

Seminarska naloga pri predmetu računalništvo

RIPtide - Univerzalno orodje za konfiguracijo omrežij

Mentor: Marko Kastelic, prof.

Avtor: Jurij Fortuna , G 3. a

Ljubljana, 24. maj 2023

Povzetek

V tej seminarski nalogi je opisana programska oprema RIPTide. Je odprtokodna in omogoča komurkoli, da s pomočjo RIPTide APIja ustvari podporo za svojo omrežno opremo ali uporablja že napisano.

Ključne besede: omrežna oprema, API, Java, vtičniki, odprtokodna programska oprema

Abstract

This seminar task describes the RIPTide software. It is open-source and allows anyone to create support for their network equipment or use what has already been written, using the RIPTide API.

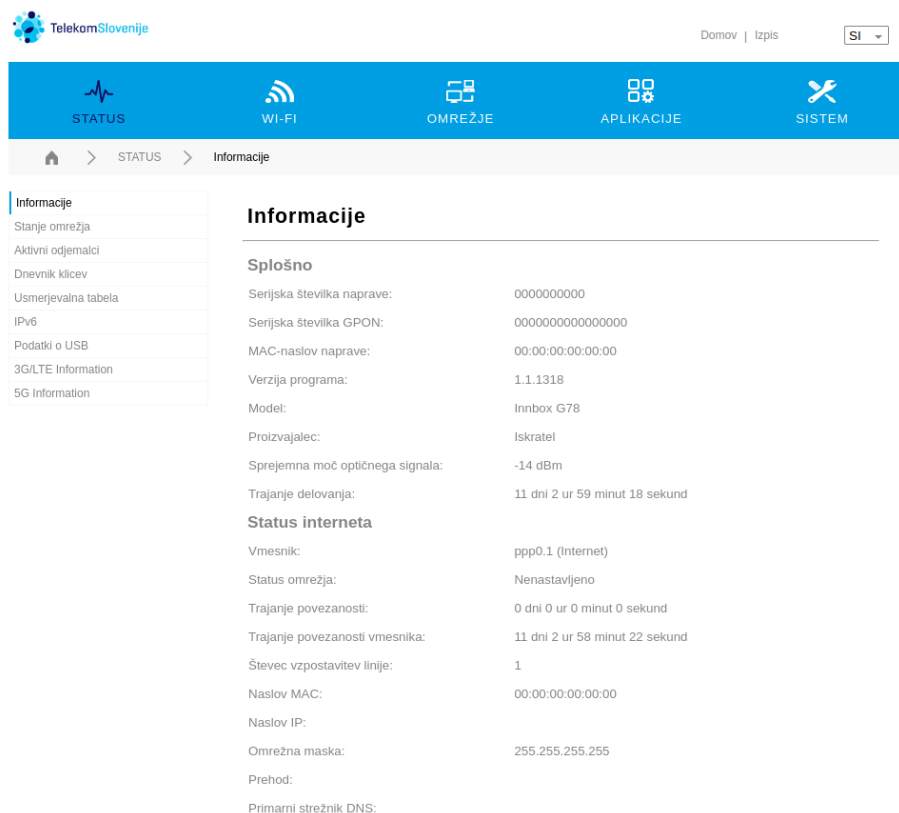
Keywords: network equipment, API, Java, plugins, open source software

Kazalo

1	Uvod	4
2	Uporabljene tehnologije	5
3	Ospredni del	6
3.1	Konfiguracija	6
3.1.1	Shranjevanje podatkov	6
3.1.2	Barve vmesnika	7
3.2	Upravljanje poverilnic	7
3.3	Rokovanje z napravami	7
3.3.1	Nalaganje Shedov	7
3.3.2	Nalaganje naprav	7
4	Aplikacijski programski vmesnik	8
4.1	Dostop do API-ja	8
4.2	Struktura API-ja	8
4.3	Vgrajena orodja	8
4.4	Komunikacija z napravami	8
4.5	Čarovniki	10
5	Shedi	11
5.1	Priprava projekta	11
5.2	Projektna struktura	11
5.2.1	Metapodatki	11
5.2.2	Gonilni razredi	11
5.2.3	Zavihki	11
5.3	Izgradnja in uporaba	11
6	Eksperimentalne funkcije	12
6.1	Topološki pogled omrežja	12
7	Zaključek	13

1 Uvod

Ideja za razvoj RIPTide-a se je pojavila, ko sem bil med konfiguracijo domačega omrežja prisiljen uporabljati tri različne programske opreme, različnih proizvajalcev. Med sabo so se nemalo razlikovale in so bile po večini nestabilne. Zato sem se odločil razviti generično programsko opremo za konfiguracijo omrežij. Deluje na principu “vtičnikov” (t.i. Shedov) za posamezne kose omrežne opreme.



Slika 1: Primer konfiguracijske strani za usmerjevalnike Telekoma Slovenije

2 Uporabljene tehnologije

Tako Shedi, kot tudi RIPTide so napisani v Javi. Java omogoča dinamično nalaganje modulov ob zagonu navideznega stroja, tudi iz zapakiranih JAR datotek. Sam RIPTide je zgrajen iz dveh glavnih delov: osrednega dela in aplikacijskega programskega vmesnika (v nadaljevanju API). Poleg tega velja omeniti Shede, ki jih lahko razvije kdorkoli z uporabo RIPTide API-ja.

3 Ospredni del

Ospredni del je odgovoren za konfiguracijo RIPTide-a ter rokovanje z uporabnikovimi poverilnicami in napravami.

3.1 Konfiguracija

Uporabnik lahko svoje naprave shrani, in si s tem prihrani čas za morebitno kasnejšo konfiguracijo. Poleg tega RIPTide omogoča spremembo barve vmesnika po uporabnikovi želji.

3.1.1 Shranjevanje podatkov

Okoljske spremenljivke, naprave in poverilnice so zapisane v objektu imenovanem `Workspace`.

```
@Data
public class Workspace implements Serializable {
    private Theme theme;
    private ArrayList<Credential> credentials;
    private ArrayList<Connection> connections;

    public Workspace() {
        theme = Theme.PRIMER_DARK;
        credentials = new ArrayList<>();
        connections = new ArrayList<>();
    }

    public Workspace(Theme theme, ArrayList<Credential>
        credentials, ArrayList<Connection> connections) {
        this.theme = theme;
        this.credentials = credentials;
        this.connections = connections;
    }
}
```

Ko uporabnik nastavitve shrani je objekt serializiran in zapisan v datoteko na uporabnikov sistem. Datotek je lahko več, kar omogoča prilagoditev vmesnika za različna okolja (npr. posebej za domačo in poslovno rabo). `Workspace` datoteke

so shranjene v JSON formatu, kar uporabnikom omogoča enostavno izmenjavo ali prenos.

3.1.2 Barve vmesnika

Uporabnik lahko vmesnik prilagodi na štiri priložene teme. V prihodnosti bo mogoče nalaganje barv iz CSS datoteke po meri.

3.2 Upravljanje poverilnic

Za konfiguracijo večine omrežnih naprav je potrebna avtorizacija. RIPTide uporabnikom omogoča varno shranjevanje poverilnic na njihovem sistemskem keyringu. Poverilnice so ob povezavi na napravo prek API-ja podani Shedu, kar pomeni, da se razvijalci teh ne rabijo ukvarjati z varnim hranjenjem poverilnic.

3.3 Rokovanje z napravami

3.3.1 Nalaganje Shedov

Nalaganje shedov poteka v dveh korakih, lociranje Sheda in branje metapodatkov. Na uporabnikovem sistemu se nahajajo v obliki JAR datotek v dveh mapah, `~/.config/RIPTide/sheds` na NIX sistemih in `%UserProfile%\ .RIPTide\sheds` na Windows sistemih.

3.3.2 Nalaganje naprav

Ob izbiri naprave, RIPTide iz Shedovih metapodatkov najprej prebere njen t.i. model path, ki izgleda približno tako: `telekomslovenije.innboxg78.ts`. Niz je sestavljen iz ID-ja Sheda, ID-ja modela naprave in ID-ja različice modela. Vsi ID-ji so edinstveni, kar pomeni, da lahko program najde Shed, v njem poišče model naprave, njegovo različico in vzpostavi povezavo z napravo prek API-ja.

4 Aplikacijski programski vmesnik

API omogoča uporabnikom, da ustvarijo lastno podporo za omrežno opremo, ki je programska oprema še ne podpira. To pomeni, da lahko skrbniki omrežij enostavno integrirajo nove naprave v svoje omrežje, ne da bi morali čakati, da prodajalec izda uradno podporo.

4.1 Dostop do API-ja

Dostop do APIja je mogoč preko SDK knjižnice. Nahaja se v RIPTide Maven repozitoriju. Ker so Javanski upravitelji paketov med seboj bolj kot ne kompatibilni, je tudi postopek dodajanja knjižnice med njimi zelo podoben. Več o projektni strukturi pa v razdelku 5.2.

4.2 Struktura API-ja

Vsi pomembni vmesniki in razredi za pisanje Shedov se nahajajo v paketu `org.riptide.sdk.sheds`. Paket je deljen na podpakete za uporabniški vmesnik, tipe naprav, čarovnike in avtentikacijo.

4.3 Vgrajena orodja

API vsebuje uporabna orodja za izdelovanje statusnih strani in formularjev, ki se nahajajo v paketu `org.riptide.sdk.sheds.ui`.

4.4 Komunikacija z napravami

Ospredni del s Shedi komunicira preko vmesnikov. Za primer vzemimo Telekomov usmerjevalnik. Ob izbiri naprave se instancira njen gonilni razred v nov objekt, ki je odgovoren za komunikacijo z napravo (postopek je opisan v razdelku 3.3.2).

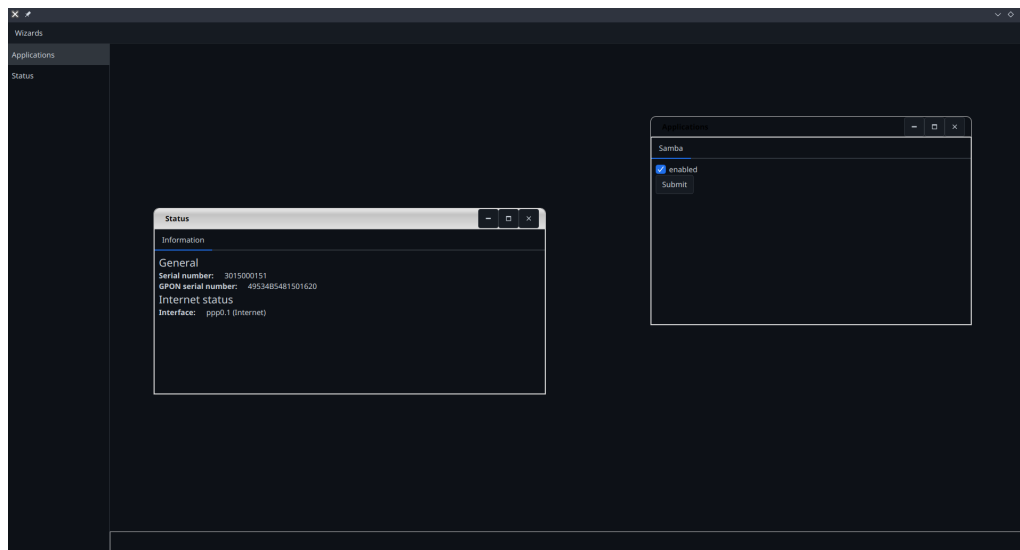
Program iz metapodatkov razbere, da naprava tipa router. To pomeni, da mora njen gonilni razred implementirati vmesnik `Router` in posledično `Device`, saj ga `Router` razširja.


```
public interface Device {
    void initialize(String address, Credential credential);
    TreeMap<String, Tab[]> getPages();
}
```

```
public interface Router extends Device {
    PATWizard patWizard();
}
```

Za tem se kliče metoda `initialize/2`, ki gonilnemu objektu poda omrežni naslov naprave in poverilnice.

Nato program od gonilnega objekta zahteva seznam konfiguracijskih strani in zavihke z metodo `getPages/1`. Strani so predstavljene s tabelami `Tab` objektov, ki so preslikane v nize. Shedi imajo običajno razrede, ki implementirajo vmesnik `Tab`, ti se instancirajo in vrnejo v metodi. Ospredni del nato z orodnim razredom `ContentRenderer` generira stran, kot je prikazana spodaj.



Slika 2: Primer generirane konfiguracijske strani

4.5 Čarovniki

Program tudi manj naprednim uporabnikom omogoča konfiguracijo njihove omrežne opreme. Na konfiguracijski strani za katerokoli omrežno napravo se v menijski vrstici nahaja gumb 'Wizards'.

5 Shedi

V prihodnosti načrtujem omogočiti razširitev tudi drugih funkcionalnosti. Zaradi tega sem se za poimenovanje vtičnikov naprav odločil uporabiti izraz 'shedi' namesto običajnega izraza 'plugin'.

5.1 Priprava projekta

5.2 Projektna struktura

5.2.1 Metapodatki

5.2.2 Gonilni razredi

5.2.3 Zavihki

5.3 Izgradnja in uporaba

6 Eksperimentalne funkcije

RIPtide je opremljen z nekaj novimi eksperimentalnimi funkcijami, ki pa jih zaradi pomanjkanja časa nisem uspel povsem dokončati in stestirati. Te omogočajo naprednim uporabnikom nove možnosti avtomacije in pregleda nad omrežjem.

6.1 Topološki pogled omrežja

7 Zaključek