Team care

Corporate Analysis Re-Engineering

ASP.NET とASP.NET Core

ASP.NET とASP.NET Core それぞれの概要、違い、開発環境等を調査してまとめる。

ASP.NET 概要

ASP.NET は、HTML、CSS、および JavaScript を使用して優れた web サイトや web アプリケーションを構築するための無料の web フレームワークです。Web API を作成し、Web ソケットなどのリアルタイム テクノロジを使用できます。

ASP.NET Core 概要

ASP.NET Core は、インターネットに接続された最新のクラウド対応アプリを構築するための、クロス プラットフォームで高パフォーマンスのオープン ソース フレームワークです。ASP.NET は、Windows以外のOSでは開発することができないという問題点がありました。その点を解決すべく、マルチプラットフォームに対応したASP.NET Core が登場しました。ASP.NET CoreはASP.NET 4.xを再設計したものです。

それぞれの違い

	ASP.NET 4.X	ASP.NET CORE
利用できるOS	Windowsのみ	Windows、macOS、Linux
利用できる言語	C#、VB、F#	C#、F#
利用できる統合開発環境	Visual Studio	Visual Studio Visual Studio for Mac Visual Studio Code
対応している機能	Webフォーム、MVC、SignalR、Web API	MVC、SignalR、Web API
サーバーアプリケーション フレームワーク	.NET Framework	.NET Core

⊠ Note

ASP.NET CoreはWebフォームに対応していません。Webフォームでは、IISというWindows用のWebサーバーを使用しているためです。

⊠ Warning

ASP.NET 4.xのすべてのアプリがASP.NET Coreで動くというわけではないので注意が必要です。

サーバーアプリケーション フレームワーク

.NET Core を選択する場合

1. クロスプラット フォームが必要である。
2. マイクロサービスが対象である。
3. Docker コンテナーを使用している。
4. 高パフォーマンスでスケーラブルなシステムが必要である。

5. １つのアプリケーションに複数の .NET バージョンが必要である。

.NET Framework を選択する場合

1. 現在、アプリで .NET Framework を使用している。
2. アプリが .NET Core で使用できないサードパーティ製の .NET ライブラリや NuGet パッケージを使用している。
3. アプリで、.NET Core で使用できない .NET テクノロジを使用している。
4. アプリで、.NET Core をサポートしていないプラットフォームを使用している。

参考文献

- ASP.NET 概要
- ASP.NET Core の概要
- ASP.NET 4.x と ASP.NET Core の選択
- サーバー アプリでの .NET Core または .NET Framework の使用
- ASP.NET Core を使ってできることとは？

DocFXを使用したドキュメントの作成

DocFXを使用したドキュメントの作成方法について記載する。

環境構築

Windows PC

- .NET Core SDK 3.1
  - dotnet CLIを使用するために必要

- Build Tools for Visual Studio 2019
  - DocFXのビルドに必要
  - 起動後に以下のコンポーネントをインストール
    - .NET Core 3.1 LTS ランタイム
    - .NET Core SDK

- DocFX
  - ドキュメントの作成に必要

  - windowsならchocoでインストールできる

    - PowerShellを管理者権限で起動して以下のコマンドでインストール

      ```
      choco install docfx -y
      ```

- NuGet
  - DocFXのプラグインをインストールするために必要

- GraphViz
  - DocFXのプラグインでPlant-umlの図を描画するために必要

  - windowsならchocoでインストールできる

    - PowerShellを管理者権限で起動して以下のコマンドでインストール

      ```
      choco install graphviz -y
      ```

- Node.js
  - DocFXのビルドに必要

- wkhtmltopdf
  - ドキュメントのPDF化に必要

  - windowsならchocoでインストールできる

    - PowerShellを管理者権限で起動して以下のコマンドでインストール

      ```
      choco install wkhtmltopdf -y
      ```

Dockerで環境構築

- 下記のDockerfileをビルドする。

```
FROM ubuntu:18.04
ENV LANG ja_JP.UTF-8
ENV PATH /docfx:${PATH}
ENV DEBIAN_FRONTEND noninteractive
ENV TZ Asia/Tokyo
ADD entrypoint.sh /usr/local/bin/docfx

# aptパッケージインストール
RUN apt-get update -y && ¥
    apt-get install -y --no-install-recommends ¥
        apt-transport-https ¥
        ca-certificates ¥
        dirmngr ¥
        fonts-ipafont ¥
        gnupg ¥
        libssl1.0-dev ¥
        nodejs ¥
        npm ¥
        nuget ¥
        wget ¥
        unzip

# ASP.NET環境構築
RUN wget https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb && ¥
    dpkg -i packages-microsoft-prod.deb && ¥
    apt-get update -y && ¥
    apt-get install -y --no-install-recommends ¥
        aspnetcore-runtime-3.1 ¥
        dotnet-sdk-3.1

# monoインストール
# ----- noproxy環境の場合 -----
# RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF && ¥

# ------ proxy環境の場合 ------
ADD key.txt .
RUN cat key.txt | apt-key add - && ¥
# ---------------------------
    echo "deb https://download.mono-project.com/repo/ubuntu stable-bionic main" | tee /etc/apt/sources.list.d/mono-official-stable.list && ¥
    apt-get update -y && ¥
    apt-get install -y --no-install-recommends ¥
        mono-complete ¥
        mono-runtime && ¥
    apt-get update -y && ¥
    apt-get upgrade -y && ¥
    rm key.txt

# wkhtmltopdfインストール
RUN wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox_0.12.5-1.bionic_amd64.deb && ¥
    apt-get install -y ./wkhtmltox_0.12.5-1.bionic_amd64.deb && ¥
    rm ./wkhtmltox_0.12.5-1.bionic_amd64.deb

# nodejsインストール
RUN npm install -g npm && ¥
```

```
RUN npm install -g npm && ¥
   npm cache clean && ¥
   npm install n -g && ¥
   n stable && ¥
   ln -sf /usr/local/bin/node /usr/bin/node

# DocFXインストール（v2.5以上の場合、metadataオプションが動作しない）
ADD https://github.com/dotnet/docfx/releases/download/v2.49/docfx.zip /
RUN unzip /docfx.zip -d /docfx && ¥
   rm /docfx.zip
RUN chmod 777 /usr/local/bin/docfx && ¥
   chmod 777 /docfx/docfx.exe && ¥
   rm -rf /var/lib/apt/lists/* /tmp/*

WORKDIR /work
```

```
#!/bin/bash

mono "/docfx/docfx.exe" "$@"
```

プロキシ環境でapt-keyコマンドが動作しなかったので こちら から公開鍵をダウンロードした。

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

xsBNBFPfqCcBCADctOzyTxfWvf40Nlb+AMkcJyb505WSbzhWU8yPmBNAJOnbwueM
sTkNMHEOu8fGRNxRWj5o/Db1N7EoSQtK3OgFnBef8xquUyrzA1nJ2aPfUWX+bhTG
1TwyrtLaOssFRz6z/h/ChUIFvt2VZCw+Yx4BiKi+tvgwrHTYB/Yf2J9+R/1O6949
n6veFFRBfgPOL0djhvRqXzhvFjJkh4xhTaGVeOnRR3+YQkblmti2n6KYl0n2kNB4
0ujSqpTloSfnR5tmJpz00WoOA9MJBdvHtxTTn8I6rVzXbm4mW9ZmB1kht/BgWaNL
alisW5AZSkQKer35wOWf0G7Gw+cWHq+I7W9pABEBAAHNOlhhbWFyaW4gUHVibGlj
IEplbmtpbnMgKGF1dG8tc2lnbmluZykgPHJlbGVuZ0B4YW1hcmluLmNvbT7CwHgE
EwECACIFAlPfqCcCGwMGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEKahmzjT
2DHvkOgH/2Hmny7VxRLLoynY+ONbf3wsIIqpbBprZb+VwsQo3uhZMLlh/kES5Ww7
3bvSlWWf0K/uGKpxsLyTLCT6xm9Gxg7e6hNHCyYiZz/u5orfzaF7LUDaG+Qfl9ge
Zj/ln9nRub8DSTRyGEbbJyNaNldgtn3ojRVTdkFAEeiHepG2BarjJZOwIkFf4Uo8
F2aQimBw9dDD6FqTSaPawguqNJxFlPU575Ymww0xotrx1J3D6k+bw0z9UYuY72JN
MMCm4CxGLGkJgt0lj5OEY2sp7rEOzBCjyCveBsGQmLTAtEM/ZHOrusPRMLY/E5pY
5nuGbLP4SGMtyNmEc0lNpr41XSTxgDDCwFwEEAECAAYFAlQIhKQACgkQyQ+cuQ4f
rQyc1wf+MCusJK4ANLWikbgiSSx1qMBveBlLKLEdCxYY+B9rc/pRDw448iBdd+nu
SVdbRoqLgoN8gHbClboP+i22yw+mga0KASD7b1mpdYB0npR3H73zbYArn3qTV8s/
yUXkIAEFUtj0yoEuv8KjO8P7nZJh8OuqqAupUVN0s3KjONqXqi6Ro3fvVEZWOUFZ
l/FmY5KmXlpcw+YwE5CaNhJ2WunrjFTDqynRU/LeoPEKuwyYvfo937zJFCrpAUMT
r/9QpEKmV61H7fEHA9oHq97FBwWfjOU0l2mrXt1zJ97xVd2DXxrZodlkiY6B76rh
aT4ZhltY1E7WB2Z9WPfTe1Y6jz4fZ8LAXAQQAQgABgUCWEyoiAAKCRABFQplW72B
An/PCAC0GkRBR3JTmG8WGeQMLb/o6Gon9cxpLnKv1GgFbHSM7XYMe7ySh5zxORwF
uECuJ5+qcA6cVe/kJAV8rewLULL9yvHK3oK7R8zoVGbFVm+lyoxiaXpkkWg21Mb8
lubiO+tA/dJc7hKQSpoI0+dmJNaNrTVwqj0tQ8e0OL9KvBOYwFbSe06bocSNPVmK
Ct0EOvpGcQfzFw5UEjJVkqFn/moUrSxj0YsJpwRXB1pOsBaQC6r9oCgUvxPf4H77
U07+ImXzxRWInVPYFSXSiBA7p+hzvsikmZEliIAia8mTteUF1GeK4kafUk6iZZUf
BlClb9sV4O9Vvv8W0VjK4Vg6O2UAzsBNBFPfqCcBCACtc7HssC9S3PxJm1youvGf
YLhm+KzMO+gloy7R32VXlZNxrkMYzaeerqSsMwxdhEjyOscT+rJbRGZ+9iPOGeh4
AqZlzzOuxQ/Lg5h+2mGVXe0Avb+A2zC56mLSQCL3W8NjABUZdknnc1Ylf9Dz05fy
4jPEttNSy+Rzte0ITLH1Hy/PKBrlF5n+G1/86f3L5n1ZZXmV3vi+rXT/OyEh9xRS
4usmR6kVh4o2XGlIzUrUjhZvb4lxrHfWgzKlWFoUSydaZDk7eikTKF692RiSSpLb
DLW2sNOdzT2eqv2B8CJRF5sLbD6BB3dAbH7KfqKiCT3xcCZhNEZw+M+GcRO/HNbn
ABEBAAHCwF8EGAECAAkFAlPfqCcCGwwACgkQpqGbONPYMe+sNQgAwjm9PJ45t7NB
NTXn1zadoQQbPqz9qAlWiII0k+zzJCTTVqgyIXJYI6zdNiB/Oh1Xajs/T9z9tL54
+LLqgtZKa0lzDOmcxn6Iujf3a1MFdYxKgaQtT2ADxAimuBoz3Y1ohxXgAs2+VISW
YoPBI+UWhYqg11zq3uwpFlYQBRgkVydCxefCxY19okNp9FPC7KJPpJkONgDAK693
Y9mOZXSq+XeGhjy3Sxesl0PYLlfV33z+vCpc2o1dDA5wuycgfqupNQlTkQm6gPOH
1jLu8Vttm4fdEtVMcqkn8dJFomo3JW3qxl7lWwjbVRg10G8LGAuBbD6CA0dGSf8P
kHFYv2XsdQ==
=45Mw
-----END PGP PUBLIC KEY BLOCK-----
```

- 作成した環境にファイルをマウントして利用する。

```
docker run --rm -it -v /`pwd`:/work <image名> bash
```

ドキュメントの作成

DocFXのファイルの説明

- docfx.json
  - DocFXで使用するファイルや設定を定義している

- index.md
  - 初期画面

- toc.yml

- WEBページの構成を定義する

- pdf/toc.yml
  - PDFの構成を定義している

- pdf/cover.yml
  - PDFのカバーページ

- **/.md
  - 各ページ

- template/*
  - デザインなどを定義している資材が格納されている

- */.json
  - REST APIの仕様をSwaggerのJSON形式で定義している

ドキュメントの作成手順

1. ディレクトリの移動

```
cd <docfx.jsonが存在するディレクトリ>
```

2. ソースコードコメントのyml化

```
docfx metadata
```

3. プラグインのインストール
   Plantumlの描画が可能になる

```
nuget install DocFx.Plugins.PlantUml -ExcludeVersion -OutputDirectory .
```

4. ドキュメントのPDF化

```
docfx pdf
```

_site_pdfフォルダにPDFが作成される。

5. ビルド

```
docfx build
```

_siteフォルダににhtml等が展開される

6. ウェブページのホスティング

```
docfx serve _site -n "*" -p 8080
```

ブラウザでlocalhost:8080にアクセスしてドキュメントを確認できる
Windowsの場合、4と5は以下のコマンドでまとめて実施できる

```
docfx --serve
```

課題

- WindowsだとPDFのフォントが汚い

  - フォントをインストールすれば、どうにかなりそう。

- WindowsだとPDFに画像が貼れない

  - PDFにすると画像が消える。

- PDFにするとnoteやwarningなどが崩れる

  - ボックスが崩れる

- Dockerfileのサイズが大きい

CIへの組み込み

GitHub Actionsで環境を構築するのはインストールするものが多くて難しいので、GitHub Actions上でDockerを使用する必要がある。
しかし、上に記載したDockerfileはビルドに約10分ほどかかり、毎回ビルドするのは避けたい。
そこで、事前にGitHub Package RegistryにビルドしたイメージをPushしておき、GitHub Actions上でそのイメージをpullして利用する。
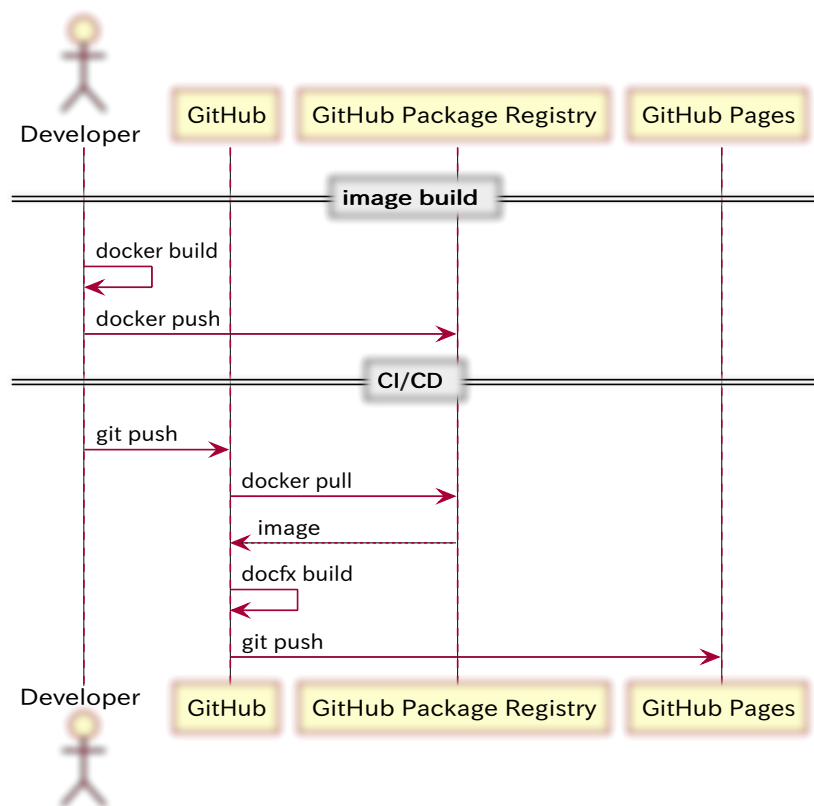
参考文献

- DocFX Getting Started
- mono Download

- Ubuntu に .NET Core SDK または .NET Core ランタイムをインストールする

テスト方法

.NET Coreの自動テストについて記載する。

.NET Coreのテスト

1. .NETのコードを書く
   - 公式チュートリアルを参考

2. .NETのテストコードを書く
   - 公式チュートリアルを参考

3. 以下のコマンドでテストコードを実行してCoverageをymlで出力する

```
dotnet test --collect:"XPlat Code Coverage"
```

.NETの自動テスト

GitHub Actionsを利用した例

1. ワークフローのymlを作成する

```yaml
name: .NET Core CI
on:
push:
  branches:
  - master
jobs:
test:
  runs-on: ubuntu-latest
  steps:
  - uses: actions/checkout@master
  - name: Setup .NET Core
    uses: actions/setup-dotnet@v1
    with:
    dotnet-version: '3.1.x'
  - name: .NET Build
    run: dotnet build
  - name: .NET Test
    id: dotnet_test
    run: |
      dotnet test --collect:"XPlat Code Coverage"
      REPORT_PATH=`find ./ -name "coverage*.xml"`
      echo ::set-output name=report_path::${REPORT_PATH}
  - name: Upload coverage to Codecov
    uses: codecov/codecov-action@v1
    with:
    token: ${{ secrets.CODECOV_TOKEN }}
    file: ${{ steps.dotnet_test.outputs.report_path }}
```

2. Gitにpushすると自動でテストが開始される。

   - codecovにカバレッジがアップロードされる。

⊠ Tip

上記コマンドでは、`./TestResults/{hash}/coverage.cobertura.xml` にレポートが出力される そのため、`find` コマンドでパスを特定してワークフロー間で変数の受け渡しをする必要がある

GitHub Actionsのワークフローコマンド
GitHub Actionsのメタデータ構文

Swagger Petstore

pet

Everything about your **Pets**

addPet

Add a new pet to the store

Request

POST /pet

Parameters

NAME	TYPE	VALUE	NOTES
*body			Pet object that needs to be added to the store

Responses

STATUS CODE	DESCRIPTION	SAMPLES
405	Invalid input	

updatePet

Update an existing pet

Request

PUT /pet

Parameters

NAME	TYPE	VALUE	NOTES
*body			Pet object that needs to be added to the store

Responses

STATUS CODE	DESCRIPTION	SAMPLES
400	Invalid ID supplied	
404	Pet not found	
405	Validation exception	

findPetsByStatus

Finds Pets by status

Multiple status values can be provided with comma separated strings

Request

```
GET /pet/findByStatus?status
```

Parameters

NAME	TYPE	VALUE	NOTES
*status	array		Status values that need to be considered for filter

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid status value	

findPetsByTags

Finds Pets by tags

Muliple tags can be provided with comma separated strings. Use tag1, tag2, tag3 for testing.

Request

```
GET /pet/findByTags?tags
```

Parameters

NAME	TYPE	VALUE	NOTES
*tags	array		Tags to filter by

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid tag value	

deletePet

Deletes a pet

Request

```
DELETE /pet/{petId}
```

Parameters

NAME	TYPE	VALUE	NOTES
api_key	string		
*petId	integer		Pet id to delete

Responses

STATUS CODE	DESCRIPTION	SAMPLES
400	Invalid ID supplied	
404	Pet not found	

getPetById

Find pet by ID

Returns a single pet

Request

> GET /pet/{petId}

Parameters

NAME	TYPE	VALUE	NOTES
*petId	integer		ID of pet to return

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid ID supplied	
404	Pet not found	

updatePetWithForm

Updates a pet in the store with form data

Request

> POST /pet/{petId}

Parameters

NAME	TYPE	VALUE	NOTES
*petId	integer		ID of pet that needs to be updated
name	string		Updated name of the pet
status	string		Updated status of the pet

Responses

STATUS CODE	DESCRIPTION	SAMPLES
405	Invalid input	

uploadFile

uploads an image

Request

POST /pet/{petId}/uploadImage

Parameters

NAME	TYPE	VALUE	NOTES
*petId	integer		ID of pet to update
additionalMetadata	string		Additional data to pass to server
file	file		file to upload

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	

store

Access to Petstore orders

addPet

Add a new pet to the store

Request

POST /pet

Parameters

NAME	TYPE	VALUE	NOTES
*body			Pet object that needs to be added to the store

Responses

STATUS CODE	DESCRIPTION	SAMPLES
405	Invalid input	

getInventory

Returns pet inventories by status

Returns a map of status codes to quantities

Request

GET /store/inventory

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	

placeOrder

Place an order for a pet

Request

POST /store/order

Parameters

NAME	TYPE	VALUE	NOTES
*body			order placed for purchasing the pet

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid Order	

deleteOrder

Delete purchase order by ID

For valid response try integer IDs with positive integer value. Negative or non-integer values will generate API errors

Request

DELETE /store/order/{orderId}

Parameters

NAME	TYPE	VALUE	NOTES
*orderId	integer		ID of the order that needs to be deleted

Responses

STATUS CODE	DESCRIPTION	SAMPLES
400	Invalid ID supplied	
404	Order not found	

getOrderById

Find purchase order by ID

For valid response try integer IDs with value >= 1 and <= 10. Other values will generated exceptions

Request

GET /store/order/{orderId}

Parameters

NAME	TYPE	VALUE	NOTES
*orderId	integer		ID of pet that needs to be fetched

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid ID supplied	
404	Order not found	

user

Operations about user

createUser

Create user

This can only be done by the logged in user.

Request

POST /user

Parameters

NAME	TYPE	VALUE	NOTES
*body			Created user object

Responses

STATUS CODE	DESCRIPTION	SAMPLES
default	successful operation	

createUsersWithArrayInput

Creates list of users with given input array

Request

POST /user/createWithArray

Parameters

NAME	TYPE	VALUE	NOTES
*body			List of user object

Responses

STATUS CODE	DESCRIPTION	SAMPLES
default	successful operation	

createUsersWithListInput

Creates list of users with given input array

Request

POST /user/createWithList

Parameters

NAME	TYPE	VALUE	NOTES
*body			List of user object

Responses

STATUS CODE	DESCRIPTION	SAMPLES
default	successful operation	

loginUser

Logs user into the system

Request

GET /user/login?username&password

Parameters

NAME	TYPE	VALUE	NOTES
*username	string		The user name for login
*password	string		The password for login in clear text

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid username/password supplied	

logoutUser

Logs out current logged in user session

Request

```
GET /user/logout
```

Responses

STATUS CODE	DESCRIPTION	SAMPLES
default	successful operation	

deleteUser

Delete user

This can only be done by the logged in user.

Request

```
DELETE /user/{username}
```

Parameters

NAME	TYPE	VALUE	NOTES
*username	string		The name that needs to be deleted

Responses

STATUS CODE	DESCRIPTION	SAMPLES
400	Invalid username supplied	
404	User not found	

getUserByName

Get user by user name

Request

GET /user/{username}

Parameters

NAME	TYPE	VALUE	NOTES
*username	string		The name that needs to be fetched. Use user1 for testing.

Responses

STATUS CODE	DESCRIPTION	SAMPLES
200	successful operation	
400	Invalid username supplied	
404	User not found	

Other APIs

updateUser

Updated user

This can only be done by the logged in user.

Request

PUT /user/{username}

Parameters

NAME	TYPE	VALUE	NOTES
*username	string		name that need to be updated
*body			Updated user object

Responses

STATUS CODE	DESCRIPTION	SAMPLES
400	Invalid user supplied	
404	User not found	

Contacts

get contacts

You can get a collection of contacts from your tenant.

Required scope: *Contacts.Read* or *Contacts.Write*

Request

```
GET /contacts?api-version
```

Parameters

NAME	TYPE	VALUE	NOTES
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES

		*Mime type:*application/json
		
		``{
		  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/Microsoft.DirectoryServices.Contact",
		  "value": [
		    {
		      "odata.type": "Microsoft.DirectoryServices.Contact",
		      "objectType": "Contact",
		      "objectId": "31944231-fd52-4a7f-b32e-7902a01fddf9",
		      "deletionTimestamp": null,
		      "city": null,
		      "companyName": null,
		      "country": null,
		      "department": null,
		      "dirSyncEnabled": null,
		      "displayName": "Marcus Breyer",
		      "facsimileTelephoneNumber": null,
		      "givenName": "Marcus",
		      "jobTitle": null,
		      "lastDirSyncTime": null,
		      "mail": "Marcus@contoso.com",
		      "mailNickname": "MarcusBreyer",
		      "mobile": null,
		      "physicalDeliveryOfficeName": null,
		      "postalCode": null,
		      "provisioningErrors": [],
		      "proxyAddresses": [
		        "SMTP:Marcus@contoso.com"
		      ],
		      "sipProxyAddress": null,
		      "state": null,```

STATUS CODE	DESCRIPTION	SAMPLES

		"streetAddress": null,
		      "surname": "Breyer",
		      "telephoneNumber": null
		    },
		    {
		      "odata.type": "Microsoft.DirectoryServices.Contact",
		      "objectType": "Contact",
		      "objectId": "35110b16-360c-4c4a-93b2-03f065fabd93",
		      "deletionTimestamp": null,
		      "city": null,
		      "companyName": null,
		      "country": null,
		      "department": null,
		      "dirSyncEnabled": null,
		      "displayName": "Yossi Ran",
		      "facsimileTelephoneNumber": null,
		      "givenName": "Yossi",
		      "jobTitle": null,
		      "lastDirSyncTime": null,
		      "mail": "Yossi@contoso.com",
		      "mailNickname": "YossiRan",
		      "mobile": null,
		      "physicalDeliveryOfficeName": null,
		      "postalCode": null,
		      "provisioningErrors": [],
		      "proxyAddresses": [
		        "SMTP:Yossi@contoso.com"
		      ],
		      "sipProxyAddress": null,
		      "state": null,
		      "streetAddress": null,
		      "surname": "Ran",
		      "telephoneNumber": null
		    },
		    {

200	OK. Indicates success. The results are returned in the response body.	"odata.type": "Microsoft.DirectoryServices.Contact",
		      "objectType": "Contact",
		      "objectId": "7163f3b8-70c9-43d2-b9e1-4467ddaf087a",
		      "deletionTimestamp": null,
		      "city": null,
		      "companyName": null,
		      "country": null,
		      "department": null,
		      "dirSyncEnabled": null,
		      "displayName": "Jeremy Nelson",
		      "facsimileTelephoneNumber": null,
		      "givenName": "Jeremy",
		      "jobTitle": null,
		      "lastDirSyncTime": null,
		      "mail": "Jeremy@contoso.com",
		      "mailNickname": "JeremyNelson",
		      "mobile": null,
		      "physicalDeliveryOfficeName": null,
		      "postalCode": null,
		      "provisioningErrors": [],
		      "proxyAddresses": [
		        "SMTP:Jeremy@contoso.com"
		      ],
		      "sipProxyAddress": null,
		      "state": null,
		      "streetAddress": null,
		      "surname": "Nelson",
		      "telephoneNumber": null
		    },
		    {
		      "odata.type": "Microsoft.DirectoryServices.Contact",

STATUS CODE	DESCRIPTION	SAMPLES

		"odata.type": "Microsoft.DirectoryServices.Contact",
		    "objectType": "Contact",
		    "objectId": "83234b5c-3a10-4108-8f36-39b1addadfdb",
		    "deletionTimestamp": null,
		    "city": null,
		    "companyName": null,
		    "country": null,
		    "department": null,
		    "dirSyncEnabled": null,
		    "displayName": "David Strome",
		    "facsimileTelephoneNumber": null,
		    "givenName": "David",
		    "jobTitle": null,
		    "lastDirSyncTime": null,
		    "mail": "David@contoso.com",
		    "mailNickname": "DavidStrome",
		    "mobile": null,
		    "physicalDeliveryOfficeName": null,
		    "postalCode": null,
		    "provisioningErrors": [],
		    "proxyAddresses": [
		      "SMTP:David@contoso.com"
		    ],
		    "sipProxyAddress": null,
		    "state": null,
		    "streetAddress": null,
		    "surname": "Strome",
		    "telephoneNumber": null
		  },
		  {
		    "odata.type": "Microsoft.DirectoryServices.Contact",
		    "objectType": "Contact",
		    "objectId": "8c1315ce-bf6f-4e26-b24f-c830606ef41c",
		    "deletionTimestamp": null,
		    "city": null,
		    "companyName": null,
		    "country": null,
		    "department": null,
		    "dirSyncEnabled": null,
		    "displayName": "Holly Holt",
		    "facsimileTelephoneNumber": null,
		    "givenName": "Holly",
		    "jobTitle": null,
		    "lastDirSyncTime": null,
		    "mail": "Holly@contoso.com",
		    "mailNickname": "HollyHolt",
		    "mobile": null,
		    "physicalDeliveryOfficeName": null,
		    "postalCode": null,
		    "provisioningErrors": [],
		    "proxyAddresses": [
		      "SMTP:Holly@contoso.com"
		    ],
		    "sipProxyAddress": null,
		    "state": null,
		    "streetAddress": null,
		    "surname": "Holt",
		    "telephoneNumber": null
		  }
		  ]
		}

get contact by id

Get a contact by using the object ID.

Required scope: *Contacts.Read* or *Contacts.Write*

Request

GET /contacts/{object_id}?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.
*api-version	string	1.6	Specifies the version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES
		*Mime type:*application/json

STATUS CODE 200	DESCRIPTION	SAMPLES

	OK Indicates success. The contact is returned in the response body.	{
		  "odata.metadata": "https://graph.windows.net/graphdir1.onmicrosoft.com/$metadata#directoryObjects/Microsoft.DirectoryServices.Contact/@Element",
		  "odata.type": "Microsoft.DirectoryServices.Contact",
		  "objectType": "Contact",
		  "objectId": "31944231-fd52-4a7f-b32e-7902a01fddf9",
		  "deletionTimestamp": null,
		  "city": null,
		  "companyName": null,
		  "country": null,
		  "department": null,
		  "dirSyncEnabled": null,
		  "displayName": "Marcus Breyer",
		  "facsimileTelephoneNumber": null,
		  "givenName": "Marcus",
		  "jobTitle": null,
		  "lastDirSyncTime": null,
		  "mail": "Marcus@contoso.com",
		  "mailNickname": "MarcusBreyer",
		  "mobile": null,
		  "physicalDeliveryOfficeName": null,
		  "postalCode": null,
		  "provisioningErrors": [],
		  "proxyAddresses": [
		    "SMTP:Marcus@contoso.com"
		  ],
		  "sipProxyAddress": null,
		  "state": null,
		  "streetAddress": null,
		  "surname": "Breyer",
		  "telephoneNumber": null
		}

update contact

Change a contact's properties.

Required scope: *Contacts.Write*

Request

PATCH /contacts/{object_id}?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	7163f3b8-70c9-43d2-b9e1-4467ddaf087a	The object ID (GUID) of the target contact.

NAME	TYPE	VALUE	NOTES
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.
bodyparam			this is request body, not real parameter

Responses

STATUS CODE	DESCRIPTION	SAMPLES
204	No Content. Indicates success. No response body is returned.	

delete contact

Delete a contact.

Required scope: *Contacts.Write*

Request

```
DELETE /contacts/{object_id}[?api-version]
```

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	7163f3b8-70c9-43d2-b9e1-4467ddaf087a	The object ID (GUID) of the target contact.
api-version	string	1.6	Specifies the version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES
204	No Content. Indicates success.	

get contact manager link

Get a link to the contact's manager.

Required scope: *Contacts.Read* or *Contacts.Write*

Request

GET /contacts/{object_id}/$links/manager?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES

200	OK. Indicates success. A link to the contact's manager is returned.	*Mime type:*application/json
		
		{
		  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/$links/manager",
		  "url": "https://graph.windows.net/myorganization/directoryObjects/35110b16-360c-4c4a-93b2-03f065fabd93/Microsoft.WindowsAzure.ActiveDirectory.Contact"
		}

404	Not Found. The requested resource was not found. This can occur if the manager property is not currently set for the specified contact. It can also have other causes, for example, a bad domain. A code and associated message is returned with the error.	*Mime type:*application/json
		
		{
		  "odata.error": {
		    "code": "Request_ResourceNotFound",
		    "message": {
		      "lang": "en",
		      "value": "Resource not found for the segment 'manager'."
		    }
		  }
		}

update contact manager

Update the contact's manager

Required scope: *Contacts.Write*

Request

PUT /contacts/{object_id}/$links/manager?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.
*bodyparam			The request body contains a single property that specifies the URL of the user or contact to add as manager.

Responses

STATUS CODE	DESCRIPTION	SAMPLES
204	No Content. Indicates success. No response body is returned.	

delete contact manager by id

Delete the contact's manager.

Required scope: *Contacts.Write*

Request

DELETE /contacts/{object_id}/$links/manager?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.

NAME	TYPE	VALUE	NOTES
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES
204	No Content. Indicates success. N response body is returned.	

get contact direct reports links

Get a links to the contact's direct reports.

Required scope: *Contacts.Read* or *Contacts.Write*

Request

GET /contacts/{object_id}/$links/directReports?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES

STATUS CODE	DESCRIPTION	SAMPLES

200	OK. Indicates success. One or more direct reports are returned.	*Mime type:* application/json
		
		 {
		  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/$links/directReports",
		  "value": [
		    {
		      "url": "https://graph.windows.net/myorganization/directoryObjects/8c1315ce-bf6f-4e26-b24f-c830606ef41c/Microsoft.DirectoryServices.Contact"
		    }
		  ]
		}

get contact memberOf links

Get a links to the contact's direct group and directory role memberships.

Required scope: *Contacts.Read* or *Contacts.Write*

Request

GET /contacts/{object_id}/$links/memberOf?api-version

Parameters

NAME	TYPE	VALUE	NOTES
*object_id	string	31944231-fd52-4a7f-b32e-7902a01fddf9	The object ID (GUID) of the target contact.
*api-version	string	1.6	The version of the Graph API to target. Beginning with version 1.5, the api-version string is represented in major.minor format. Prior releases were represented as date strings: '2013-11-08' and '2013-04-05'. Required.

Responses

STATUS CODE	DESCRIPTION	SAMPLES

200	OK. Indicates success. One or more groups and/or directory roles are returned.	*Mime type:*application/json
		
		``json
		{
		  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/$links/memberOf",
		  "value": [
		    {
		      "url": "https://graph.windows.net/myorganization/directoryObjects/12345678-b942-47c9-a10e-a4bee353ce60/Microsoft.DirectoryServices.Group"
		    }
		  ]
		}
		``

*Mime type:*application/json

```json
{
  "odata.metadata": "https://graph.windows.net/myorganization/$metadata#directoryObjects/$links/memberOf",
  "value": [
    {
      "url": "https://graph.windows.net/myorganization/directoryObjects/
```