Technology Tarena 达内科技

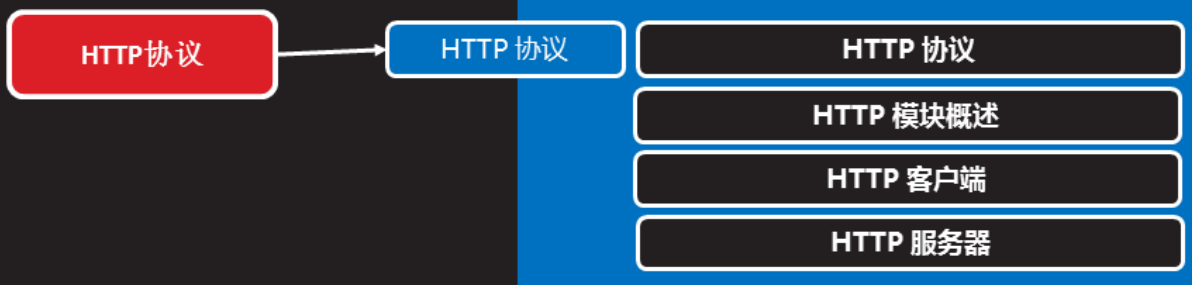
前端全栈 后台开发

Node.js Unit04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	HTTP协议
	10:30 ~ 11:20	Express框架
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑

HTTP协议

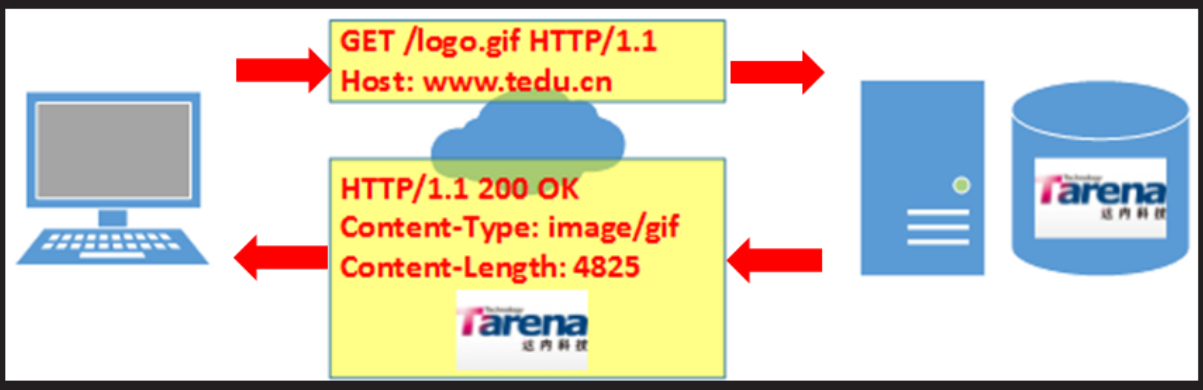


HTTP 协议

HTTP协议概述

- 浏览器与Web服务器之间的通信消息格式遵守HTTP协议。它是典型的基于“请求-响应”模型的协议，只有客户端发出了请求消息，服务器才会给出响应消息，并且一个请求消息只会得到一个响应消息。

知识讲解



请求消息

Technology Tarena 达内科技

- 请求消息是客户端构建，发送给Web服务器的消息；
- 一个请求消息由四部分构成：

  (1) 请求起始行

  (2) 请求头部

  (3) CRLF

  (4) 请求主体(可能没有)

知识讲解

请求方法

Technology Tarena 达内科技

- 请求起始行中的请求方法描述了当前请求的目的；
- 常见的请求方法有：
  - GET：表示客户端想“获取”服务器上的指定资源
  - POST：表示客户端想向服务器“传递并保存”数据
  - PUT：一般用于表示客户端想向服务器“传递并更新”数据
  - DELETE：一般表示客户端想从服务器“删除”指定数据

知识讲解

(1)只有 POST 和 PUT 请求才能有请求主体；
(2) HTML表单只能使用发起 GET 和 POST 请求；XHR 对象可以发起任意的 HTTP 请求。

响应消息

Technology Tarena 达内科技

- 响应消息是Web服务器根据客户端的请求，返回给客户端的消息；
- 一个响应消息由四部分构成：
  (1) 响应起始行
  (2) 响应头部
  (3) CRLF
  (4) 响应主体(可能没有)

知识讲解

响应状态码

Technology Tarena 达内科技

- 响应起始行中的响应状态码用于标识响应消息的状态，可能有如下五类可能：
  - 1xx：请求-响应继续进行
  - 2xx：成功的响应
  - 3xx：响应重定向到其它地址
  - 4xx：客户端请求错误
  - 5xx：服务器端运行错误

知识讲解

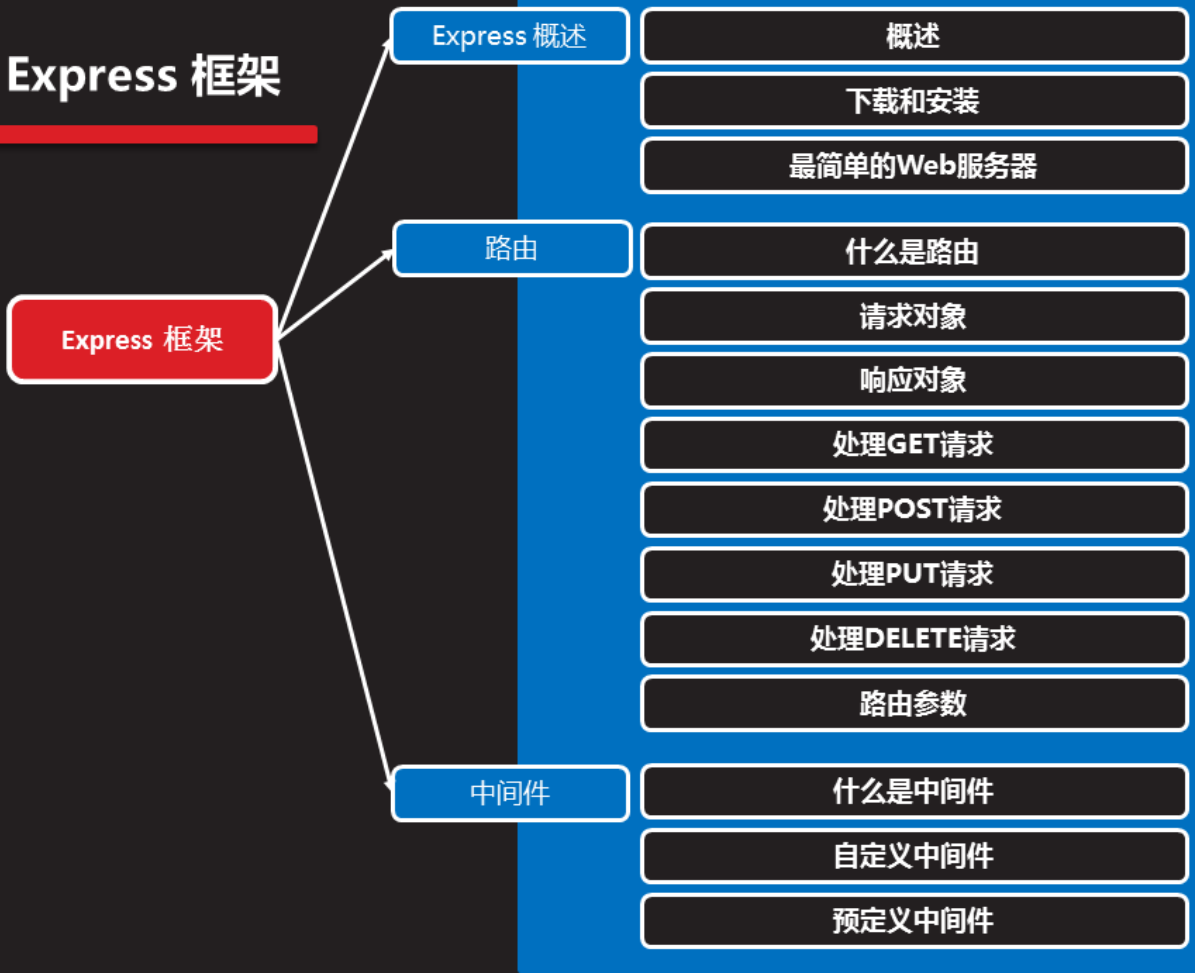
Tarena Technology 达内科技

HTTP请求消息和响应消息

课堂练习

- 使用HTML表单向服务器发起GET请求消息；
- 使用HTML表单向服务器发起POST请求消息；
- 使用XHR向服务器发起GET请求消息；
- 使用XHR向服务器发起POST请求消息；
- 使用XHR向服务器发起PUT请求消息；
- 使用XHR向服务器发起DELETE请求消息；

Express 框架



Express 概述

概述

- Express 是一个基于 Node.js 平台的极简、灵活的 Web 应用开发框架，提供一系列强大的特性，帮助你创建各种 Web 和移动设备应用。

知识讲解

- 丰富的 HTTP 快捷方法和任意排列组合的中间件，让你创建健壮、友好的 API 变得既快速又简单。

- Express 不对 Node.js 已有的特性进行二次抽象，只是在它之上扩展了 Web 应用所需的基本功能。

下载和安装

Tarena Technology 达内科技

知识讲解

- 可以使用 npm 包管理器下载 Express 到当前项目的依赖路径中，立即开始使用 Express 提供的API：

```
npm   install   express
```

- 也可以为当前项目创建包描述文件： package.json 文件，下载 Express ，并把它添加到当前项目的依赖列表中：

```
npm   init
npm   install  express  --save
```

最简单的Web服务器

Tarena Technology 达内科技

知识讲解

- 使用 Express 框架可以很方便的创建基于 HTTP 协议的Web服务器：

```
const express  = require( 'express' );

var app = express( );          //创建 Express 应用程序

var server = app.listen( 80, function( ){
  console.log('start listening ... ')
  console.log( server.address( ).address );
  console.log( server.address( ).port );
});
```

Technology Tarena 达内科技

使用 Express 框架构建服务器应用

- 使用 Express 框架构建服务器应用
- 使用客户端测试

课堂练习

Technology Tarena 达内科技

路由

什么是路由

Technology Tarena 达内科技

- HTTP 客户端发来请求消息，服务器端程序根据请求方法和请求 URI 确定一个响应方法，在 Express 中称为定义了一个“路由”，即一个路由分为三部分：

知识讲解

```
app.METHOD( PATH, HANDLER )
```

路由方法：对应于某个 HTTP 请求方法，如get、post、put、delete等

路由路径：定义了请求的目标，如 /、/index.html、/list、/user/:id、/users?等

路由句柄：是一个请求处理函数，参数是一个请求对象和响应消息对象

请求对象

Technology Tarena 达内科技

- Express 处理请求消息的异步回调函数中，第一个形参是一个请求消息的描述对象，从此对象中可以读取请求消息中的数据：

知识讲解

```
app.get( '/', function( req, res ){
    req.query           //查询字符串数据对象
    req.params          //路由参数对象
    req.cookies         //客户端出示的Cookie数据对象
    req.xhr             //判定是否为XHR请求
    ...
});
```

响应对象

Technology Tarena 达内科技

- Express 处理请求消息的异步回调函数中，第二个形参用于设置向客户端输出的响应数据：

知识讲解

```
app.get( '/', function( req, res ){
    res.status( 200 )    //设置响应状态码
    res.type( 'json' )    //修改 Content-Type 响应头
    res.set( header, value )                //设置响应消息头
    res.send( 'response body' )             //发送响应消息
    res.sendFile( 'path/of/file' )          //发送文件
    res.json( obj )                         //发送JSON响应
    res.redirect( 'path' )                  //响应重定向
    res.cookie( name, value )               //设置Cookie
});
```

处理GET请求

Technology Tarena 达内科技

- 根据 HTTP 协议规定，GET 请求表示客户端想“获得”指定资源。可以使用 express 路由方法中的 get( ) 方法处理客户端提交的 GET 请求：

知识讲解

```
app.get( '/', function( req, res ){
        //req.query              //获取查询字符串数据
});
```

```
app.get( '/user/:uid', function( req, res ){
        //req.params             //读取路由参数
});
```

处理POST请求

Technology Tarena 达内科技

知识讲解

- 根据 HTTP 协议规定，POST 请求表示客户端想“提交并保存”数据到服务器，所以都是有请求主体的。可以使用 express 路由方法中的 post( ) 方法处理客户端提交的 POST 请求：

```
app.post( '/user', function( req, res ){
    req.on( 'data', function(data){
        var body = querystring.parse(data.toString());
    });
});
```

处理PUT请求

Technology Tarena 达内科技

知识讲解

- 根据 HTTP 协议规定，PUT 请求表示客户端想“提交并更新”数据给服务器（且多次重复操作产生的效果是一样的）所以是有请求主体的。可以使用 express 路由方法中的 put( ) 方法处理客户端提交的 PUT 请求：

```
app.put( '/user', function( req, res ){
    req.on( 'data', function(data){
        var body = querystring.parse(data.toString());
    });
});
```

注意：HTML表单只能发起 GET 和 POST 请求，页面中只有使用XHR 才能发起 PUT 和 DELETE 请求！

处理DELETE请求

Technology Tarena 达内科技

知识讲解

- 根据 HTTP 协议规定，DELETE 请求表示客户端想“删除”服务器上的指定资源（没有请求主体）。可以使用 express 路由方法中的 delete( ) 方法处理客户端提交的 DELETE 请求：

```
app.delete( '/user/:uid', function( req, res ){
    //req.params.uid
});
```

路由参数

Technology Tarena 达内科技

知识讲解

- 客户端在向服务器提交请求数据时，除了可以使用传统的查询字符串方式，Express 还允许在请求 URI 中包含请求参数：

```
app.get( '/user/:uid/:pno', function( req, res ){
        req.params.uid
        req.params.pno
});
```

```
app.get( '/user/:uid?/:pno?', function( req, res ){
        req.params.uid
        req.params.pno
});
```

Technology Tarena 达内科技

处理各种请求消息

- 使用 Express 框架处理 GET 请求
- 使用 Express 框架处理 POST 请求
- 使用 Express 框架处理 PUT 请求
- 使用 Express 框架处理 DELETE 请求

课堂练习

Technology Tarena 达内科技

中间件

什么是中间件

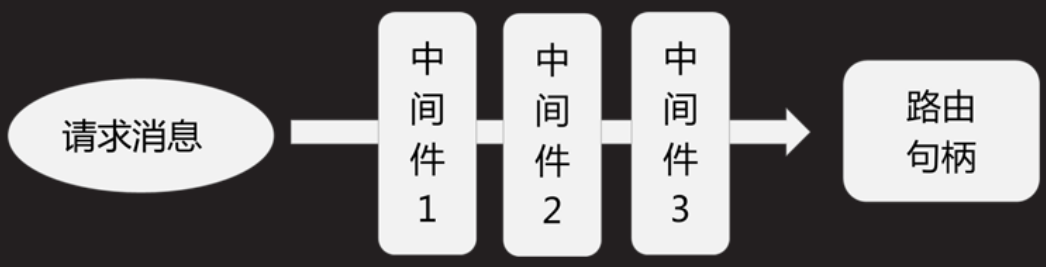
知识讲解

- Express 是一个自身功能极简，完全是由路由和中间件构成一个的 web 开发框架：从本质上来说，一个 Express 应用就是在调用各种中间件。
- 中间件（Middleware） 是一个函数，它可以访问请求对象（req），响应对象（res），和 web 应用中处于请求-响应循环流程中的其它中间件。
- 中间件的功能包括：
  - 执行任何代码；
  - 修改请求和响应对象；
  - 终结请求-响应循环；
  - 调用堆栈中的下一个中间件。

什么是中间件（续1）

知识讲解

- 可以在路由句柄之前或之后声明多个中间件函数，组成一种链式结构；
- 请求消息会按照声明的顺序依次提交给每个中间件函数；
- 中间件或路由句柄内部可以控制next()回调函数的执行以决定是否继续执行下一个中间件函数：



自定义中间件

知识讲解

- 定义中间件函数：

```
function middlewareFn( req, res, next ){
        //...中间件功能代码
        //...处理req和res对象
        next( );//继续调用下一个中间件或路由句柄
}
```

- 使用中间件：

```
app.use( [path], middlewareFn );
//其中path参数可选，指定中间件应用到的请求路径
//middlewareFn必需，指定要执行的中间件函数
```

自定义中间件（续1）

知识讲解

- 下面是一个典型的用于处理 POST 请求消息的中间件：

```
function bodyParser( ){
   return function( req, res, next ){
        req.on( 'data' , function(data){
            req.body = querystring.parse( data.toString() );
            next( );    //继续调用下一个中间件或路由句柄
        });
   }
}
```

```
app.use( bodyParser( ) );
app.post( '/user', function(req, res){
   //从 req.body 中读取数据
})
```

预定义中间件

Tarena Technology 达内科技

- Express 3.x 中预定义了很多中间件；但为了便于维护，Express 4.x 中剥离了大多数的中间件，让它们成为独立的 NPM 包，仅保留了一个 express.static 中间件：

知识讲解

```
var options = {
        dotfiles: 'ignore',
        etag: false,
        extensions: ['htm', 'html'],
        index: false,
        maxAge: '1d',
        redirect: false,
}
app.use(express.static('public', options));
```

使用自定义中间件

Tarena Technology 达内科技

- 创建一个loginCheck中间件，若客户端访问了/admin目录下的任何内容，必须已经登录；否则就自动跳转回管理员登录界面。

课堂练习

Technology Tarena 达内科技

总结和答疑