Raspberry Pi

Folder

- OS: Raspbian
- Raspbian Installation Guideline Energie AG (German) (Node Red not used)

Installing General Stuff

Timezone muss auf Österreich gestellt sein! => sudo raspi-config

- install python3 and modules
  - sudo apt-get install python3 python3-venv python3-pip
- install python module Typer (used for connecting raspberry to a network)
  - pip3 install typer
- install python module wifi
  - pip3 install wifi
- install python module web3
  - pip3 install web3
- Install avahi-daemon (mDNS)
  - sudo apt-get install avahi-daemon
  - Edit /etc/avahi/avahi-daemon.conf
    - `use-ipv6=no`
    - `publish-workstation=yes`

Installing .NET

© https://edi.wang/post/2019/9/29/setup-net-core-30-runtime-and-sdk-on-raspberry-pi-4

- move to home directory
  - cd ~
- Download
  - wget https://download.visualstudio.microsoft.com/download/pr/e41a177d-9f0b-4afe-97a4-53587cd89d00/c2c897aa6442d49c1d2d86abb23c20b2/dotnet-sdk-6.0.202-linux-arm.tar.gz
  - wget https://download.visualstudio.microsoft.com/download/pr/adc5bbf5-6cf6-4da6-be27-60de0b8739e5/fecb289bd70834203f2397c18c82bbde/aspnetcore-runtime-6.0.4-linux-arm.tar.gz
- Unpack Downloads
  - mkdir dotnet-arm32
  - tar zxf dotnet-sdk-6.0.202-linux-arm.tar.gz -C $HOME/dotnet-arm32
  - tar zxf aspnetcore-runtime-6.0.4-linux-arm.tar.gz -C $HOME/dotnet-arm32
- Environment Variables
  - nano ~/.bashrc
  - LD_LIBRARY_PATH=/usr/local/lib

    - export LD_LIBRARY_PATH
    - DOTNET_ROOT=$HOME/dotnet-arm32
    - export DOTNET_ROOT
    - PATH=$PATH:$HOME/dotnet-arm32
    - export PATH

Setup VPN

- DDNS
    - create account at noip.com
    - https://www.noip.com/support/knowledgebase/install-ip-duc-onto-raspberry-pi/
- VPN
    - https://pimylifeup.com/raspberry-pi-wireguard/
- Router
    - enable port forwarding
    - enable DDNS
- Smartphone
    - install WireGuard

Install C++ Library (slave) from Energie AG

Energie AG installation guide (German)

Use the two .7z files in this folder: ExternalComponents\smart-meter-slave\Energie AG\Linux_CPP

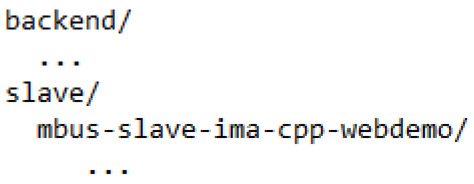
AES Key

Request AES Key from Energie AG

change it in the config.ini file of the slave

Directory Structure

You must have the following directory structure in your folder {YOUR_FOLDER}. Otherwise, the backend will not work.



```
backend/
  ...
slave/
  mbus-slave-ima-cpp-webdemo/
     ...
```

- backend
    - unzipped backend

Startup

- start slave
    - cd {YOUR_FOLDER}/slave/mbus-slave-ima-cpp-webdemo

- start backend
  - cd ../../backend
  - dotnet SmartMeterAPI.dll --urls http://*:8080

Linux systemd services

e-community-local/services copy to raspberry (home/pi/services).

Follow readme.txt instructions and install c# and slave service as system service.

Consent Management – Blockchain

GoQuorum Ethereum Client https://github.com/ConsenSys/quorum

- setup (Blockchain node and cross-compile to Raspberry architecture (arm-v7)
  - git clone https://github.com/ConsenSys/quorum.git
  - env GO111MODULE=on CGO_ENABLED=1 CC=gcc go build -o build/bin/geth-linux-arm-7 ./cmd/geth
- create blockchain node (Script) and copy to the raspberry $HOME/blockchain/data
  - ./create_node.sh
- run linux system service (e-community-blockchain.service – install Script)
  - ./install.sh

Smart Contract IDE https://remix.ethereum.org/

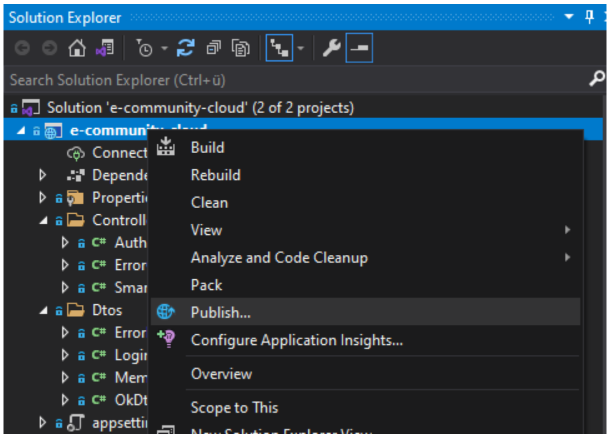
Deployment of Smart Contracts https://trufflesuite.com/

Backend

Download Visual Studio: https://visualstudio.microsoft.com/de/downloads/

Publishing in general

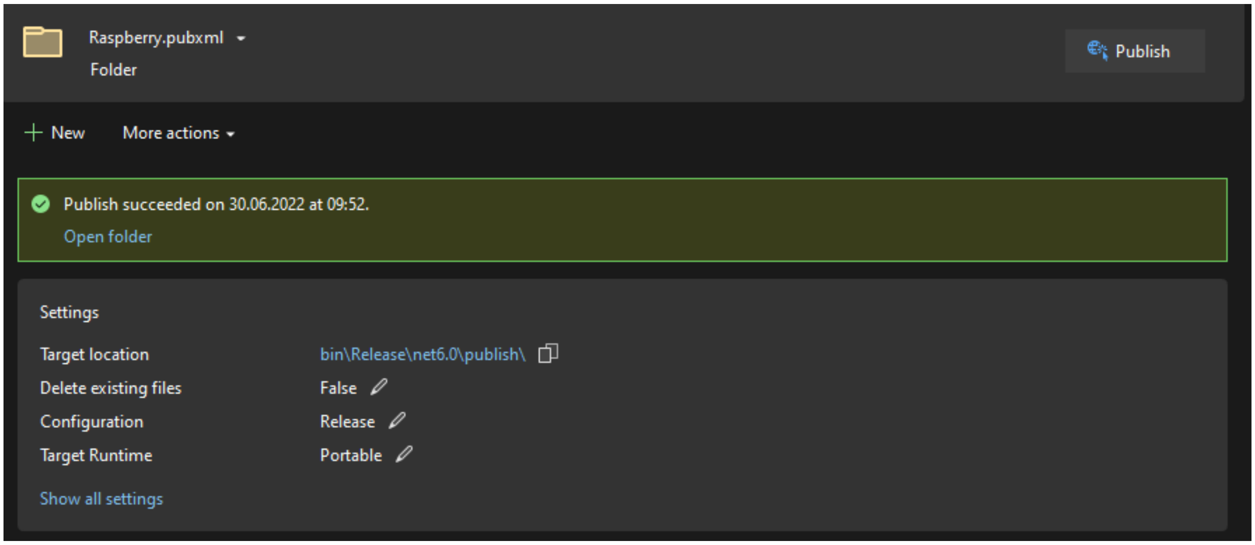
Right Click on SmartMeterAPI Project



Local Backend

Folder

Publish



Raspberry.pubxml
Folder

Publish succeeded on 30.06.2022 at 09:52.

Settings

Target location	bin\Release\net6.0\publish\
Delete existing files	False
Configuration	Release
Target Runtime	Portable

Select a location and press publish. Copy the selected folder onto your raspberry's backend folder (zip would help). Attention: stop backend before

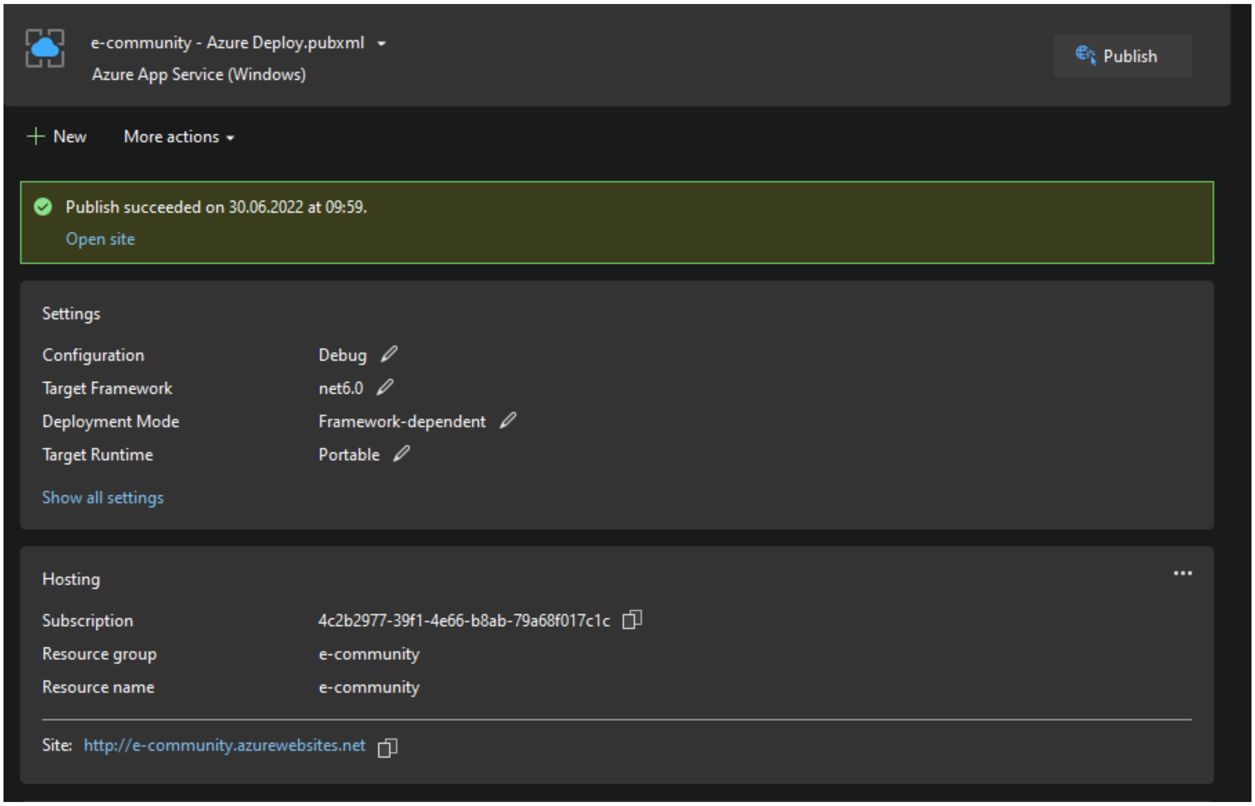
Cloud Backend

Folder

DB changes

- change model classes as desired
- Package Manager Console in …Context folder
  - Add-Migration {some_name}
  - Update-Database

Publish



e-community - Azure Deploy.pubxml
Azure App Service (Windows)

Publish succeeded on 30.06.2022 at 09:59.

Settings

Configuration	Debug
Target Framework	net6.0
Deployment Mode	Framework-dependent
Target Runtime	Portable

Hosting

Subscription	4c2b2977-39f1-4e66-b8ab-79a68f017c1c
Resource group	e-community
Resource name	e-community

Site: http://e-community.azurewebsites.net

Android

Folder

Download Android Studio: https://developer.android.com/studio

- run on phone
  - run project with android studio (activate USB Debugging on your phone)

Web

Folder

- publish
  - ng build --prod in root directory
  - copy dist folder onto web server

Continue Working …

Every Class/Method is documented with inline comments (XML Documentation in backend and Javadoc in android). Please make use of them.

The current project state is working with:

- Azure SQL Database
- Azure App Service
- Database local
    - SQLite3 (locally stored)
    - e-community-local.db
- Backend
    - Entity Framework Core 6.0.9
        - DB Connection
        - SqlServer, Tools, Design
    - .NET Core 6.0.4
        - REST service
        - …Controller
    - SignalR 1.18.3 and SignalR.Client 6.0.9
        - Local <-> Cloud
        - Cloud <-> End Device
        - …Hub
    - Identity.EntityFrameworkCore 6.0.9
    - Serilog 6.0.1, Serilog.Sinks.File 5.0.0
        - Logging mechanism
    - Swashbuckle 6.4.0
        - Swagger
    - Only Cloud
        - JwtBearer 6.0.9 and OpenIDConnect 6.0.9
            - Token generation
        - OpenIdConnnect 6.0.4
            - Authentication
        - MailKit 3.4.1
            - sending mails
        - FirebaseAdmin 2.3.0
            - Firebase Cloud Messaging
    - Only Local
        - Sqlite 6.0.4
            - connection to SQLite database
        - Newtonsoft.Json
            - JSON parser
- Android
    - Android 13
    - see build.gradle
- Web: Angular CLI 13.1.3