**一、安装**

**配置文件**：congfig文件夹中的application.yml文件

server:

port: 9999

servlet:

encoding:

charset: UTF-8

force: true

enabled: true

spring:

datasource:

jdbc-url: jdbc:mysql://127.0.0.1:3306/zebone?useSSL=false&useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull&transformedBitIsBoolean=true&serverTimezone=GMT%2B8&nullCatalogMeansCurrent=true&allowPublicKeyRetrieval=true

driver-class-name: com.mysql.cj.jdbc.Driver

username: root

password: 123456

jpa:

hibernate:

ddl-auto: update

show-sql: true

second-datasource:

# orcl

# jdbc-url: jdbc:oracle:thin:@127.0.0.1:1521/orcl

# driver-class-name: oracle.jdbc.driver.OracleDriver

# username: nhis

# password: nhis

# sqlserver

jdbc-url: jdbc:sqlserver://192.168.1.36:1433;DatabaseName=nhis\_test

driver-class-name: com.microsoft.sqlserver.jdbc.SQLServerDriver

username: sa

password: 123456

# mysql

# jdbc-url: jdbc:mysql://127.0.0.1:3306/zb?useSSL=false&useUnicode=true&characterEncoding=utf-8&zeroDateTimeBehavior=convertToNull&transformedBitIsBoolean=true&serverTimezone=GMT%2B8&nullCatalogMeansCurrent=true&allowPublicKeyRetrieval=true

# driver-class-name: com.mysql.cj.jdbc.Driver

# username: root

# password: 123456

HASOR\_DATAQL\_DATAWAY: true

HASOR\_DATAQL\_DATAWAY\_ADMIN: true

HASOR\_DATAQL\_DATAWAY\_API\_URL: /api/

HASOR\_DATAQL\_DATAWAY\_UI\_URL: /interface-ui/

**1、设置服务器端口**

server:

port: 9999

**prot**:端口号

**2、设置数据库连接**

**datasource**：主数据源

**second-datasource**：第二数据源

**jdbc-url**：数据库连接地址

**username**：数据库用户名

**password**：数据库密码

**3、安装nginx**

配置文件：nginx目录下的conf/nginx.conf

#user nobody;

worker\_processes 1;

#error\_log logs/error.log;

#error\_log logs/error.log notice;

#error\_log logs/error.log info;

#pid logs/nginx.pid;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#gzip on;

server {

listen 80;

server\_name 192.168.1.9;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html/dist;

index index.html index.htm;

}

location /api/ {

proxy\_pass http://127.0.0.1:9999;

}

location /interface-ui/ {

proxy\_pass http://127.0.0.1:9999;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

# another virtual host using mix of IP-, name-, and port-based configuration

#

#server {

# listen 8000;

# listen somename:8080;

# server\_name somename alias another.alias;

# location / {

# root html;

# index index.html index.htm;

# }

#}

# HTTPS server

#

#server {

# listen 443 ssl;

# server\_name localhost;

# ssl\_certificate cert.pem;

# ssl\_certificate\_key cert.key;

# ssl\_session\_cache shared:SSL:1m;

# ssl\_session\_timeout 5m;

# ssl\_ciphers HIGH:!aNULL:!MD5;

# ssl\_prefer\_server\_ciphers on;

# location / {

# root html;

# index index.html index.htm;

# }

#}

}

修改http下server的listen节点，设置为可用端口号，此端口号为客户端访问端口号

修改server\_name为当前服务器ip

location /api/ {

proxy\_pass http://127.0.0.1:9999;

}

location /interface-ui/ {

proxy\_pass http://127.0.0.1:9999;

}

修改proxy\_pass 转发地址，为服务端访问ip:port

**4、启动客户端，nginx目录下的nginx.exe**

**5、启动服务端，strat.bat**

**二、初始化数据**

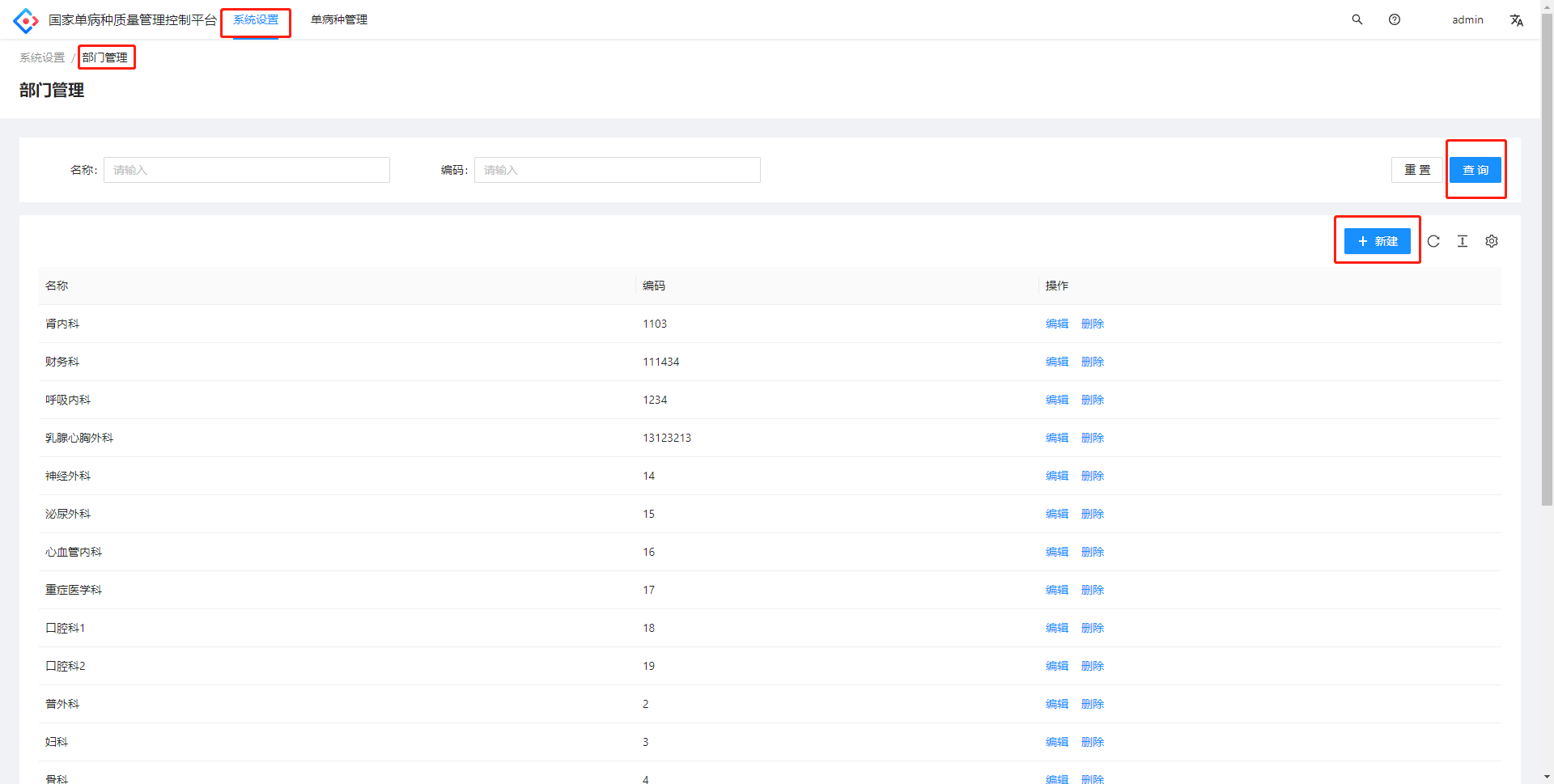
INSERT INTO js\_sys\_user (user\_code, login\_code, user\_name, password, email, mobile, phone, sex, avatar, sign, wx\_openid, mobile\_imei, user\_type, ref\_code, ref\_name, mgr\_type, pwd\_security\_level, pwd\_update\_date, pwd\_update\_record, pwd\_question, pwd\_question\_answer, pwd\_question\_2, pwd\_question\_answer\_2, pwd\_question\_3, pwd\_question\_answer\_3, pwd\_quest\_update\_date, last\_login\_ip, last\_login\_date, freeze\_date, freeze\_cause, user\_weight, status, create\_by, create\_date, update\_by, update\_date, remarks, corp\_code, corp\_name, extend\_s1, extend\_s2, extend\_s3, extend\_s4, extend\_s5, extend\_s6, extend\_s7, extend\_s8, extend\_i1, extend\_i2, extend\_i3, extend\_i4, extend\_f1, extend\_f2, extend\_f3, extend\_f4, extend\_d1, extend\_d2, extend\_d3, extend\_d4) VALUES ('admin', 'admin', '系统管理员', '$2a$10$KftQvo2Km6q9gKr7mnMejOXNP/fbgMnxbB2AW3.Tcx1vK7BBz3Mgy', null, null, null, null, null, null, null, null, 'none', null, null, '1', 1, '2020-12-07 23:34:09', null, null, null, null, null, null, null, null, '0:0:0:0:0:0:0:1', '2021-01-27 11:04:10', null, null, 0, '0', 'system', '2020-12-07 23:34:10', 'system', '2020-12-07 23:34:10', '客户方使用的系统管理员，用于一些常用的基础数据配置。', '110', 'JeeSite', null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null);

用户名：admin

密码：123

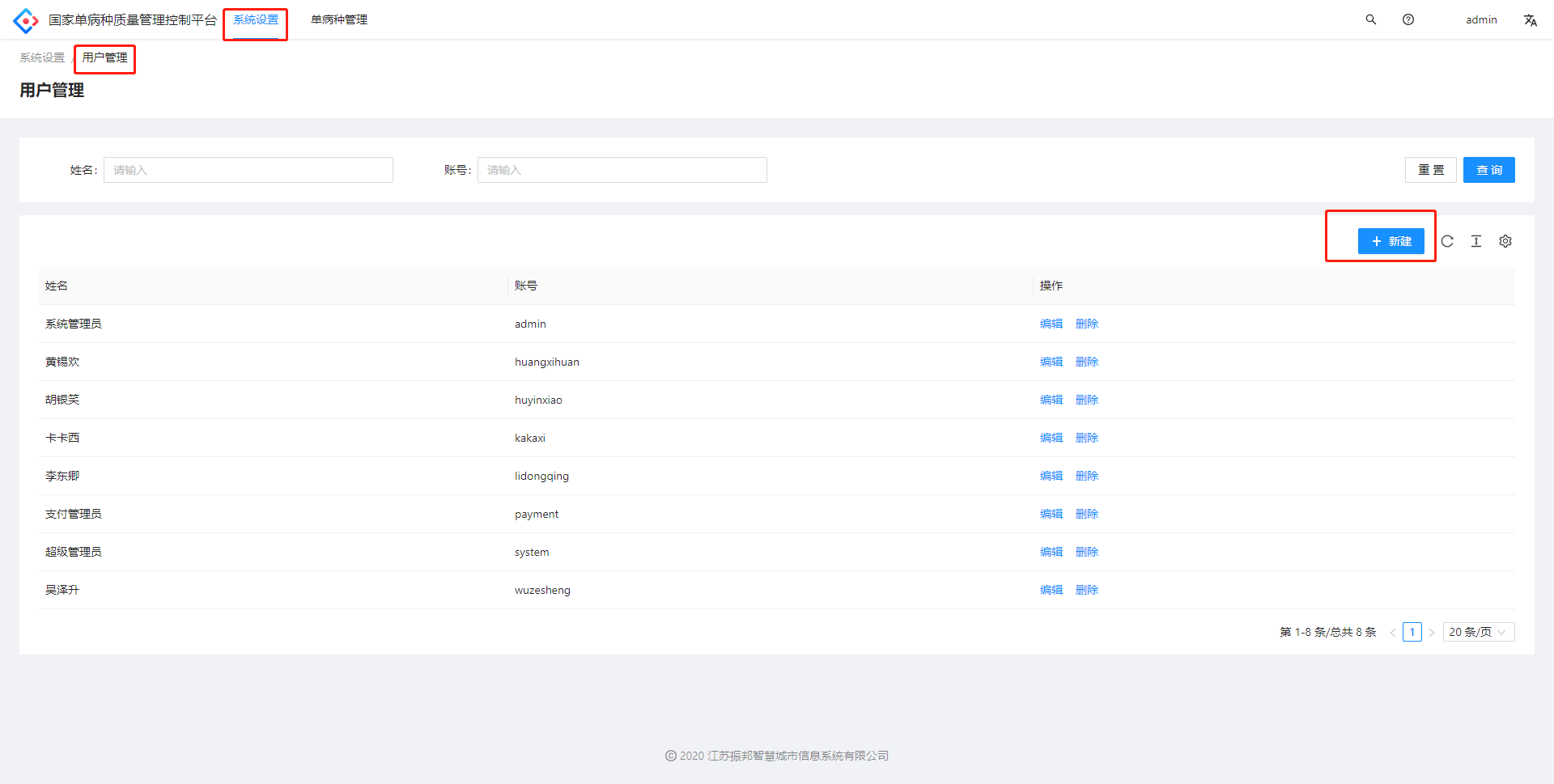
**三、功能介绍**

**1、部门机构**



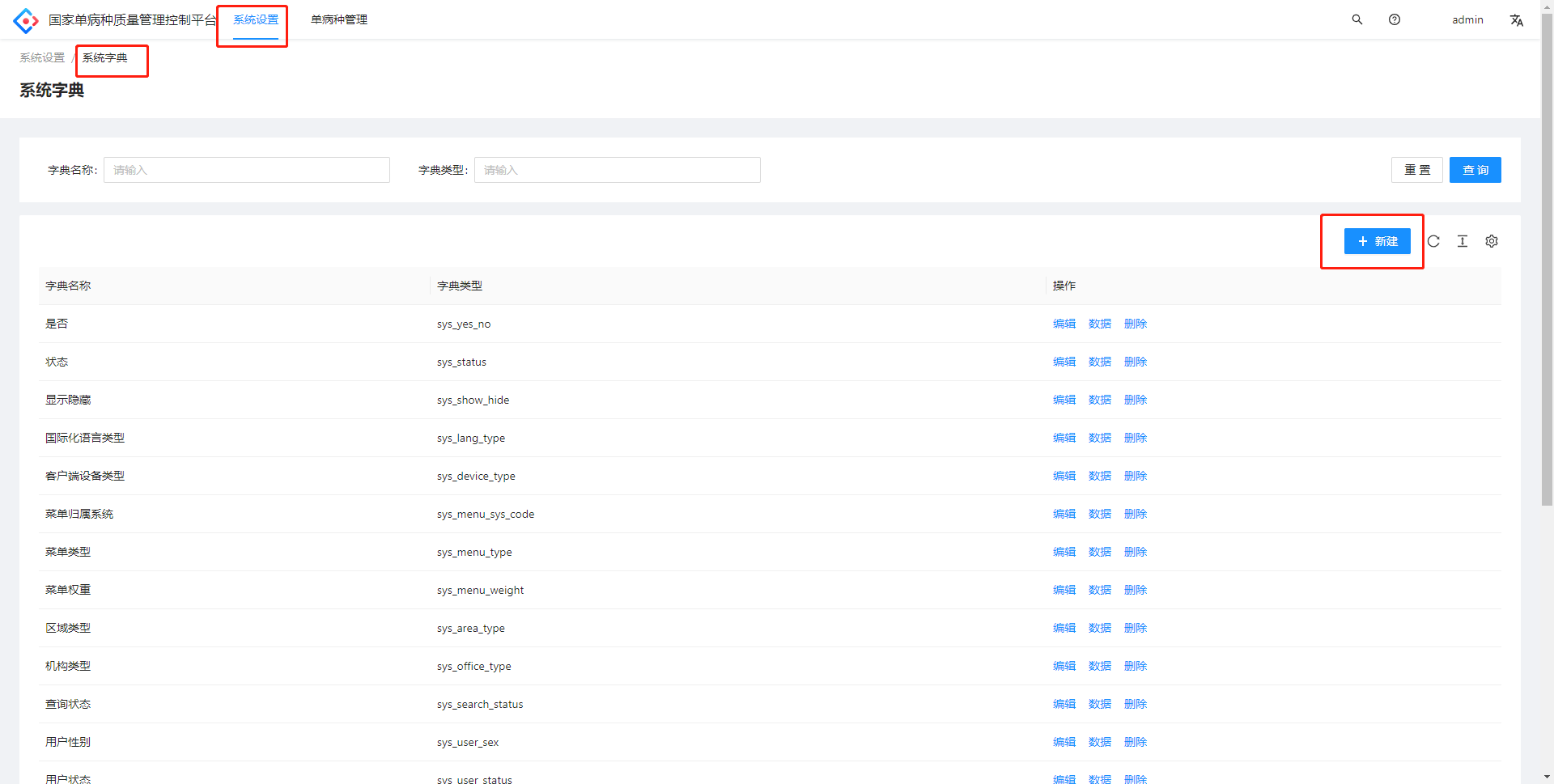
如图所示，在系统设置菜单下的部门管理下，可进行部门机构的创建，编辑，删除，查询等操作

**2、用户管理**



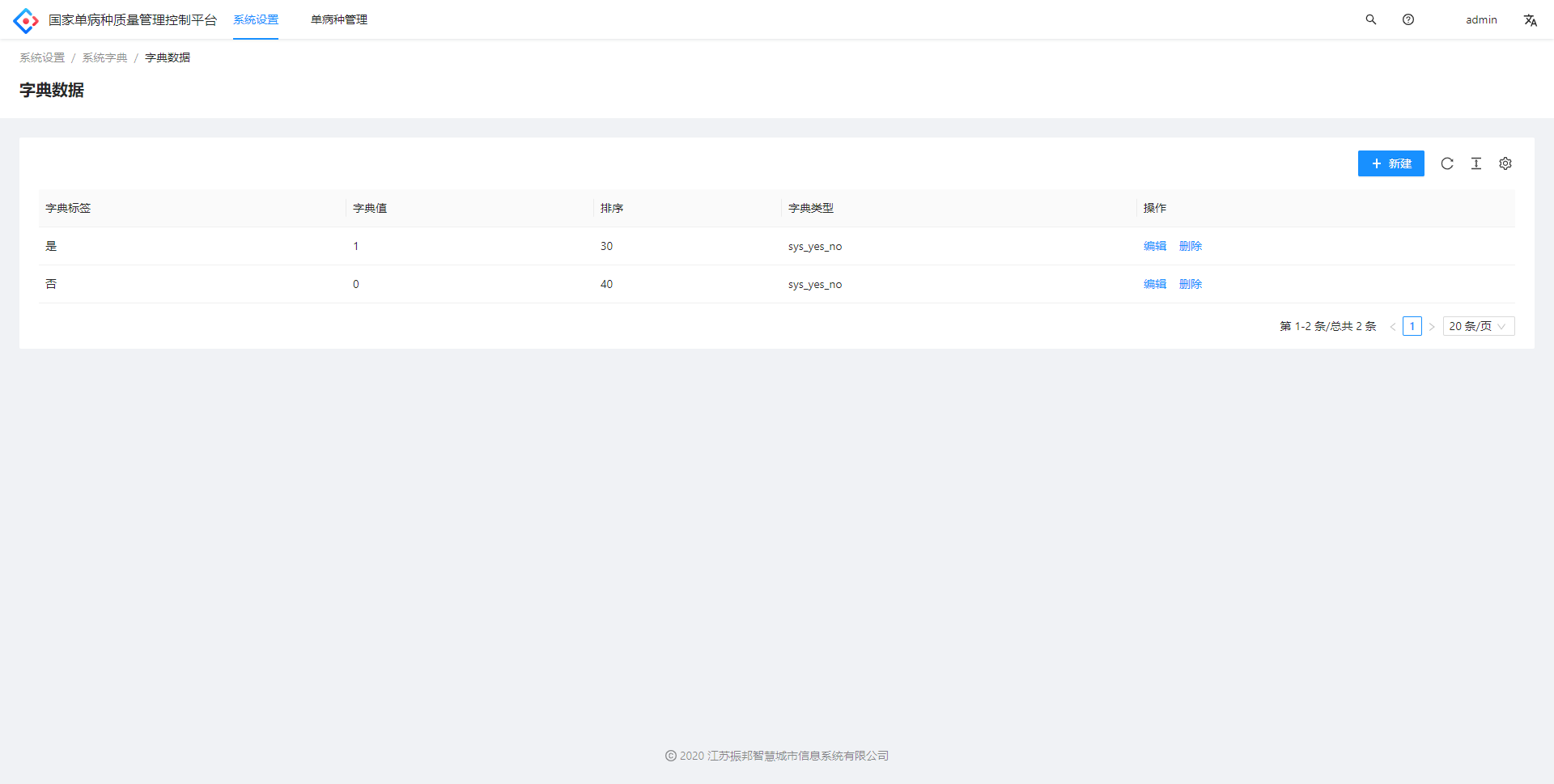
如图所示，在系统设置菜单下的用户管理下，可进行部门机构的创建，编辑，删除，查询等操作

**3、数据字典**



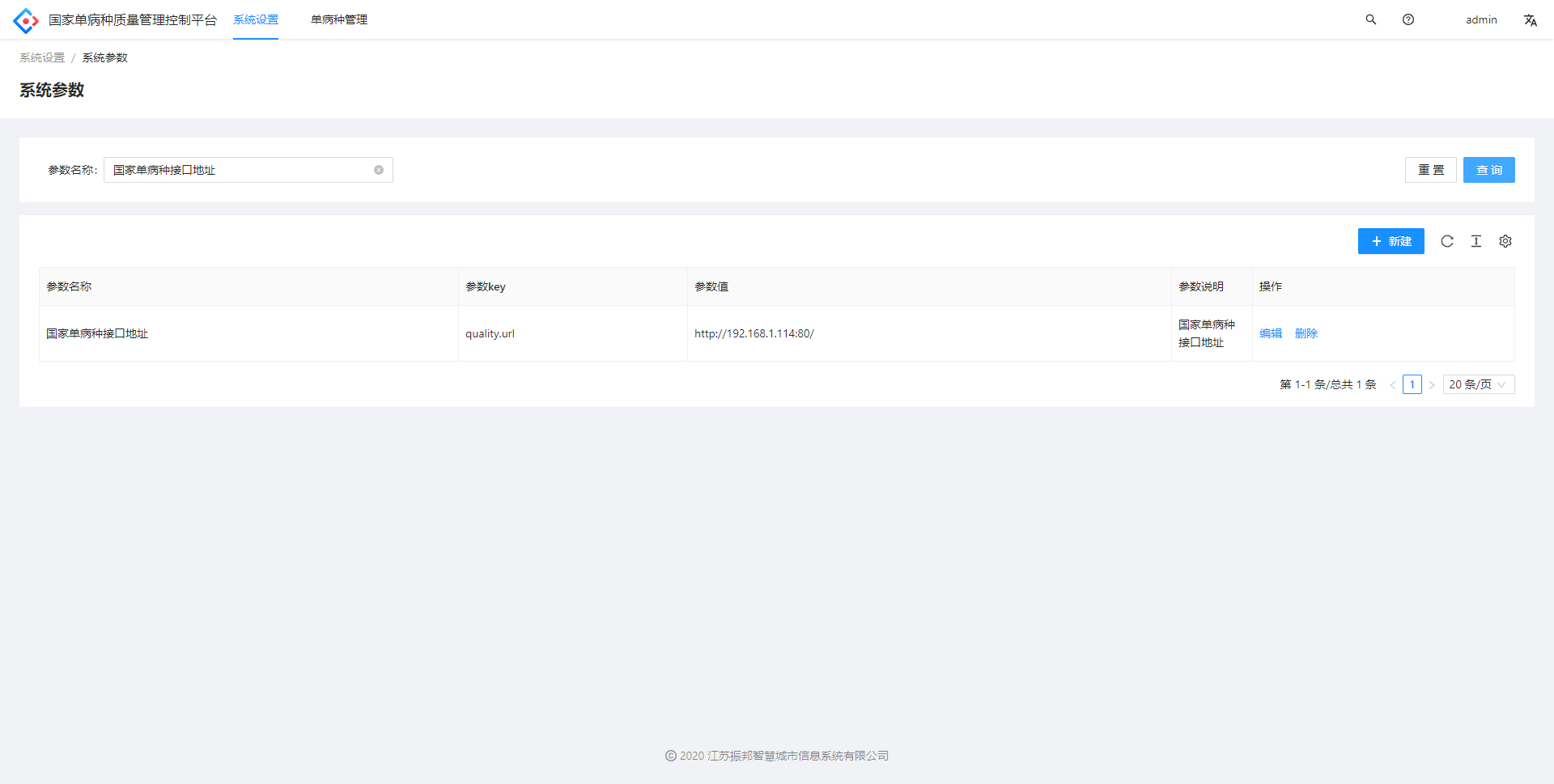
如图所示，在系统设置菜单下的系统字典中，可进行字典项的创建，编辑，删除，查询等操作，单病种文档中所有下拉选项，复选项的编码为字典类型，名称自定义

创建好字典类型后，点击数据按钮，进入字典数据界面，如下图所示：



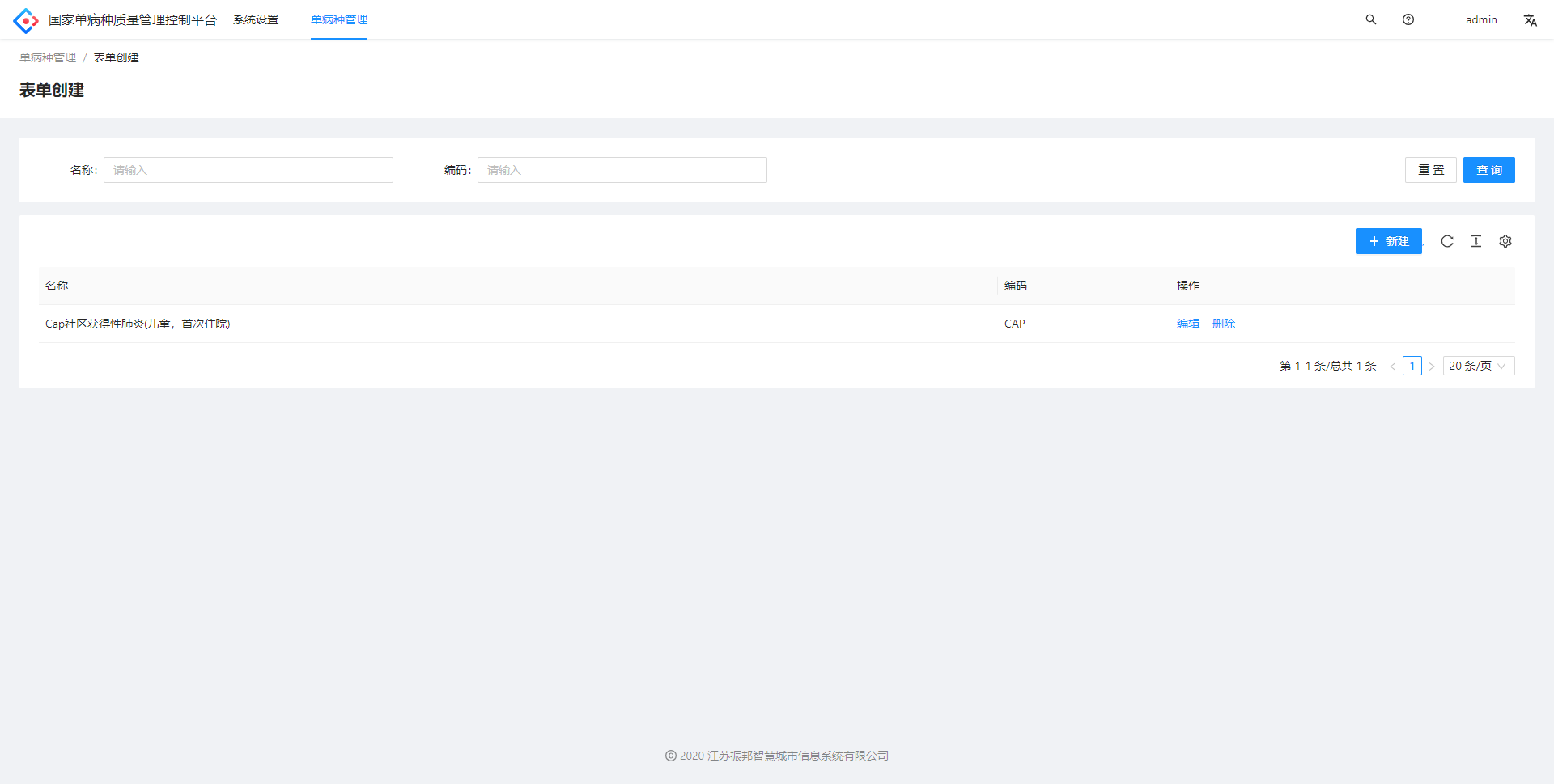
按照单病种文档，值为字典值，标签为要显示出的内容名称

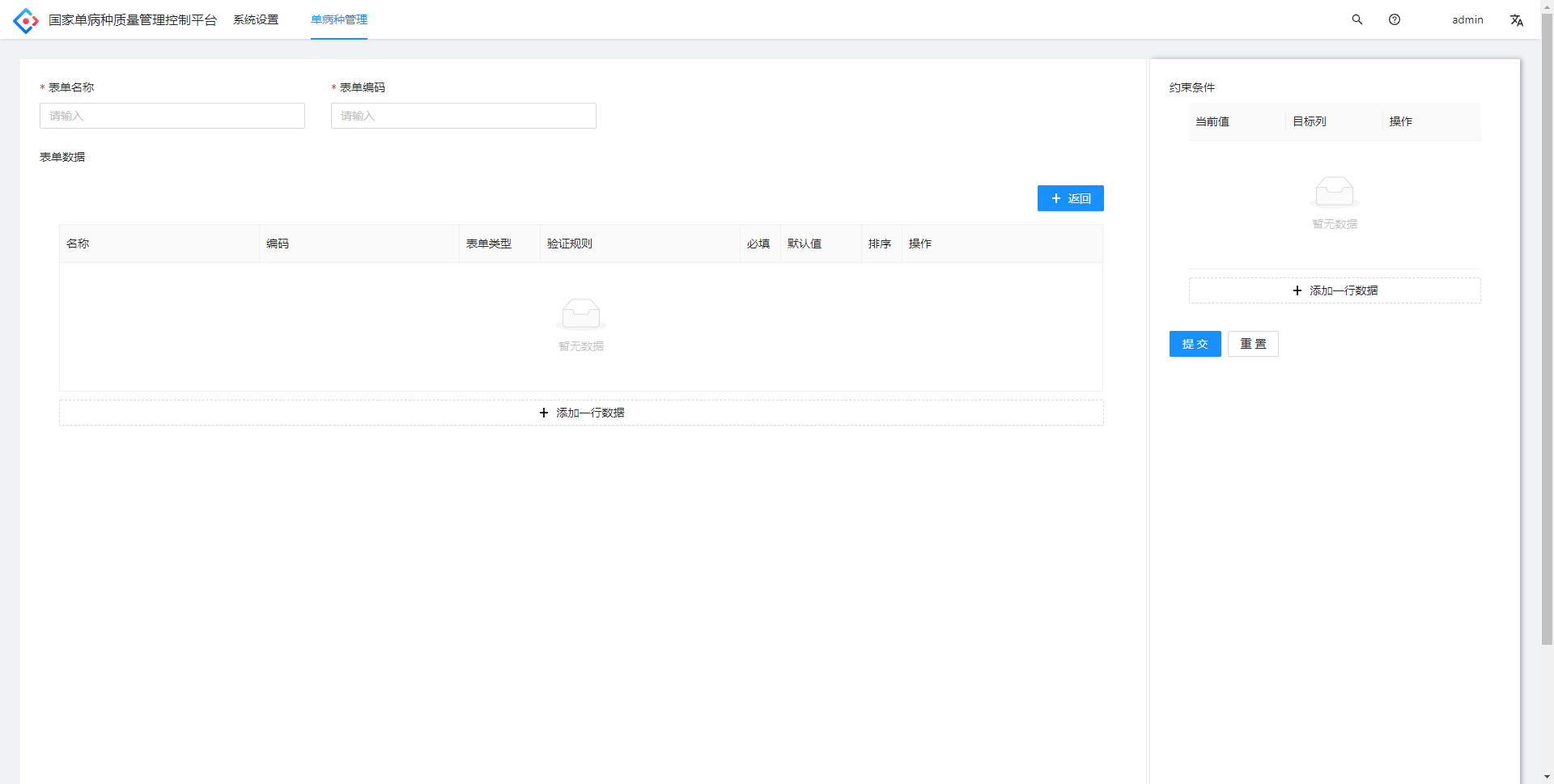
**4、系统参数**



进入系统设置-系统参数，添加单病种上报接口参数，参数key:quality.url,参数值：单病种上报接口地址，注意端口号后跟/

**四、表单创建**



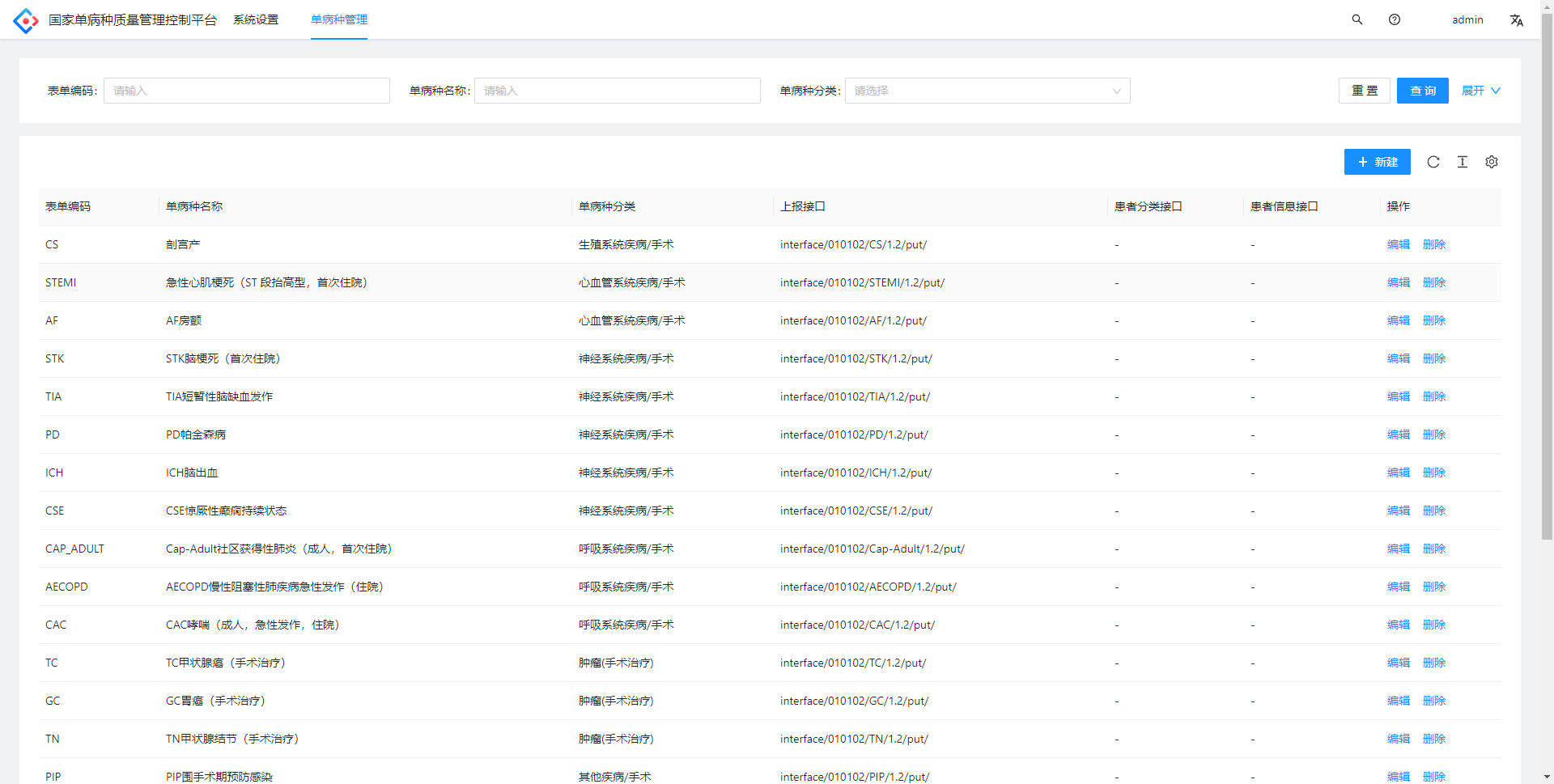


在单病种管理-表单常见菜单中，点击新建按钮，创建表单，表单名称与表单编码必填，表单编码统一大写病种编码。

点击列表下方添加一行数据，设置表单项名称（数据采集项），编码（字段名称），表单类型根据单病种文档选择对应项，必填项根据文档进行勾选，排序（序号），点击保存后设置验证规则。

根据文档中的说明，对有校验的列进项必填项动态设置。

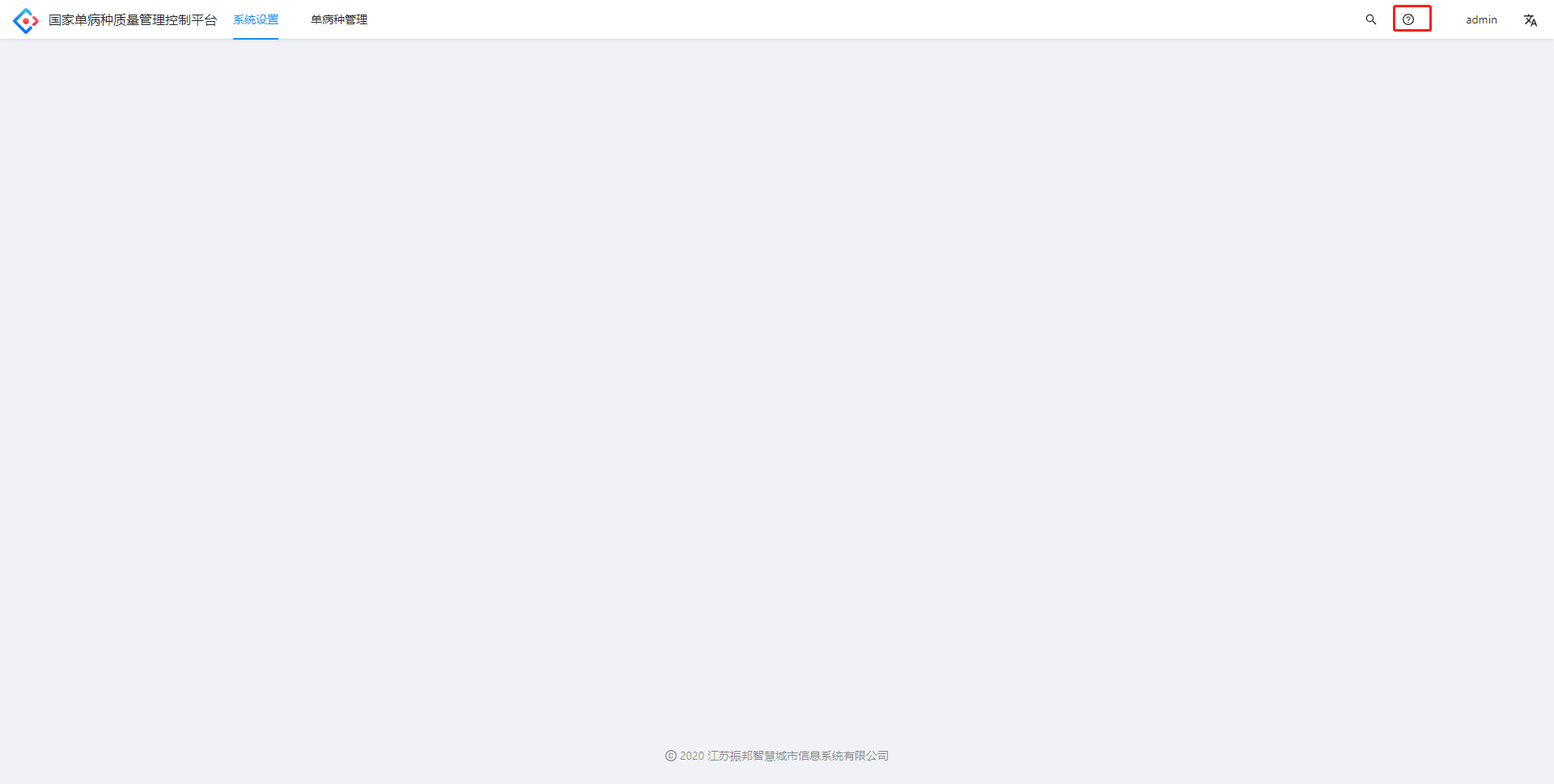
**五、单病种信息设置**



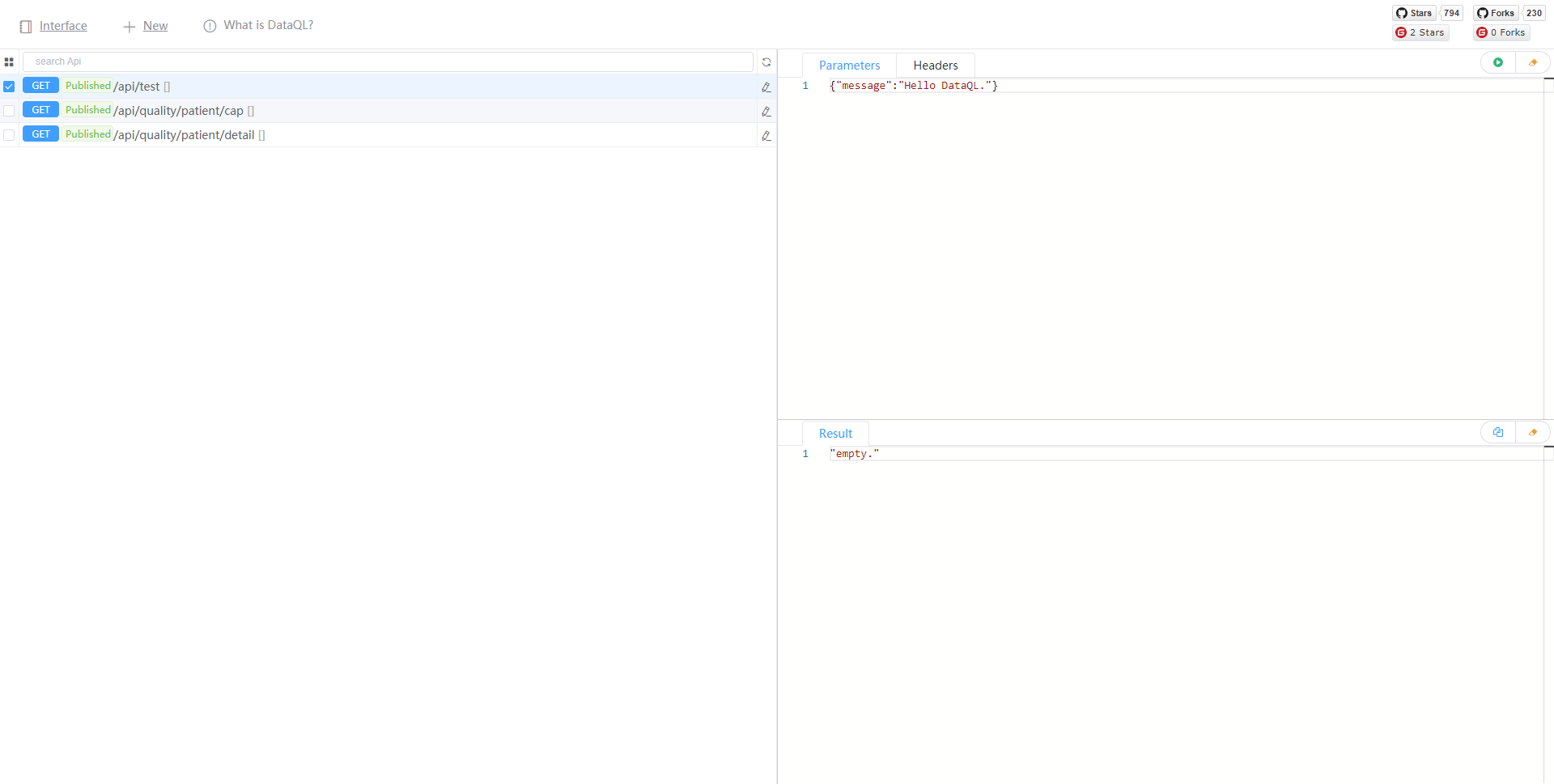
表单创建好后，需要进行单病种信息设置，在单病种设置功能中，需要设置之前创建的表单编码，名称，选择分类，填写上报数据接口，患者分类接口为每个病种筛选出符合上报条件的患者，此接口为自定义接口，在DataWay功能中又运维人员自定义创建，患者信息接口从当前HIS系统中取出需要上报的数据填充表单，同样由运维人员自定义创建接口。

**六、自定义接口创建**

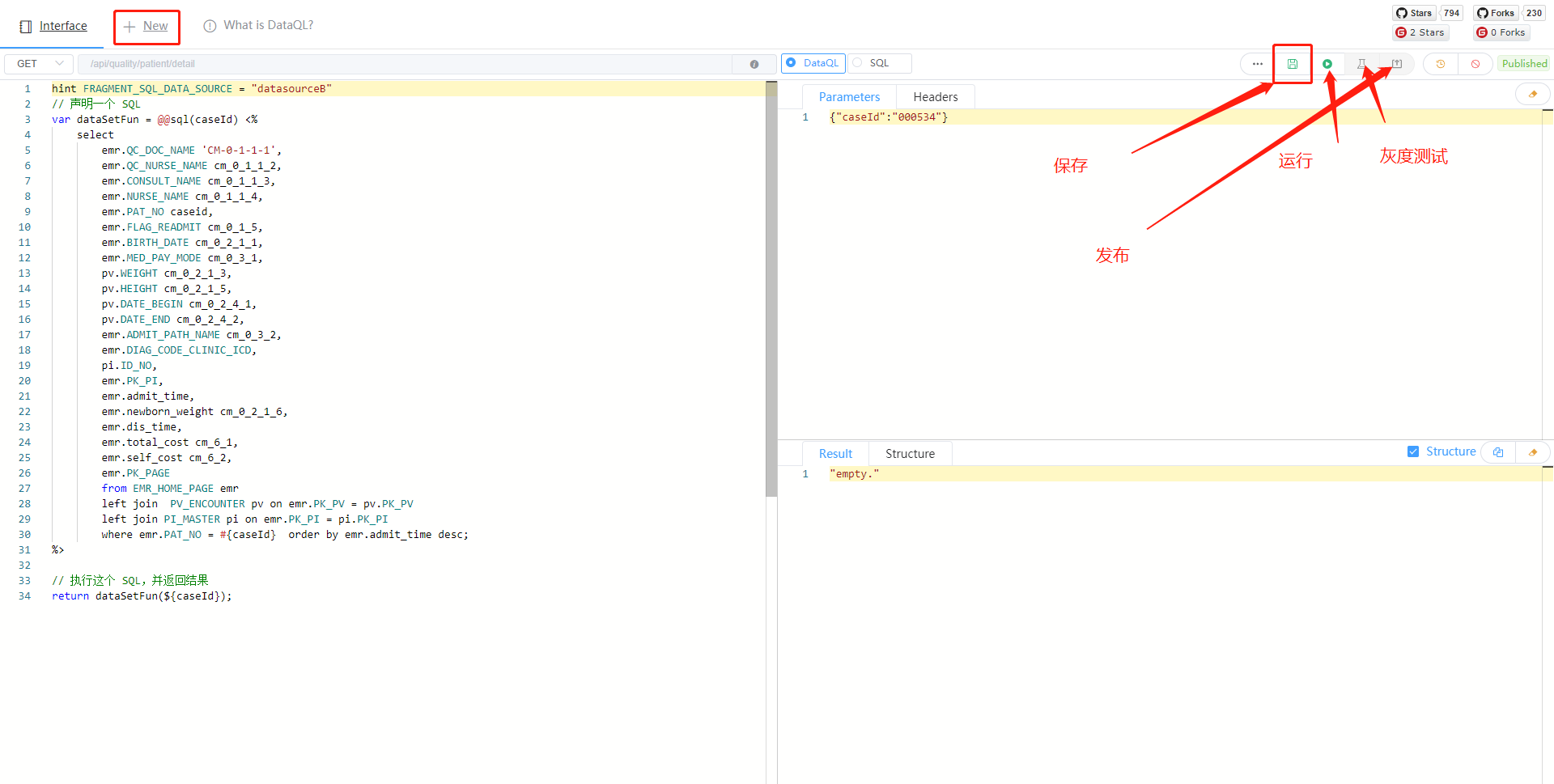
使用DataWay功能，目的是从第二数据源中查询数据。



点击导航右侧的问号图标，进入DataWay管理界面。用户名:admin,密码：admin。入下图所示：



右侧是自定义接口的列表，点解+New按钮，进入新建接口界面，如下图：



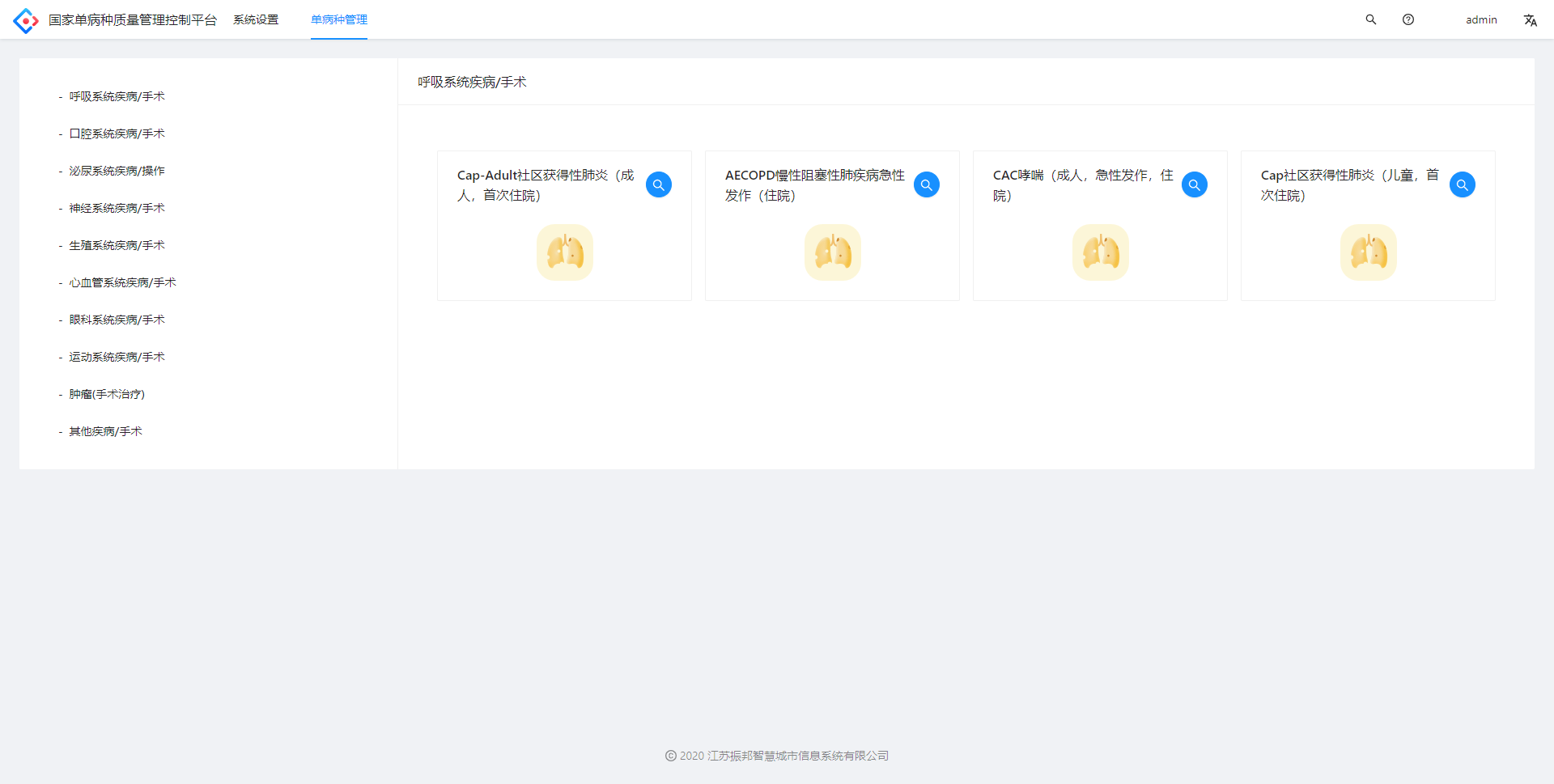
注：所有自定义接口需要设置为GET请求方式

在接口地址栏填写接地址名称，需要以api作为前缀。

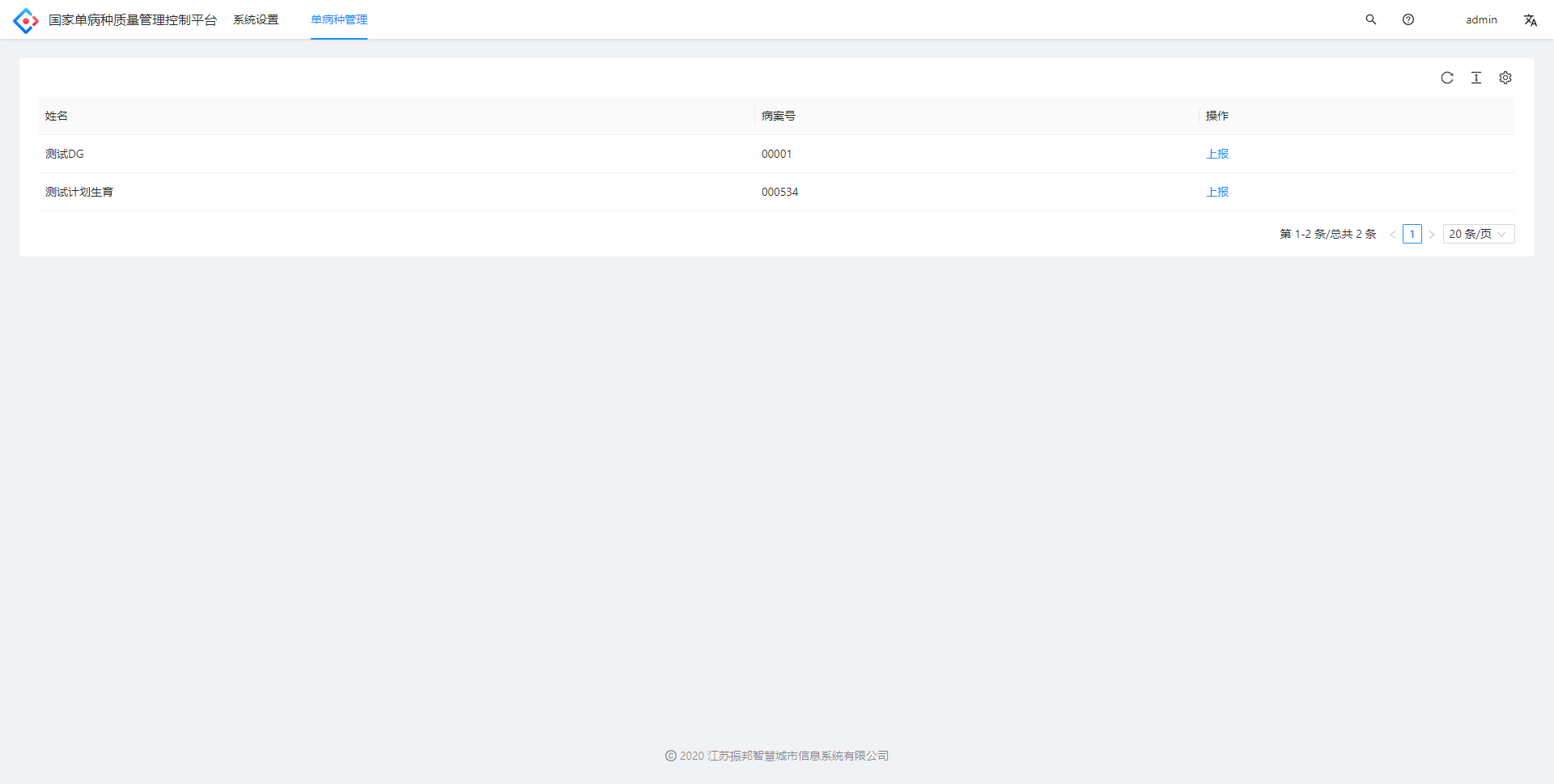
在右侧功能区域，由运维人员编写sql脚本，所谓此自定义接口的数据源。

按照保存，运行，灰度测试，发布四个执行顺序发布自定义接口，发布后将接口地址填写到病种设置中

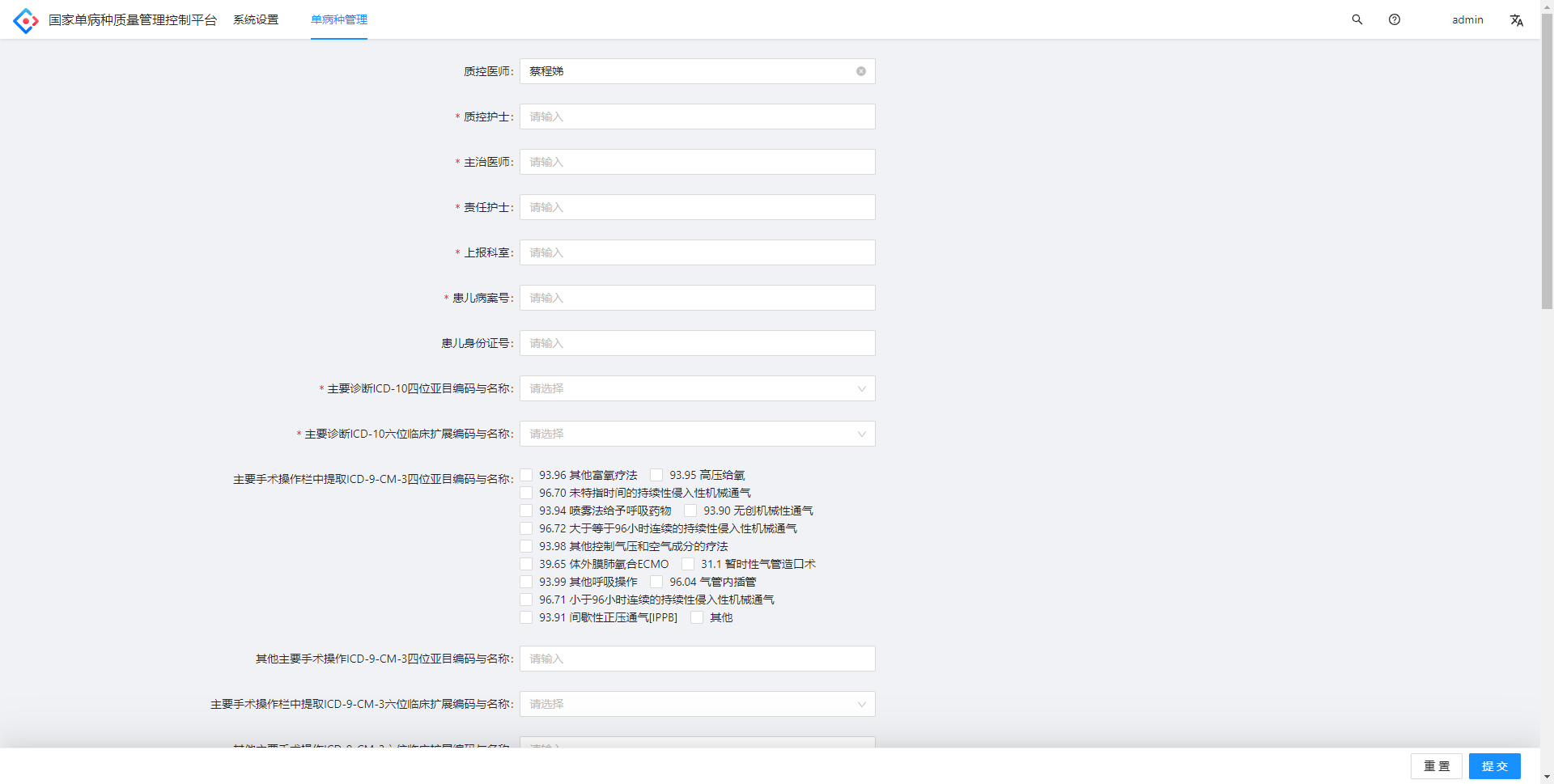
**七、指标填报**



在指标填报界面中，选择需要填报的病种，点击卡片右上角按钮，进入患者选择界面，如下图：



点击上报按钮，进入表单页面，



填写表单项，点击提交上报数据