

lab1.1

安装java

由于我的电脑上本来就装好了Java，这里演示一下安装完成的效果

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS F:\桌面\一些文件\通讯\支付宝前端营\自己的尝试> java -v
Unrecognized option: -v
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
PS F:\桌面\一些文件\通讯\支付宝前端营\自己的尝试> java -version
java version "9.0.4"
Java(TM) SE Runtime Environment (build 9.0.4+11)
Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)
PS F:\桌面\一些文件\通讯\支付宝前端营\自己的尝试> java -version
java version "1.8.0_321"
Java(TM) SE Runtime Environment (build 1.8.0_321-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.321-b07, mixed mode)
PS F:\桌面\一些文件\通讯\支付宝前端营\自己的尝试> █
```

安装tomcat

1. 在官网下载对应版本的安装包

Tomcat Connectors

Tomcat Native

Taglibs

Archives

Documentation

Tomcat 10.1 (alpha)

Tomcat 10.0

Tomcat 9.0

Tomcat 8.5

Tomcat Connectors

Tomcat Native

Wiki

Migration Guide

Presentations

Specifications

Problems?

Security Reports

Find help

FAQ

Mailing Lists

Bug Database

IRC

Get Involved

Overview

Source code

Buildbot

Translations

Tools

Media

Twitter

YouTube

Blog

Misc

Who We Are

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file which contains the OpenPGP keys of Tomcat's Release Managers. We also provide SHA-512 checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

Mirrors

You are currently using <https://d1cdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available.

Other mirrors:

10.0.20

Please see the [README](#) file for packaging information. It explains what every distribution contains.

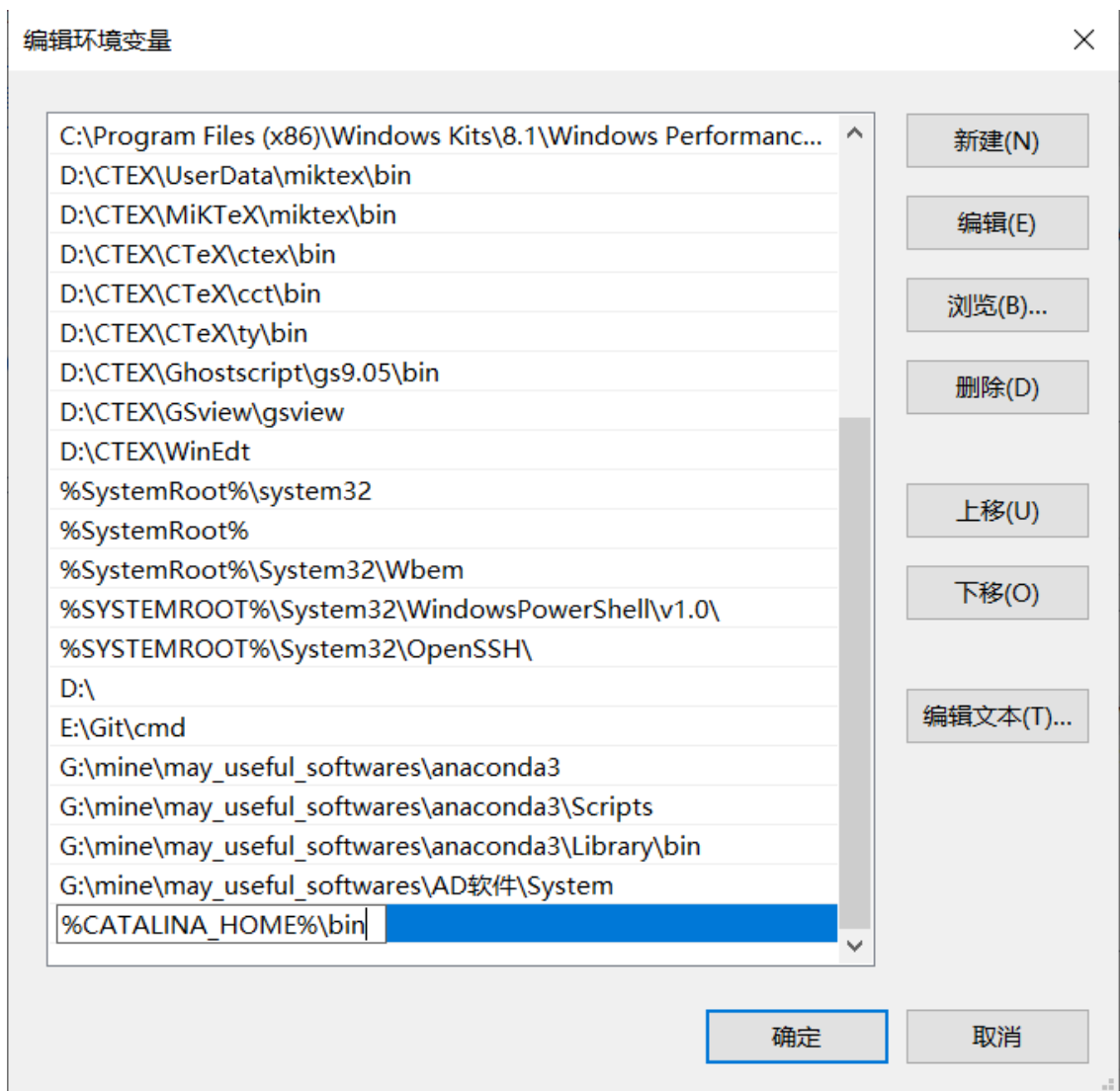
Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows.zip \(pgp, sha512\)](#)
 - [64-bit Windows.zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

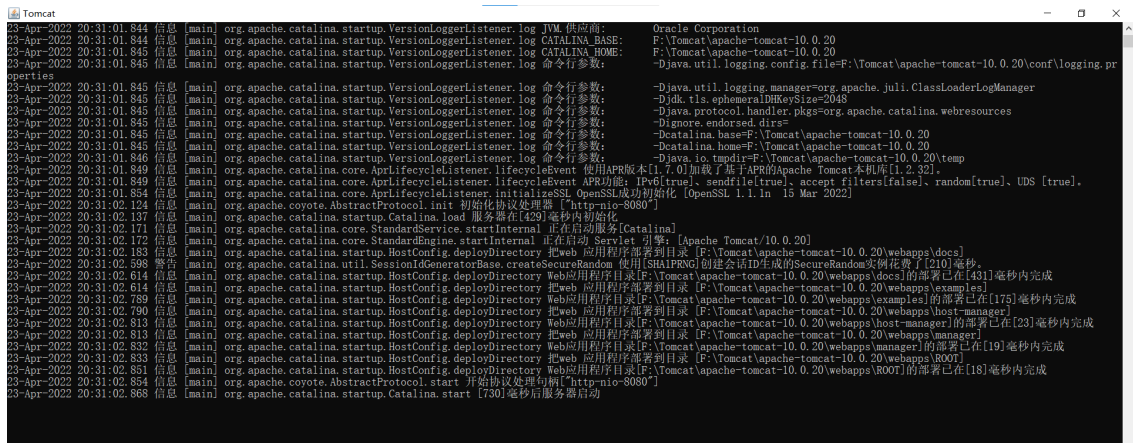
Source Code Distributions

[tar.gz \(pgp, sha512\)](#)

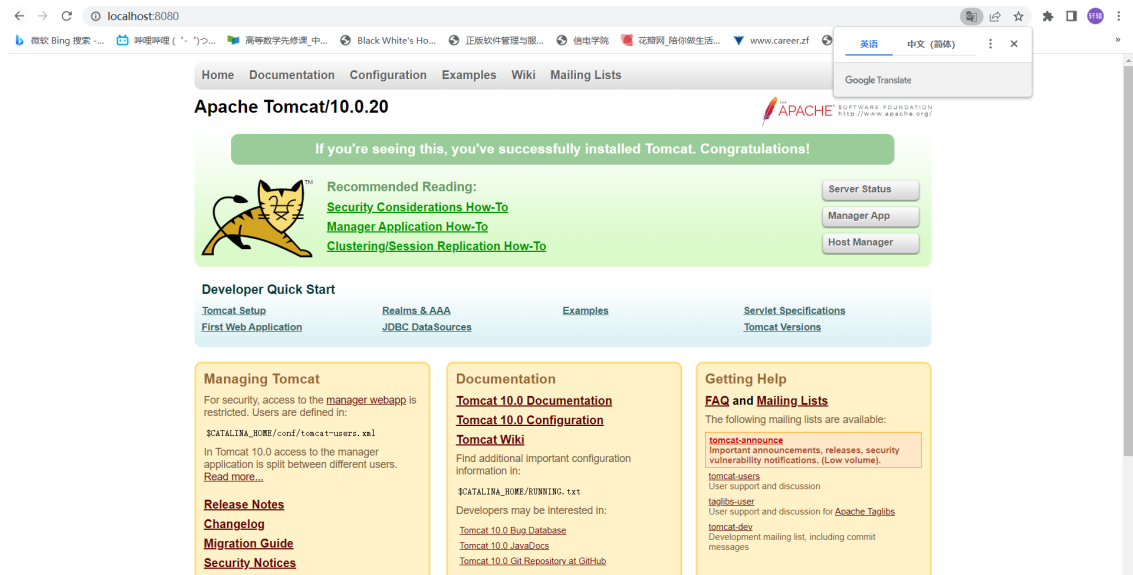
2. 编辑环境变量



3. 启动tomcat

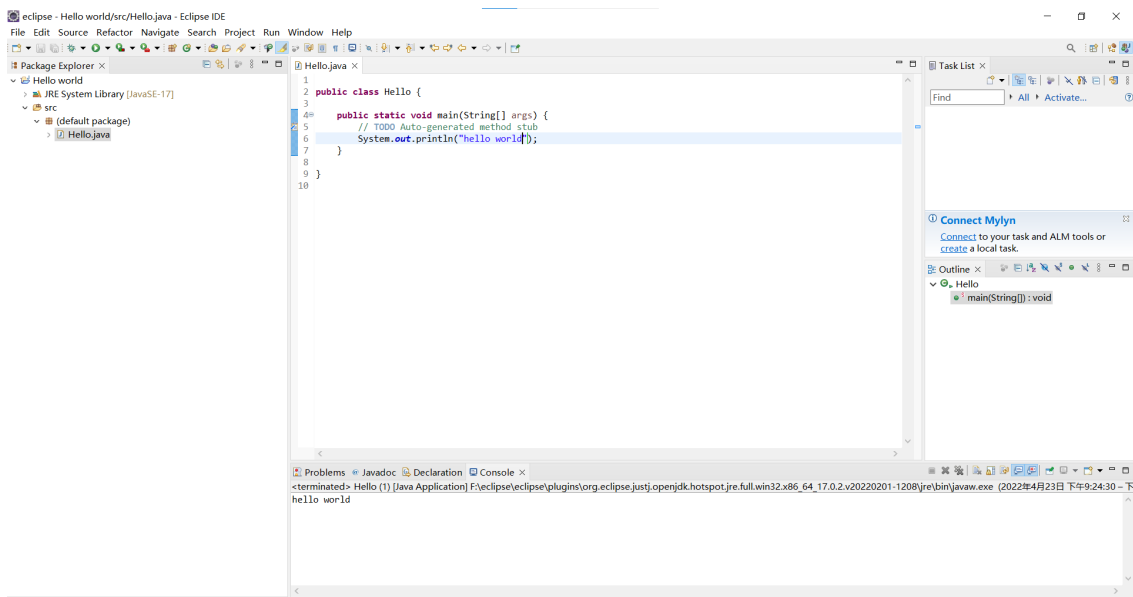


4. 观察结果



安装eclipse

1. 在官网下载安装包并安装
2. 开发hello world



lab1.2

应用场景

个人博客

个人程序是怎么设计的

- 前端：React+Typescript
- 后端：tomcat+mysql
- 服务器：腾讯云轻量应用服务器+TencentDB

数据库

前端访问api以及数据库访问

一共有如下api

- `/api/uploadBlog`；这个api用于上传文章，上传的同时还会生成一张新的表用于储存这篇文章的评论数据

```
Complexity is 7 It's time to do something...
app.post('/api/uploadBlog',function(req,res) {
  const id = Date.now();
  console.log("上传博客中...");
  console.log(req.body);
  const sql = `insert into blog_content(title,content,time,tags,id) values(?,?,?,?,?)`;
  const createCommentsSql = `create table comment_${id}(id varchar(100) primary key ,name varchar(100),content varchar(500))`;
  const addSqlParams = [req.body.title,
    req.body.content,
    req.body.time,
    req.body.tag,
    id];
  connection.query(sql,addSqlParams,function(err) {
    if(err){
      console.log(err);
      return;
    }
  });
  console.log("Everything is cool!");
  connection.query(createCommentsSql,function(err) {
    if(err){
      console.log(err);
      return;
    }
  });
});
```

- `/api/login`；这个api用于登录，实现了管理员登录，登录后能在home上传文章的功能

```
Complexity is 8 It's time to do something...
app.post('/api/login',function(req,res){
  console.log("登录中...");
  const userName = req.body.userName;
  const password = req.body.password;
  const sql = 'select * from user where userName = ? and password = ?';
  const queryParams = [userName,password];
  connection.query(sql,queryParams,function(err,result) {
    console.log(result);
    if(err){
      result = JSON.stringify({
        code:400,
      });
      res.end(result);
      return;
    }
    else{
      if(result.length === 0){
        result = JSON.stringify({
          code:401,
        });
        res.end(result);
        return;
      }
      result = JSON.stringify({
        code:200,
        token:new Date().getTime(),
        data:result
      });
      res.end(result);
      return;
    }
  });
});
```

- `/api/getPosts`；这个api用于访问 `blog_content` 表，用于获取文章内容

Complexity is 4 Everything is cool!

```
app.get('/api/getPosts',function(req,res){  
  console.log("收到请求,拉取数据...");  
  const sql = 'select * from blog_content';  
  Complexity is 3 Everything is cool!  
  connection.query(sql,function(err,result) {  
    if(err) {  
      console.log(err);  
      return;  
    }  
    const result1 = JSON.stringify({  
      code: 200,  
      data:result  
    });  
    //console.log(result1);  
    res.end(result1);  
  });  
});
```

- `/api/getComments` ; 这个api用于获取评论数据

Complexity is 4 Everything is cool!

```
app.get('/api/getComments',function(req,res){  
  console.log("收到请求,拉取数据...");  
  console.log("url的解码:",req.query);  
  const sql = `select * from comment_${req.query.id}`;  
  Complexity is 3 Everything is cool!  
  connection.query(sql,function(err,result) {  
    if(err) {  
      console.log(err);  
      return;  
    }  
    const result1 = JSON.stringify({  
      code: 200,  
      data:result  
    });  
    //console.log(result1);  
    res.end(result1);  
  });  
});
```

- `/api/postComments` ; 用于上传评论数据


```

Complexity is 7 It's time to do something...
app.post('/api/postComments',function(req,res){
  //console.log(req.body);
  const dateNow = new Date();
  const logWrite = `${dateNow.getFullYear()}-${dateNow.getMonth()+1}-${dateNow.getDate()} ${dateNow.getHours()}:${dateNow.getMinutes()}:${dateNow.getSeconds()}`;
  //fs.writeFile('./log.txt',logWrite);
  console.log("收到请求...");
  if(!req.body.userName || !req.body.content){
    res.end(JSON.stringify({
      code:400,
      msg:'参数错误'
    }));
    return;
  }
  const addSqlParams = [];
  const addSql = `insert into comment_${req.query.id}(id,name,content) values(?,?,?)`;
  var id = Date.now();
  addSqlParams.push(id.toString());
  addSqlParams.push(req.body.userName);
  addSqlParams.push(req.body.content);
  console.log(addSqlParams);
  connection.query(addSql,addSqlParams,function(err) {
    if(err) {
      console.log(err);
      return;
    }
  });
});

```

数据库设计

使用 `blog_content` 表储存文章数据，这个表中有 `id`, `title`, `time`, `content`, `tags` 等列，其中 `id` 是主键，当我通过 `/api/postBlog` 上传文章时，后端会同步在数据库中生成一张 `comment_${id}` 的表，这样就实现了每篇文章的评论区都相互对应

`comment_${id}` 的表中则有 `id`, `name`, `content` 等列，用于储存用户的评论信息

部署网站

运行 `npm run build` 打包项目，设置 `homepage: '.'`；将打包好的文件发布到服务器上，tomcat的 `server.xml` 的相应配置如下：

```

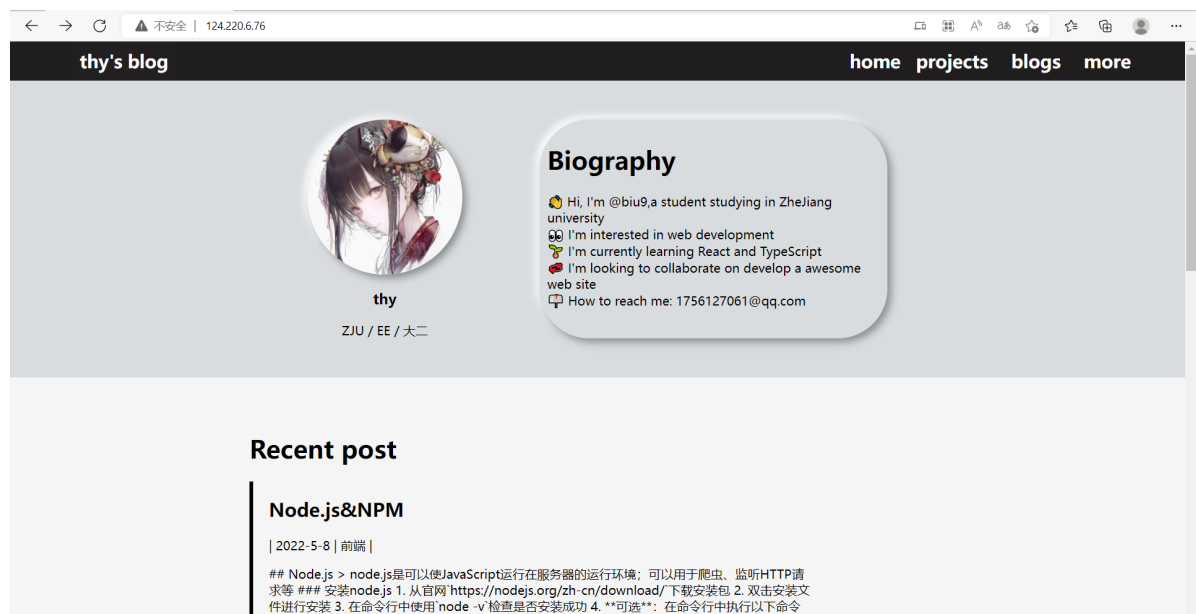
<Context path="" docBase="/usr/src/sp2022/apache-tomcat-10.0.20/webapps/blog"
debug="0" reloadable="true" />

```

现在可以访问 `http://124.220.6.76` 来观察效果

实现截图

网站首页



thy's blog

homeprojectsblogsmore

Node.js&NPM

| 2022-5-8 | 前端 |

Node.js

node.js是可以使JavaScript运行在服务器的运行环境；可以用于爬虫、监听HTTP请求等

安装node.js

1. 从官网<https://nodejs.org/zh-cn/download/>下载安装包

2. 双击安装文件进行安装

3. 在命令行中使用node -v 检查是否安装成功

4. 可选：在命令行中执行以下命令可以安装cnpm，指向国内淘宝镜像以提高一些资源的访问速度，npm install -g cnpm --registry=https://registry.npm.taobao.org

hello Node.js!

新建一个server.js文件，在文件中写入以下代码

```
const hello = 'Hello';
const to = 'Node.js';
console.log(`${hello},${to}!`);
```

在命令行中运行node server.js，可以观察到终端输出Hello,Node.js!

内置模块

npm install xxx -g会安装在全局目录下

但是由于package.json文件只规定了包的大版本，容易出现协作双方安装的小版本不同的情况；package-lock.json用于处理这种情况，当npm会优先安装package-lock.json中的包的版本

Comments

姓名:

请输入你的名字

内容:

请输入你的评论

提交

thy

再看看评论

前端React+typescript+material UI
后端Node.js+express+TencentDB
服务器-腾讯云轻量应用服务器+tomcat
@All Rights Reserved 2022

home界面

thy's blog

homeprojectsblogsmore

标题

时间

标签

内容

