<!DOCTYPE html>

<html><head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">

<title>校园慕课网</title>

<link rel="stylesheet" href="css/OriginCSS.css" type="text/css" />

<link rel="stylesheet" href="css/iconfont.css">

</head>

<body>

<div id="container" style="width:100%;background: #000000">

<ul>

<li><a class="active" href="#home">主页</a></li>

<li><a href="#news">文章</a></li>

<li><a href="#contact">视频</a></li>

<li><a href="#about">笔记</a></li>

</ul>

<div class="header">

<h1 style="margin:0;">第一章-第1节</h1></div>

<div class="content" id="content">

<div class="leftcontent">

<button id="toggle" class="button"><i class="iconfont icon-xueyuan-mulu"></i><br>章节</button>

<div class="menu">

<b class="mulu">目录</b>

<dl>

<dt><b class="chapter"> 第一章</b></dt>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第1节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第2节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第3节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第4节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第5节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第6节</a></b></dd>

<dt><b class="chapter"> 第二章</b></dt>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第1节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第2节</a></b></dd>

<dd><b class="section">——<a href="www.jslink.com" target="\_blank"style="color: white">第3节</a></b></dd>

</dl>

</div>

</div>

<div class="player">

<video id="myVideo"><source src="test.mp4" type="video/mp4"></video>

<div class="controls">

<div class="buttonzone">

<a href="javascript:" class="switch"><i id="switch" class="iconfont2 icon-bofang"></i></a>

<div class="time">

<span class="currentTime">00:00:00</span>

\

<span class="totalTime">00:00:00</span>

</div>

<div class="bgpanel">

<div id="volume">

<div id="setVolume">

<div id="setVolumeBtn"></div>

</div>

</div>

</div>

<a href="javascript:" class="speed"><div id="speed">1.0</div></a>

<a href="javascript:" class="expand"><i class="iconfont2 icon-quanping"></i></a>

</div>

<div class="progresszone">

<div class="chartzone" id="chartzone">

<canvas id="chart" style="display: block;width: 931px;height: 20px;"></canvas>

</div>

<div id="progress" class="progress">

<div id="elapse" class="elapse">

<div id="setProgressBtn"></div>

</div>

</div>

</div>

</div>

</div>

<div class="rightcontent">

<button id="toggle2" class="button2"><i class="iconfont icon-kecheng"></i><br>相关</button>

<div class="menu2" >

相关推荐课程

</div>

</div>

</div>

<div class="content2">

<div class="shade" style="height: 20px;"></div>

<div class="comment">

<div class="item">

<div class="leftpart">

<div class="portrait"><i class="iconfont icon-touxiang" style="color:#B7B7B7"></i></div>

<div class="name">xxx</div>

</div>

<div class="rightpart"></div>

</div>

<div class="item">

<div class="leftpart">

<div class="portrait"><i class="iconfont icon-touxiang" style="color:#B7B7B7"></i></div>

<div class="name">xxx</div>

</div>

<div class="rightpart">qweqwe</div>

</div>

<div class="item">

<div class="leftpart">

<div class="portrait"><i class="iconfont icon-touxiang" style="color:#B7B7B7"></i></div>

<div class="name">xxx</div>

</div>

<div class="rightpart">qweqwe</div>

</div>

<div class="item">

<div class="leftpart">

<div class="portrait"><i class="iconfont icon-touxiang" style="color:#B7B7B7"></i></div>

<div class="name">xxx</div>

</div>

<div class="rightpart">qweqwe</div>

</div>

</div>

</div>

<div id="footer" style="background-color:#8487FF;clear:both;text-align:center;">

页脚</div>

</div>

<script src='js/Chart.min.js'></script>

<script src="js/jquery.js"></script>

<script>

$(document).ready(function(){

$('#toggle').click(function(){

$('.menu').animate({width:'toggle'},350);//两侧折叠菜单的展开和折叠

});

$('#toggle2').click(function(){

$('.menu2').animate({width:'toggle'},350);

});

});

</script>

<script>

var xmlhttp = new XMLHttpRequest();

var labels = [], data=[];

xmlhttp.onreadystatechange = function() {

var myObj;

if (this.readyState == 4 && this.status == 200) {//判断网页运行状态

myObj = JSON.parse(this.responseText);

for (var i = 0; i < myObj.weight.length; i++) {//将JSON格式的txt文件中的数据赋给data，同时创建对应数量的空标签

labels[i] = " ";

data[i] = myObj.weight[i];

}

var ctx = document.getElementById("chart").getContext('2d');//使用jQuery.canvas以及chart.js来构成曲线，实际上是表格

var gradientStroke = ctx.createLinearGradient(500, 0, 100, 0);//设置曲线的样式，包括颜色渐变等等

gradientStroke.addColorStop(0, "#ff6c00");

gradientStroke.addColorStop(1, "#ff3b74");

var gradientBkgrd = ctx.createLinearGradient(0, 100, 0, 400);//曲线周围的光晕

gradientBkgrd.addColorStop(0, "rgba(244,94,132,0)");

gradientBkgrd.addColorStop(1, "rgba(249,135,94,0)");

var draw = Chart.controllers.line.prototype.draw;

Chart.controllers.line = Chart.controllers.line.extend({

draw: function() {

draw.apply(this, arguments);

var ctx = this.chart.chart.ctx;

var \_stroke = ctx.stroke;

ctx.stroke = function() {

ctx.save();

ctx.shadowColor = 'rgba(244,94,132,0.8)';

ctx.shadowBlur = 8;

ctx.shadowOffsetX = 0;

ctx.shadowOffsetY = 6;

\_stroke.apply(this, arguments)

ctx.restore();

}

}

});

var chart = new Chart(ctx, {

type: 'line',

data: {

labels:labels,//在这里labels设置为之前存好的labels1

datasets: [{

backgroundColor: gradientBkgrd,//线条下方颜色

borderColor: gradientStroke,//线条颜色：混合色

data:data,//这里data已经设置为JSON中读取的data

pointBorderColor: "rgba(255,255,255,0)",

pointBackgroundColor: "rgba(255,255,255,0)",

pointBorderWidth: 0,

pointHoverRadius: 8,

pointHoverBackgroundColor: gradientStroke,

pointRadius: 1,

borderWidth: 1,//修改曲线宽度

pointHitRadius: 0,//点显示的触发范围,设为0则永远不会显示

}]

},

options: {

legend: {display: false},

scales: {

xAxes: [{

display: false,//不让x轴显示

gridLines: {display:false}

}],

yAxes: [{

display: false,//不让y轴显示

gridLines: {display:false}//取消网格线

}]

}

}

});

}

};

xmlhttp.open("GET", "json\_demo.txt", true);//method：请求的类型；GET 或 POST，url：文件在服务器上的位置，async：true（异步）或 false（同步）

xmlhttp.send();//

</script>

<script>

var totalVolume = document.getElementById("volume");

var volume = document.getElementById("setVolume");

var setVolumeBtn = document.getElementById("setVolumeBtn");

//拖动设置音量

setVolumeBtn.onmousedown = function () {

document.onmousemove = function () {

changeVolume();

};

};

//拖动后弹起鼠标不做任何操作

document.onmouseup = function () {

document.onmousemove = null;

};

//点击音量条跳至相应音量

totalVolume.onclick = function () {

changeVolume();

};

//鼠标进入音量条时出现按钮

totalVolume.onmouseover = function () {

setVolumeBtn.style.display = "block";

};

//鼠标离开音量条时按钮消失

totalVolume.onmouseout = function () {

setVolumeBtn.style.display = "none";

};

//音量改变

function changeVolume(event) {

var e = event || window.event;

var volumeWidth;

volumeWidth = e.clientX - totalVolume.offsetLeft-237.5;

if (volumeWidth < 0) {

volumeWidth = 0;

} else if (volumeWidth > totalVolume.offsetWidth) {

volumeWidth = totalVolume.offsetWidth;

}

volume.style.width = volumeWidth + "px";

setVolume();

}

//音量-宽度转换

function setVolume() {

var theTotalWidth = totalVolume.offsetWidth;

var theWidth = volume.offsetWidth;

document.getElementById("myVideo").volume = theWidth / theTotalWidth;

}

</script>

<script>

$(function ($) {

var video = $("video")[0];

var totalProgress = document.getElementById("progress");

var progress = document.getElementById("elapse");

var setProgressBtn = document.getElementById("setProgressBtn");

$(".speed").on("click", function () {

if (video.playbackRate==1.0) {

video.playbackRate=1.25;

$(".speed").html(1.25);

return;

}

if (video.playbackRate==1.25) {

video.playbackRate=1.5;

$(".speed").html(1.5);

return;

}

if (video.playbackRate==1.5) {

video.playbackRate=2.0;

$(".speed").html(2.0);

return;

}

if (video.playbackRate==2.0) {

video.playbackRate=0.75;

$(".speed").html(0.75);

return;

}

if (video.playbackRate==0.75) {

video.playbackRate=1.0;

$(".speed").html(1.0);

}

});

$(".switch").on("click", function () {

if (video.paused) {

video.play();

document.getElementById("switch").className="iconfont2 icon-zanting";

/\*移除暂停样式 添加播放样式\*/

} else {

video.pause();

document.getElementById("switch").className="iconfont2 icon-bofang";

/\*移除播放样式，添加暂停样式\*/

}

});

$(".expand").on("click", function () {

/\*全屏>>不同浏览器添加不同前缀\*/

if (video.requestFullscreen) {

video.requestFullscreen();

} else if (video.webkitRequestFullscreen) {

video.webkitRequestFullscreen();

} else if (video.mozRequestFullScreen) {

video.mozRequestFullScreen();

}

});

video.oncanplay = function () {//一开始设置初始的00：00：00值

setTimeout(function () {

/\*1.将视频文件设置为显示\*/

video.style.display = "block";

/\*2.获取当前文件的总时长 : 计算出 时分秒

\* duration:是以秒为单位，同时获取了小数值

\* \*/

var total = video.duration;

/\*3.计算时分秒\*/

var result = getResult(total);

/\*4.将计算结果展示在DOM元素中\*/

$(".totalTime").html(result);

}, 2000);

};

function getResult(totalTime) {//将输入的totaltime转换为时：分：秒的格式并返回

/\*时\*/

var hour = Math.floor(totalTime / 3600);

/\*补0操作\*/

hour = hour < 10 ? "0" + hour : hour;

/\*分\*/

var minute = Math.floor(totalTime % 3600 / 60);

minute = minute < 10 ? "0" + minute : minute;

/\*秒\*/

var second = Math.floor(totalTime % 60);

second = second < 10 ? "0" + second : second;

/\*返回结果\*/

return hour + ":" + minute + ":" + second;

}

video.ontimeupdate = function () {//只要播放进度改变，就改变进度条宽度以及00：00：00

/\*1.获取当前的播放时间\*/

var current = video.currentTime;

/\*2.计算出时分秒\*/

var result = getResult(current);

/\*3.将结果展示在指定的DOM中\*/

$(".currentTime").html(result);

/\*4.设置当前播放进度条样式 例如0.12 >> 0.12\*100 = 12 + % >> 12% \*/

var percent = current / video.duration \* 100 + "%";

$(".elapse").css("width", percent);

};

video.onended = function () {//视频结束后，进度置0，改变图标

video.currentTime = 0;

video.pause();

document.getElementById("switch").className="iconfont2 icon-bofang";

};

//鼠标进入进度条时出现按钮

totalProgress.onmouseover = function () {

setProgressBtn.style.display = "block";

};

//鼠标离开进度条时按钮消失

totalProgress.onmouseout = function () {

setProgressBtn.style.display = "none";

};

var timeDrag = false;

$(".progress").on("mousedown", function (e) {

/\*1.获取当前鼠标相对于父元素的left值--- 偏移值\*/

var offset = e.offsetX;

/\*2.计算偏移值相对于总父元素宽度的比例\*/

var percent = offset / $(this).width();

/\*3.计算这个位置对应的视频进度值\*/

/\*4.设置当前视频的currentTime\*/

video.currentTime = percent \* video.duration;

video.pause();

document.onmousemove = function () {

changeProgress();

};

timeDrag = true;

});

$(".setProgressBtn").on("mousedown", function () {

video.pause();

document.onmousemove = function () {

changeProgress();

};

timeDrag = true;

});

$(document).on("mouseup", function () {

if(timeDrag) {

timeDrag = false;

video.play();

}

});

//进度改变

function changeProgress(event) {

var e = event || window.event;

var progressWidth;

progressWidth = e.clientX - totalProgress.offsetLeft-237.5;

if (progressWidth < 0) {

progressWidth = 0;

} else if (progressWidth > totalProgress.offsetWidth) {

progressWidth = totalProgress.offsetWidth;

}

progress.style.width = progressWidth + "px";

var theTotalWidth = totalProgress.offsetWidth;//视频总长度

var theWidth = progress.offsetWidth;//已经走过的elapse的长度

var percent = theWidth / theTotalWidth;

video.currentTime = percent \* video.duration;

}

})

</script>

<script type="text/javascript">

var startFlag = 0;

var endFlag = 0;

var startTime = 0;

var endTime = 0;

var my\_video = document.getElementById("v1");

my\_video.addEventListener("mousedown", mouseDown);

my\_video.addEventListener("mouseup", mouseUp);

function mouseDown() {

startFlag = 0;

my\_video.addEventListener("seeking", getSeekStart);

}

function mouseUp() {

endFlag = 0;

getSeekEnd();

}

function getSeekStart() {

if ((my\_video.seeking) && (startFlag != 1)) {

startTime = my\_video.currentTime;

startFlag = 1;

}

}

function getSeekEnd() {

if (endFlag != 1) {

endTime = my\_video.currentTime;

//传值：Url,Speed,StartTime,EndTime

video\_request = false;

if (window.XMLHttpRequest) {

video\_request = new XMLHttpRequest();

} else if (window.ActiveXObject) {

try {

video\_request = new ActiveXObject("Msxml2.XMLHTTP");

} catch (e) {

try {

video\_request = new ActiveXObject(

"Microsoft.XMLHTTP");

} catch (e) {}

}

}

video\_request.open('POST',window.location.href,true);

video\_request.setRequestHeader("Content-Type",\"application/x-www-form-urlencoded");//创建JSON文件

<%

StringBuilder sb = new StringBuilder(request.getRemoteAddr().toString());

for(int i=0;i<sb.length();i++){

if(sb.charAt(i)==':'){

sb.setCharAt(i, '.');

}

}

String ip = sb.toString();

%>

var ip = "<%=ip%>";

var param = new String();

param="url="+window.location.href;

param+="&id="+ip;

param+="&length="+my\_video.duration;

param+="&startTime="+startTime;

param+="&endTime="+endTime;

param+="&speed="+my\_video.defaultPlaybackRate;//写入参数

video\_request.send(param);

<%

Video video = new Video();

int flag\_video = 0;

if(request.getParameter("url")!=null){

video.setUrl(request.getParameter("url"));

flag\_video++;

}

if(request.getParameter("id")!=null){

video.setId(request.getParameter("id"));

flag\_video++;

}

if(request.getParameter("length")!=null){

video.setLength(Double.parseDouble(request.getParameter("length")));

flag\_video++;

}

if(request.getParameter("startTime")!=null){

video.setStartTime(Double.parseDouble(request.getParameter("startTime")));

flag\_video++;

}

if(request.getParameter("endTime")!=null){

video.setEndTime(Double.parseDouble(request.getParameter("endTime")));

flag\_video++;

}

if(request.getParameter("speed")!=null){

video.setSpeed(Double.parseDouble(request.getParameter("speed")));

flag\_video++;

}

if(flag\_video==6){

GetLine.newInstase().getLine(video);

flag\_video=0;

}

%>//检查格式

endFlag = 1;

}

}

</script>

</body>

</html>

@font-face {font-family: "iconfont";

src: url('iconfont.eot?t=1577428319212'); /\* IE9 \*/

src: url('iconfont.eot?t=1577428319212#iefix') format('embedded-opentype'), /\* IE6-IE8 \*/

url('data:application/x-font-woff2;charset=utf-8;base64,') format('woff2'),

url('iconfont.woff?t=1577428319212') format('woff'),

url('iconfont.ttf?t=1577428319212') format('truetype'), /\* chrome, firefox, opera, Safari, Android, iOS 4.2+ \*/

url('iconfont.svg?t=1577428319212#iconfont') format('svg'); /\* iOS 4.1- \*/

}

@font-face {font-family: "iconfont";

src: url('iconfont.eot?t=1577435114430'); /\* IE9 \*/

src: url('iconfont.eot?t=1577435114430#iefix') format('embedded-opentype'), /\* IE6-IE8 \*/

url('data:application/x-font-woff2;charset=utf-8;base64,') format('woff2'),

url('iconfont.woff?t=1577435114430') format('woff'),

url('iconfont.ttf?t=1577435114430') format('truetype'), /\* chrome, firefox, opera, Safari, Android, iOS 4.2+ \*/

url('iconfont.svg?t=1577435114430#iconfont') format('svg'); /\* iOS 4.1- \*/

}

.iconfont {

font-family: "iconfont" !important;

font-size: 35px;

font-style: normal;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

.iconfont2 {

font-family: "iconfont" !important;

font-size: 28px;

font-style: normal;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

color: #ffffffc2;

}

.icon-bofang:before {

content: "\e630";

}

.icon-zanting:before {

content: "\e635";

}

.icon-touxiang:before {

content: "\e620";

}

.icon-kecheng:before {

content: "\e609";

}

.icon-quanping:before {

content: "\e622";

}

.icon-xueyuan-mulu:before {

content: "\e6c9";

}

package hadoop.video.service.impl;

public class Video {

private String id;

private String url;

private double length;

private double startTime;

private double endTime;

private double speed;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public double getLength() {

return length;

}

public void setLength(double length) {

this.length = length;

}

public double getStartTime() {

return startTime;

}

public void setStartTime(double startTime) {

this.startTime = startTime;

}

public double getEndTime() {

return endTime;

}

public void setEndTime(double endTime) {

this.endTime = endTime;

}

public double getSpeed() {

return speed;

}

public void setSpeed(double speed) {

this.speed = speed;

}

}

package hadoop.video.util;

import java.util.Date;

import hadoop.video.service.impl.Video;

public class GetLine {

public static GetLine newInstase(){

return new GetLine();

}

public void getLine(Video video){

String s = new String();

s=video.getUrl()+"-"+video.getId()+"-"+video.getLength()+"-"+video.getStartTime()+"-"+video.getEndTime()+"-"+video.getSpeed()+"+";

String path = new String();

path = "C:/Users/84492/Desktop/test/test"+video.getId();

TextTransfer.newInstanse().save(path, s);

}

}

package hadoop.video.util;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class TextTransfer {

public static TextTransfer newInstanse(){

return new TextTransfer();

}

public String read(File file){

String result = readText(file);

return result;

}

public boolean save(String path, String content){

return writeText(path, content);

}

private String readText (File file){

StringBuilder result = new StringBuilder();

try{

BufferedReader br = new BufferedReader(new FileReader(file));

String s =null;

while((s = br.readLine())!=null){

result.append(s+System.lineSeparator());

}

br.close();

}catch(Exception e){

e.printStackTrace();

}

return result.toString();

}

private boolean writeText (String path, String content){

try{

File textName = new File(path);

String result = new String();

if(!textName.exists()){

boolean hasFile = textName.createNewFile();

if(hasFile){

result = content;

}

}else{

result = readText(textName).trim()+content;

}

try(FileWriter writer = new FileWriter(textName);BufferedWriter out = new BufferedWriter(writer)){

out.write(result);

out.flush();

}

}catch(IOException e){

e.printStackTrace();

}

return false;

}

}