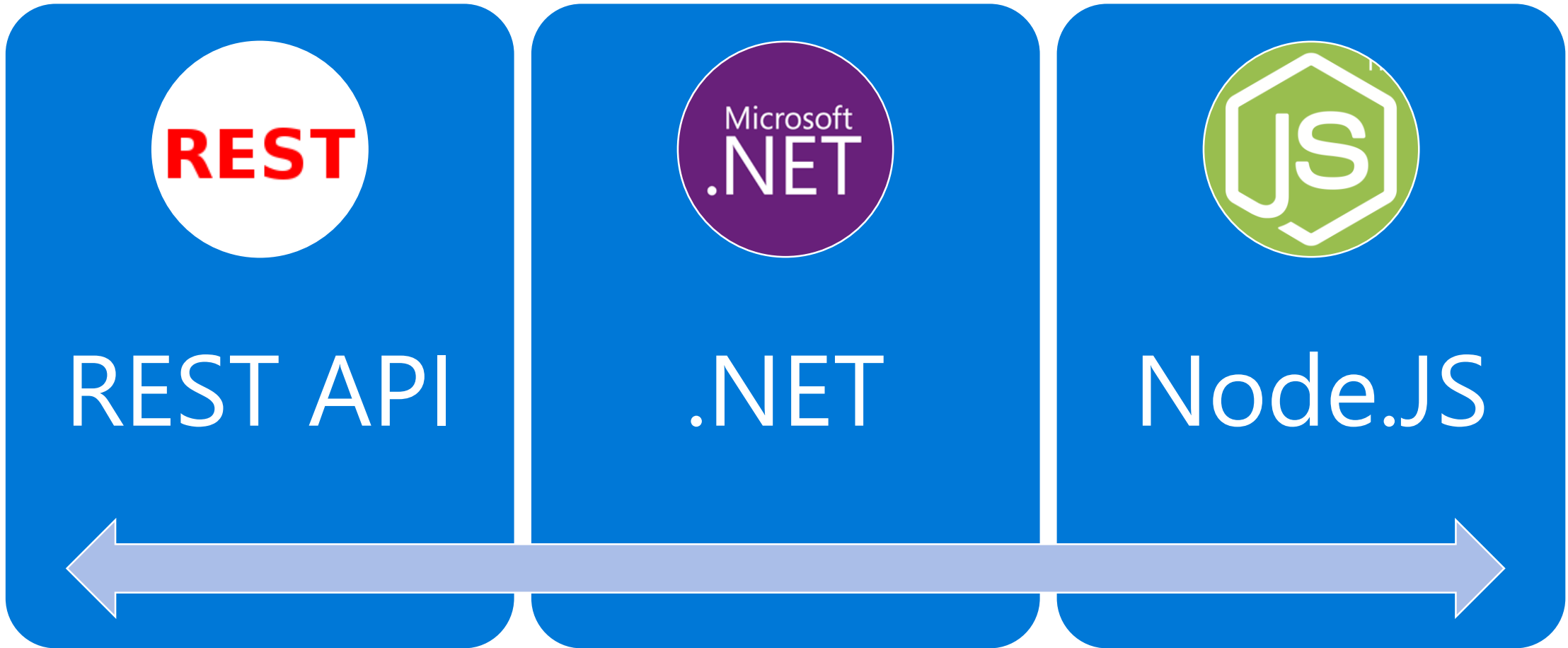
FormFlow, LUIS, ...

# Microsoft Bot Framework





# Поддерживаемые языки/технологии



<https://docs.botframework.com/>

# Что необходимо

## Visual Studio 2015 Update 3

С обновлёнными обновлениями

## Bot Application Template

<http://aka.ms/bf-bc-vstemplate>

Установить в %USERPROFILE%\Documents\Visual Studio 2015\Templates\ProjectTemplates\Visual C#

## Bot Framework Emulator

<https://aka.ms/bf-bc-emulator>

<https://github.com/evangelism/ModernAI/tree/master/SimpleCommandBot>

 Демонстрация

# С чего начать

<https://github.com/evangelism/ModernAI>

**SimpleCommandBot**

# Боты не имеют состояния

Бот – это Web API

Это хорошо, поскольку позволяет легко масштабировать бота в случае большой нагрузки

Для поддержки состояния нужно использовать механизм Cookies

Bot Framework берет это на себя!

# Идентификаторы в Activity

<b>Свойство</b>	<b>Описание</b>
From	Адрес пользователя для данного канала связи
Conversation	Идентификатор беседы
From + Conversation	Уникальный контекст для пользователя в беседе

# Сервис СОСТОЯНИЯ

```
var State = activity.GetStateClient();  
var UserData = await State.BotState.GetUserDataAsync(activity.ChannelId, activity.From.Id);  
var x = UserData.GetProperty<string>("name");  
  
UserData.SetProperty<string>("name", "John Doe");  
await State.BotState.SetUserDataAsync(activity.ChannelId, activity.From.Id, UserData);
```

<b>СВОЙСТВО</b>	<b>Get (+Async)</b>	<b>Set (+Async)</b>
From	GetUserData	SetUserData
Conversation	GetConversationData	SetConversationData
From + Conversation	GetPrivateConversationData	SetPrivateConversationData



# Bot Builder SDK

Дополнительные классы для упрощения построения ботов

Открытый исходный код на

<http://github.com/Microsoft/botbuilder>

Установка через NuGet: Microsoft.Bot.Builder  
или в составе шаблона

# Использование Bot Framework Dialogs

Диалог – это фрагмент беседы

Диалоги можно объединять между собой в более сложные беседы

Как правило, диалог ведет к какому-то результату, представляющемуся объектом

Диалог описывается интерфейсом  
`IDialog<T>`

Диалог может иметь внутреннее состояние

Все классы должны быть `[Serializable]`

 Демонстрация

# AgeBot

<https://github.com/evangelism/ModernAISamples> / **AgeBot**

# Объединение диалогов

В стиле LINQ

```
var query = from x in new PromptDialog.PromptString("String 1", "p1", 1)
            from y in new PromptDialog.PromptString("String 2", "p2", 1)
            select string.Join(" ", x, y);
```

В явном виде

```
await Conversation.SendAsync(activity, () =>
    new WeatherDialog()
        .ContinueWith<WeatherParam, string>(async (ctx, wpa) =>
            {
                var WP = await wpa;
                return new ChoiceDialog($"Subscribe?", new string[] { "Yes", "No" });
            })
        .ContinueWith<string, string>(async (ctx, aa) =>
            {
                var a = await aa;
                if (a.ToLower() == "yes") return new StringDialog("What is your name?");
            }
        ));
```

# FormFlow

```
[Serializable]
```

```
public class WeatherParam
```

```
{
```

```
    public static IForm<WeatherParam> BuildForm()
```

```
    {
```

```
        return new FormBuilder<WeatherParam>()
```

```
            .Message("Welcome to weather bot")
```

```
            .Build();
```

```
    }
```

```
    [Prompt("Which city do you want to know the weather in?")]
```

```
    public string Location { get; set; }
```

```
    public DateTime When { get; set; }
```

```
    public Measurement MeasurementType { get; set; }
```

```
}
```

# Ручное создание формы

```
return new FormBuilder<SandwichOrder>()
    .Message("Welcome to the sandwich order bot!")
    .Field(nameof(Sandwich))
    .Field(nameof(Toppings))
    .Message("For sandwich you have selected {Toppings}.")
    .Confirm(async (state) =>
        {
            var cost = CalculateCost(this);
            return new PromptAttribute($"Total for your sandwich is ${cost:F2}, ok?");
        })
    .AddRemainingFields()
    .Message("Thanks for ordering a sandwich!")
    .OnCompletionAsync(processOrder)
    .Build();
```

 Демонстрация

# Понимание естественного языка: LUIS

<https://github.com/evangelism/ModernAISamples> / **AgeBot.LUIS**

# Основные шаги по публикации бота

Отлаживаем бота локально в эмуляторе

Публикуем бота в облако как Azure App Service

Регистрируем бота в <http://botframework.com>

Обновляем App Key и App Secret в Web.Config и пере-публикуем в облако

Проверяем работу удалённо в эмуляторе

Регистрируем каналы связи на <http://botframework.com>



**1** Искусственный интеллект стремительно развивается и теперь доступен всем

**2** Чат-боты – это следующее поколение приложений. Bot Framework облегчает их создание

**3** Microsoft сегодня – это не про Windows, а про цифровую трансформацию бизнеса с помощью технологий

🔑 Q&A

# Чат-боты и когнитивные сервисы

Дмитрий Сошников

<http://twitter.com/shwars>

