1. 后台主页

1.1 获取用户的基本信息

1. 导入需要的脚本：

```
<!-- 导入 jQuery -->
<script src="/assets/lib/jquery.js"></script>
<!-- 导入自己封装的 baseAPI -->
<script src="/assets/js/baseAPI.js"></script>
<!-- 导入自己的 js 文件 -->
<script src="/assets/js/index.js"></script>
```

2. 定义 getUserInfo 函数：

```
// 获取用户的基本信息
function getUserInfo() {
  $.ajax({
    method: 'GET',
    url: '/my/userinfo',
    // headers 就是请求头配置对象
    headers: {
      Authorization: localStorage.getItem('token') || ''
    },
    success: function(res) {
      if (res.status !== 0) {
        return layui.layer.msg('获取用户信息失败！')
      }
      // 调用 renderAvatar 渲染用户的头像
      renderAvatar(res.data)
    }
  })
}
```

1.2 渲染用户头像

1. 定义 renderAvatar 函数：

```
// 渲染用户的头像
function renderAvatar(user) {
  // 1. 获取用户的名称
  var name = user.nickname || user.username
  // 2. 设置欢迎的文本
  $('#welcome').html('欢迎  ' + name)
  // 3. 按需渲染用户的头像
  if (user.user_pic !== null) {
    // 3.1 渲染图片头像
    $('.layui-nav-img')
      .attr('src', user.user_pic)
      .show()
    $('.text-avatar').hide()
  } else {
    // 3.2 渲染文本头像
```

```
        $('.layui-nav-img').hide()
        var first = name[0].toUpperCase()
        $('.text-avatar')
          .html(first)
          .show()
      }
    }
```

1.3 统一为有权限的接口设置headers请求头

1. 在 baseAPI的 `ajaxPrefilter` 中添加如下代码：

```
    // 统一为有权限的接口，设置 headers 请求头
    if (options.url.indexOf('/my/') !== -1) {
      options.headers = {
        Authorization: localStorage.getItem('token') || ''
      }
    }
```

1.4 实现退出功能

1. 修改退出的 `<a>` 链接如下：

```
<a href="javascript:;" id="btnLogout"><span class="iconfont icon-tuichu">
</span>退出</a>
```

2. 实现退出功能：

```
    var layer = layui.layer

    // 点击按钮，实现退出功能
    $('#btnLogout').on('click', function() {
      // 提示用户是否确认退出
      layer.confirm('确定退出登录?', { icon: 3, title: '提示' },
function(index) {
        //do something
        // 1. 清空本地存储中的 token
        localStorage.removeItem('token')
        // 2. 重新跳转到登录页面
        location.href = '/login.html'

        // 关闭 confirm 询问框
        layer.close(index)
      })
    })
```

1.5 控制用户的访问权限

1. 在调用有权限接口的时候，指定 `complete` 回调函数：

```
      // 不论成功还是失败，最终都会调用 complete 回调函数
      complete: function(res) {
        // console.log('执行了 complete 回调：')
        // console.log(res)
        // 在 complete 回调函数中，可以使用 res.responseJSON 拿到服务器响应回来的数据
        if (res.responseJSON.status === 1 && res.responseJSON.message === '身份认证失败！') {
          // 1. 强制清空 token
          localStorage.removeItem('token')
          // 2. 强制跳转到登录页面
          location.href = '/login.html'
        }
      }
```

1.6 优化权限控制的代码

1. 将权限控制的代码，从每个请求中，抽离到 `ajaxPrefilter` 中：

```
  // 注意：每次调用 $.get() 或 $.post() 或 $.ajax() 的时候，
  // 会先调用 ajaxPrefilter 这个函数
  // 在这个函数中，可以拿到我们给Ajax提供的配置对象
  $.ajaxPrefilter(function(options) {
    // 在发起真正的 Ajax 请求之前，统一拼接请求的根路径
    options.url = 'http://ajax.frontend.itheima.net' + options.url

    // 统一为有权限的接口，设置 headers 请求头
    if (options.url.indexOf('/my/') !== -1) {
      options.headers = {
        Authorization: localStorage.getItem('token') || ''
      }
    }

    // 全局统一挂载 complete 回调函数
    options.complete = function(res) {
      // console.log('执行了 complete 回调：')
      // console.log(res)
      // 在 complete 回调函数中，可以使用 res.responseJSON 拿到服务器响应回来的数据
      if (res.responseJSON.status === 1 && res.responseJSON.message === '身份认证失败！') {
        // 1. 强制清空 token
        localStorage.removeItem('token')
        // 2. 强制跳转到登录页面
        location.href = '/login.html'
      }
    }
  })

```

2. 基本资料

2.1 创建基本资料对应的页面

1. 新建 `/user/user_info.html` 并初始化如下：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <!-- 导入 layui 的样式 -->
    <link rel="stylesheet" href="/assets/lib/layui/css/layui.css" />
    <!-- 导入自己的样式 -->
    <link rel="stylesheet" href="/assets/css/user/user_info.css" />
  </head>
  <body>
    <!-- 卡片区域 -->
    <div class="layui-card">
      <div class="layui-card-header">修改用户信息</div>
      <div class="layui-card-body">
        卡片式面板面板通常用于非白色背景色的主体内<br />
        从而映衬出边框投影
      </div>
    </div>
  </body>
</html>
```

2. 新建 `/assets/css/user/user_info.css` 并初始化如下：

```
html,
body {
  margin: 0;
  padding: 0;
}

body {
  background-color: #f2f3f5;
  padding: 15px;
}
```

3. 修改 `index.html` 中对应的 `<a>` 链接：

```
<a href="/user/user_info.html" target="fm"><i class="layui-icon layui-icon-app"></i>基本资料</a>
```

2.2 绘制基本资料对应的表单

1. 编写如下的表单结构：

```
<!-- form 表单区域 -->
        <form class="layui-form" action="">
          <div class="layui-form-item">
            <label class="layui-form-label">登录名称</label>
            <div class="layui-input-block">
              <input type="text" name="username" required lay-verify="required" placeholder="请输入登录名称" autocomplete="off" class="layui-input" readonly />
            </div>
          </div>
```

```
            <div class="layui-form-item">
              <label class="layui-form-label">用户昵称</label>
              <div class="layui-input-block">
                <input type="text" name="nickname" required lay-verify="required|nickname" placeholder="请输入用户昵称" autocomplete="off" class="layui-input" />
              </div>
            </div>
            <div class="layui-form-item">
              <label class="layui-form-label">用户邮箱</label>
              <div class="layui-input-block">
                <input type="text" name="email" required lay-verify="required|email" placeholder="请输入用户邮箱" autocomplete="off" class="layui-input" />
              </div>
            </div>
            <div class="layui-form-item">
              <div class="layui-input-block">
                <button class="layui-btn" lay-submit lay-filter="formDemo">提交修改</button>
                <button type="reset" class="layui-btn layui-btn-primary">重置</button>
              </div>
            </div>
          </form>
```

2. 在页面底部导入如下的脚本：

```
    <!-- 导入 layui 的 js -->
    <script src="/assets/lib/layui/layui.all.js"></script>
    <!-- 导入 jquery -->
    <script src="/assets/lib/jquery.js"></script>
    <!-- 导入自己的 js -->
    <script src="/assets/js/user/user_info.js"></script>
```

3. 在 `user_info.js` 中编写如下的代码：

```
$(function() {
  var form = layui.form

  form.verify({
    nickname: function(value) {
      if (value.length > 6) {
        return '昵称长度必须在 1 ~ 6 个字符之间！'
      }
    }
  })
})
```

2.3 获取用户的基本信息

1. 导入 `baseAPI`：

```
<script src="/assets/js/baseAPI.js"></script>
```

2. 在 `user_info.js` 中定义并调用 `initUserInfo` 函数：

```
    initUserInfo()
```

```
2
3       // 初始化用户的基本信息
4       function initUserInfo() {
5         $.ajax({
6           method: 'GET',
7           url: '/my/userinfo',
8           success: function(res) {
9             if (res.status !== 0) {
10              return layer.msg('获取用户信息失败！')
11            }
12            console.log(res)
13          }
14        })
15      }
```

2.4 使用form.val方法快速为表单赋值

1. 为表单指定 `lay-filter` 属性：

```
1     <form class="layui-form" lay-filter="formUserInfo"></form>
```

2. 调用 `form.val()` 方法为表单赋值：

```
1     form.val('formUserInfo', res.data)
```

3. 使用隐藏域保存用户的 `id` 值：

```
1     <!-- form 表单区域 -->
2     <form class="layui-form" lay-filter="formUserInfo">
3       <!-- 这是隐藏域 -->
4       <input type="hidden" name="id" value="" />
5
6       <!-- 省略其他代码 -->
7     </form>
```

2.5 实现表单的重置效果

1. 阻止表单的默认重置行为，再重新获取用户信息即可：

```
1       // 重置表单的数据
2       $('#btnReset').on('click', function(e) {
3         // 阻止表单的默认重置行为
4         e.preventDefault()
5         initUserInfo()
6       })
```

2.6 发起请求更新用户的信息

1. 阻止表单的默认提交行为，并发起数据请求：

```
1       // 监听表单的提交事件
2       $('.layui-form').on('submit', function(e) {
3         // 阻止表单的默认提交行为
4         e.preventDefault()
5         // 发起 ajax 数据请求
6         $.ajax({
7           method: 'POST',
```

```
        url: '/my/userinfo',
        data: $(this).serialize(),
        success: function(res) {
          if (res.status !== 0) {
            return layer.msg('更新用户信息失败！')
          }
          layer.msg('更新用户信息成功！')
          // 调用父页面中的方法，重新渲染用户的头像和用户的信息
          window.parent.getUserInfo()
        }
      })
    })
```

2. 注意：`<iframe>` 中的子页面，如果想要调用父页面中的方法，使用 `window.parent` 即可。

3. 重置密码

3.1 渲染重置密码的页面结构

1. 在 `/user/user_pwd.html` 页面中编写如下的结构：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="/assets/lib/layui/css/layui.css" />
    <link rel="stylesheet" href="/assets/css/user/user_pwd.css" />
  </head>
  <body>
    <!-- 卡片区域 -->
    <div class="layui-card">
      <div class="layui-card-header">修改密码</div>
      <div class="layui-card-body">
        <form class="layui-form">
          <div class="layui-form-item">
            <label class="layui-form-label">原密码</label>
            <div class="layui-input-block">
              <input type="password" name="oldPwd" required lay-verify="required" placeholder="请输入原密码" autocomplete="off" class="layui-input" />
            </div>
          </div>
          <div class="layui-form-item">
            <label class="layui-form-label">新密码</label>
            <div class="layui-input-block">
              <input type="password" name="newPwd" required lay-verify="required" placeholder="请输入新密码" autocomplete="off" class="layui-input" />
            </div>
          </div>
          <div class="layui-form-item">
            <label class="layui-form-label">确认新密码</label>
```

```
              <div class="layui-input-block">
                <input type="password" name="rePwd" required lay-verify="required" placeholder="请再次确认密码" autocomplete="off" class="layui-input" />
              </div>
            </div>
            <div class="layui-form-item">
              <div class="layui-input-block">
                <button class="layui-btn" lay-submit lay-filter="formDemo">修改密码</button>
                <button type="reset" class="layui-btn layui-btn-primary">重置</button>
              </div>
            </div>
          </form>
        </div>
      </div>
    </body>
  </html>
```

2. 在 `/assets/css/user/user_pwd.css` 中编写如下的样式：

```
  html,
  body {
    margin: 0;
    padding: 0;
  }

  body {
    padding: 15px;
    background-color: #f2f3f5;
  }

  .layui-form {
    width: 500px;
  }
```

3.2 为密码框定义校验规则

1. 定义如下的三个校验规则：

```
  $(function() {
    var form = layui.form

    form.verify({
      pwd: [/^[\S]{6,12}$/, '密码必须6到12位，且不能出现空格'],
      samePwd: function(value) {
        if (value === $('[name=oldPwd]').val()) {
          return '新旧密码不能相同！'
        }
      },
      rePwd: function(value) {
        if (value !== $('[name=newPwd]').val()) {
          return '两次密码不一致！'
        }
      }
    })
```

```
})
```

2. 在 body 结束标签之前导入如下的 `script` 标签：

```
<!-- 导入 layui 的 js -->
<script src="/assets/lib/layui/layui.all.js"></script>
<!-- 导入 jQuery -->
<script src="/assets/lib/jquery.js"></script>
<!-- 导入 baseAPI -->
<script src="/assets/js/baseAPI.js"></script>
<!-- 导入自己的 js -->
<script src="/assets/js/user/user_pwd.js"></script>
```

3. 为密码框分别添加对应的校验规则：

```
 <form class="layui-form">
   <div class="layui-form-item">
     <label class="layui-form-label">原密码</label>
     <div class="layui-input-block">
       <input type="password" name="oldPwd" required lay-verify="required|pwd" placeholder="请输入原密码" autocomplete="off" class="layui-input" />
     </div>
   </div>
   <div class="layui-form-item">
     <label class="layui-form-label">新密码</label>
     <div class="layui-input-block">
       <input type="password" name="newPwd" required lay-verify="required|pwd|samePwd" placeholder="请输入新密码" autocomplete="off" class="layui-input" />
     </div>
   </div>
   <div class="layui-form-item">
     <label class="layui-form-label">确认新密码</label>
     <div class="layui-input-block">
       <input type="password" name="rePwd" required lay-verify="required|pwd|rePwd" placeholder="请再次确认密码" autocomplete="off" class="layui-input" />
     </div>
   </div>
   <div class="layui-form-item">
     <div class="layui-input-block">
       <button class="layui-btn" lay-submit lay-filter="formDemo">修改密码</button>
       <button type="reset" class="layui-btn layui-btn-primary">重置</button>
     </div>
   </div>
 </form>
```

3.3 发起请求实现重置密码的功能

```
$('.layui-form').on('submit', function(e) {
  e.preventDefault()
  $.ajax({
    method: 'POST',
    url: '/my/updatepwd',
```

```
        data: $(this).serialize(),
        success: function(res) {
          if (res.status !== 0) {
            return layui.layer.msg('更新密码失败！')
          }
          layui.layer.msg('更新密码成功！')
          // 重置表单
          $('.layui-form')[0].reset()
        }
      })
    })
```

4. 更换头像

4.1 初步渲染更换头像页面的结构

1. 创建 `/user/user_avatar.html` 页面：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="/assets/lib/layui/css/layui.css" />
    <link rel="stylesheet" href="/assets/css/user/user_avatar.css" />
  </head>
  <body>
    <!-- 卡片区域 -->
    <div class="layui-card">
      <div class="layui-card-header">更换头像</div>
      <div class="layui-card-body">
        卡片式面板面板通常用于非白色背景色的主体内<br />
        从而映衬出边框投影
      </div>
    </div>
  </body>
</html>

```

2. 美化基本样式：

```
  html,
  body {
    margin: 0;
    padding: 0;
  }

  body {
    padding: 15px;
    background-color: #f2f3f5;
  }

```

3. 修改 `index.html` 中对应链接的属性：

```
  <a href="/user/user_avatar.html" target="fm"><i class="layui-icon layui-icon-app"></i>更换头像</a>
```

4.2 实现裁剪区域图片的替换

```
// 为文件选择框绑定 change 事件
$('#file').on('change', function(e) {
  // 获取用户选择的文件
  var filelist = e.target.files
  if (filelist.length === 0) {
    return layer.msg('请选择照片！')
  }

  // 1. 拿到用户选择的文件
  var file = e.target.files[0]
  // 2. 将文件，转化为路径
  var imgURL = URL.createObjectURL(file)
  // 3. 重新初始化裁剪区域
  $image
    .cropper('destroy') // 销毁旧的裁剪区域
    .attr('src', imgURL) // 重新设置图片路径
    .cropper(options) // 重新初始化裁剪区域
})
```

4.3 将裁剪后的头像上传到服务器

```
// 为确定按钮，绑定点击事件
$('#btnUpload').on('click', function() {
  // 1. 要拿到用户裁剪之后的头像
  var dataURL = $image
    .cropper('getCroppedCanvas', {
      // 创建一个 Canvas 画布
      width: 100,
      height: 100
    })
    .toDataURL('image/png') // 将 Canvas 画布上的内容，转化为 base64 格式的字符串
  // 2. 调用接口，把头像上传到服务器
  $.ajax({
    method: 'POST',
    url: '/my/update/avatar',
    data: {
      avatar: dataURL
```

```
        },
        success: function(res) {
          if (res.status !== 0) {
            return layer.msg('更换头像失败！')
          }
          layer.msg('更换头像成功！')
          window.parent.getUserInfo()
        }
      })
    })
```

4.4 设置头部区域的快捷方式

1. 打开 `index.html`，修改头部 `个人中心` 下的三个快捷方式如下：

```
<dl class="layui-nav-child">
  <dd><a href="/user/user_info.html" target="fm">基本资料</a></dd>
  <dd><a href="/user/user_avatar.html" target="fm">更换头像</a></dd>
  <dd><a href="/user/user_pwd.html" target="fm">重置密码</a></dd>
</dl>
```

5. 文章分类

5.1 创建并显示文章分类页面

1. 创建 `/article/art_cate.html` 页面，并初始化如下的UI结构：

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="/assets/lib/layui/css/layui.css" />
    <link rel="stylesheet" href="/assets/css/article/art_cate.css" />
  </head>
  <body>
    <!-- 卡片区域 -->
    <div class="layui-card">
      <div class="layui-card-header">
        <span>文章类别管理</span>
        <button type="button" class="layui-btn layui-btn-normal layui-btn-sm">添加类别</button>
      </div>
      <div class="layui-card-body">
        <table class="layui-table">
          <colgroup>
            <col />
            <col />
            <col width="200" />
          </colgroup>
          <thead>
```

```
              <tr>
                <th>分类名称</th>
                <th>分类别名</th>
                <th>操作</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td>贤心</td>
                <td>2016-11-29</td>
                <td>人生就像是一场修行</td>
              </tr>
              <tr>
                <td>许闲心</td>
                <td>2016-11-28</td>
                <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…
</td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </body>
  </html>
```

2. 定义 `/assets/css/article/art_cate.css` 美化样式：

```
  html,
  body {
    margin: 0;
    padding: 0;
  }

  body {
    padding: 15px;
    background-color: #f2f3f5;
  }

  .layui-card-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
  }

```

3. 修改 `index.html` 中对应的 `<a>` 链接：

```
   <a href="/article/art_cate.html" target="fm"><i class="layui-icon layui-
icon-app"></i>文章类别</a>
```

5.2 获取并使用模板引擎渲染表格的数据

1. 在页面底部导入模板引擎：

```
   <script src="/assets/lib/template-web.js"></script>
```

2. 定义模板：

```
<!-- 表格数据的模板 -->
<script type="text/html" id="tpl-table">
  {{each data}}
  <tr>
    <td>{{$value.name}}</td>
    <td>{{$value.alias}}</td>
    <td>
      <button type="button" class="layui-btn layui-btn-xs">编辑</button>
      <button type="button" class="layui-btn layui-btn-danger layui-btn-xs">删除</button>
    </td>
  </tr>
  {{/each}}
</script>
```

3. 发起请求获取数据：

```
initArtCateList()

// 获取文章分类的列表
function initArtCateList() {
  $.ajax({
    method: 'GET',
    url: '/my/article/cates',
    success: function(res) {
      var htmlStr = template('tpl-table', res)
      $('tbody').html(htmlStr)
    }
  })
}
```

5.3 使用layer.open实现弹出层效果

1. 导入 `layer`：

```
var layer = layui.layer
```

2. 为按钮添加 `id` 属性：

```
<button type="button" class="layui-btn layui-btn-normal layui-btn-sm" id="btnAddCate">添加类别</button>
```

3. 在按钮的点击事件中，通过 `layer.open()` 展示弹出层：

```
// 为添加类别按钮绑定点击事件
$('#btnAddCate').on('click', function() {
  layer.open({
    type: 1,
    area: ['500px', '250px'],
    title: '添加文章分类',
    content: 'ABC'
  })
})
```

5.4 在弹出层中渲染form表单结构

1. 在页面中定义如下的 `script` 标签：

```
<script type="text/html" id="dialog-add">
  <form class="layui-form" id="form-add">
    <div class="layui-form-item">
      <label class="layui-form-label">分类名称</label>
      <div class="layui-input-block">
        <input type="text" name="name" required  lay-verify="required" placeholder="请输入分类名称" autocomplete="off" class="layui-input">
      </div>
    </div>
    <div class="layui-form-item">
      <label class="layui-form-label">分类别名</label>
      <div class="layui-input-block">
        <input type="text" name="alias" required  lay-verify="required" placeholder="请输入分类别名" autocomplete="off" class="layui-input">
      </div>
    </div>
    <div class="layui-form-item">
      <div class="layui-input-block">
        <button class="layui-btn" lay-submit lay-filter="formDemo">确认添加</button>
        <button type="reset" class="layui-btn layui-btn-primary">重置</button>
      </div>
    </div>
  </form>
</script>
```

2. 通过 `content` 属性指定内容：

```
layer.open({
  type: 1,
  area: ['500px', '250px'],
  title: '添加文章分类',
  content: $('#dialog-add').html()
})
```

5.5 实现添加文章分类的功能

1. 发起Ajax请求：

```
// 通过代理的形式，为 form-add 表单绑定 submit 事件
$('body').on('submit', '#form-add', function(e) {
  e.preventDefault()
  $.ajax({
    method: 'POST',
    url: '/my/article/addcates',
    data: $(this).serialize(),
    success: function(res) {
      if (res.status !== 0) {
        return layer.msg('新增分类失败！')
      }
      initArtCateList()
      layer.msg('新增分类成功！')
```

```
            // 根据索引，关闭对应的弹出层
            layer.close(indexAdd)
          }
        })
      })
```

2. 预先保存弹出层的索引：

```
      // 为添加类别按钮绑定点击事件
      var indexAdd = null
      $('#btnAddCate').on('click', function() {
        indexAdd = layer.open({
          type: 1,
          area: ['500px', '250px'],
          title: '添加文章分类',
          content: $('#dialog-add').html()
        })
      })
```