

## 勘误：

1.方法重载：TypeScript中可以实现方法重载，但需要先声明在实现

```
function f(a: string):string;
function f(a: number):number;
function f(a: any):any {
    if(typeof a === 'string'){...}
    else {...}
}
```

2.函数可选参数不能有默认值

```
// 这里“='Harden'”是不允许的，一个参数不能既可选又有默认值，不合逻辑
function buildName(first: string = 'James', last?: string) {
    return first + last;
}
```

3.组件不能同名：

```
// 这两个同名
interface Result1<T, U> {
    success: boolean;
    data?: T;
    message?: U;
}

class Result2<T> {
    constructor(public success: boolean, public data: T) {
    }
}
```

## 用户登录、注册

1.添加登录表单

1) 表单校验：模板驱动表单校验

第一步：导入FormsModule

第二步：创建数据模型LoginUser

第三步：表单中绑定该模型[(ngModel)]="model.phone"

关于输入输出属性，输出属性(click)="onClick(\$event)"，输入属性[title]="myName"

双向绑定，即输入又输出[(ngModel)]="model.phone"

模板引用变量：在当前模板上下文定义一个变量，，若不赋值，表示dom元素；

若赋值为ngModel，则可以访问当前元素校验状态；若赋值为ngForm，则可以访问整个表单校验状态

数据校验基于H5表单校验，常用的有：required,min,max,minlength,maxlength,pattern,  
type=email,type=number,type=url

校验状态：valid,invalid,touched,untouched,dirty,pristine

errors里面的字段名就是上面的h5校验名字，如：phone.errors.required

条件语句：\*ngIf="expr"

## 2.发送登录请求

### 1) 配置代理，防止跨域

proxy.conf.json:

```
{
  "/api": {
    "target": "http://localhost:8080",
    "secure": false
  }
}
```

package.json:

```
"start": "ng serve --proxy-config proxy.conf.json",
```

### 2) 导入HttpClient模块才能使用http请求

```
@NgModule({
  imports: [
    FormsModule, // 使用模板驱动表单
  ],
})
export class AppModule {
}
```

### 3) 请求后台接口一般写入服务中

创建：ng g s user

使用：UserService中使用HttpClient服务

```
export class UserService {
  url = '/api/users/';

  // 依赖注入：构造函数注入服务
  constructor(private http: HttpClient) {
  }

  login(user: LoginUser) {
    // 返回的结果是Observable对象
    return this.http.post<Result<User>>(this.url + 'login', user);
  }
}
```

使用：LoginComponent中使用UserService

```
constructor(private userService: UserService) { // 注入服务
  this.model = new LoginUser();
}

login() {
  // 使用服务
  // 发送请求
  this.userService.login(this.model).subscribe()
}
```

4) 组件中注入服务并调用之

3.后台创建表和接口

4.session处理

1) 安装: npm i -S express-session

2) 配置 app.use(session({...}))

```
const session = require('express-session');
app.use(session({
  secret: 'its a secret',
  resave: false,
  saveUninitialized: false
})))
```

3) session使用

- 赋值 req.session.xx = xx;
- 删除 delete req.session.xx

\*\*\*快捷键

导入: alt+enter