前端门户系统

访问：http://edufront.lagou.com/

用户名：15510792995 密码：111111

- 页面不需要我们自己开发，使用提供的页面即可
- 运行项目 `npm run serve`

1、首页显示全部课程

- Index.vue

```
<script>
import Header from "./Header/Header"; // 顶部登录条
import Footer from "./Footer/index"; // 顶部登录条
export default {
  name: "Index",
  components: {
    Header,
    Footer,
  },
  data() {
    return {
      courseList: [], // 课程集合
      activeName: "allLesson",
    };
  },
  created() {
    this.getCourseList();  // 当组件创建完成，调用获取课程方法
  },
  methods: {
    changeCourseTab(tabName) {
      this.classSelect = tabName;
      sessionStorage && sessionStorage.setItem("courseTab", tabName);
    },
    gotoDetail() {
      this.$router.push({ name: "Course", params: { courseid: 1 } });
    },
    getCourseList() {
      return this.axios
        .get("http://localhost:8002/course/getAllCourse")
        .then((result) => {
          console.log(result);
          this.courseList = result.data; // 将返回的数据 赋值给 当前组件中的集合
        })
        .catch((error) => {
          console.log(error);
          this.$message.error("获取数据失败!");
        });
    },
```

```
  },
};
</script>
```

```
<!-- 课程信息展示开始 -->
<li class="course-li"  v-for="(item,index) in courseList" :key="index">
  <!-- 课程封面图 -->
  <img
    :src="item.courseImgUrl"
    class="teacher-portrait hover-pointer"
  />
  <!-- 课程文字信息 -->
  <div class="content-main">
    <!-- 课程标题 -->
    <div class="content-title hover-pointer">
      <div
        class="p-title"
        style="text-align:left;"
        @click="gotoDetail"
      >
        <span>
          {{item.courseName}}
        </span>
      </div>
      <!-- 作者和职称 -->
      <p class="p-title-buy text-overflow">
        <span class="p-author-span">
          {{item.teacher.teacherName}}
        </span>
        <span class="p-author-line" />
        <span class="p-author-span">
          {{item.teacher.position}}
        </span>
      </p>
      <p></p>
      <!-- 课程简单描述 -->
      <p class="p-describe" style="text-align:left;">
        {{item.brief}}
      </p>
    </div>
    <!-- 课程前两个章节信息 -->
    <ul class="content-course" style="text-align:left;">
      <!-- 章节1 : 免费试看，通常是第一章的前两节课 -->
      <li
        class="content-course-lesson text-overflow"
        style="width:300px"
        v-for="(lesson , index) in
item.courseSections[0].courseLessons.slice(0,2)" :key="index">
        <!-- 免费试看图标 -->
        <img
          src="@/assets/course-list/free-course.png"
          class="free-label hover-pointer"
        />
        <span class="theme-span hover-pointer">
          {{lesson.theme}}
          </span>
      </li>
```

```
    </ul>
    <!-- 价格信息 -->
    <div class="content-price" style="text-align:left;">
      <p class="content-price-p">
        <span class="content-price-orange-sm">￥</span>
        <span class="content-price-orange">{{ item.discounts }}</span>
        <span class="current-price">
          <span class="current-price-unite">￥</span>
          {{ item.price }}
        </span>
        <span class="activity-name">成就自己</span>
        <span class="content-price-buy">{{ item.sales }}人购买</span>
      </p>
      <div class="btn btn-green btn-offset">立即购买</div>
    </div>
  </div>
</li>
<!-- 课程信息结束 -->
```

2、登录

```
<script>
export default {
  name: "Header",

  props: {},
  data() {
    return {
      isLogin: false, // 登录状态，true：已登录，false：未登录
      userDTO:null, // 用来保存登录的用户信息
      isHasNewMessage: false, // 是否有新的推送消息
      dialogFormVisible: false, // 是否显示登录框，true：显示，false：隐藏
      phone: "", // 双向绑定表单 手机号
      password: "", // 双向绑定表单 密码
    };
  },
  computed: {
  },
  watch: {
  },
  mounted() {
  },
  created(){
    // 当刷新页面，组件创建成功之后，立刻检测本地储存中是否存在用户对象
    this.userDTO = JSON.parse( localStorage.getItem("user") );
    if(this.userDTO != null){
        this.isLogin = true; // 已登录
    }
  },
  methods: {
    goToSetting() {
      this.$router.push("/setting"); // 跳转个人设置页面
    },
```

```
    goToLogin() {
      this.dialogFormVisible = true; // 显示登录框
    },
    login(){ // 前去登录
      return this.axios
      .get("http://localhost:8002/user/login" , {
        params:{
          phone:this.phone,
          password:this.password
        }
      })
      .then( (result)=>{
        console.log( result );
        this.dialogFormVisible = false ; //关闭登录框
        this.userDTO = result.data; // 保存返回数据中的用户对象信息
        this.isLogin = true; // 更新登录状态
        localStorage.setItem("user", JSON.stringify(this.userDTO)); // 将登录成功
的对象信息保存到本地储存中
      } )
      .catch( (error)=>{
        this.$message.error("登录失败！");
      });
    },
    goToLoginWX() {
      alert("微信登录");
    },
    toToIndex() {
      this.$router.push("/"); //回到首页
    },
    toToNotic(){
    },
    logout(){ //登出
      localStorage.setItem("user", null); // 将登录成功的对象信息保存到本地储存中
      this.isLogin = false; // 更新登录状态
      alert('谢谢使用，再见！');
    }
  },
};
</script>
```

3、已购课程

```
<script>
import Header from "./Header/Header"; //顶部登录条
import Footer from "./Footer/index"; //顶部登录条
export default {
  name: "Index",
  components: {
    Header,
    Footer,
  },
  data() {
    return {
      activeName: "allLesson",
```

```
      courseList:[],  // 课程集合
      myCourseList:[], // 已登录用户购买的课程集合
      user:null, // 登录的用户对象
      isLogin:false // 登录状态
    };
  },
  created() {
    this.getCourseList(); //当组件创建完毕，就调用获取课程的方法
    this.user = JSON.parse( localStorage.getItem("user") );
    if(this.user != null){
      this.getCourseByUserId(); //查询已经用户购买过的课程列表
      this.isLogin = true; // 改变登录状态
    }
  },
  methods: {
    changeCourseTab(tabName) {
      this.classSelect = tabName;
      sessionStorage && sessionStorage.setItem("courseTab", tabName);
    },
    gotoDetail() {
      this.$router.push({ name: "Course", params: { courseid: 1 } });
    },
    getCourseList(){ // 去dubbo服务获取全部课程的数据
      return this.axios
      .get("http://localhost:8002/course/getAllCourse")
      .then((result) => {
        console.log(result);
        this.courseList = result.data;
      }).catch( (error)=>{
        this.$message.error("获取课程信息失败！");
      } );
    },
    getCourseByUserId(){
    return this.axios

.get("http://localhost:8002/course/getCourseByUserId/"+this.user.content.id)
      .then((result) => {
        console.log( result);//购买的
        this.myCourseList = result.data;
      }).catch( (error)=>{
        this.$message.error("获取课程信息失败！");
      } );
    }
  },
};
</script>
```

```
<el-tab-pane label="已购" name="hasPay">
  <div v-if="!isLogin">
    <img src="@/assets/course-list/no-login@2x.png" class="no-data-icon"/>
    <div class="no-data-title">您还没有登录</div>
    <div class="no-data-title">登录后即可查看已购课程</div>
    <div class="btn btn-yellow btn-center">立即登录</div>
  </div>
  <div v-if="isLogin">
   <ul class="course-ul-pc">
        <!-- 已购买的课程信息展示开始 -->
```

```
        <li>...省略...</li>
        <!-- 已购买的课程信息结束 -->
  </ul>
  </div>
</el-tab-pane>
```

4、课程详情-基本信息

Index.vue

```
<div @click="gotoDetail(item)">
    <span>
        {{item.courseName}}
    </span>
</div>

<script>
    gotoDetail(item) {
      // 将课程对象传递到课程详情组件
      this.$router.push({ name: "Course", params: { course: item } });
    }
</script>
```

Course.vue

```
<script>
data() {
    return {
      activeName: "intro",
      course:null
    };
},
created(){
    this.course = this.$route.params.course;  // 从路由中获得参数对象赋值给本组件的参数
}
</script>
```

```
<img
  class="course-img"
  :src="course.courseImgUrl"
  alt="课程图片"
/>
<div class="conent-wrap">
  <div class="name" style="text-align:left;">
    {{course.courseName}}
  </div>
  <div class="des text-omit" style="text-align:left;">
    {{course.brief}}
  </div>
  <div class="title">
    <div class="teacher-name text-omit">
      讲师：{{course.teacher.teacherName}}
```

```
      <span class="line"></span>
     {{course.teacher.position}}
    </div>
  </div>
  <div class="lesson-info">
    <div class="boook-icon backgroun-img-set"></div>
    <div class="time">100 讲 / {{course.totalDuration}} 课时</div>
    <div class="person-icon backgroun-img-set"></div>
    <div class="person">{{course.sales}}人已购买</div>
  </div>
</div>
```

5、课程详情-共计多少讲

```
<div class="time">{{totalLessons}} 讲 / {{course.totalDuration}} 课时</div>
```

```
<script>
data() {
    return {
      activeName: "intro",
      course:null,
      totalLessons:0, // 本门课程的总节数(多少讲)
    };
  },
  created(){
    this.course = this.$route.params.course;  // 从路由中获得参数对象赋值给本组件的参数

    // 计算多少节课要讲
    let x = 0;
    for(let i = 0; i< this.course.courseSections.length; i++){
        let section = this.course.courseSections[i]; //每一章
        for( let j = 0; j<section.courseLessons.length ; j++){
          x++;
        }
    }
    this.totalLessons = x;
  }
</script>
```

6、课程详情-实现课程信息描述

html不识别数据库的html标签，必须使用v-html才可以

```
<el-tab-pane label="课程信息" name="intro">
    <div v-html="course.courseDescription" class="content-p pc-background">
</div>
```

7、课程详情-显示章节目录

```
<el-tab-pane label="目录" name="directory">
    <div class="class-menu-contaniner list-page-container more-sections more-
sections-padding">
      <!-- 章 开始 -->
      <div v-for="section in course.courseSections" >
        <div class="section-name single-line">
          {{section.sectionName}}
        </div>
        <div class="class-menu-block">
          <!-- 每节课 开始 -->
          <div
            class="class-level-one over-ellipsis"
            @click="watchCourse(1)"
            v-for="(lesson , index) in section.courseLessons" :key="index">
            <div class="text-wrap">
              <div class="content">{{lesson.theme}}</div>
              <div class="item-status-wrap item-status-wrap-list">
                  <div class="item-status test-watch">试看</div>
              </div>
            </div>
          </div>
          <!-- 每节课 结束 -->
        </div>
      </div>
      <!-- 章 结束 -->
    </div>
</el-tab-pane>
```

8、课程详情-显示全部留言

```
<script>
data() {
    return {
      activeName: "intro",
      course:null,
      totalLessons:0, // 本门课程的总节数
      commentList:null,  // 所有留言
    };
  },
  created(){
    this.course = this.$route.params.course;  // 从路由中获得参数对象赋值给本组件的参
数

    // 计算多少节课要讲
    let x = 0;
    for(let i = 0; i< this.course.courseSections.length; i++){
        let section = this.course.courseSections[i]; //每一章
        for( let j = 0; j<section.courseLessons.length ; j++){
          x++;
        }
```

```
    }
    this.totalLessons = x;

    // 获得所有留言
    this.getComment();

  },
  methods: {
    //播放视频
    watchCourse(index) {
      alert("观看第【" + index + "】节课程视频！");
    },
    buy(courseid) {
      alert("购买第【" + courseid + "】门课程成功，加油！");
    },
    getComment(){
      return this.axios

.get("http://localhost:8002/course/comment/getCourseCommentList/"+this.course.id
+"/1/20")
      .then((result) => {
        //console.log(result);
        this.commentList = result.data;
      }).catch( (error)=>{
        this.$message.error("获取留言信息失败！");
      } );
    }
  }
</script>
```

```
<!-- 留言板 开始-->
<div class="message">
  <div class="message-topic">
    <div class="message-topic-title normal-font">精选留言</div>
  </div>
  <div>
    <div class="message-edit">
      <textarea rows="20" style="border:0px;resize: none;"
        contenteditable="true"
        placeholder="分享学习心得、思考感悟或者给自己一个小鼓励吧！"
        class="edit-div pcStyle"
      ></textarea>
      <div class="message-edit-count">
        <span class="message-edit-count-cur">0</span>
        <span class="message-edit-count-max">/2000</span>
      </div>
    </div>

    <div class="message-edit-footer flex">
      <button class="message-edit-btn disableBg" @click="saveComment">发表留言
</button>
    </div>
  </div>

  <!-- 留言 开始 -->
  <div class="message-list" v-for="(comment , index) in commentList"
:key="index">
```

```
    <div class="message-list-title">
      <div class="message-list-title-left">
        <div class="message-list-title-left-name">{{comment.userName}}</div>
        <div class="message-list-title-left-tag"></div>
      </div>
      <div class="message-list-title-right">
        <img class="message-list-title-right-icon"
src="
EVHcExnZ2dzc3NmZmZqampmZmZmZmZnZ2dnZ2dnZ2dmZmZoaGhmZmZmZmZl+8SAAAAADXRSTlMA/AbsF
tilbj5YwSqJPyESoQAAAZxJREFUOMt1lTtLA1EQha8xRhPTBEmhuKCCoNgoIlYLMcRKBG0sxIUgCDaBS
DohEO0FEbQyIBZaBazERvAPWCwxPnP+i3tnrlGTmVPswn73NXNm7hrzq9m9kZ2ckTUUABifkOEBrK7li
R7BMRFOA/uFc+BUgnV8mFisEW5IsIFi9FzBuwR91KJnAm8S9EIbxSBeBRZHk86MrBQJWjymJUC3nlugS
yk+SQyhANfxos+s4krfMODZvmbw2cuSCHNGi3PAfUygXYiU79ryyw1ibf0xZ9intBsz6SBadx24iiZXz
8kPxCiTtYdLPzKTVFkkLQAZO/VikwYW/x/wHohcT/MiPQE8W9frxJrlbpiw4xvA0vbNmWyhj2Nrhmy+B
7nEyTsN0rIaJAc0SDWqwX7rhAYfMa/Dui0bDZbwZAwUGNjWUWActnUUyN2hwDTaOkxRaSiwj6pRhjHKg
TazSkWlwBK1jgIpBwrkHCgwyZ0oQ86BAjkHCjziGOKE8YBvCA/5KacOm6sgrHFAotouT6J23bkkLbsND
jM9yt7yP+IbQYga5De+eBMAAAAASUVORK5CYII=" alt="">
        <div class="message-list-title-right-praise">{{comment.likeCount}}</div>
      </div>
    </div>
    <div class="message-list-content">
      {{comment.comment}}
    </div>
  </div>
  <!-- 留言 结束 -->

</div>
<!-- 留言板 结束-->
```

9、课程详情-章节状态

```
<script>
import Header from "./Header/Header"; //顶部登录条
import Footer from "./Footer/index"; //顶部登录条
export default {
  name: "Course",
  components: {
    Header,
    Footer,
  },
  data() {
    return {
      activeName: "intro",
      course:null,
      totalLessons:0, // 本门课程的总节数
      commentList:null,  // 所有留言
      isLogin:false, // false 未登录
      myCourseList:[], // 我购买过的课程列表
      isBuy:false, // false 未购买
      user:null, // 登录的用户
    };
  },
  created(){
```

```
    this.course = this.$route.params.course;  // 从路由中获得参数对象赋值给本组件的参数

    // 检测是否登录
    this.user = JSON.parse( localStorage.getItem("user") );
    if( this.user != null ){
        this.isLogin = true; // 已登录
        this.getMyCourseList(); // 调用查询我购买的课程方法
    }

    // 计算多少节课要讲，略。。。
    // 获得所有留言，略。。。
  },
  methods: {
    //章节，课程索引，文件播放地址
    watchCourse(i,index,file) {
      if(i==1 && (index==0||index==1)){
        alert("观看第【" + index + "】节课程视频！"+file);
        this.$router.push({name: "videoDetail",params: { file: file }});
      }else{
        if(!this.isLogin){
            alert("请先登录！");
        }else{
          if(!this.isBuy){
            alert("请购买解锁才能观看本视频！");
          }else{
            alert("观看第【" + index + "】节课程视频！"+file);
            this.$router.push({name: "videoDetail",params: { file: file }});
          }
        }
      }
    },
    buy(courseid) {
      alert("购买第【" + courseid + "】门课程成功，加油！");
    },
    getComment(){
     // 获取全部留言，省略。。。
    },
    saveComment(){},
    getMyCourseList(){
      return this.axios

.get("http://localhost:8002/course/getCourseByUserId/"+this.user.content.id)
      .then((result) => {
        console.log("获得我的课程");
        console.log(result);
        this.myCourseList = result.data;

        // 检查我是否购买过本门课程
        console.log(this.myCourseList.length);
        for(let i = 0; i< this.myCourseList.length ; i++){
          if(this.myCourseList[i].id == this.course.id){
              this.isBuy = true;//标记购买过本次课程
              break;
          }
        }
      }).catch( (error)=>{
        this.$message.error("获取课程信息失败！");
```

```
      } );
    }
  },
};
</script>
```

```
<el-tab-pane label="目录" name="directory">
<div class="class-menu-contaniner list-page-container more-sections more-
sections-padding">
  <!-- 第一章 开始 -->
  <div v-for="section in course.courseSections.slice(0,1)" >
    <div class="section-name single-line">
      {{section.sectionName}}
    </div>
    <div class="class-menu-block">
      <!-- 每节课 开始 -->
      <div
        class="class-level-one over-ellipsis"
        @click="watchCourse(1,i,lesson.courseMedia.fileEdk)"
        v-for="(lesson , i) in section.courseLessons" :key="i">
        <div class="text-wrap">
          <div class="content">{{lesson.theme}}</div>
          <div class="item-status-wrap item-status-wrap-list">
            <!--第一章，前两节-->
            <div v-if="i == 0 || i==1">
              <!--未登录，试看-->
              <div v-if="!isLogin" class="item-status test-watch">试看</div>
              <!--已登录，未购买，试看-->
              <div v-else-if="isLogin && !isBuy" class="item-status test-watch">
试看</div>
              <!--已登录，已购买，播放-->
              <div v-else class="item-status test-watch">播放</div>
            </div>
            <!--第一章，除了前两节后面的-->
            <div v-if="i != 0 && i!=1">
              <!--未登录，锁-->
              <div v-if="!isLogin" class="item-status lock"></div>
              <!--已登录，未购买，锁-->
              <div v-else-if="isLogin && !isBuy" class="item-status lock"></div>
              <!--已登录，已购买，播放-->
              <div v-else class="item-status test-watch">播放</div>
            </div>
          </div>
        </div>
      </div>
      <!-- 每节课 结束 -->
    </div>
  </div>
  <!-- 第一章 结束 -->
  <!-- 其余章 开始 -->
  <div v-for="section in
course.courseSections.slice(1,course.courseSections.length)" >
    <div class="section-name single-line">
      {{section.sectionName}}
    </div>
    <div class="class-menu-block">
      <!-- 每节课 开始 -->
```

```
      <div
        class="class-level-one over-ellipsis"
        @click="watchCourse(2,j,lesson.courseMedia.fileEdk)"
        v-for="(lesson , j) in section.courseLessons" >
        <div class="text-wrap">
          <div class="content">{{lesson.theme}}</div>
          <div class="item-status-wrap item-status-wrap-list">
            <!--已登录，已购买，播放-->
            <div v-if="isLogin&&isBuy" class="item-status test-watch">播放</div>
            <!--未登录，锁-->
            <div v-else-if="!isLogin" class="item-status lock"></div>
            <!--已登录，未购买，锁-->
            <div v-if="isLogin && !isBuy" class="item-status lock"></div>
          </div>
        </div>
      </div>
      <!-- 每节课 结束 -->
    </div>
  </div>
  <!-- 其余章 结束 -->
</div>
</el-tab-pane>
```

10、在课程详情页点击视频播放

Course.vue

```
<div
    class="class-level-one over-ellipsis"
    @click="watchCourse(2,lesson.id,index,lesson.courseMedia)"
    v-for="(lesson , index) in section.courseLessons" :key="index">
<script>
   //播放视频 ( 章节，小节课编号，每节课的索引，每节课的视频对象)
   watchCourse(status,lessonid,index,media) {
      this.$router.push({ name: "videoDetail", params: { course: this.course ,
lessonid:lessonid } });
</script>
```

videoDetail.vue

```
<script>
data() {
    return {
      myvideo: null, // 播放器对象
      isplay: false, //是否在播放
      nowTime: "00:00", //当前播放时间
      totalTime: "00:00", //总时长
      course:null, // 课程
      lessonid:0, // 当前播放视频的课节id
    };
  },
  computed: {},
  created() {
```

```
    // 判断登录(暂无)

    this.course = this.$route.params.course;
    this.lessonid = this.$route.params.lessonid; // 点击的视频编号
  },
  mounted() {
    this.myvideo = document.getElementById("myvideo");
    this.initplay();   // 通过id找到点击视频的地址并播放
  },
  methods:{
    // 初始化时播放的视频
    initplay(){
      //1.在课程信息中查找即将播放的小节视频的编号
      for( let i = 0 ; i< this.course.courseSections.length;i++ ){
          let section = this.course.courseSections[i];
          for(let j = 0; j<section.courseLessons.length ; j++){
              let lesson = section.courseLessons[j]; // 每节课
              if(lesson.courseMedia!=null){
                if(this.lessonid == lesson.courseMedia.lessonId){
                  console.log("视频地址：" + lesson.courseMedia.fileEdk);
                  //2.将小节视频的地址 赋值 给播放器，进行播放
                  this.myvideo.src = lesson.courseMedia.fileEdk;
                  return;
                }
              }
          }
       }
     }
  }
</script>
```

11、播放页的信息显示

```
<script>
export default {
  name: "videoDetail",
  components: {},
  data() {
    return {
      myvideo: null, // 播放器对象
      isplay: false, //是否在播放
      nowTime: "00:00", //当前播放时间
      totalTime: "00:00", //总时长
      course:null, // 课程
      lessonid:0, // 当前播放视频的课节id
      lessonName:null, // 当前播放的视频名称
      isLogin:false, // false 未登录
      isBuy:false, // false 未购买
    };
  },
  computed: {},
  created() {
    // 判断登录(暂无)
    this.user = JSON.parse(localStorage.getItem("user"));
```

```
    if(this.user != null){
       this.isLogin = true; // 已登录
    }

    // 从上一级页面的请求中获得课程对象和小节编号
    this.course = this.$route.params.course;
    this.lessonid = this.$route.params.lessonid;
    this.isBuy = this.$route.params.isBuy;
  },
  mounted() {
    this.myvideo = document.getElementById("myvideo");
    this.initplay(); // 初始化播放的视频
  },
  methods: {
    play() {
      this.isplay = !this.isplay;
      if (this.isplay) {
        this.myvideo.play();
      } else {
        this.myvideo.pause();
      }
    },
    // 获取视频的时间是秒为单位，格式化城00：00的格式
    formatTime(time) {
      let mm = Math.floor((time % 3600) / 60);
      let ss = Math.floor(time % 60);
      mm = mm < 10 ? "0" + mm : mm;
      ss = ss < 10 ? "0" + ss : ss;
      return `${mm}:${ss}`;
    },
    //获取初始化信息
    getInit() {
      if (!this.myvideo) {
        //获取失败，显示0
        this.totalTime = this.formatTime(0);
      } else {
        //获取视频总时长
        this.totalTime = this.formatTime(this.myvideo.duration);
      }
    },
    //播放时显示当前播放时间和总时长
    handlerNowTime() {
      if (!this.myvideo) {
        this.nowTime = `${this.formatTime(0)}`;
      }
      this.nowTime = this.formatTime(this.myvideo.currentTime || 0);
    },
    //返回
    goBack() {
      //this.$router.push({ name: "Course" });
      this.$router.go(-1);
    },
    //播放课程
    playLesson(mp4) {
      //alert("播放视频! " + mp4);
      //this.myvideo.src = mp4;
    },
    // 初始化时播放的视频
```

```
    initplay(){
      //1.在课程信息中查找即将播放的小节视频的编号
      for( let i = 0 ; i< this.course.courseSections.length;i++ ){
          let section = this.course.courseSections[i];
          for(let j = 0; j<section.courseLessons.length ; j++){
              let lesson = section.courseLessons[j]; // 每节课
              if(lesson.courseMedia!=null){
                if(this.lessonid == lesson.courseMedia.lessonId){
                  console.log("视频地址：" + lesson.courseMedia.fileEdk);
                  this.lessonName = lesson.theme;
                  //2.将小节视频的地址 赋值 给播放器，进行播放
                  this.myvideo.src = lesson.courseMedia.fileEdk;
                  return;
                }
              }
          }
      }
    }
  },
};
</script>
```

```
<div class="content-container">

  <!-- 第一章 开始 -->
  <div v-for="section in course.courseSections.slice(0,1)">
    <div class="content-label">
      <div class="content-label-title single-line">{{section.sectionName}}</div>
      <img
        class="arrow-icon"

 src="
VEVHcEz/////////////////////////8dS1W+AAAAB3RSTlMAgwWyMeGgBsFrrQAAAFVJREFUKM9j
YBiugMUBQOi8EF2EsbxcAF1ReXkhuqJiczRl4uVGyqjKgIoUmFCVARUxMKAoAyliYEBRBlaEqiwcpAik
rBShKglCqyFUMcEYCgwjBAAAeaoQrHtg6QoAAAAASUVORK5CYII="
        alt=""
      />
    </div>

    <!-- 第一章 开始 -->
    <div class="content-sections">
      <div class='content-section'
          v-for="(lesson , index) in section.courseLessons" :key="index">
        <!-- 第一章，前两节 -->
        <div v-if="index<2">
          <div class="section-item clearfix">
            <span class="kw-icon-video section-type-icon"><i class="el-icon-
video-play"></i></span>
            <span class="section-dec">{{lesson.theme}}</span>
            <span class="section-status-icon pause-play">试看</span>
          </div>
          <div class="section-duration">
            <span v-if="lesson.courseMedia != null">时长：
{{lesson.courseMedia.duration}}</span>
            <span v-else>时长：无媒体文件</span>
          </div>
        </div>
```

```
        <div v-if="index>1">
          <div class="section-item clearfix">
            <span class="kw-icon-video section-type-icon fl"><i class="el-icon-
video-play"></i></span>
            <span class="section-dec need-update">{{lesson.theme}}</span>
            <!-- 未登录 => 锁 -->
            <span  v-if="!isLogin" class="section-status-icon pause-play">未解锁
</span>
            <!-- 已登录，未购买 => 锁 -->
            <span v-else-if="isLogin && !isBuy" class="section-status-icon
pause-play">未解锁</span>
            <!-- 已登录，已购买 => 播放 -->
            <span v-else class="section-status-icon pause-play">播放</span>
          </div>
          <div class="section-duration">
            <span v-if="lesson.courseMedia != null">时长：
{{lesson.courseMedia.duration}}</span>
            <span v-else>时长：无媒体文件</span>
          </div>
        </div>
      </div>
    </div>

  </div>
  <!-- 第一章 结束 -->
  <!-- 其余章 开始 -->
  <div v-for="section in
course.courseSections.slice(1,course.courseSections.length)">
    <div class="content-label">
      <div class="content-label-title single-line">{{section.sectionName}}</div>
      <img
        class="arrow-icon"

 src="
VEVHcEz/////////////////////////8dS1W+AAAAB3RSTlMAgwWyMeGgBsFrrQAAAFVJREFUKM9j
YBiugMUBQOi8EF2EsbxcAF1ReXkhuqJiczRl4uVGyqjKgIoUmFCVARUxMKAoAyliYEBRBlaEqiwcpAik
rBShKglCqyFUMcEYCgwjBAAAeaoQrHtg6QoAAAAASUVORK5CYII="
        alt=""
      />
    </div>
    <div class="content-sections">
      <div class="content-section" v-for="(lesson , index) in
section.courseLessons" :key="index">
        <div>
          <div class="section-item clearfix">
            <span class="kw-icon-video section-type-icon fl"><i class="el-icon-
video-play"></i></span>
            <span class="section-dec need-update">{{lesson.theme}}</span>
            <!-- 未登录 => 锁 -->
            <span  v-if="!isLogin" class="section-status-icon pause-play">未解锁
</span>
            <!-- 已登录，未购买 => 锁 -->
            <span v-else-if="isLogin && !isBuy" class="section-status-icon
pause-play">未解锁</span>
            <!-- 已登录，已购买 => 播放 -->
            <span v-else class="section-status-icon pause-play">播放</span>
          </div>
          <div class="section-duration">
```

```
            <span v-if="lesson.courseMedia != null">时长：
{{lesson.courseMedia.duration}}</span>
            <span v-else>时长：无媒体文件</span>
          </div>
        </div>
      </div>
    </div>

  </div>
  <!-- 其余章 结束 -->
</div>
```

12、当前播放的视频高亮突出

```
<div class="content-sections">
  <div :class="{
      'content-section': lesson.id != lessonid,
      'content-section content-section-choose': lesson.id == lessonid,
      }"
      v-for="(lesson , index) in section.courseLessons"
      :key="index"
    >
    <div>
      <div class="section-item clearfix">
        <span :class="{
          'kw-icon-video section-type-icon': lesson.id != lessonid,
          'kw-icon-video section-type-icon lv': lesson.id == lessonid,
          }"><i class="el-icon-video-play"></i></span>
        <span :class="{
          'section-dec': lesson.id != lessonid,
          'section-dec lv': lesson.id == lessonid,
          }">{{lesson.theme}}</span>
        <!-- 未登录 => 锁 -->
        <span  v-if="!isLogin" class="section-status-icon pause-play">未解锁
</span>
        <!-- 已登录，未购买 => 锁 -->
        <span v-else-if="isLogin && !isBuy" class="section-status-icon pause-
play">未解锁</span>
        <!-- 已登录，已购买 => 播放 -->
        <span v-else-if="lesson.id != lessonid" class="section-status-icon
pause-play">播放</span>
        <span v-else-if="lesson.id == lessonid" class="section-status-icon
pause-play"></span>
      </div>
      <div class="section-duration">
        <span v-if="lesson.courseMedia != null">时长：
{{lesson.courseMedia.duration}}</span>
        <span v-else>时长：无媒体文件</span>
      </div>
    </div>
  </div>
</div>
```

13、点击课标题切换视频并播放

```
<div :class="{
  'content-section': lesson.id != lessonid,
  'content-section content-section-choose': lesson.id == lessonid,
  }"
  v-for="(lesson , index) in section.courseLessons"
  :key="index"
  @click="playLesson(lesson)"
>
```

```
<script>
   //播放课程
    playLesson(lesson) {
      this.lessonid = lesson.id;
      this.myvideo.src = lesson.courseMedia.fileEdk;
      this.myvideo.play();
      this.isplay = !this.isplay;
    },
</script>
```

14、留言点赞

14.1 修改dao

```
@Data
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class CourseComment implements Serializable {
    private static final long serialVersionUID = 922554392538715061L;
    // 一条留言：N个点赞
    private List<CourseCommentFavoriteRecord> favoriteRecords ;
```

```
<resultMap type="com.lagou.entity.CourseComment" id="commentMap">
    <result property="id" column="cc_id" jdbcType="OTHER"/>
    <result property="courseId" column="course_id" jdbcType="INTEGER"/>
    <result property="sectionId" column="section_id" jdbcType="INTEGER"/>
    <result property="lessonId" column="lesson_id" jdbcType="INTEGER"/>
    <result property="userId" column="cc_user_id" jdbcType="INTEGER"/>
    <result property="userName" column="user_name" jdbcType="VARCHAR"/>
    <result property="parentId" column="parent_id" jdbcType="INTEGER"/>
    <result property="isTop" column="is_top" jdbcType="OTHER"/>
    <result property="comment" column="comment" jdbcType="VARCHAR"/>
    <result property="likeCount" column="like_count" jdbcType="INTEGER"/>
    <result property="isReply" column="is_reply" jdbcType="OTHER"/>
    <result property="type" column="type" jdbcType="INTEGER"/>
    <result property="status" column="status" jdbcType="INTEGER"/>
    <result property="createTime" column="cc_create_time" jdbcType="TIMESTAMP"/>
    <result property="updateTime" column="cc_update_time" jdbcType="TIMESTAMP"/>
```

```
    <result property="isDel" column="cc_is_del" jdbcType="OTHER"/>
    <result property="lastOperator" column="last_operator" jdbcType="INTEGER"/>
    <result property="isNotify" column="is_notify" jdbcType="OTHER"/>
    <result property="markBelong" column="mark_belong" jdbcType="OTHER"/>
    <result property="replied" column="replied" jdbcType="OTHER"/>
    <!-- N个点赞-->
    <collection property="favoriteRecords"
ofType="com.lagou.entity.CourseCommentFavoriteRecord">
        <result property="id" column="ccfr_id" jdbcType="OTHER"/>
        <result property="userId" column="ccfr_user_id" jdbcType="INTEGER"/>
        <result property="commentId" column="comment_id" jdbcType="INTEGER"/>
        <result property="isDel" column="ccfr_is_del" jdbcType="OTHER"/>
        <result property="createTime" column="ccfr_create_time"
jdbcType="TIMESTAMP"/>
        <result property="updateTime" column="ccfr_update_time"
jdbcType="TIMESTAMP"/>
    </collection>
</resultMap>

<!--分页查询某门课程的全部留言-->
<select id="getCommentsByCourseId" resultMap="commentMap">
    select
    cc.id cc_id,`course_id`,`section_id`,`lesson_id`,cc.user_id
cc_user_id,`user_name`,`parent_id`,`is_top`,`comment`,`like_count`,`is_reply`,`t
ype`,`status`,cc.create_time cc_create_time ,cc.update_time cc_update_time
,cc.is_del cc_is_del,`last_operator`,`is_notify`,`mark_belong`,`replied` ,
    ccfr.id ccfr_id,ccfr.user_id ccfr_user_id,comment_id,ccfr.is_del
ccfr_is_del,ccfr.create_time ccfr_create_time,ccfr.update_time ccfr_update_time
    from course_comment cc left join course_comment_favorite_record ccfr
    on cc.id = ccfr.comment_id
    where cc.is_del = 0
    and course_id = #{courseid}
    order by is_top desc , like_count desc , cc.create_time desc
    limit #{offset}, #{pageSize}
</select>
```

```
@Test
public void getComment(){
    List<CourseComment> list = courseCommentDao.getCommentsByCourseId(7, 1, 20);
    for(CourseComment comment : list){
        System.out.println("【"+comment.getUserName()+"】"+"留言：" +
comment.getComment());
        for(CourseCommentFavoriteRecord fr: comment.getFavoriteRecords()){
            System.out.println("--->" + fr.getUserId());
        }
    }
}
```

14.2 修改controller

```
@GetMapping("comment/getCourseCommentList/{courseid}/{pageIndex}/{pageSize}")
public List<CourseComment> getCommentsByCourseId(@PathVariable("courseid")
Integer courseid,@PathVariable("pageIndex") Integer
pageIndex,@PathVariable("pageSize") Integer pageSize){
    int offset = (pageIndex-1)*pageSize;
    List<CourseComment> list = commentService.getCommentsByCourseId(courseid,
offset, pageSize);
    System.out.println("获取第"+courseid+"门课程的留言：共计"+list.size()+"条");
    return list;
}
```

14.3 页面点赞之后的显示样式

```
<!-- 已赞 -->
<div v-if="JSON.stringify(comment.favoriteRecords).indexOf( user.content.id ) >=
0" class="message-list-title-right">
    <img class="message-list-title-right-icon" src="已赞图片过长，省略" alt="">
    <div class="message-list-title-right-praise">{{comment.likeCount}}</div>
</div>
<!-- 没点过赞 -->
<div v-else class="message-list-title-right">
    <img class="message-list-title-right-icon" src="未赞图片过长，省略" alt="">
    <div class="message-list-title-right-praise">{{comment.likeCount}}</div>
</div>
```

14.4 页面调用点赞和取消赞

```
<!-- 已赞 -->
<div @click="cancelzan(comment)" v-
if="JSON.stringify(comment.favoriteRecords).indexOf( user.content.id ) >= 0"
class="message-list-title-right">
    <img class="message-list-title-right-icon" src="已赞图片过长，省略" alt="">
    <div class="message-list-title-right-praise">{{comment.likeCount}}</div>
</div>
<!-- 没点过赞 -->
<div @click="zan(comment)" v-else class="message-list-title-right">
    <img class="message-list-title-right-icon" src="未赞图片过长，省略" alt="">
    <div class="message-list-title-right-praise">{{comment.likeCount}}</div>
</div>
```

```
<script>
// 点赞
    zan(comment){
      return this.axios

.get("http://localhost:8002/course/comment/saveFavorite/"+comment.id+"/"+this.us
er.content.id)
      .then((result) => {
        // console.log(result);
        // 重新获取本门课的全部留言信息
        this.getComment();
      }).catch( (error)=>{
```

```
        this.$message.error("点赞失败！");
      } );
    },
    // 取消赞
    cancelzan(comment){
      return this.axios

.get("http://localhost:8002/course/comment/cancelFavorite/"+comment.id+"/"+this.
user.content.id)
      .then((result) => {
        // console.log(result);
        // 重新获取本门课的全部留言信息
        this.getComment();
      }).catch( (error)=>{
        this.$message.error("取消赞失败！");
      } );
    },
</script>
```

```
<!--分页查询某门课程的全部留言-->
<select id="getCommentsByCourseId" resultMap="commentMap">
    select
    cc.id cc_id,`course_id`,`section_id`,`lesson_id`,cc.user_id
cc_user_id,`user_name`,`parent_id`,`is_top`,`comment`,`like_count`,`is_reply`,`t
ype`,`status`,cc.create_time cc_create_time ,cc.update_time cc_update_time
,cc.is_del cc_is_del,`last_operator`,`is_notify`,`mark_belong`,`replied` ,
    ccfr.id ccfr_id,ccfr.user_id ccfr_user_id,comment_id,ccfr.is_del
ccfr_is_del,ccfr.create_time ccfr_create_time,ccfr.update_time ccfr_update_time
    from course_comment cc left join course_comment_favorite_record ccfr
    on cc.id = ccfr.comment_id
    where cc.is_del = 0
    and ccfr.is_del = 0  <!-- 点赞表也要过滤掉被删除的数据 -->
    and course_id = #{courseid}
    order by is_top desc , like_count desc , cc.create_time desc
    limit #{offset}, #{pageSize}
</select>
```

15、发表留言

- Course.vue

```
// 发表留言
saveComment(){
   return this.axios
    .get("http://localhost:80/course/comment/saveCourseComment",{
      params:{
        courseid:this.course.id,
        userid:this.user.content.id,
        username:this.user.content.name,
        comment:this.comment,
      }
    })
    .then((result) => {
```

```
      // console.log(result);
      // 重新获取本门课的全部留言信息
      this.getComment();
    }).catch( (error)=>{
      this.$message.error("发表留言失败！");
    } );
},
```

- web消费方

```
public interface CommentService {
    /**
     * 保存留言
     * @param comment 留言内容对象
     * @return 受影响的行数
     */
    Integer saveComment(CourseComment comment);
```

```
@RestController
@RequestMapping("course")
public class CommentController {

    @Reference // 远程消费
    private CommentService commentService;


    @GetMapping("comment/saveCourseComment")
    public Object saveCourseComment(Integer courseid,Integer userid,String
username,String comment) throws UnsupportedEncodingException {
        username = new String( username.getBytes("ISO-8859-1"),"UTF-8" );
        comment = new String( comment.getBytes("ISO-8859-1"),"UTF-8" );
        System.out.println("昵称：" + username);
        CourseComment courseComment = new CourseComment();
        courseComment.setCourseId(courseid); // 课程编号
        courseComment.setSectionId(0); // 章节编号 （预留字段，为项目的2.0版本保留）
        courseComment.setLessonId(0);// 小节编号（预留字段，为项目的2.0版本保留）
        courseComment.setUserId(userid); // 用户编号
        courseComment.setUserName(username); // 用户昵称
        courseComment.setParentId(0); //没有父id（预留字段，为项目的2.0版本保留）
        courseComment.setComment(comment);// 留言内容
        courseComment.setType(0); // 0用户留言（预留字段，为项目的2.0版本保留）
        courseComment.setLastOperator(userid); //最后操作的用户编号
        Integer i = commentService.saveComment(courseComment);
        return i;
    }
```

- service提供方

```
<!--保存留言-->
<insert id="saveComment">
   insert  into
`course_comment`(`course_id`,`section_id`,`lesson_id`,`user_id`,`user_name`,`par
ent_id`,`is_top`,`comment`,`like_count`,`is_reply`,`type`,`status`,`create_time`
,`update_time`,`is_del`,`last_operator`,`is_notify`,`mark_belong`,`replied`)
   values
   (#{courseId},#{sectionId},#{lessonId},#{userId},#{userName},#{parentId},0,#
{comment},0,0,#{type},0,sysdate(),sysdate(),0,#{lastOperator},1,0,0)
</insert>
```