

Popis aktuálního stavu IoT Platformy

Architektura

Architektura uživatelské části je klasická server-client. Klientem je zde webová aplikace, backend je v NodeJS a komunikace probíhá přes Rest API (+ Web Socket pro aktuální hodnoty). Backend je připojený na Mqtt broker přes který komunikuje s jednotlivými zařízeními - získává data a odesílá příkazy. Pro ukládání statických dat jako jsou uživatelská data, konfigurace zařízení atd. Je využita MongoDB databáze. Pro časové řady, data získávaná ze zařízení v průběhu času, je použita InfluxDB databáze. Pro vizualizaci dat používám Grafanu - webová aplikace umožňuje zobrazení dat pouze za posledních 24h. Pro automatizaci jsem využil Node-red, který pomocí vizuálního programování dokáže celkem hezky vytvářet reakce na událost - dopsal jsem si do něj autentizační plugin, takže se uživatel přihlásí přes stejný účet který má na Platformě.

Backend

Git monorepo: <https://github.com/founek2/IOT-Platforma>

Serverová část je rozdělena na 3 micro služby:

- Rest API (backend) - současně služba slouží jako Gateway - zpřístupňuje ostatní služby přes API
- obsluha MQTT (backend-mqtt) - ukládá data a odesílá příkazy
- autorizace (backend-auth) - řeší přihlášení uživatelů, validaci tokenů a oprávnění

Snažil jsem se celou architekturu přetvořit na MicroServices, ale stále to ještě není úplně hotové - chybí dokončení přesunu zodpovědností (některé služby pracují s daty ke kterým by měli přistupovat přes jiné služby). Technologicky je použit webový framework ExpressJS a knihovna Mongoose pro práce s Mongo databází. Součástí služby pro autorizaci je API, které provolává Authentication plugin pro RabbitMQ (Mqtt broker) - přes toto je řízen přístup k jednotlivým tématům (topics) pro uživatele a zařízení -> poměrně vysoká úroveň bezpečnosti.

Frontend

Git monorepo: <https://github.com/founek2/IOT-Platforma/tree/develop/packages/frontend-ng>

Frontend je napsaný s použitím knihoven React, Redux Toolkit a Material UI. Asi před půl rokem jsem ho celý přepsal aby používal novější technologický stack, vylepšil kešovací vrstvu a přidal možnost změny pořadí zobrazených ovládacích prvků a jednotlivých místností zobrazených v aplikaci - dříve toto nešlo měnit žádným způsobem. Aplikace obsahuje následující funkce:

- Registrace uživatelů
- Přihlášení jménem a heslem
- Přihlášení přes OAuth přes seznam.cz
- Správa vlastních zařízení
- Přidání/odebrání práv k zařízení
- Přehled budov/místností ve kterých se zařízení nachází
- Ovládání jednotlivých zařízení a zobrazení hodnot za posledních 24h
- Správa uživatelů pro administrátory
- Nastavení Push notifikací

Koncová zařízení

Git: <https://github.com/founek2/IoT-Platform-library.git>

Git: <https://github.com/founek2/IOT-Platforma-bots/tree/master>

Koncová zařízení jsou primárně HW čipy ESP8266 nebo již nově podporuji i ESP32. Mám napsanou knihovnu pro Arduino framework pomocí které se zadefinuje chování daného zařízení - jaké hodnoty bude posílat, co lze ovládat atd... Následně po spuštění zařízení vytvoří wifi hotspot, uživatel se k němu připojí a zadá k jaké wifi se má zařízení připojit, url IoT Platformy a své uživatelské jméno. Následně se zařízení připojí k platformě a uživatel ve webové aplikaci uvidí nové zařízení, které si může přidat. Následně proběhne automatická výměna API klíče, které si zařízení uloží a ho používá pro přístup k MQTT aby bylo ověřené. Veškerá komunikace je šifrovaná.

Dalším druhým koncových zařízení jsou boti. Aplikace napsané v NodeJS, která se chová jako zařízení - část používám pro testování kdy posílají náhodná data nebo jsem je využíval pro performance testing. Dalším druhem botů je tzv. Bridge - slouží pro překlad mezi nějakým rozhraním a mojí platformou. Aktuálně mám napsané bridge pro Sony Bravia televizi, Intex výřivku a Solax Invertor (solární elektrárna). Nově jsem napsal prvotní verzi bridge pro zigbee2mqtt. Aktuálně otestováno že funguje pro Sonoff spínač (zigbee relátka na 230V).

Hosting

Git: <https://github.com/founek2/IOT-Platforma-hosting>

Platforma je celá dockerizovaná a její nasazení probíhá pomocí připraveného docker-compose viz. repozitář. Stačí vyplnit .env soubor a celá platforma se spustí - jednotlivé služby včetně InfluxDB, MongoDB, RabbitMQ, Grafana a Node-red.

