Подигнување на сервер на локална околина

ФАКУЛТЕТ ЗА ЕЛЕКТРОТЕХНИКА И ИНФОРМАЦИСКИ ТЕХНОЛОГИИ

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGIES

Тамара Митевска

- E-mail: tamara_mitevska@hotmail.com
- LinkedIn: https://www.linkedin.com/in/tamara-mitevska-988737197/
- Github: https://github.com/TMitevska

Содржина на втор дел од курс

1. Подигнување на сервер на локална околина
- ❑ Запознавање со NodeJS
- ❑ Подигнување на сервер на работна околина
- ❑ Далечинско пристапување
- ❑ Практични примери

2. Работа во облак
- ❑ Вовед за облак
- ❑ Запознавање со околината на неколку облаци
- ❑ Вовед во бази со податоци, разлики помеѓу релациони и нерелациони бази
- ❑ Практични примери

Тамара Митевска
Факултет за електротехника и информациски технологии

3.Синхронизација во реално време помеѓу Raspberry Pi и други апликации поврзани на облак

- Работа со готови апликации
- Споделување на податоци помеѓу сите уреди вклучени на ист облак
- Градење на паметна соба или уред
- Практичен проект

Тамара Митевска
Факултет за електротехника и информациски технологии

Методологија

1. 80% теорија - 20% вежби

2. Секоја презентација има вежби кои треба сами да ги изработите

3. Презентации:
https://github.com/TMitevska/raspberryPiCourse

4. Решенија од вежби:
https://github.com/TMitevska/raspberryPiCourse

Вовед во Node.js

Тамара Митевска
Факултет за електротехника и информациски технологии

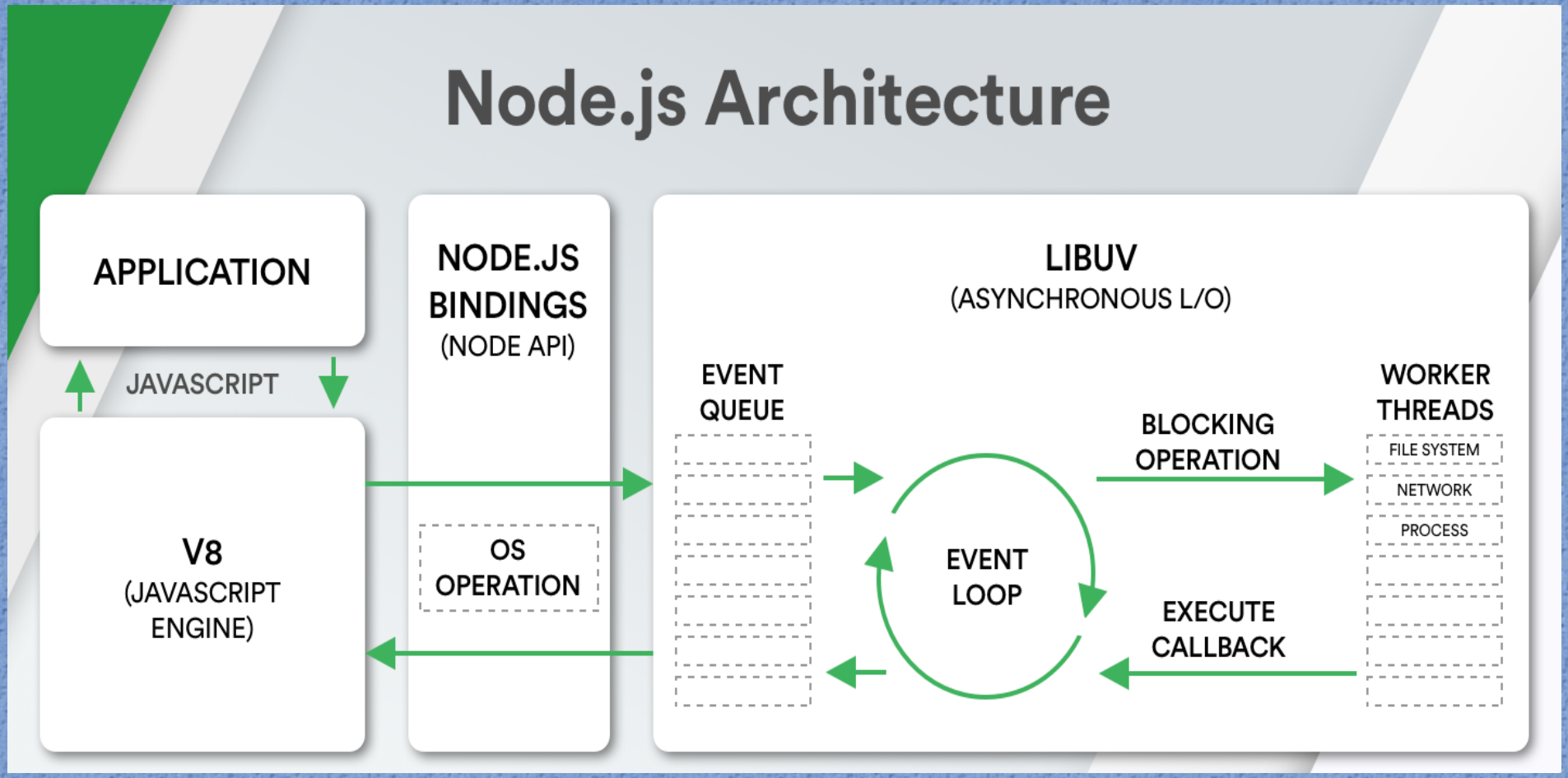
Што е Node.js?

- Node.js е една од најкористените платформи за преведување на JavaScript код во машински код разбирлив за секој компјутер. Базиран на V8 JavaScript механизмот користен во Google Chrome.
- Оваа платформа е бесплатна и е креирана во 2009 година од страна на Ryan Dahl.
- Node.js е повеќенаменска платформа која најчесто се користи за изработка на брзи и скалабилни веб апликации и сервиси. Сите Node.js апликации се пишуваат во JavaScript, а потоа може да се пуштат во Node.js runtime околина на било кој оперативен систем.

Node.js = Runtime Environment + JavaScript Library

Тамара Митевска
Факултет за електротехника и информациски технологии

Node.js Architecture



Тамара Митевска
Факултет за електротехника и информациски технологии

Карактеристики на Node.js

- Asynchronous and Event Driven - сите API-а на Node.js се асинхрони, односно не се блокирачки. Тоа значи дека серверот базиран на Node.js никогаш не чека одговор на некое барање од некое API. Серверот се префрла на следнито API откако ќе го повика претходното, а механизмот за известување за настани на Node.js му помага на серверот да добие одговорот на барањето од претходниот повик на некое API и соодветно да се справи со него.
- Брз - Бидејќи е базиран на V8 JavaScript Engine на Google Chrome, библиотеката Node.js е многу брза во извршувањето на кодот.
- Single Threaded, но многу скалабилен - Node.js механизнот се справува со секое барање последнователно, односно кога ќе заврши со претходно барање преминува на следното. Механизмот за настани му помага на серверот да одговори на не-блокирачки начин и го прави серверот многу скалабилен за разлика од традиционалните сервери кои создаваат повеќе нишки за да се справат со барањата. Node.js користи една нишка и истата може да се справи со поголем број барања отколку традиционалните сервери како Apache HTTP серверот.
- No Buffering - апликациите на Node.js никогаш не ставаат податоци во бафери.

Тамара Митевска
Факултет за електротехника и информациски технологии

Што може да направи Node.js?

- Node.js може да генерира динамична содржина на страница
- Node.js може да креира, отвора, чита, пишува, брише и затвора датотеки на сервер
- Node.js може да собира податоци за некоја форма од апликација
- Node.js може да додава, брише, модифицира податоци во база на податоци
- Датотеките на Node.js имаат наставка ".js"

Тамара Митевска
Факултет за електротехника и информациски технологии

Каде да се користи Node.js?

Node.js е совршен избор при креирање на:

- Апликации за влезни / излезни уреди
- Апликации за пренос на податоци
- Апликации во реално време
- Апликации базирани на JSON API
- Single Page апликации

Каде да не се користи Node.js?

Не е препорачливо да се користи Node.js за апликации кои бараат искористување на максимум ресурси на процесор.

Тамара Митевска
Факултет за електротехника и информациски технологии

Кој го користи Node.js?

Според податоците на github, многу проекти, апликации и компании кои користат Node.js. Оваа листа ги вклучува eBay, General Electric, GoDaddy, Microsoft, PayPal, Uber, Wikipins, Yahoo! и многу други.

Тамара Митевска
Факултет за електротехника и информациски технологии

Зошто Node.js?

Node.js користи асинхроно програмирање!

Заедничка задача за веб-сервер може да биде отворање датотека на серверот и враќање на содржината до клиентот.

Еве како PHP или ASP се справува со барање за датотека:

- Се испраќа задачата до датотечниот систем на компјутерот.
- Се чека додека датотечниот систем се отвори и ја чита датотеката.
- Се враќа содржината до клиентот.
- Подготвен е да се справи со следното барање.

Еве како Node.js се справува со барање за датотека:

- Се испраќа задачата до датотечниот систем на компјутерот.
- Подготвен е да се справи со следно барање.
- Кога датотечниот систем ќе ја отвори и ќе ја прочита датотеката, серверот ја враќа содржината до клиентот.
- Node.js го елиминира чекањето и едноставно продолжува со следното барање.

Node.js е single-threaded, не-блокирачки јазик кој нуди асинхроно програмирање, што е многу ефикасно во меморијата.

Тамара Митевска
Факултет за електротехника и информациски технологии

HTTP барање

- Verb: Кажува за кој HTTP метод станува збор (GET, POST, DELETE, PUT)
- URI: Uniform Resource Identifer (URI) патека
- HTTP version: верзија - “HTTP v1.1”.
- Request header: Содржи метадата (прегледувач, верзија на прегледувач, дополнителни информации за барањето)
- Request body: порака што се праќа

Тамара Митевска
Факултет за електротехника и информациски технологии

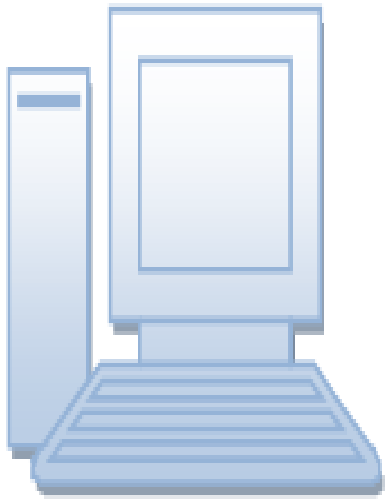
HTTP одговор

- Status/response code: Кажува дали е добиен бараниот ресурс. E.g. 404 значи дека ресурсот не е пронајден, додека 200 значи дека е пронајден
- HTTP version: верзија “HTTP v1.1”.
- Response header: Содржи метадата како должина на содржина, тип, тип на сервер и слично
- Response body: Ресурс

(2) Browser sends a request message

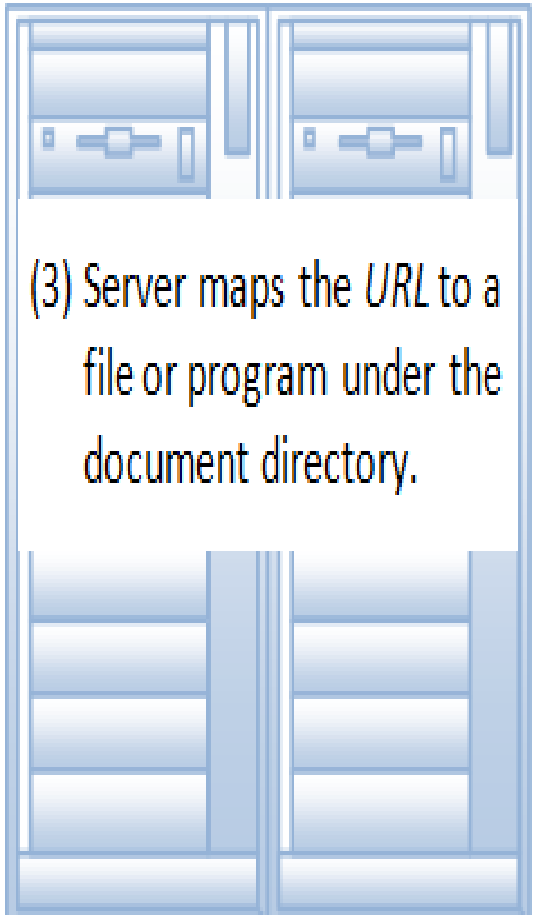
(1) User issues URL from a browser

http://host:port/path/file



GET *URL* HTTP/1.1
Host: *host:port*
.....
.....

(3) Server maps the *URL* to a file or program under the document directory.



(4) Server returns a response message

HTTP/1.1 200 OK
.....
.....
.....

(5) Browser formats the response and displays

Client (Browser)

HTTP (Over TCP/IP)

Server (@ *host:port*)

Тамара Митевска
Факултет за електротехника и информациски технологии

HTTP методи

- GET - Се користи за повлекување на ресурси
- POST - Се користи за креирање на нов ресурс
- PUT - Се користи за изменување на веќе постоечки ресурс или доколку истиот не постои креира нов ресурс
- DELETE - Се користи за бришење на ресурс

HTTP метод	URI	Description
GET	/course (http://feit.com/course)	Преземање на сите курсеви
GET	/course/1 (http://feit.com/cours/1)	Преземање на курс со id 1
POST	/courses (http://feit.com/courses)	Креирање на нов курс
PUT	/courses/1 (http://feit.com/courses/1)	Промена на курс со id 1
DELETE	/courses/1 (http://feit.com/courses/1)	Бришење на курс со id 1

JSON

JSON (JavaScript Object Notation) е формат кој ги дефинира податоците кои се праќаат од/до некое REST API.
JSON објектот изгледа како JavaScript објект и се составен од парови име/вредност.


```
{
    "data": {
        "id": 1,
        "name": "Dr. Irma Mohr",
        "email": "schumm.thora@example.com",
        "created_at": "2017-09-05 09:15:30",
        "updated_at": "2017-09-05 09:15:30"
    },
    "version": "1.0",
    "success": true
}
```


Тамара Митевска
Факултет за електротехника и информациски технологии

Далечинско пристапување

Тамара Митевска
Факултет за електротехника и информациски технологии

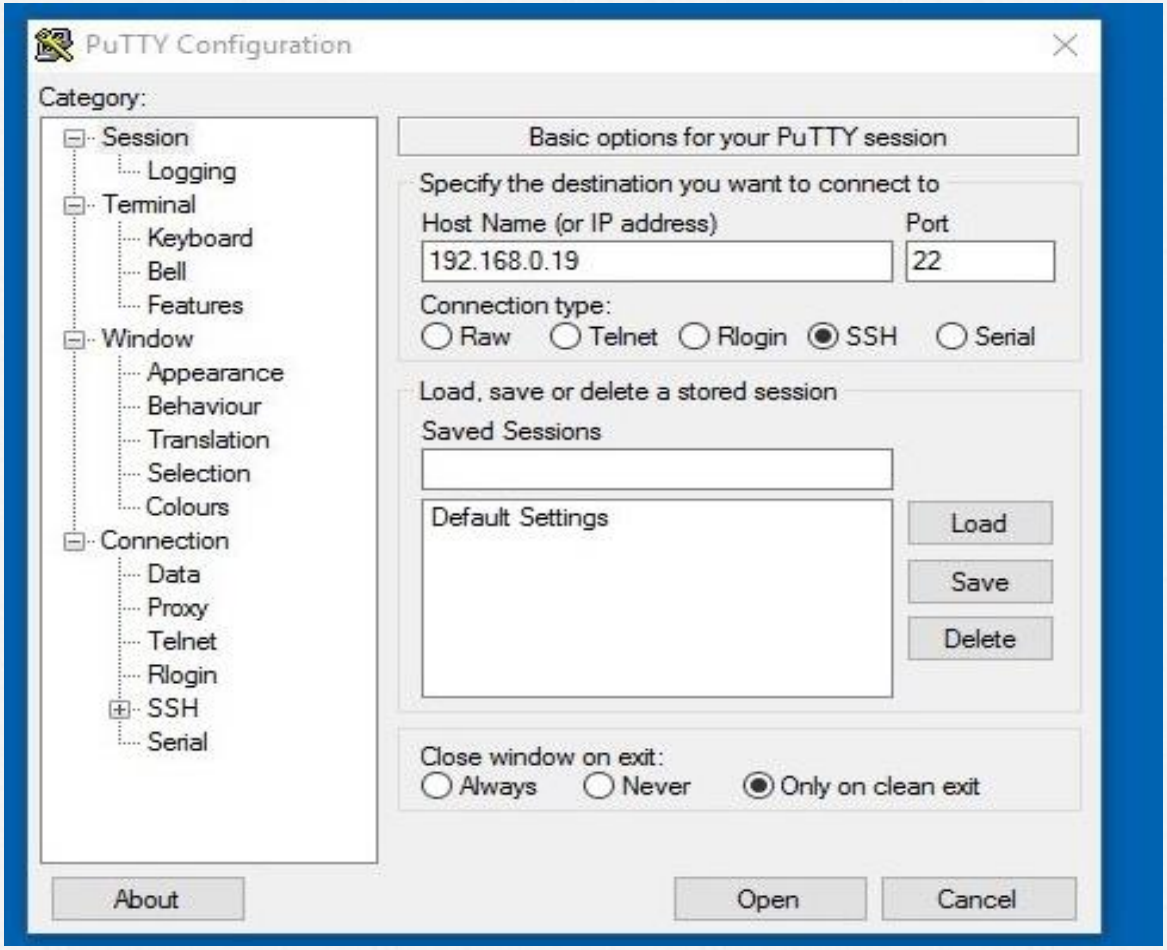
Понекогаш треба да пристапите до Raspberry Pi без да го поврзете со монитор. Можеби Raspberry Pi е вграден во нешто како робот, или можеби ќе сакате да прегледате некои информации од него од друго место или пак можеби едноставно немате резервен монитор.

Постојат повеќе начини за далечинско пристапување и тоа:

- **SSH**
- **Putty**
- **TightVNC**

Тамара Митевска
Факултет за електротехника и информациски технологии

Putty



Тамара Митевска
Факултет за електротехника и информациски технологии

SSH

SSH („Secure Shell“) е шифрирана технологија за вмрежување која ви овозможува да управувате со компјутерите од командната линија преку мрежа.

SSH е корисен ако сакате брзо да се поврзете со Raspberry Pi од конзола на друг компјутер. Исто така е идеален за лесни дистрибутивни инсталации кои немаат графички интерфејси и проекти кои немаат екран (како што се роботи). Особено е корисно кога креирате проекти на Интернет на нештата (IoT), бидејќи тие можат да бидат вградени во друг хардвер.

Linux и macOS поддржуваат SSH и нема потреба од активација пред негово користење.

Windows поддржува SSH, но треба претходно да се активира. Кликнете на:

Search-->Manage Optional Features-->Add a feature-->Open SSH Client (Beta)

Тамара Митевска
Факултет за електротехника и информациски технологии

Активирање на SSH на Raspbian

Од безбедносни причини, Secure Shell не е вклучен стандардно во Raspbian. На вашиот Raspberry Pi, изберете:

Menu > Preferences > Raspberry Pi Configuration > Interfaces > Enable SSH

или

sudo systemctl enable ssh
sudo systemctl start ssh

За поврзување на Raspberry Pi во мрежа потребно е негово поврзување на интернет преку wireless LAN или преку Ethernet кабел. За да се дознае IP адреса потребно е во конзола да се напише следнава команда:

ping raspberrypi.local
192.168.0.41

За далечинско пристапување до Raspberry Pi потребно е од локален компјутер да се внесе следнава комада:

ssh pi@[IP]
ssh pi@192.168.0.41

Тамара Митевска
Факултет за електротехника и информациски технологии

TightVNC

Windows:

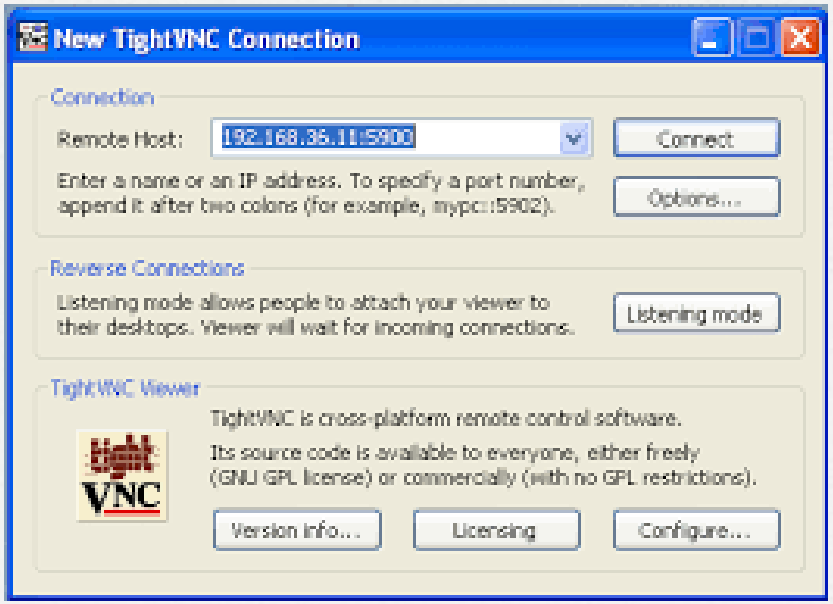
https://www.tightvnc.com/download.php

Linux:

sudo apt-get update

sudo apt-get install tightvncserver

vncserver



Тамара Митевска
Факултет за електротехника и информациски технологии

Прашања?

БЛАГОДАРАМ НА ВНИМАНИЕТО! 

Тамара Митевска
Факултет за електротехника и информациски технологии