**Polaris Platform Develop Setup**

**Developer Workstation Configuration Guide**

# Build your coding environment -- guidance

1. Create a Visual Studio account using your @pwc.com email address.  
   <https://signup.live.com/signup>  
     
   https://pwc-gx-humancapital.visualstudio.com/
2. Sign up for a Visual Studio subscription using the account created in the previous step. Follow the step2 in the document for Visual Studio subscription. <https://docs.google.com/document/d/1RKaMoBHwyAPLa5Y8sQVH1cYnC2qlh27dbtLsUKBAmdc/edit>
3. Ensure the following software is installed on the workstation:  
   1. SQL Server Express 2017 ([Installation Guide](#_td1e8je3jege))
   2. SQL Server Management Studio 18 ([Installation Guide](#_sjzptxpqk7x7)) ***OR***  
      Toad for SQL Server ([Installation Guide](#_uykvy5dt9yz5))
   3. Visual Studio 2019 ([Installation Guide](#_1rtre4qm9thx)) Version >=16.8.0
   4. Docker Desktop ([Installation Guide](#_ln2ejhux9yvh)) (Ignore)
   5. Azure Cosmos DB Emulator ([Installation Guide](#_wszt4ey2spc1))
   6. .NET 5.0 Install ([Installation Guide](#_v5zghlk1lb3x))
   7. Python2.7 INstall (Installation Guide)
   8. Node.js ([Installation Guide](#_mj2rtflgl2e5))
   9. @EasyDevops/Angular-Auth-Plugin ([Installation Guide](#_k27rdto9b5l0))
   10. Angular 8 (or latest version) ([Installation Guide](#_v84rhqxcan6j))
   11. Visual Studio Code ([Installation](#_8z7ztjh8cvsl))

# Install Redis Cache Locally

To make sure that everyone has Redis Cache installed locally so that we don't miss any scenarios when developing as well as increase performance locally.

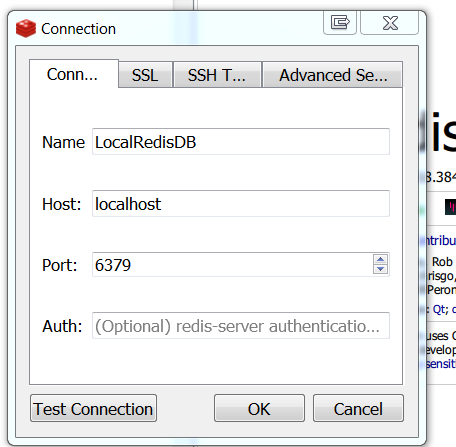
And here is the link to download Redis Desktop Manager tool that will let you view the cache:

<https://drive.google.com/drive/folders/1PryG9D6Pd7Z9qvIZgBlVoWU8Qjl7g3CB>

Use the Redis-x64-3.2.100.msi installer to install Redis.

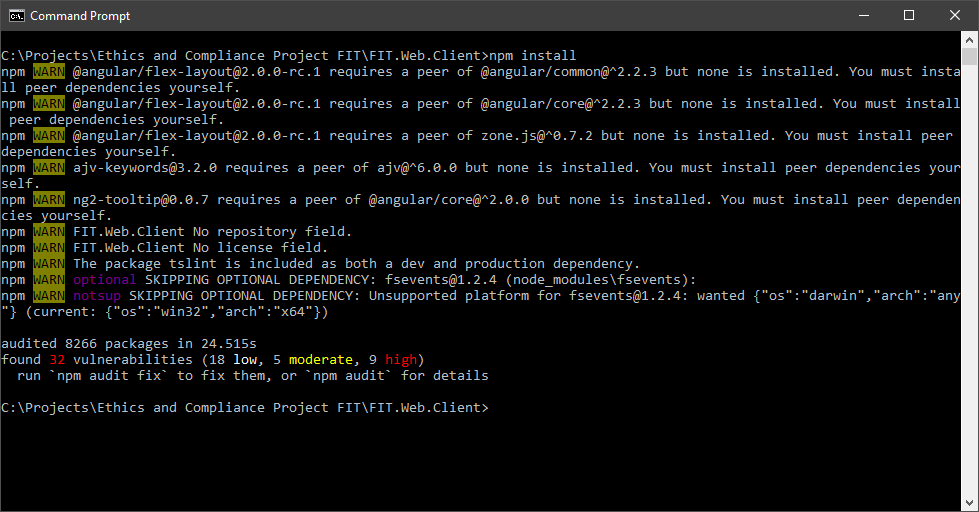
Then use the file redis-desktop-manager-0.9.3.817.zip for installing the manager tools.

When creating a new connection in the desktop manager it should look like:



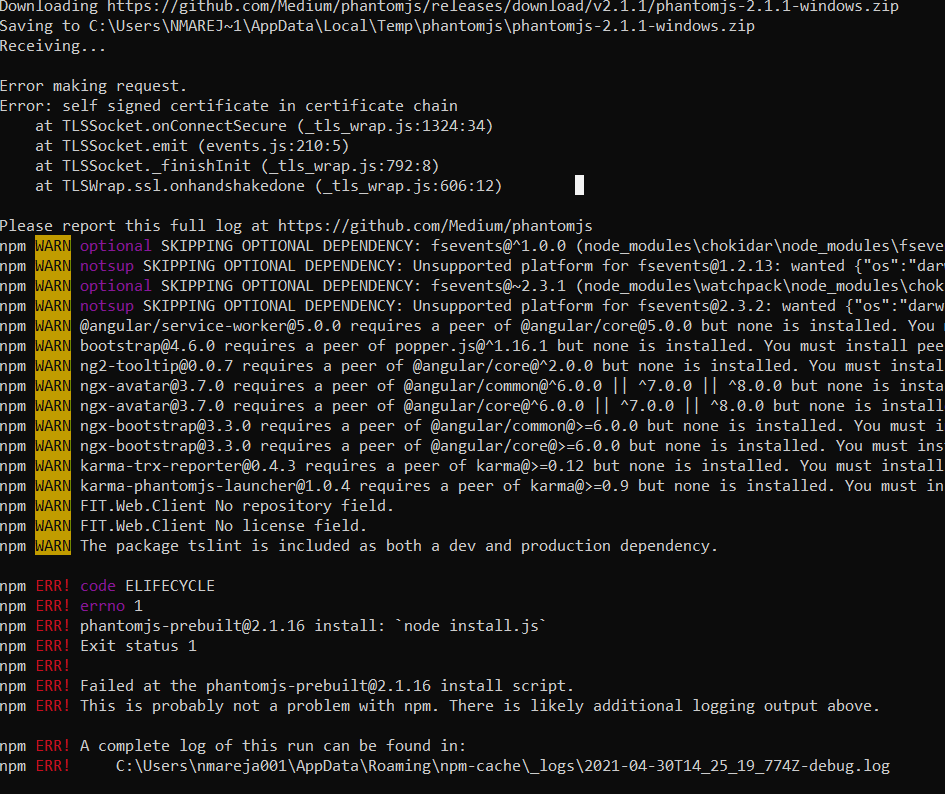
The name can be changed to whatever makes sense for your local env.

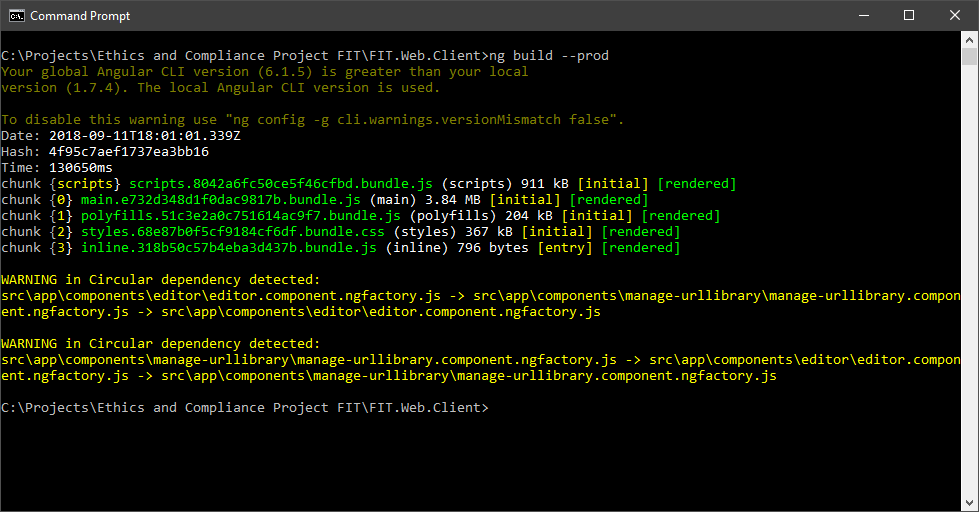
If you have any questions please reach out to Sarathy or Rex.

1. Clone the git repository for Polaris @   
   **https://pwc-gx-humancapital.visualstudio.com/Polaris/\_git/Polaris***\* After cloning the repository, you should replace all instances of %20 in the directory name with a space character. While not required, this may prevent certain issues from arising.*
2. Create a branch with your change in one of the following formats (braces=replace, brackets=optional):   
   **feature/US-USERSTORYNUMBER\_some-description**   
   Examples:  
   **feature/US-123\_create-new-awesome-feature**
3. Open the FIT solution file (**FIT.sln**) from the cloned repository.
4. Build the FIT .NET Solution
5. Run this before running npm install
   1. **$Env:NODE\_TLS\_REJECT\_UNAUTHORIZED=0**
6. In the FIT.Web.Client directory, run the following command on the command prompt:  
   **npm install**  
     
   This will install all the required Node.js dependencies.  
   

If you get self signed certificate error, then goto [Troubleshooting Tips](#_v6hx51lyzh9b)

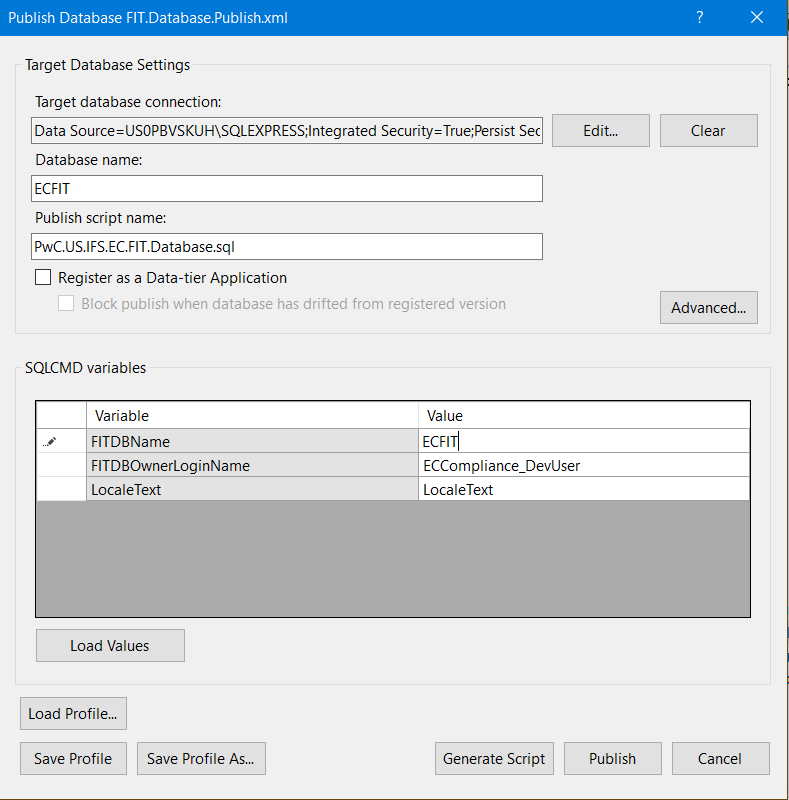
Running “npm set strict-ssl false” will fix the self signed certificate error.



1. In the ***FIT.Web.Client*** directory, run the following command on the command prompt:  
   **ng build --prod**  
     
   This will build and copy the client-side artifacts into the webroot direction.  
   

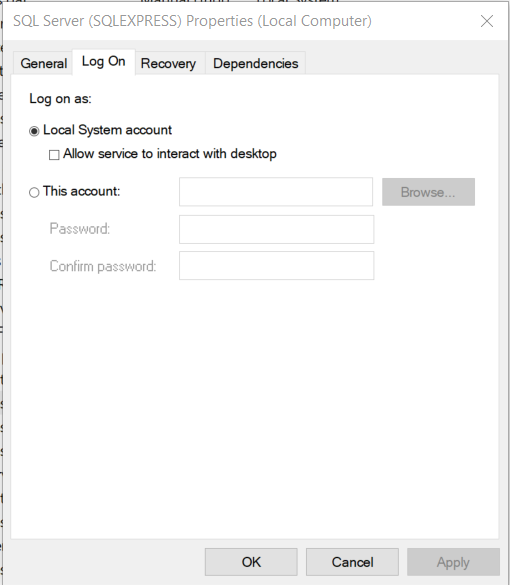
Currently overflows the javascript heap. Can workaround by executing this instead:

node --max-old-space-size=8192 node\_modules/@angular/cli/bin/ng build --prod

1. Within *Visual Studio*, expand ***FIT.Database*** in the *Solution Explorer* then double-click on the ***FIT.Database.Publish.xml*** file to setup a local test database.  
     
   1. Click the ***Edit*** button and select the local *SQL Express* server.
   2. Click the ***Load Values*** button to populate the *SQLCMD variables*.  
      
   3. Make sure that the FITDBName is “ECFIT”. Also the FITDBOwnerLoginName should be your id, ie. “PCWGLB\dbrooks016” (Signin to Desktop with pwc email address for this PCWGLB domain to work)
   4. Prior to performing the next step go to the “Services” App in windows and make sure that SQLEXPRESS is running.



If it is not, then right click on it and select Properties. Then select the Log On tab and change the Log on as: to Local System account.



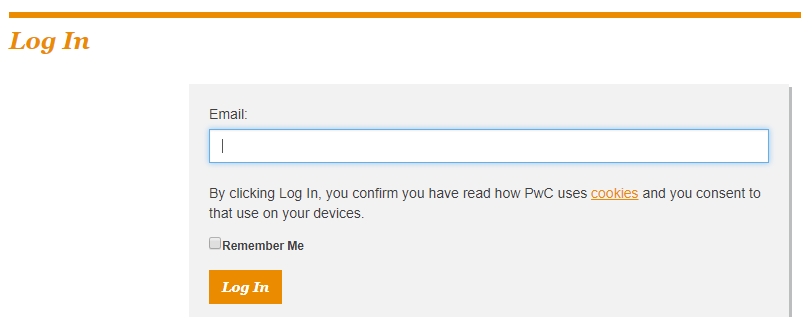
Click OK. Then Right click on SQLEXPRESS and Choose Start.

* 1. Click the ***Publish*** button and await the creation of the database.

If the ***Publish*** fails, then just try a 2nd time.

1. In *Visual Studio* set the startup projects to *Api* and *Client*:  
   1. Right-click on the solution in the *Solution Explorer* and select ***Set Startup Projects…***
   2. Select *Multiple startup projects:*
   3. Set ***FIT.Web.Api*** to ***Start***
   4. Set ***FIT.Web.Client*** to ***Start without debugging***

13. Press F5 in Visual Studio with the FIT.Web.API as the startup project. This will launch the api @ <http://localhost:20208>



Enter any @pwc.com email address.

# Data Model

You can review and print the most recent data model from please follow the guidance given in this model for creation of tables (Specific guidance was given for the Event and Event related tables.)

<https://drive.google.com/open?id=0BwvpnVvg7WmAb3BSamhmck84Z0k>

# Docker Desktop

This is needed to build and test docker compose files.

The installation files can be found at the following URL:

<https://www.docker.com/products/docker-desktop>

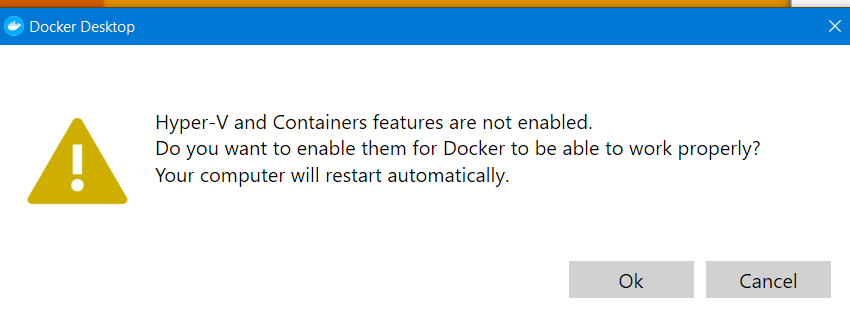
Sign up for a docker account to download the docker desktop.

Skip the tutorial steps.

Unselect ***Use Windows containers instead of Linux Containers***

During installation of Docker Desktop, you will be asked to log out. Log out of windows and login back.

Select ***OK*** to enable Hyper-V and Container features:



# Microservices - Gateway

Using Git Bash

**git clone**

**https://pwc-gx-humancapital@dev.azure.com/pwc-gx-humancapital/Polaris/\_git/ms-gateway**

Switch to **develop** branch.

Run Visual Studio using the **Thycotic Administrator** account.

Open **Polaris.Gateway.sln** file in Visual Studio 2019 in administrator mode, and wait for Nuget packages to restore.

Build and Run the solution.

# Software Installation Guides

## SQL Server Express 2017

The installation files can be found at the following URL:

<https://go.microsoft.com/fwlink/?linkid=853017>

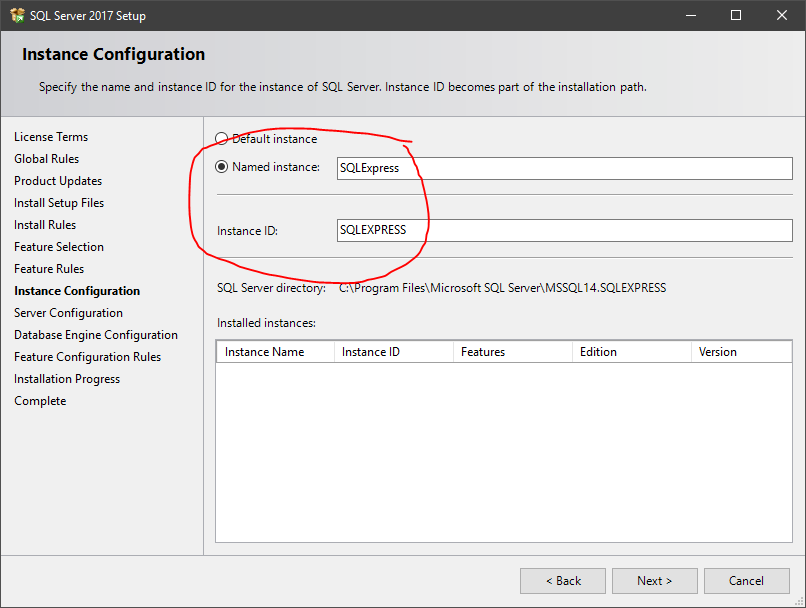
You’ll need to run the installer using Thycotic Administrator.

Ensure that a custom installation is performed so that FileStream can be installed.

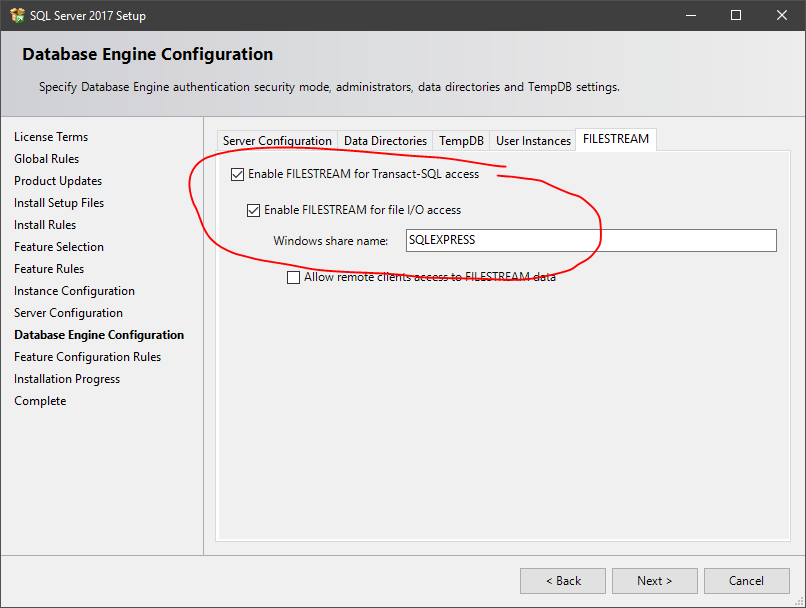
Select the following options for the custom installation:

* Database Engine Services
  + SQL Server Replication
  + Full-Text and Semantic Extractions for Search
* Client Tools Connectivity
* Client Tools Backwards Compatibility
* Client Tools SDK
* SQL Client Connectivity SDK

Ensure the ID of the SQL Express server is set to ***SQLEXPRESS***, if this is not possible then changes will have to be made to ***web.config*** files in the project codebase.



***FILESTREAM*** must be enabled during the *Database Engine Configuration* step of the installation process.



## SQL Server Management Studio 17

The installation files can be found at the following URL:

<https://go.microsoft.com/fwlink/?linkid=2014306>

## Toad for SQL Server 6.7

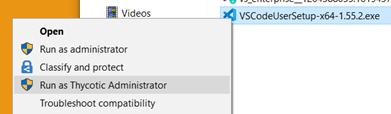
The installation of *Toad for SQL Server 6.7* is handled through the PwC ***Software Installer*** utility.

## Visual Studio 2019

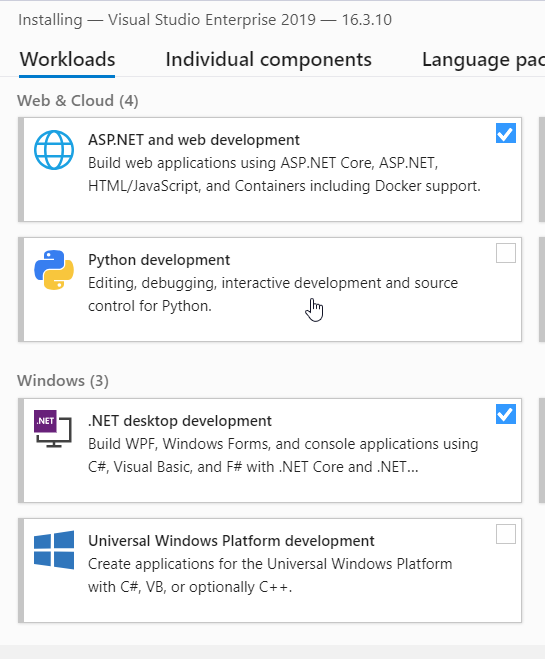
<https://visualstudio.microsoft.com/vs/>

Download the enterprise edition.

Run the install under the **Thycotic Administrator** account.



The ***.NET desktop development*** and ***ASP.NET and web development*** workloads must be installed to allow the FIT solution to be opened.

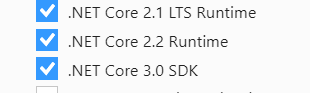


Under **Individual Components** select ***.NET > >NET Framework 4.6.2 SDK*** and ***.NET Framework 4.6.2 targeting pack*** for compatibility with the code base.

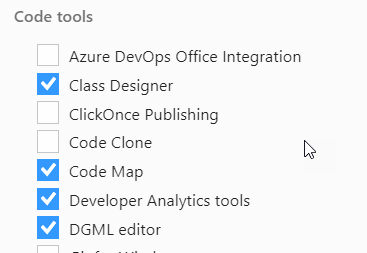


Select **.NET Core 3.0 SDK**

**Update, we just need 3.0 and 5.0 SDK, no longer using 2.x**



Optionally select ***Code Tools > Class Designer*** and ***Code Tools > Code Map*** for improved code visualizations.



## Visual Studio Code

Download and install:

<https://code.visualstudio.com/>

Recommended Extensions to install:

Jest by Orta

TSLint by Microsoft

Angular Language Service by Angular

Angular Snippets (Version 11) by John Papa

## Azure Cosmos DB Emulator

The installation files can be found at the following URL:

<https://aka.ms/cosmosdb-emulator>

Installing and running this emulator requires that it be started using the Thycotic Administrator.

## .NET 5.0 Install

Verify that .NET 5.0 SDK is installed in your workstation using the following command in the powershell:

**dotnet --version**

If you do not have version 5.0 then install it/.

The installation files can be found at the following URL:

<https://dotnet.microsoft.com/download/dotnet/5.0>

## Python 2.7.18 Install

## The angular client app leverages the NPM package node-sass version 4.13.1. At the time of writing this (5/26/21), node-sass needs to be built locally using node-gyp and visual studio 2017 build tools. In order for this process to work, it requires python 2.7.18 installed.

Download python here (Windows x86-64 MSI installer):  
<https://www.python.org/downloads/release/python-2718/>

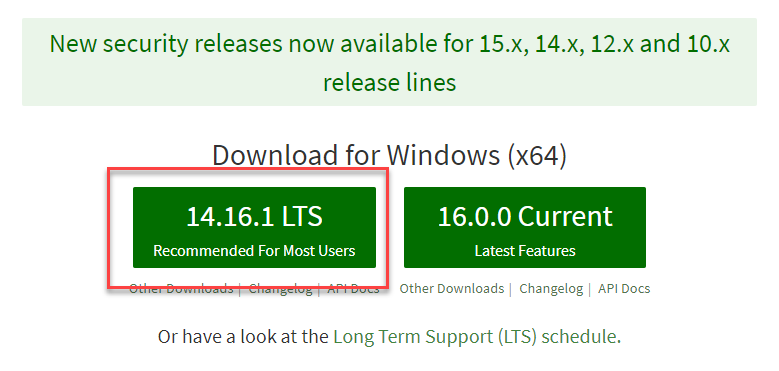
Install as Thycotic Administrator. Ensure that the install adds it to the Path. This option is in the features box towards the bottom.

## Node.js

The installation files can be found at the following URL:

<https://nodejs.org/en/>

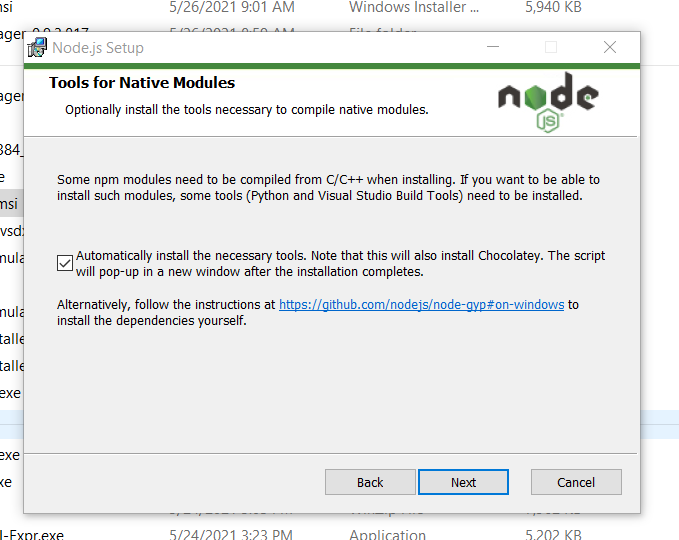
Download the installation files for Node.js version 8 or 14.16.1 LTS or latest LTS version.



When installing node, ensure to check “Automatically install the necessary tools” when prompted to install the C/C++ Visual Studio Build tools.

Ensure chocolatey is installed as well.

Make sure all instances of visual studio installer are closed



## @EasyDevops/Angular-Auth-Plugin

1. Copy the below login details to connect JFrog into the file .npmrc file under c:\Users\{your guid}.

If the .npmrc file does not exist then use Git Bash and goto folder c:\Users\{your guid} and use the touch command to create an .npmrc file and save it. Paste the following text into this file and save it.

\_auth = YmVuamFtaW4uamFja3NvbkB1cy5wd2MuY29tOkFLQ3A1ZTMxNzc0a0ZqVm12aE1Ma1h0Q01LdDVMenBqdHpzcER3WXBocXJnOFQ4WEgxRGlhVjdMYzZmZVVpbmpKeWIzaHhrdTY=

email = ifs-albatross-service-user@nowhere.com

registry=https://artifacts.pwc.com/artifactory/api/npm/ifs-npm/

always-auth = true

@easydevops:registry=https://artifacts.pwc.com/artifactory/api/npm/ifs-npm/

@quicksuite:registry=<https://artifacts.pwc.com/artifactory/api/npm/ifs-npm/>

strict-ssl=false

2. Generate an **Entrust IdentityGuard Token** that’s needed for JFrog Artifactory’s Multi-Factor Authentication (MFA)  
<https://wwwpwcnetwork.pwc.myshn.net/hub?id=kb_article&sys_id=a01a1c4ddbb1bb041b209407db961996&shn-direct>

3. HOW-TO setup **JFrog Artifactory** token and update npm configuration

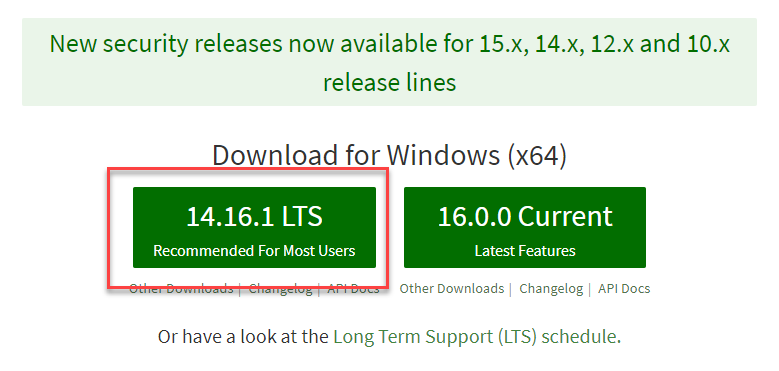
**NOTE:** When clicking on the below link you’ll be directed to a Login page. When prompted with the Login page select the **PwC Login** button option. <https://innovation-tech.pwc.com/confluence/display/AL/HOW-TO+setup+Artifactory+token+and+update+npm+configuration>

## 

## Angular

*Requires Node.js to be installed.*

Download the installation files for Node.js version 14.16.1 LTS or latest LTS version.



From the command-line interface, execute the following command to install Angular CLI globally:

**npm install -g @angular/cli**

If the above command fails with the following errors:  
  
npm ERR! code SELF\_SIGNED\_CERT\_IN\_CHAIN

npm ERR! errno SELF\_SIGNED\_CERT\_IN\_CHAIN

npm ERR! request to https://registry.npmjs.org/@angular%2fcli failed, reason: self signed certificate in certificate chain

Then run the following command and try again:  
**npm config set registry="http://registry.npmjs.org/"**

## 

## Local Dev Env Setup

Create entry in Users; require UserGuid, EmployeeID, FirstName, LastName

/development needs Full Name, Employee ID, and Employee Guid

Use <https://navigator.pwc.com/> to get your employee id.

Execute the below script by updating with your Guid/EmployeeID/FirstName/LastName:

DECLARE @guid nvarchar(50) = 'sganesan012'

DECLARE @UserID INT

SET @UserID = (SELECT TOP 1 UserID from dbo.[User] WHERE UserGuid = @guid)

-- Admin

IF (@UserID IS NULL)

BEGIN

INSERT INTO dbo.[User] ([UserGuid]

,[EmployeeID]

,[FirstName]

,[LastName]

,[CreatedBy]

,[UpdatedBy]

,[LastUpdate]

,[LastEmailAddressUpdate])

VALUES

(@guid,

'00090085934',

'Sarathy',

'Ganesan'

,'System'

,'System'

,GETUTCDATE()

,GETUTCDATE()

)

SET @UserID = (SELECT TOP 1 UserID from dbo.[User] WHERE UserGuid = @guid)

END

IF NOT EXISTS (SELECT \* FROM dbo.[UserRole] WHERE UserID = @UserID AND RoleID = 2)

BEGIN

INSERT INTO [dbo].[UserRole]

([UserID]

,[RoleID]

,[CreatedDate]

,[CreatedBy]

,[LastUpdated]

,[UpdatedBy])

VALUES

(@UserID

,2

,GETUTCDATE()

,'Admin'

,GETUTCDATE()

,'Admin')

END

## Troubleshooting Tips

If you get an error message “**SSL certificate problem: self signed certificate in certificate chain**” then follow the instructions from Sharepoint

<https://pwceur.sharepoint.com/sites/VisualStudioTeamServices/SitePages/SSL%20certificate%20problem%20self%20signed%20certificate%20in%20certificate%20chain.aspx>

If you receive the Cypress App could not be downloaded error, enter one of the following depending on your shell:

CMD: **SET NODE\_TLS\_REJECT\_UNAUTHORIZED=0**

Powershell: **$Env:NODE\_TLS\_REJECT\_UNAUTHORIZED=0**

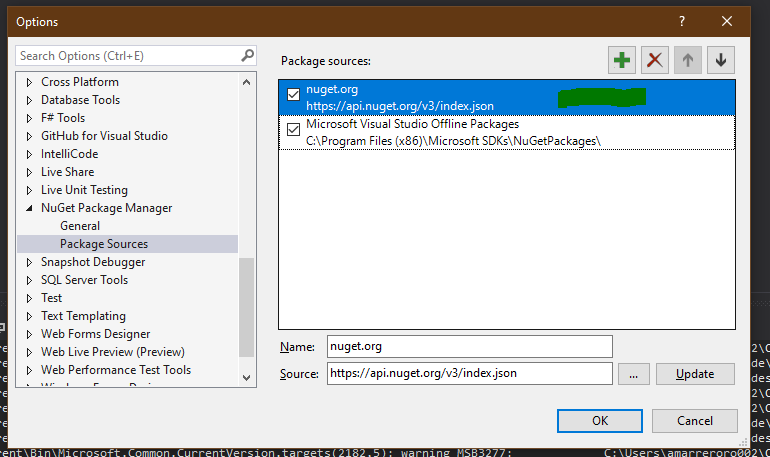
**--**

If you get `npm install` error with phantomjs, do the following:

npm install phantomjs-prebuilt@2.1.16 --ignore-scripts

**--**

If you Open a Visual Studio solution such as Fit.Sln and the build gives a lot of errors. Possible Fix: Make sure this nuget.org package source is set.g



Here is the link to the install of the Windows wrapper - <https://github.com/MicrosoftArchive/redis/releases> or here <https://redis.io/download>