

云计算与大数据应用开发

实验六：云计算应用开发（二）

丁烨

dingye@dgut.edu.cn

计算机科学与技术学院

2024-05-27



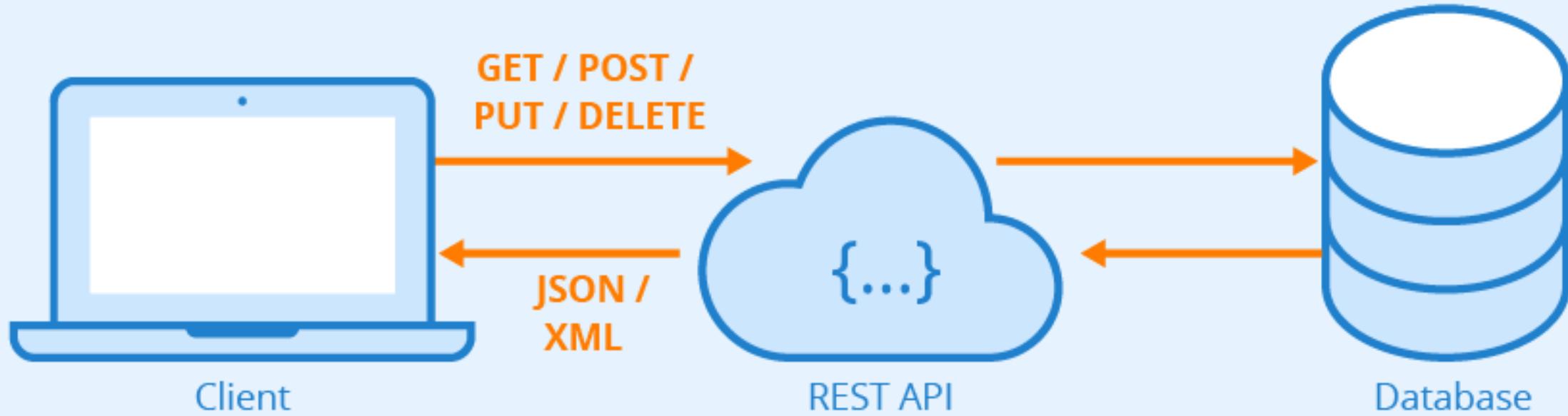
東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

- ❖ 表现层状态转换 (Representational State Transfer, REST)
- ❖ 由 Roy Thomas Fielding 于 2000 年提出来的一种互联网应用程序架构
- ❖ 目的是便于不同应用程序在互联网中互相传递信息
- ❖ 允许客户端发出统一的、无状态的资源标识符访问和操作网络资源
- ❖ 相对于其它种类的互联网应用程序架构，例如 SOAP (Simple Object Access Protocol) ， REST 的无状态特性降低了多平台应用程序的开发强度
- ❖ REST 是目前 SaaS 的主流技术架构

{ REST }

- ❖ 接口统一
- ❖ 客户端和服务端通常为 **JSON** 或 **XML**
- ❖ 服务端和数据库通常为 **ORM**

- ❖ 无状态
- ❖ 如有需要，客户端每次服务请求都需要携带验证信息
- ❖ 服务端每次数据请求都需要携带验证信息



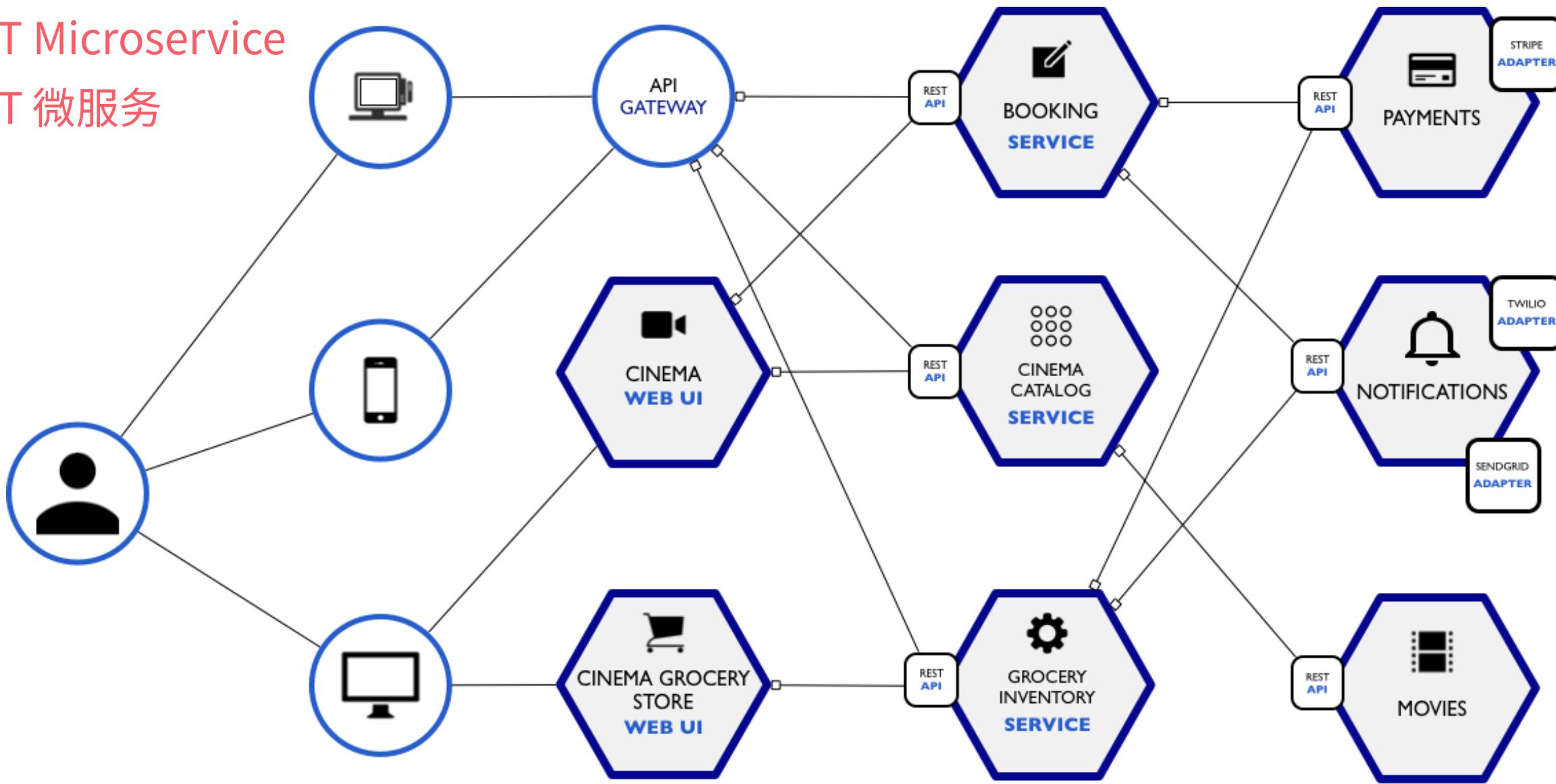
- ❖ REST 的优势
- ❖ 客户端、服务端、数据库可以任选架构，只要接口统一即可
- ❖ 客户端包括：PC 网页端、移动网页端，Windows / Mac / Linux 桌面应用程序、Android / iOS 移动端应用程序、微信小程序等
- ❖ 数据库包括：RDBMS 数据库，如 SQLite、MySQL、PostgreSQL、Oracle 等；NoSQL 数据库，如 Redis、MongoDB、Impala、Cassandra、HBase 等
- ❖ 服务端：即“应用程序接口”

- ❖ 不同的组件可以选择不同的运行环境，包括操作系统和硬件
- ❖ 不同的组件可以同时运行（例如，一个用户可以同时使用手机和电脑登陆“微信”）
- ❖ 只要接口统一即可

- ❖ REST 的优势
- ❖ REST 架构非常适合云计算时代的互联网应用
- ❖ 为了适应用户需求，大部分应用程序都包含不止一个终端
- ❖ REST 架构使得应用开发完全分离：前端工程师、后端工程师、数据库工程师
- ❖ Docker 为 REST 架构提供了非常方便的部署方案
- ❖ 在 REST 架构中，每个组件都可以非常方便的扩大服务规模（Scale Up）或减小服务规模（Scale Down）

REST 应用程序架构

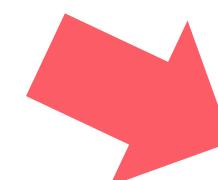
- ❖ REST Microservice
- ❖ REST 微服务



- ❖ 互联网应用前端 (User Interface, UI)
- ❖ 又称 “客户端”
- ❖ 与用户交互，并将用户操作转换为 API 请求
- ❖ 通过 API 与服务器通信，并取得用户希望获取的数据
- ❖ 通过丰富的客户端引擎将冰冷的数据展示为互联网应用
- ❖ UI 和 API 的统一接口通常为 JSON 或 XML



- ❖ Vue.js
- ❖ <https://vuejs.org/>
- ❖ 简称 Vue，是一个基于 JavaScript / TypeScript 的开源 UI 框架
- ❖ 也是一个创建单页应用的互联网应用程序框架
- ❖ 在 GitHub 上目前排名前 10，最高曾超过 React 排名第 3



Github Ranking

Top 100 Stars

Ranking	Project Name	Stars	Forks	Language	Open Issues	Description	Last Commit
1	freeCodeCamp	364269	31646	TypeScript	141	freeCodeCamp.org's open-source codebase and curriculum. Learn to code for free.	2023-04-10T23:46:23Z
2	free-programming-books	275490	54685	None	26	Freely available programming books	2023-04-09T21:57:47Z
3	996.ICU	265522	21428	None	0	Repo for counting stars and contributing. Press F to pay respect to glorious developers.	2022-12-26T03:57:50Z
4	coding-interview-university	254163	67018	None	44	A complete computer science study plan to become a software engineer.	2023-04-03T15:47:07Z
5	awesome	248618	25386	None	25	Awesome lists about all kinds of interesting topics	2023-04-10T08:13:23Z
6	developer-roadmap	236429	33607	Astro	228	Interactive roadmaps, guides and other educational content to help developers grow in their careers.	2023-04-10T22:28:10Z
7	public-apis	235504	26747	Python	42	A collective list of free APIs	2023-04-10T12:03:18Z
8	system-design-primer	217278	38624	Python	175	Learn how to design large-scale systems. Prep for the system design interview. Includes Anki flashcards.	2023-04-10T01:01:51Z
9	react	205592	42824	JavaScript	959	The library for web and native user interfaces	2023-04-11T02:45:44Z
10	vue	203121	33568	TypeScript	365	This is the repo for Vue 2. For Vue 3, go to https://github.com/vuejs/core	2023-04-06T14:05:35Z

- ❖ Node.js
- ❖ <https://nodejs.org/>



- ❖ 一个能够在服务器端运行 JavaScript 的开源、跨平台运行环境
- ❖ 由 OpenJS Foundation 维护，亦为 Linux 基金会的项目
- ❖ 采用 Google 开发的 V8 运行代码，使用事件驱动、非阻塞和异步输入输出模型等技术来提高性能，可优化应用程序的传输量和规模
- ❖ 在 Node.js 出现之前，JavaScript 通常作为客户端程序设计语言使用，以 JavaScript 编写的程序通常在用户的浏览器上运行
- ❖ Node.js 的出现使得 JavaScript 也能用于服务端编程
- ❖ 目前几乎所有的互联网公司都使用了 Node.js 及其相关技术

- ❖ 安装 Node.js
- ❖ <https://nodejs.org/en/download/>

Downloads

Latest LTS Version: 18.15.0 (includes npm 9.5.0)

Download the Node.js source code or pre-built installer for your platform, and start developing today.

LTS

Recommended For Most Users

Current

Latest Features



Windows Installer

node-v18.15.0-x64.msi



macOS Installer

node-v18.15.0.pkg



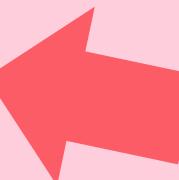
Source Code

node-v18.15.0.tar.gz

- ❖ 使用 APT 在 Ubuntu 22.04 上安装 Node.js
- ❖ <https://github.com/nodesource/distributions>
- ❖ curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
- ❖ sudo apt install nodejs

```
~ » node -v  
v16.15.0
```

```
~ » npm -v  
8.5.5
```



- ❖ NPM
- ❖ <https://www.npmjs.com/>
- ❖ Node.js 的一个软件包管理器
- ❖ 2020 年 4 月被 GitHub 收购
- ❖ 截止到 2016 年 7 月，NPM 的 Registry 上已经注册了超过 280,000 个模块
- ❖ NPM 是 Node.js 的默认软件包管理器，安装 Node.js 通常会一并安装 NPM



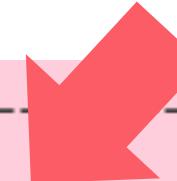
安装 Vue.js

- ❖ 阿里云 NPM 镜像
- ❖ <http://www.npmmirror.com/>

- ❖ 配置阿里云 NPM 镜像：
- ❖ `npm install -g cnpm --registry=https://registry.npmmirror.com`

- ❖ 配置完毕之后可以使用 `cnpm` 替换 `npm`，例如：
- ❖ `cnpm i -g @vue/cli`

- ❖ Vue CLI
- ❖ <https://cli.vuejs.org/>
- ❖ Vue.js 的 CLI (命令行用户界面, Command Line Interface)
- ❖ 安装 Vue CLI
- ❖ `npm i -g @vue/cli`



```
~ » vue -V  
@vue/cli 4.5.11
```

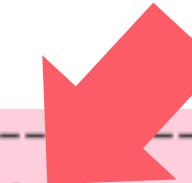
❖ 使用 Vue CLI 创建一个新的 Vue 项目

❖ vue create hello-world

```
~/Workspace/course » vue create hello-world
```

```
? Your connection to the default npm registry seems to be slow.
```

```
Use https://registry.npm.taobao.org for faster installation? Yes
```



```
Vue CLI v4.3.1
```

```
? Please pick a preset: default (babel, eslint)
```

```
Vue CLI v4.3.1
```

```
* Creating project in /mnt/e/Workspace/course/hello-world.  
* Initializing git repository...  
⚙️ Installing CLI plugins. This might take a while...
```

43 packages are looking for funding
run `npm fund` for details

- 📝 Invoking generators...
- 📦 Installing additional dependencies...

added 53 packages from 36 contributors in 9.664s

46 packages are looking for funding
run `npm fund` for details

- +. Running completion hooks...

- 📄 Generating README.md...



- 🎉 Successfully created project `hello-world`.
- 👉 Get started with the following commands:

```
$ cd hello-world  
$ npm run serve
```



- ❖ 启动 Vue 服务
- ❖ cd hello-world && npm run serve

DONE Compiled successfully in 11837ms

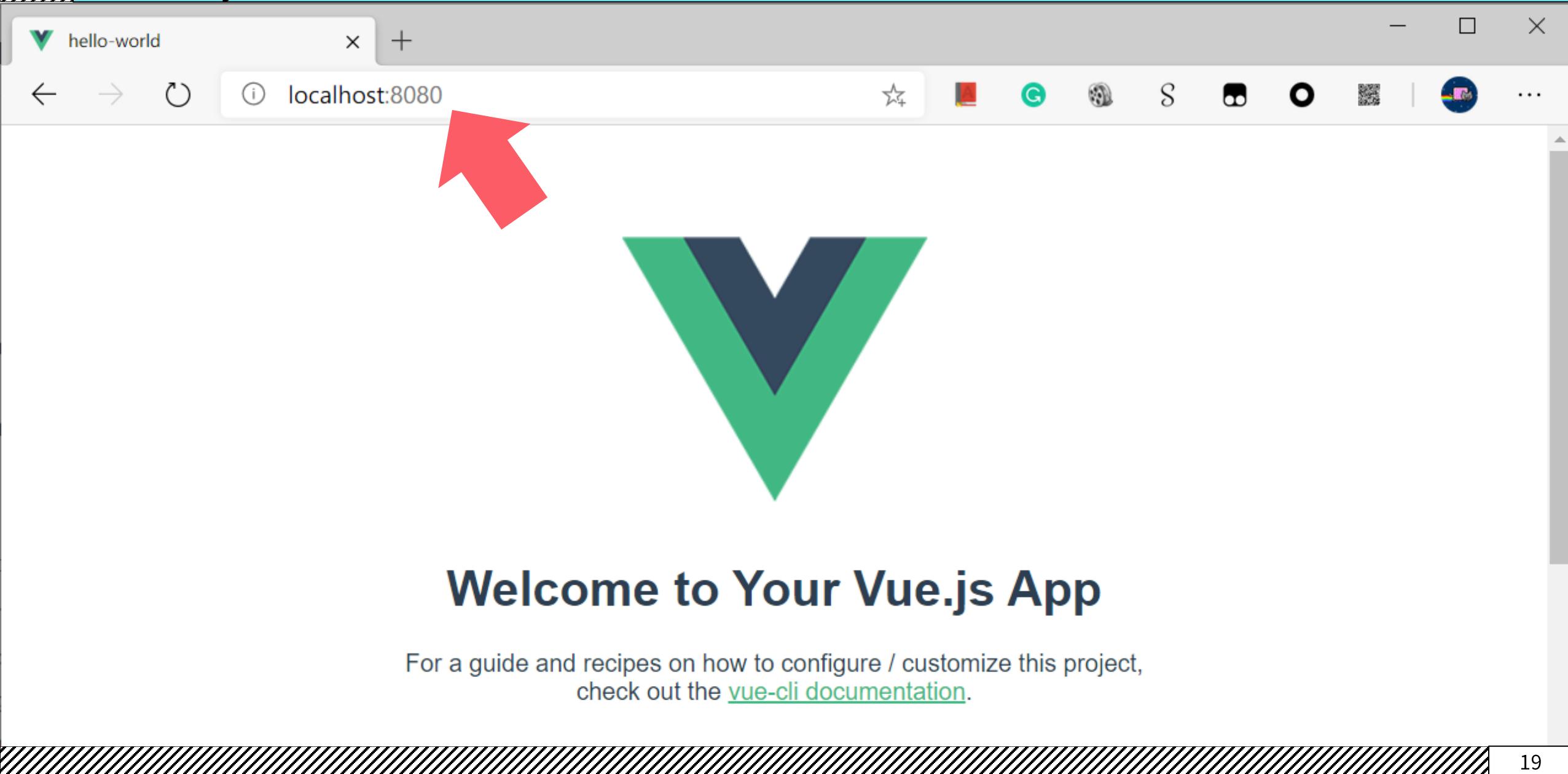
App running at:

- Local: <http://localhost:8080/>
- Network: <http://172.20.254.114:8080/>



Note that the development build is not optimized.
To create a production build, run `npm run build`.

使用 Vue.js



A screenshot of a web browser window. The title bar says "使用 Vue.js". The address bar shows "localhost:8080". A large red arrow points from the top left towards the address bar. The main content area displays the "Hello World" page of a Vue.js application, featuring the large green and blue "V" logo and the text "Welcome to Your Vue.js App". Below it is a descriptive paragraph about the vue-cli documentation.

localhost:8080

Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,
check out the [vue-cli documentation](#).

The screenshot shows the WebStorm IDE interface with a Vue.js project open. The project structure on the left includes files like `App.vue`, `main.js`, and `babel.config.js`. The `App.vue` file is currently selected and displayed in the main editor window. Several large red arrows are overlaid on the screen, indicating a flow from the project structure towards the code editor and back again, suggesting a user navigating between the two views.

```
<template>
  <div id="app">
    
    <HelloWorld msg="Welcome to Your Vue.js App"/>
  </div>
</template>
<script>
import HelloWorld from './components/HelloWorld.vue'

export default {
  name: 'App',
  components: {
    HelloWorld
  }
}
</script>
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help hello-world - HelloWorld.vue - WebStorm

hello-world > src > components > HelloWorld.vue >

Project

hello-world E:\Workspace\course\hello-world

- node_modules library root
- public
- src
 - assets
 - components
 - HelloWorld.vue

App.vue x HelloWorld.vue x

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
    <p...>
    <h3>Installed CLI Plugins</h3>
    <ul...>
    <h3>Essential Links</h3>
    <ul...>
    <h3>Ecosystem</h3>
    <ul...>
  </div>
</template>
<script>
export default {
  name: 'HelloWorld',
  props: {
    msg: String
  }
}
</script>
<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
  h3 {
```

ADD CONFIGURATION... ▶ ⚙️ ⚡ 🌐

Project Structure

Favorites

Git Terminal TODO

59:1 LF UTF-8 2 spaces* master M

WS File Edit View Navigate Code Refactor Run Tools VCS Window Help hello-world - User.vue - WebStorm

hello-world > src > components > User.vue >

Project ▾ hello-world E:\Workspace\course\hello-world > node_modules library root > public > src > assets > components > HelloWorld.vue > User.vue > App.vue > HelloWorl

ADD CONFIGURATION... ▶ ⚙️ ⚡ Git: M

Project Structure

File Structure

Favorites

Scratches and Consoles

Git Terminal TODO

10:21 CRLF UTF-8 4 spaces master M

```
<template>
  <div>
    <label for="username">User Name: </label>
    <input id="username"/>
  </div>
</template>

<script>
  export default {
    name: "User"
  }
</script>

<style scoped>
</style>
```

WS File Edit View Navigate Code Refactor Run Tools VCS Window Help hello-world - App.vue - WebStorm

hello-world > src > App.vue >

Project Structure ADD CONFIGURATION ▶ ⚙️ ⚡ Git: M

File tree:

- Project
- hello-world E:\Workspace\course\hello-world
 - node_modules library root
 - public
 - src
 - assets
 - components
 - HelloWorld.vue
 - User.vue
 - App.vue

Code editor (App.vue):

```
<template>
  <div id="app">
    <User/>
  </div>
</template>

<script>
  import User from './components/User.vue'

  export default {
    name: 'App',
    components: {
      User
    }
  }
</script>

<style>
  #app {
    font-family: Avenir, Helvetica, Arial, sans-serif;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-align: center;
    color: #2c3e50;
    margin-top: 60px;
  }
</style>
```

Bottom status bar: 14:4 LF UTF-8 2 spaces* master M

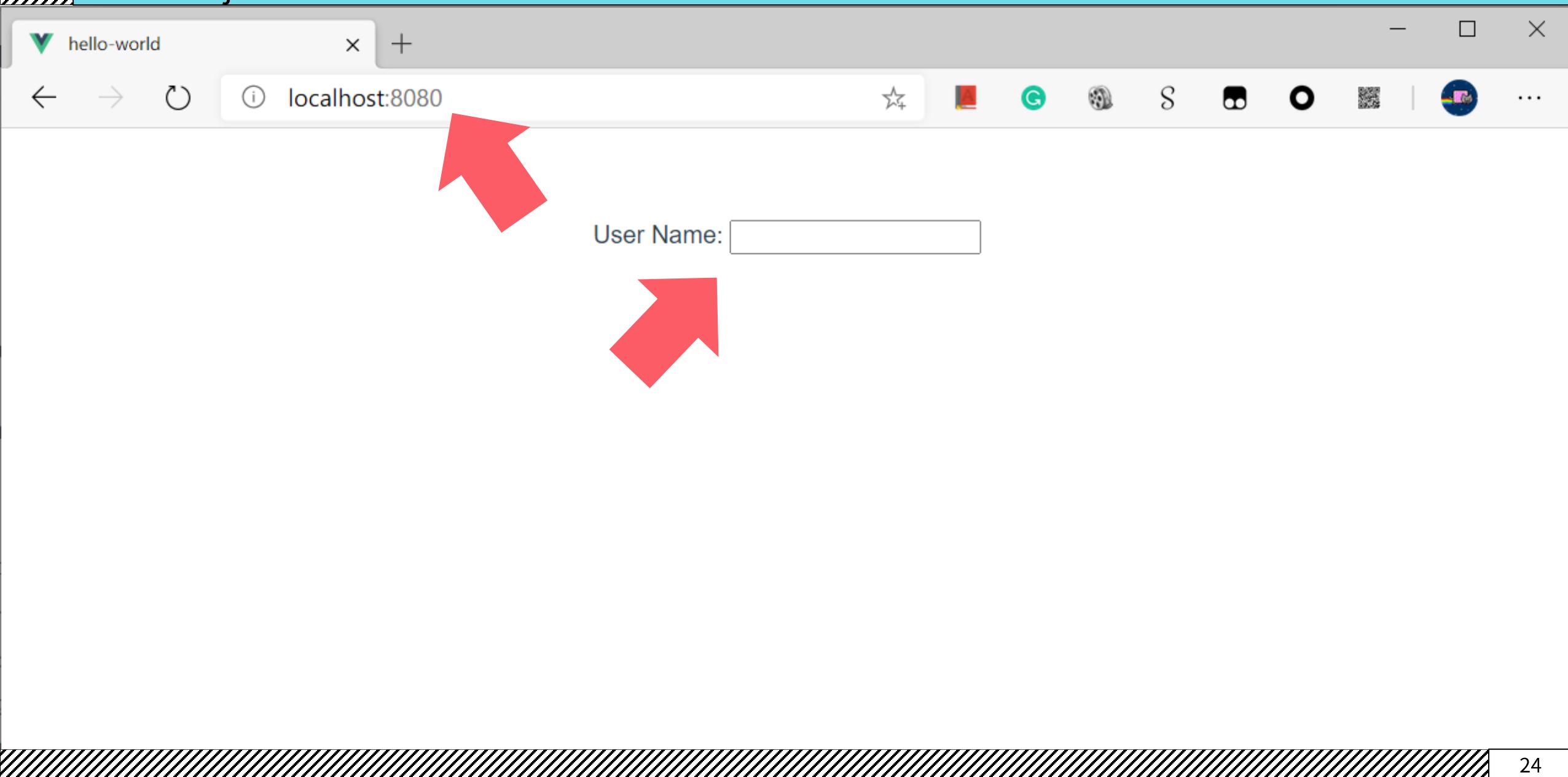
The screenshot shows the WebStorm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Title Bar:** hello-world - App.vue - WebStorm.
- Toolbar:** ADD CONFIGURATION, navigation icons (▶, ⚙️, ⚡), Git status (M).
- Project Structure:** Shows the project tree with files like node_modules, public, src (containing assets, components, App.vue, main.js, .gitignore, babel.config.js, package.json, package-lock.json, README.md), and External Libraries.
- Code Editor:** The main window displays the contents of App.vue. The code includes a template section with a div and a User component, a script section importing User from './components/User.vue' and defining the App component with a User component, and a style section for the #app element.
- Status Bar:** At the bottom, it shows the time (14:4), line separator (LF), encoding (UTF-8), whitespace settings (2 spaces*), the current branch (master), and a message (M).

Red arrows are overlaid on the screenshot, pointing from the code editor back towards the project tree, specifically highlighting the import statement and the components definition in the script block.

Vue.js

使用 Vue.js



WS File Edit View Navigate Code Refactor Run Tools VCS Window Help hello-world - User.vue - WebStorm

hello-world > src > components > User.vue >

Project ▾

hello-world E:\Workspace\course\hello-world

- > node_modules library root
- > public
- src
 - > assets
 - components
 - HelloWorld.vue
 - User.vue

App.vue ▾

App.vue ▾

App.vue ▾

```
<template>
  <div>
    <label for="username">User Name: </label>
    <input id="username"
      v-model="user"
      @change="password=Math.random().toString(36).substr(2, 10)">
  </div>
</template>
<script>
  export default {
    name: "User",
    data() {
      return {
        user: "",
        password: ""
      }
    }
  }
</script>
<style scoped>
```

ADD CONFIGURATION... ▶ ⚙️ ⌂ Git: M

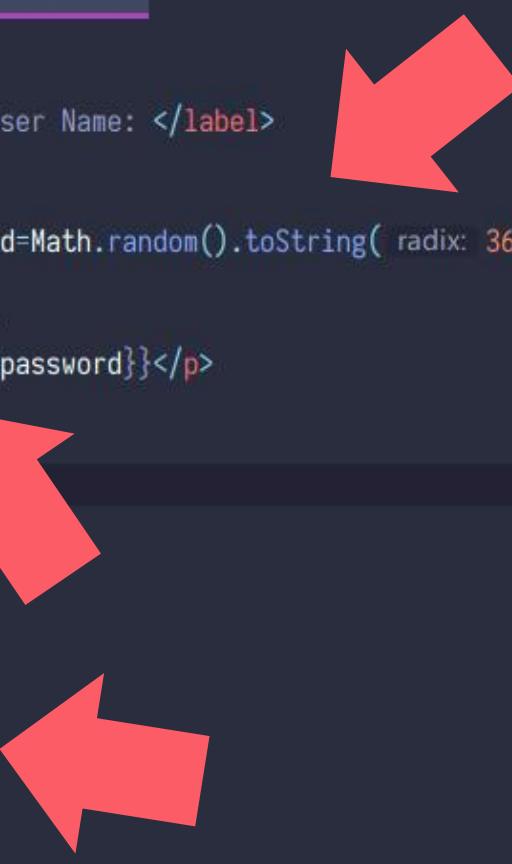
Project Structure

Favorites

Scratches and Consoles

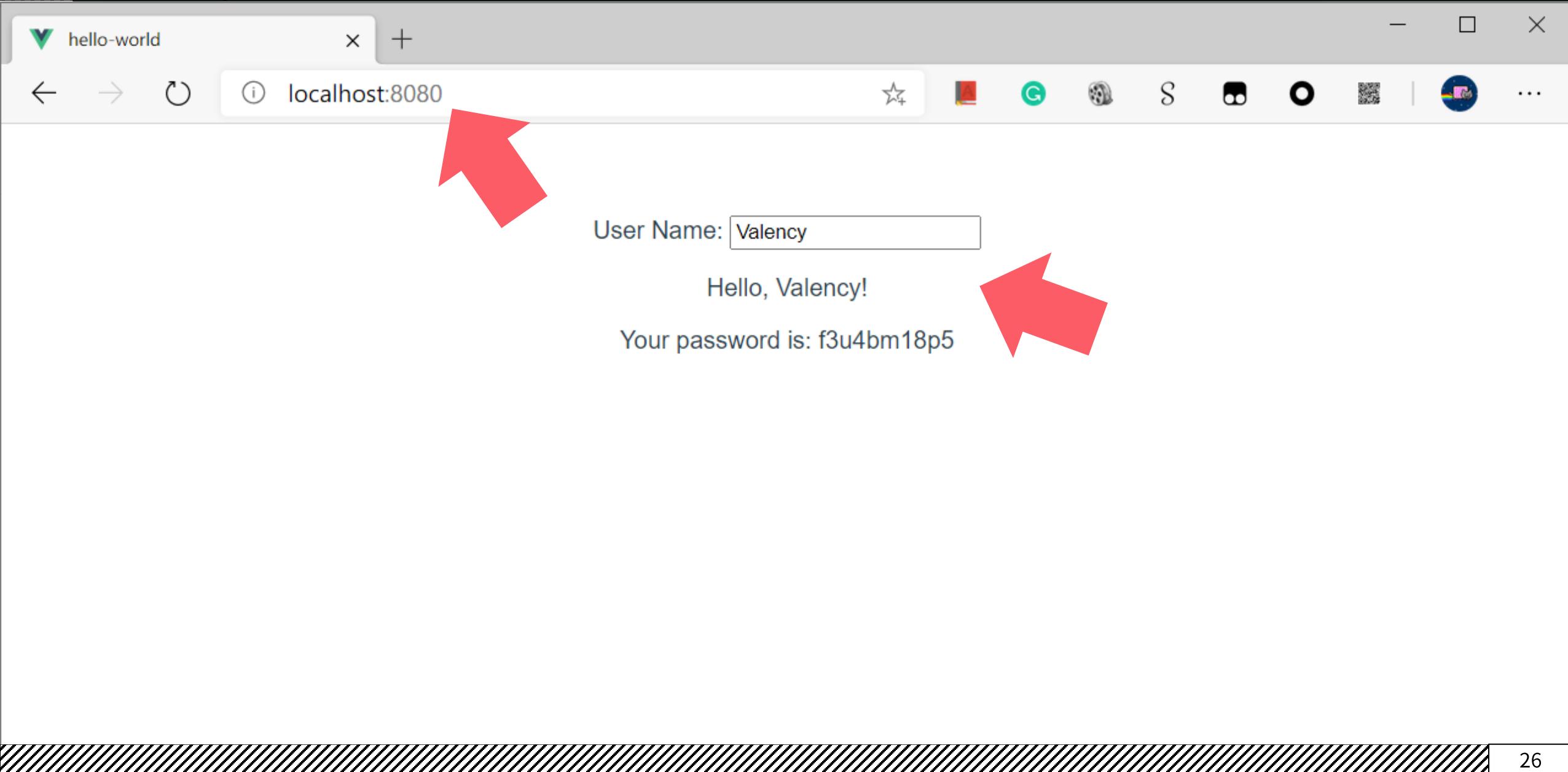
Git Terminal TODO

12:1 CRLF UTF-8 4 spaces master M



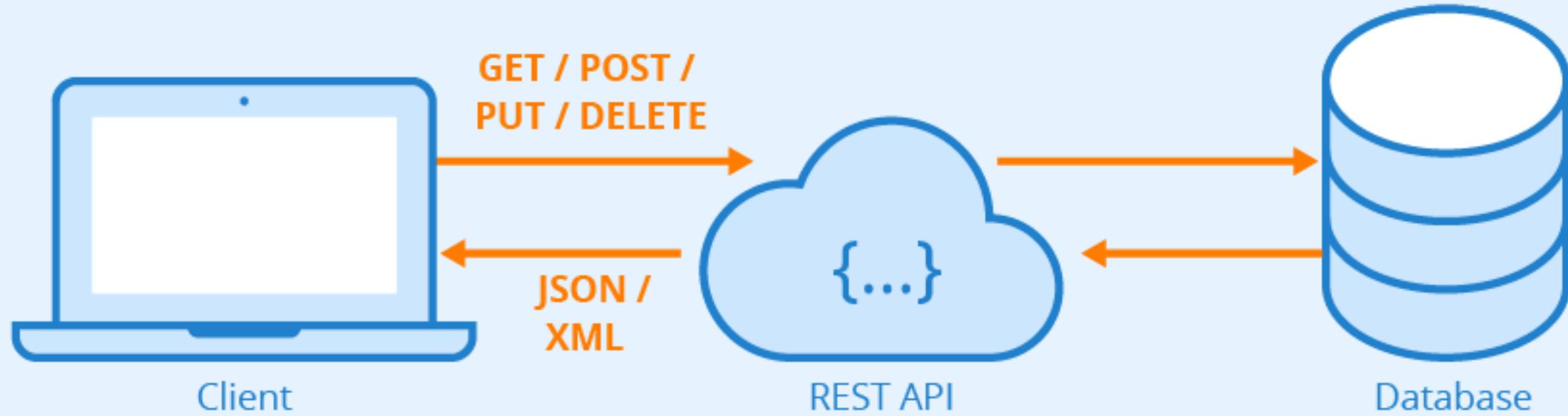
Vue.js

使用 Vue.js



连接 UI 和 API

REST API



连接 UI 和 API

REST API

POST ▼ http://localhost:5000/

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	valency	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time:

Pretty Raw Preview **Serialize** JSON ▾

```
1 {  
2   "hello": "valency"  
3 }
```

连接 UI 和 API

REST API



localhost:5000/valency/?t=15886



localhost:5000/valency/?t=1588692642



```
[  
  {  
    "id": 3,  
    "name": "valency"  
  },  
  {  
    "id": 4,  
    "name": "valency"  
  }]  
]
```

安装 Axios

- ❖ Axios
- ❖ <https://github.com/axios/axios>
- ❖ 一个基于 JavaScript 的 REST API 客户端
- ❖ 安装 Axios:
- ❖ `npm i --save axios`

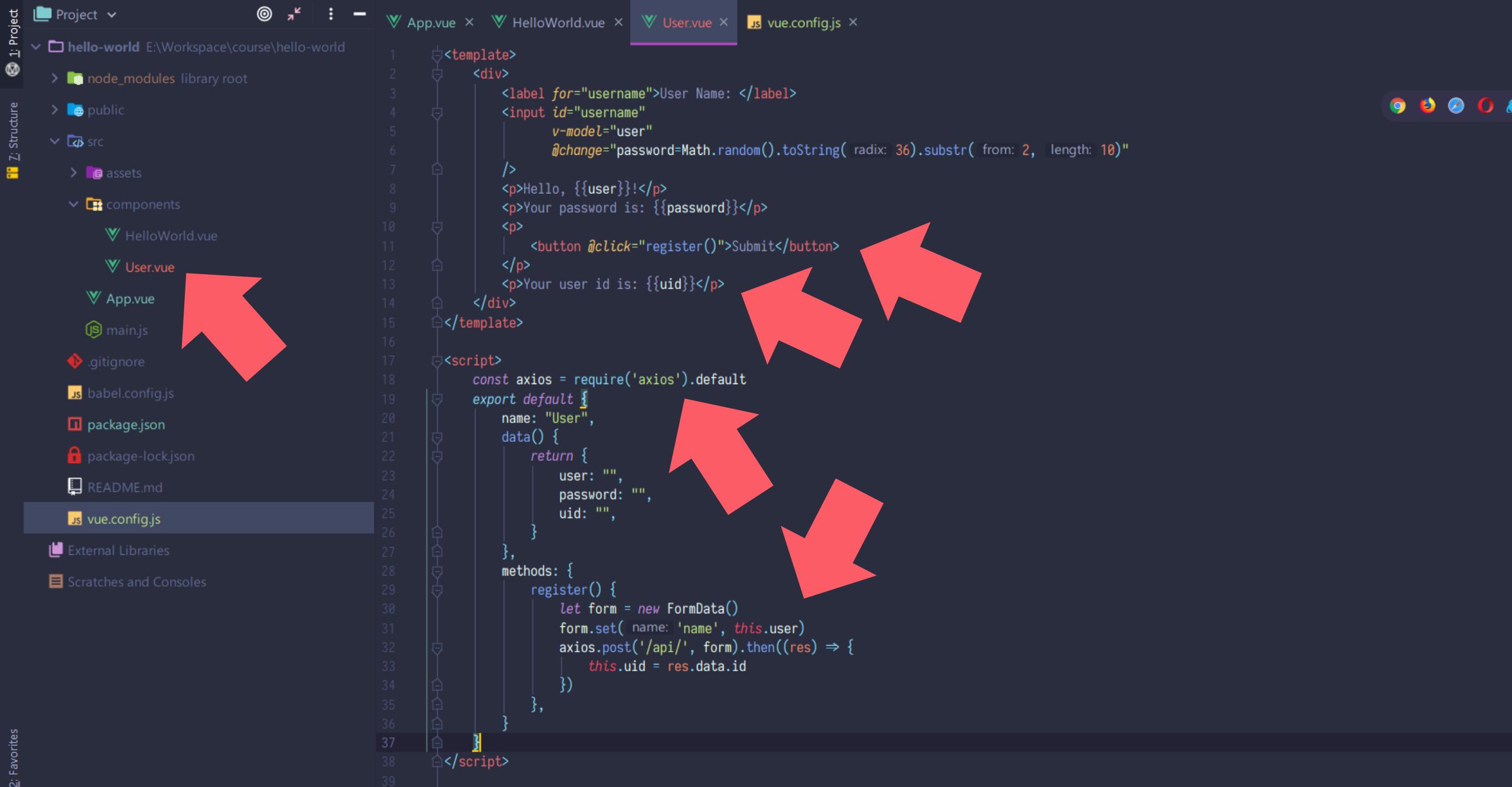
axios

npm v0.19.2 build passing coverage 94% install size 419 kB download

Promise based HTTP client for the browser and node.js

Features

- Make XMLHttpRequests from the browser
- Make http requests from node.js
- Supports the Promise API
- Intercept request and response
- Transform request and response data
- Cancel requests
- Automatic transforms for JSON data
- Client side support for protecting against XSRF



- ❖ 跨域资源共享 (Cross-Origin Resource Sharing, CORS)
- ❖ 用于让网页的受限资源能够被其他域名的页面访问的一种机制
- ❖ 通过 CORS 机制，页面能够自由地使用不同源 (Cross-origin) 的资源
- ❖ 一些跨域的请求 (特别是 Ajax) 常常会被同源策略 (Same-origin Policy) 禁止
- ❖ 简单来讲，就是在没有开启 CORS 的情况下，不能用 8080 端口的互联网应用程序去访问 5000 端口的互联网应用程序

```
✖ Access to XMLHttpRequest at 'http://localhost:5000/' from origin 'http://localhost:8080' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.          localhost/:1
✖ ▶ POST http://localhost:5000/ net::ERR_FAILED                                     xhr.js?a169:178
✖ ▶ Uncaught (in promise) Error: Network Error                                createError.js?c5b9:16
    at createError (createError.js?c5b9:16)
    at XMLHttpRequest.handleError (xhr.js?a169:83)
```

>

- ❖ CORS 并不安全，一般不建议开启，而应当使用代理 (Proxy) 的方式访问资源

hello-world > js vue.config.js >

Project

Structure

Tool

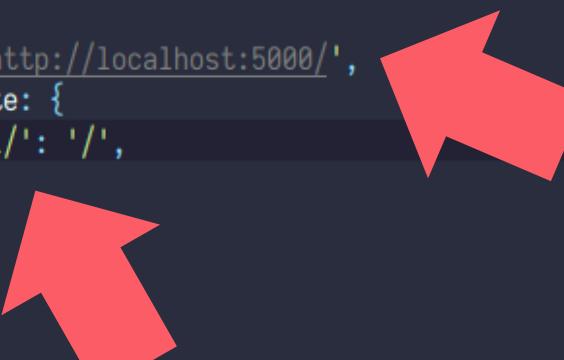
Project

hello-world E:\Workspace\course\hello-world

- node_modules library root
- public
- src
 - assets
 - components
 - HelloWorld.vue
 - User.vue
 - App.vue
 - main.js
- .gitignore
- babel.config.js
- package.json
- package-lock.json
- README.md
- vue.config.js

App.vue × HelloWorld.vue × User.vue × js vue.config.js ×

```
1 module.exports = {
2   devServer: {
3     proxy: {
4       '^/api': {
5         target: 'http://localhost:5000/',
6         pathRewrite: {
7           '^/api/': '/',
8         },
9       },
10      },
11    },
12  };
13 }
```



External Libraries

Scratches and Consoles

User Name:

Hello, valency!

Your password is: 3m34jr5mrg

Your user id is: 6



Elements

Console

Sources

Network

Performance

Memory

Application

Security

Audits

AdBlock

Preserve log Disable cache

Online



Filter

 Hide data URLs All | XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies

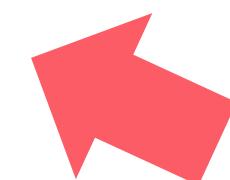
Name

 content.min.css info?t=1589824017459 api/

x Headers Preview Response Initiator Timing Cookies

{id: 6}

id: 6



File Edit View Navigate Code Refactor Run Tools VCS Window Help hello-world - User.vue - WebStorm

ADD CONFIGURATION... ▶ ⚙️ ⌂ Git: 🌐 ➔ ⌂

Project ▾ App.vue ▾ HelloWorld.vue ▾ User.vue ▾ vue.config.js

hello-world > src > components > User.vue

hello-world E:\Workspace\course\hello-world

node_modules library root

public

src

assets

components

HelloWorld.vue

User.vue

App.vue

main.js

.gitignore

babel.config.js

package.json

package-lock.json

README.md

vue.config.js

External Libraries

Scratches and Consoles

Favorites

```
<input id="username" v-model="user" @change="password=Math.random().toString(radix: 36).substr(from: 2, length: 10)" />
<p>Hello, {{user}}!</p>
<p>Your password is: {{password}}</p>
<p>
    <button @click="register()">Submit</button>
</p>
<p>Your user id is: {{uid}}</p>
<p>
    <button @click="query()">Query</button>
</p>
<p v-for="i in users" :key="i.id">{{i.id}} - {{i.name}}</p>
</div>
</template>
<script>
    const axios = require('axios').default
    export default {
        name: "User",
        data() {
            return {
                user: "",
                password: "",
                uid: "",
                users: []
            }
        },
        methods: {
            register() {...},
            query() {
                axios.get('/api/' + this.user + '?t=' + ~~(new Date().getTime() / 1000)).then((res) => {
                    this.users = res.data
                })
            }
        }
    }
</script>
```

The image shows a screenshot of the WebStorm IDE interface. At the top, there's a menu bar with options like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the menu is a title bar for 'hello-world - User.vue - WebStorm'. The main workspace has several tabs open: 'Project', 'App.vue', 'HelloWorld.vue', 'User.vue' (which is currently selected), and 'vue.config.js'. On the left, the 'Project' tool window displays the project structure, including 'src' and 'components' folders containing files like 'HelloWorld.vue', 'User.vue', 'App.vue', and 'main.js'. A 'vue.config.js' file is also listed. The bottom navigation bar includes links for 'External Libraries' and 'Scratches and Consoles'. On the far right, there are browser icons for Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera. The central part of the screen shows the code for 'User.vue'. The code uses Vue.js syntax, including template tags like <input>, <p>, and <button>, and script tags with data binding like 'v-model' and 'v-for'. Several large red arrows are overlaid on the image, pointing from the left side towards the 'User.vue' code, specifically highlighting the 'User.vue' tab and the first few lines of the template and script blocks.

The screenshot shows a web browser window with a login form and a developer tools Network tab.

Browser Header: The address bar shows "localhost:8080".

Form Content:

- User Name: valency
- Hello, valency!
- Your password is: sadu23p7j6
- Submit

Developer Tools Network Tab:

- Filter: content.min.css, info?t=15898251, ?t=1589825121
- Headers, Preview, Response, Initiator, Timing, Cookies
- Preview pane shows an array of objects:

 - [{"id": 3, "name": "valency"}, {"id": 4, "name": "valency"}, {"id": 6, "name": "valency"}]

- Details pane shows the array elements:

 - 0: {"id": 3, "name": "valency"}
 - 1: {"id": 4, "name": "valency"}
 - 2: {"id": 6, "name": "valency"}

- ❖ Vue.js 官方教程：
- ❖ <https://cn.vuejs.org/guide/introduction.html>

- ❖ Vue CLI 官方教程：
- ❖ <https://cli.vuejs.org/zh/guide/>



尤雨溪

前端框架Vue.js 作者，独立开源开发者，现居美国新泽西，现在全职开发和维护 Vue.js。

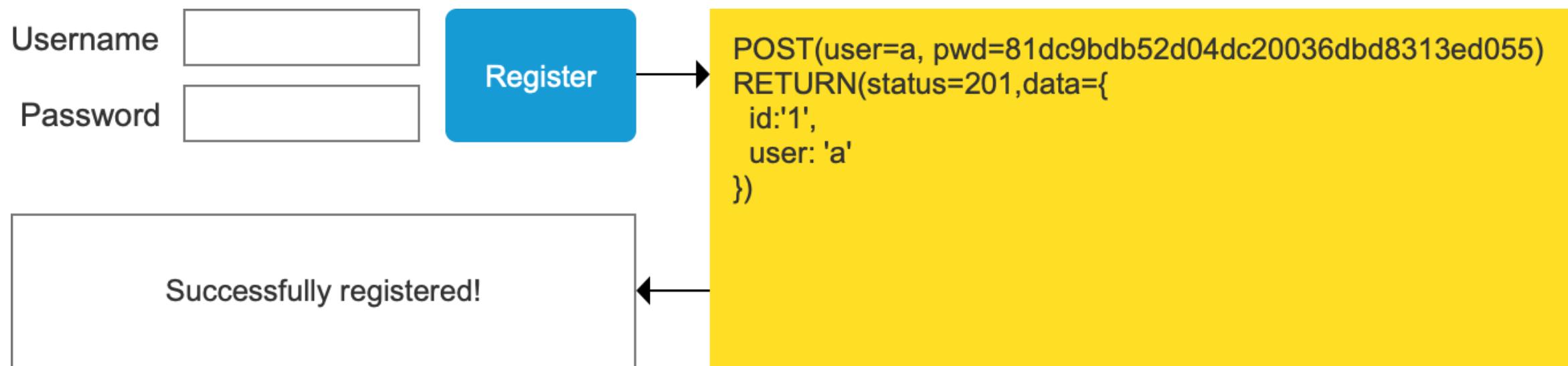
本期嘉宾

尤雨溪 (**Evan You**)，前端框架**Vue.js** 作者，独立开源开发者，现居美国新泽西。曾就职于 **Google Creative Labs** 和 **Meteor Development Group**。由于工作中大量接触开源的 **JavaScript** 项目，最后自己也走上了开源之路，现在全职开发和维护 **Vue.js**。

- ❖ Axios 官网及教程：
<https://github.com/axios/axios>
- ❖ Vue CLI 官网关于代理的教程：
<https://cli.vuejs.org/config/#devserver-proxy>
- ❖ Vue 使用的代理模块 (http-proxy-middleware)：
<https://github.com/chimurai/http-proxy-middleware>

- ❖ 使用任意一个 REST UI 框架撰写一个简单的用户管理系统
- ❖ REST UI 需要与上节实验的 REST API 通过标准接口（例如 JSON）通信
- ❖ 至少需要包含以下几个功能：
 - ❖ **注册**：通过用户名和密码注册新账户
 - ❖ **登录**：通过用户名和密码登录
 - ❖ **验证登录状态**：通过 ID 和 Token 验证用户登录状态
 - ❖ **登出**：通过 ID 和 Token 登出

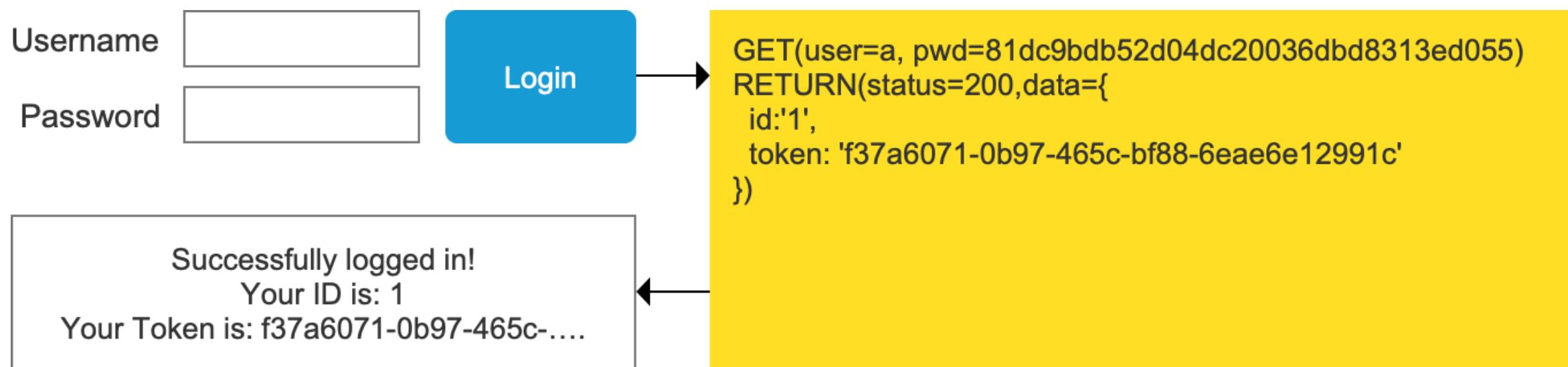
- ❖ 注册
- ❖ 展示用户名和密码输入框
- ❖ 点击“注册”按钮通过 API 注册，并显示注册是否成功



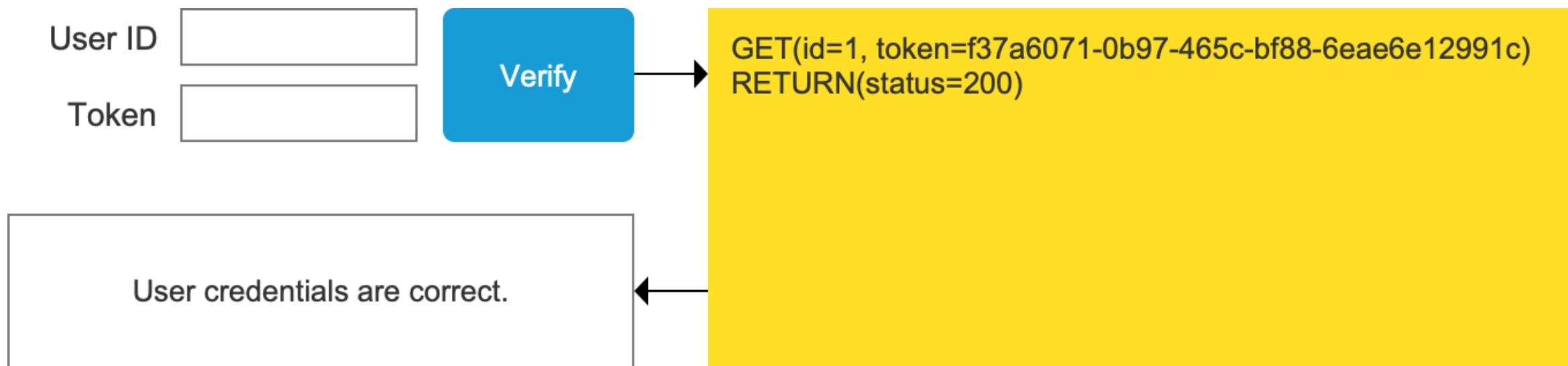
❖ 登录

❖ 展示用户名和密码输入框

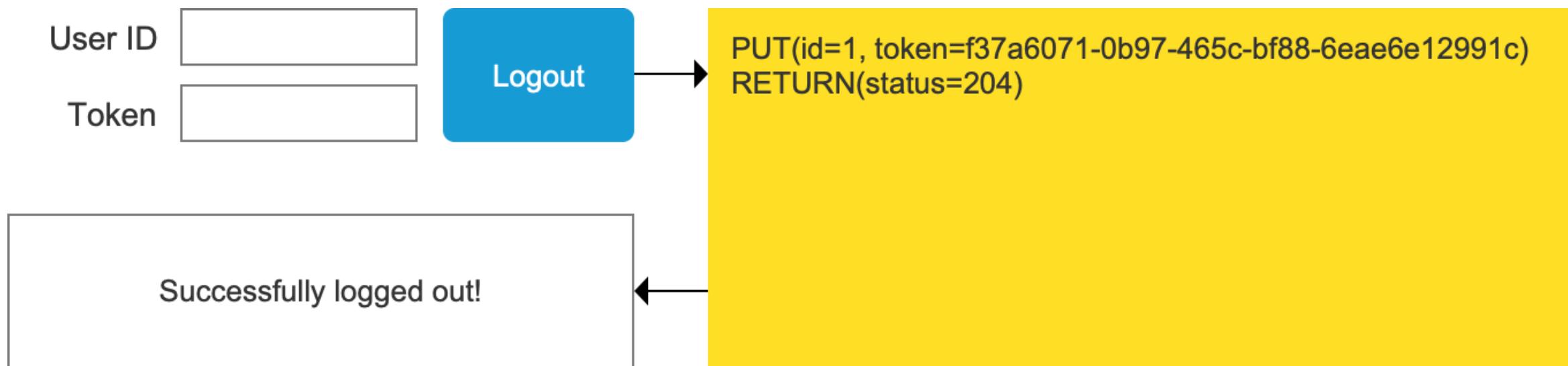
❖ 点击“登录”按钮通过 API 登录，并返回 ID 和 Token（可记录到 Cookie 中）



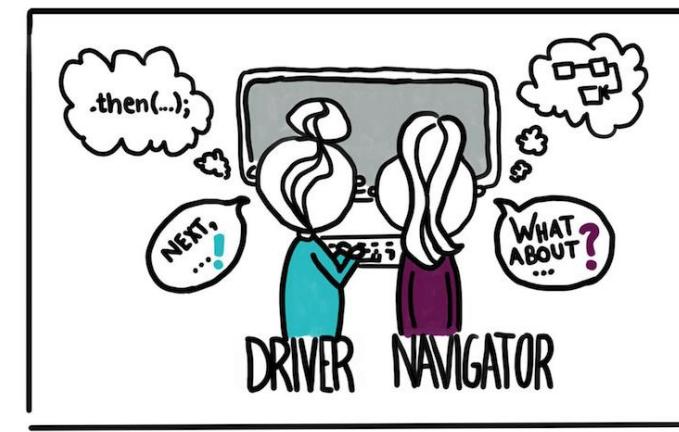
- ❖ 验证登录状态
- ❖ 如已使用 Cookie:
 - ❖ 展示“验证”按钮，点击之后通过 API 返回验证是否成功
- ❖ 如未使用 Cookie:
 - ❖ 展示 ID 和 Token 输入框，点击“验证”按钮通过 API 返回验证是否成功



- ❖ 登出
- ❖ 如已使用 Cookie:
 - ❖ 展示“登出”按钮，点击之后通过 API 返回登出是否成功
- ❖ 如未使用 Cookie:
 - ❖ 展示 ID 和 Token 输入框，点击“登出”按钮通过 API 返回登出是否成功



- ❖ 结对编程 (Pair Programming)
 - ❖ 是一种敏捷软件开发的方法，两个程序员在一个计算机上共同工作。一个人输入代码，而另一个人审查他输入的每一行代码。输入代码的人称作驾驶员 (Driver) ，审查代码的人称作导航员 (Navigator) 。两个程序员经常互换角色。
 - ❖ 在结对编程中，观察员同时考虑工作的战略性方向，提出改进的意见，或将来可能出现的问题以便处理。这样使得驾驶者可以集中全部注意力在完成当前任务的“战术”方面。观察员当作安全网和指南。结对编程对开发程序有很多好处。比如增加纪律性，写出更好的代码等。
 - ❖ 结对编程是极限编程的组成部分。
-
- ❖ 考虑到本次实验较为复杂，**两人一组提交实验报告即可**
 - ❖ 注意：**两人都需要提交实验报告，内容务必保持一致**
 - ❖ **如果不愿意结对编程，也可以一人独立完成实验**



作业

- ❖ 在作业系统中下载并完成本实验课对应实验报告
- ❖ <https://hw.dgut.edu.cn/>
- ❖ 注意：所有标识为 * 的地方都需要填写
- ❖ 截止日期：2024-06-03 23:59:59

课程名称：云计算与大数据应用开发

学期：2023 年春季

实验名称	虚拟化技术			实验序号	1
姓 名	***	学 号	***	班 级	***
实验地点	***	实验日期	***	指导老师	丁烨
教师评语	-			实验成绩	-
				分制	100
同组同学	无				

四、 实验作业及分析

4.1 实验过程

1) *** 请将详细实验过程填写在此处 ***

4.2 实验结果

*** 请将实验结果截图填写在此处 ***

五、 实验总结

*** 请撰写一段 200 字左右的实验总结 ***



