



H3 BPM 安装及快速使用手册

V10.7.3

二〇一九年十一月

版权所有(c)2010-2019 深圳奥哲网络科技有限公司

目录

1	系统安装.....	3
1.1	系统介绍.....	3
1.2	服务器安装必备软件.....	3
1.3	客户端必备软件.....	3
1.4	注意事项.....	3
2	系统安装.....	3
2.1	安装过程.....	4
2.1.1	Engine 端配置.....	错误！未定义书签。
2.1.2	Portal 端配置.....	4
2.2	系统启动.....	8
2.2.1	启动 Engine.....	11
2.2.2	启动 Portal.....	12
2.3	关闭系统.....	12
2.4	激活系统.....	15
3	系统登录.....	17
3.1	系统首页.....	17
3.2	H3 管理中心.....	17
4	新建流程.....	19
4.1	需求概述.....	19
4.2	新建流程包.....	19
4.3	数据建模.....	21
4.4	设计表单.....	22
4.5	流程建模.....	24
5	流程模拟.....	31
5.1	新增用例.....	31
5.2	用例运行.....	35

6	发起流程.....	37
---	-----------	----

1 系统安装

1.1 系统介绍

本系统分为 Engine 端与 Portal 端，Portal 端负责与用户进行交互，Engine 负责数据的操作。目前版本 Portal 与 Engine 既可以单机部署，也可以分机部署。

1.2 服务器安装必备软件

在使用该 workflow 软件之前，有以下是一些软件是必须安装：

- JDK 8.0 以上版本(必须，安装包已自带，无须再次安装)
- Tomcat 7(必须，安装包已自带，无须再次安装)
- 数据库二选其一即可：1、SQL SERVER 2008 及以上版本 2、Oracle11g 及以上版本，ODAC 112.3.0 版本（安装 Oracle 时必选） 3、MYSQL5.7 及以上版本
- Nginx(必须，安装包已自带，无须再次安装)
- Zookeeper（双引擎必备，安装包已自带，无需再次安装）

1.3 客户端必备软件

- 浏览器：Chrome 浏览器；IE9 以上

1.4 注意事项

- 数据库请勿使用 8.x 或之前版本的 H3 数据库；
- 如果要与原版本共存，那么此版本引擎端口号请勿与旧版本引擎端口号相同，旧版本端口号默认为 8212，此版本引擎端口号可在配置文件中更改，具体更改步骤请看下章节提示说明；
- 如使用 oracle 数据库，确保数据库编码为 **AL32UTF8**，确保数据库最大游标数大于 1000
- 如使用 mysql 数据库，确保数据库编码为 **UTF-8**，**设置表名不区分大小写**，设置允许远程访问，设置允许数据包大小大于 10M。（`lower_case_table_names=1`，`max_allowed_packet=10M`）
- nginx 配置文件 http{} 里面添加 `client_max_body_size 10m`

2 系统安装

2.1 安装过程

将 JAVA 版的 H3 BPM 压缩包解压到对应的目录，解压后的效果如下图：

名称	修改日期	类型	大小
h3bpm-engine	2019/1/18 22:33	文件夹	
h3bpm-portal	2019/1/18 22:30	文件夹	
jdk1.8.0_172	2019/1/18 22:30	文件夹	
nginx-1.14.2	2018/12/4 22:53	文件夹	
zookeeper-3.4.13	2018/7/1 7:36	文件夹	
dist.zip	2019/1/18 22:36	好压 ZIP 压缩文件	3,486 KB

其中 h3bpm-engine 为 Engine 端所在目录，h3bpm-portal 为 Portal 端所在目录，jdk1.8.0_172 为 jdk 所在目录，dist 是移动端静态文件，nginx 代理服务安装包，zookeeper 为注册中心。

解压 dist.zip 文件至当前目录

2.1.1 引擎端配置

进入 h3bpm-engine 目录，编辑 h3bpm_db_config.xml 文件，此文件为配置 H3BPM 的数据库信息，用户需自行指定 SharedService 节点下相应的信息，**注意：只有在 SqlServer 数据库情况下才需配置 H3CloudName 与 H3CloudLogName，其中 H3CloudName 为业务数据库名称，H3CloudLogName 为日志记录库名称，配置它们名称时需确保名称不与原实例中其他数据库名重复即可。**

SqlServer、Oracle 和 MySql 参考配置如下图：

Oracle：

```
<?xml version="1.0" encoding="UTF-8"?>
<Config>
  <CurrentCode>DefaultVessel</CurrentCode>

  <!--oracle-->
  <SharedService DBServer="127.0.0.1" DBName="orcl" H3CloudName="orcl"
    H3CloudLogName="orcl" DBUser="h3bpm" DBPassword="h3password"
    DBType="Oracle" DBConnString="jdbc:oracle:thin:@127.0.0.1:1521:orcl"/>

  <!--sqlserver-->
  <!--
  <SharedService DBServer="127.0.0.1" DBName="h3bpm" H3CloudName="h3bpm"
    H3CloudLogName="h3bpm" DBUser="sa" DBPassword="h3password"
    DBType="SqlServer" DBConnString="jdbc:sqlserver://127.0.0.1:1433;DatabaseName=h3bpm" />
  -->

  <License MachineCode="" Key="" LicenseType="Product" UserLimit="0" WorkflowLimit="0" VesselLimit="0" EngineLimit="0"
    LanguageLimit="0" BizBus="true" BizRule="true" BPA="true" Apps="true" Portal="true" Mobile="true" WeChat="true" SAP="true"
    ExpiredDate="9998-12-31"/>
</Config>
```

SqlServer:

```
<?xml version="1.0" encoding="UTF-8"?>
<Config>
  <CurrentCode>DefaultVessel</CurrentCode>

  <!--oracle-->
  <!--
  <SharedService DBServer="127.0.0.1" DBName="orcl" H3CloudName="orcl"
    H3CloudLogName="orcl" DBUser="h3bpm" DBPassword="h3password"
    DBType="Oracle" DBConnString="jdbc:oracle:thin:@127.0.0.1:1521:orcl"/>
  -->

  <!--sqlserver-->
  <SharedService DBServer="127.0.0.1" DBName="h3bpm" H3CloudName="h3bpm"
    H3CloudLogName="h3bpm" DBUser="sa" DBPassword="h3password"
    DBType="SqlServer" DBConnString="jdbc:sqlserver://127.0.0.1:1433;DatabaseName=h3bpm" />

  <License MachineCode="" Key="" LicenseType="Product" UserLimit="0" WorkflowLimit="0" VesselLimit="0" EngineLimit="0"
    LanguageLimit="0" BizBus="true" BizRule="true" BPA="true" Apps="true" Portal="true" Mobile="true" WeChat="true" SAP="true"
    ExpiredDate="9998-12-31"/>
</Config>
```

MySql:

```
<Config>
  <CurrentCode>DefaultVessel</CurrentCode>

  <!-- ORACLE -->
  <!--<SharedService DBServer="127.0.0.1" DBName="orcl" DBUser="ousf" DBPassword="ousf"
    DBType="Oracle"
    DBConnString="jdbc:oracle:thin:@127.0.0.1:1521:orcl"
    H3CloudName="orcl" H3CloudLogName="orcl"/>-->

  <!-- Mysql -->
  <SharedService DBServer="127.0.0.1" DBName="h3bpm" DBUser="root" DBPassword="123456" DBType="MySql"
    DBConnString="jdbc:mysql://127.0.0.1:3306/h3bpm"
    H3CloudName="mysql" H3CloudLogName="mysql"/>

  <License MachineCode="" Key="" LicenseType="Develop" UserLimit="0" WorkflowLimit="0" VesselLimit="0"
    EngineLimit="0" LanguageLimit="0" BizBus="true" BizRule="true" BPA="true" Apps="true" Portal="true"
    Mobile="true" WeChat="true" SAP="true" ExpiredDate="9998-12-31"/>
</Config>
```

名称	说明
DBServer	数据库服务器 ip, 默认为 127.0.0.1
DBName	数据库名称
DBUser	连接数据库的用户名

DBPassword	连接数据库的密码
DBType	数据库类型，可选项(大小写严格区分): 1, Oracle; 2, SqlServer; 3, MySql
DBConnString	数据库连接 url, 1, Oracle: jdbc:oracle:thin:@数据库 ip:端口号:数据库名称, 2, SqlSever: jdbc:sqlserver://数据库 ip:端口号;DatabaseName=数据库名称 3, MySql: jdbc:mysql:// 数据库 ip:端口号;DatabaseName=数据库名称
H3CloudName	业务数据库名称, SqlServer 情况下才需配置, 不与数据库中实例中其他数据库名称相同即可
H3CloudLogName	业务日志数据库名称, SqlServer 情况下才需配置, 不与数据库中实例中其他数据库名称相同即可

配置完相应信息后保存即可。

最后编辑 h3bpm_engine.properties 文件, 此文件为 Engine 配置文件, 如下图,

如果是单引擎, 可以不使用注册中心, 只需配置 BPMEngine 为 Engine 机器所在 IP 与端口, 格式为: 引擎 ip 地址:端口;即可, 单引擎也可使用注册中心

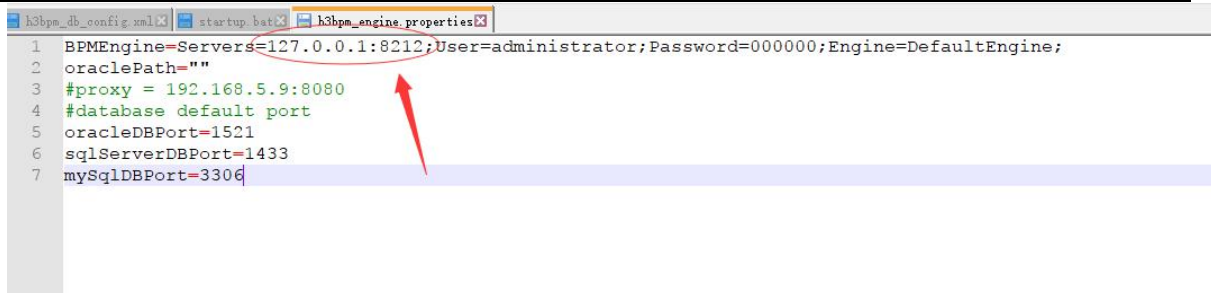
如果是双引擎, 则需要使用注册中心, 除了单引擎所需的配置外, 同时须配置 registry.center.enable 为 true, registry.center.address 注册中心地址, zookeeper 服务器 ip: 端口, 默认端口为 2181

```
BPMEngine=Servers=127.0.0.1:8212;User=administrator;Password=000000;Engine=DefaultEngine;
oraclePath=D:\app\zhaozm\product\11.2.0\dbhome_1\NETWORK\ADMIN
#proxy = 192.168.5.9:8080
#database default port
oracleDBPort=1521
sqlServerDBPort=1433
mySqlDBPort=3306

#registry center is enable or not to get engine service
registry.center.enable=false
#registry center address
registry.center.address=127.0.0.1:2181

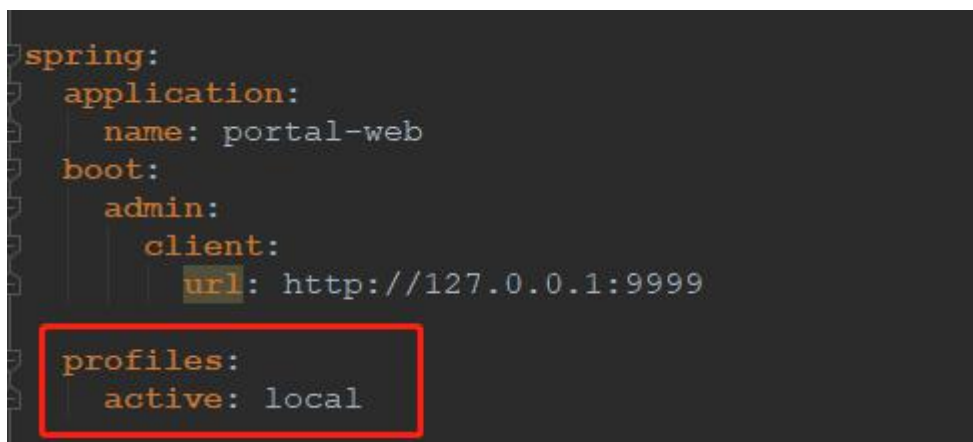
# max threads the engine will use
max.engine.threads=24
```

注意: 如果数据库端口不是默认端口, 此文件也需要作相应修改。



2.1.2 Portal 端配置

修改配置文件前，请先查看 application.yml 文件中 spring.profiles.active 配置了哪个配置文件，如下图，配置了 local，则默认会查找后缀为 local 的 application.yml，即 application-local.yml，若没有找到则默认使用 application.yml 配置文件



1、单引擎，并且没有使用注册中心

进入 Portal 端所在目录，修改 application-xx.yml 文件，**注意：请确保 Tomcat 端口唯一并不被占用并且防火墙规则中允许对外开放此端口**，如下图：



引擎连接地址，默认修改引擎 ip 和端口就可以了，密码使用 MD5 加密


```
h3:
  system:
    version: @project.version@
  engine:
    server: Servers=127.0.0.1:8212;User=administrator;Password=670b14728ad9902aecba32e22fa4f6bd;Engine=DefaultEngine; #引擎连接地址
```

名称	说明
引擎地址信息	格式为：Servers=引擎 ip 地址:端口;User=用户名;Password=密码;Engine=引擎编码 注意：这里的信息需要与引擎端 h3bpm_engine.properties 文件配置的地址信息保持一致。
Tomcat 端口	用户打开 BPM 管理系统时 URL 中的端口号，请确保端口唯一并不被占用并且防火墙规则中允许对外开放此端口 注意：请不要将此端口与引擎端口配置成一样。

2、双引擎，或者使用了注册中心

修改 application-xx.yml 文件

H3.engine.server 无需配置，必须得配置 zookeeper.enable 为 true，
zookeeper.server 注册中心地址，zookeeper 服务器 ip: 端口，默认端口为 2181，
与引擎端配置同步

```
# 注册中心
zookeeper:
  enabled: false
  server: 127.0.0.1:2181

h3:
  system:
    version: @project.version@
  engine:
    server: Servers=127.0.0.1:8212;User=administrator;Password=000000
#新版移动端
```

2.1.3 前端部署

前端代码使用 nginx 部署，请求后台接口通过 nginx 转发。

转发规则是：只转发非静态资源和服务端生成的图片。

部署需要修改 nginx.conf 文件：

1. root 改成前端代码根目录。

```
server {
    listen      80 default_server;
    server_name 172.18.0.65;

    # 前端源码地址
    root        D:\h3-portal\frontend;
    client_max_body_size 100M;
    # Load configuration files for the default server block.
    #include /etc/nginx/default.d/*.conf;

    location / {
        try_files $uri $uri/ /index.html;

        expires 7d;
        log_not_found off;
        access_log off;
    }

    #后台接口使用代理转发（非静态资源）
    location ~* ^(?:\.png|\.jpg|\.gif|\.jpeg|\.js|\.html|\.htm|\.json|\.css|\.woff|\.ttf|\.otf|\.eot|\.svg|\.ico)
    {
        #proxy_pass http://47.98.49.203:8085;
        proxy_pass http://172.18.0.65:8083;
    }
    #文件上传使用后台临时图片
    location ~* attached|Portal/TempImages|TempFiles {
        #proxy_pass http://47.98.49.203:8085;
        proxy_pass http://172.18.0.65:8083;
    }
    error_page 404 /404.html;
    location = /40x.html {
    }
}
```

1. 修改接口代理地址 proxy_pass 为 portal 的访问地址（2 处）

```
server {
    listen      80 default_server;
    server_name 172.18.0.65;

    # 前端源码地址
    root        D:\h3-portal\frontend;
    client_max_body_size 100M;
    # Load configuration files for the default server block.
    #include /etc/nginx/default.d/*.conf;

    location / {
        try_files $uri $uri/ /index.html;

        expires 7d;
        log_not_found off;
        access_log off;
    }

    #后台接口使用代理转发（非静态资源）
    location ~* ^(?:\.png|\.jpg|\.gif|\.jpeg|\.js|\.html|\.htm|\.json|\.css|\.woff|\.ttf|\.otf|\.eot|\.svg|\.ico|\.ico)*$ {
        #proxy_pass http://47.98.49.203:8085;
        proxy_pass http://172.18.0.65:8083;
    }
    #文件上传使用后台临时图片
    location ~* attached|Portal/TempImages|TempFiles {
        #proxy_pass http://47.98.49.203:8085;
        proxy_pass http://172.18.0.65:8083;
    }
    error_page 404 /404.html;
    location = /40x.html {
    }
}
```

2.1.4 Nginx 配置

1、修改 nginx/conf/nginx.conf 文件，参考安装包自带的配置，根据当前服务器配置修改以下红色标注的地方

```

http {
    include      mime.types;
    default_type application/octet-stream;

    #定义日志格式
    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;

    server_names_hash_bucket_size 64;

    upstream portal_server {
        server 192.168.8.141:8081 weight=2 max_fails=2 fail_timeout=30s;
    }

    server {
        listen      8082;
        server_name 192.168.8.141;

        location / {
            proxy_pass http://portal_server/;
            proxy_set_header Host $http_host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        location /Mobile {
            alias C:/Users/Administrator/Desktop/dist/;
            #try_files $uri $uri/ /index.html;
        }
    }
}

```

修改为portal服务的ip:端口

当前服务器的ip或域名, 以及监听的端口

dist.zip文件解压后的路径

修改 dist 目录下 serverConfig.json 文件, 修改为 nginx 当前监听使用的 ip: 端口

```
{ "baseUrl": "http://192.168.8.141:8082" }
```

2.1.5 新版移动端

安装包默认使用了老版本移动端, 要使用新版本需要修改配置

使用新版本配置:

修改 application.yml

```

#新版移动端
# scanBasePackages : OThinker.H3.Controller.*,com.h3bpm.mobile.controller,com.h3bpm.base.controller.handler
#旧版移动端
scanBasePackages: OThinker.H3.Controller.*,OThinker.H3.Mobile.Controller,com.h3bpm.base.controller.handler

```

注释掉老版的接口扫描，使用新版本的接口扫描

2.2 系统启动

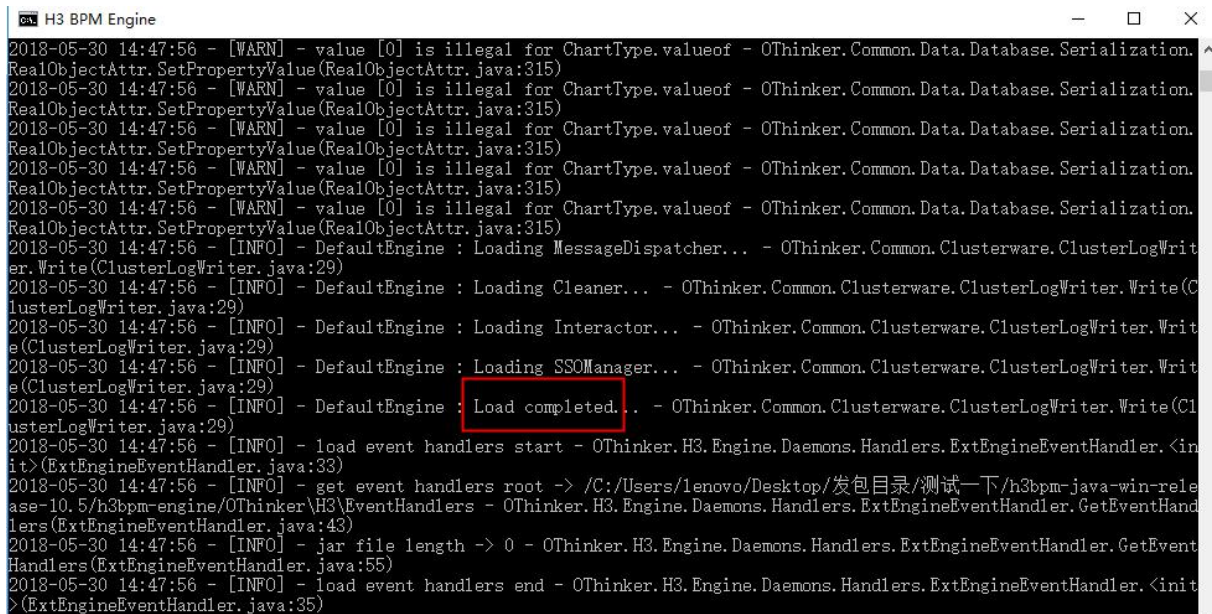
2.2.1 启动 zookeeper

进入 zookeeper 解压目录，进入 bin 目录，双击 zkServer.bat 启动 zookeeper。出现下面窗口则表示启动成功。

```
C:\Windows\system32\cmd.exe
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.version=1.8.0_191
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.vendor=Oracle Corporation
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.home=C:\Program Files\Java\jdk1.8.0_191\jre
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.class.path=C:\Wind\zookeeper-3.4.13\bin\..\build\classes;C:\Wind\zookeeper-3.4.13\bin\..\build\lib\*;C:\Wind\zookeeper-3.4.13\bin\..\zookeeper-3.4.13.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\audience-annotations-0.5.0.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\jline-0.9.94.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\log4j-1.2.17.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\netty-3.10.6.Final.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\slf4j-api-1.7.25.jar;C:\Wind\zookeeper-3.4.13\bin\..\lib\slf4j-log4j12-1.7.25.jar;C:\Wind\zookeeper-3.4.13\bin\..\conf
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.library.path=C:\Program Files\Java\jdk1.8.0_191\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Program Files\Java\jdk1.8.0_191\bin;C:\Program Files\Java\jdk1.8.0_191\jre\bin;C:\Wind\mysql-5.7.24-winx64\bin;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\iCLS;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\NVIDIA Corporation\PhysX Common;C:\Wind\apache-maven-3.6.0\bin;C:\Program Files\TortoiseSVN\bin;C:\Users\Wind\AppData\Local\Microsoft\WindowsApps;.
2019-01-19 18:10:41.073 [myid:] - INFO [main:Environment@100] - Server environment:java.io.tmpdir=C:\Users\Wind\AppData\Local\Temp\
2019-01-19 18:10:41.074 [myid:] - INFO [main:Environment@100] - Server environment:java.compiler=(NA)
2019-01-19 18:10:41.074 [myid:] - INFO [main:Environment@100] - Server environment:os.name=Windows 10
2019-01-19 18:10:41.075 [myid:] - INFO [main:Environment@100] - Server environment:os.arch=amd64
2019-01-19 18:10:41.075 [myid:] - INFO [main:Environment@100] - Server environment:os.version=10.0
2019-01-19 18:10:41.075 [myid:] - INFO [main:Environment@100] - Server environment:user.name=Wind
2019-01-19 18:10:41.075 [myid:] - INFO [main:Environment@100] - Server environment:user.home=C:\Users\Wind
2019-01-19 18:10:41.076 [myid:] - INFO [main:Environment@100] - Server environment:user.dir=C:\Wind\zookeeper-3.4.13\bin
2019-01-19 18:10:41.082 [myid:] - INFO [main:ZooKeeperServer@836] - tickTime set to 2000
2019-01-19 18:10:41.082 [myid:] - INFO [main:ZooKeeperServer@845] - minSessionTimeout set to -1
2019-01-19 18:10:41.082 [myid:] - INFO [main:ZooKeeperServer@854] - maxSessionTimeout set to -1
2019-01-19 18:10:41.203 [myid:] - INFO [main:ServerCnxnFactory@117] - Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory
2019-01-19 18:10:41.204 [myid:] - INFO [main:NIOServerCnxnFactory@89] - binding to port 0.0.0.0/0.0.0.0:2181
```

2.2.2 启动 Engine

进入 Engine 所在目录，双击运行 startup.bat 即可启动 Engine，启动过程中可查看控制台实时输出，当出现 Load Completed 字样时，表明 Engine 启动完成，如下图：



```
H3 BPM Engine
2018-05-30 14:47:56 - [WARN] - value [0] is illegal for ChartType.valueOf - OThinker.Common.Data.Database.Serialization.
RealObjectAttr.SetPropertyValue(RealObjectAttr.java:315)
2018-05-30 14:47:56 - [WARN] - value [0] is illegal for ChartType.valueOf - OThinker.Common.Data.Database.Serialization.
RealObjectAttr.SetPropertyValue(RealObjectAttr.java:315)
2018-05-30 14:47:56 - [WARN] - value [0] is illegal for ChartType.valueOf - OThinker.Common.Data.Database.Serialization.
RealObjectAttr.SetPropertyValue(RealObjectAttr.java:315)
2018-05-30 14:47:56 - [WARN] - value [0] is illegal for ChartType.valueOf - OThinker.Common.Data.Database.Serialization.
RealObjectAttr.SetPropertyValue(RealObjectAttr.java:315)
2018-05-30 14:47:56 - [WARN] - value [0] is illegal for ChartType.valueOf - OThinker.Common.Data.Database.Serialization.
RealObjectAttr.SetPropertyValue(RealObjectAttr.java:315)
2018-05-30 14:47:56 - [INFO] - DefaultEngine : Loading MessageDispatcher... - OThinker.Common.Clusterware.ClusterLogWriter.
Write(ClusterLogWriter.java:29)
2018-05-30 14:47:56 - [INFO] - DefaultEngine : Loading Cleaner... - OThinker.Common.Clusterware.ClusterLogWriter. Write(C
lusterLogWriter.java:29)
2018-05-30 14:47:56 - [INFO] - DefaultEngine : Loading Interactor... - OThinker.Common.Clusterware.ClusterLogWriter. Writ
e(ClusterLogWriter.java:29)
2018-05-30 14:47:56 - [INFO] - DefaultEngine : Loading SSOManager... - OThinker.Common.Clusterware.ClusterLogWriter. Writ
e(ClusterLogWriter.java:29)
2018-05-30 14:47:56 - [INFO] - DefaultEngine : Load completed... - OThinker.Common.Clusterware.ClusterLogWriter. Write(Cl
usterLogWriter.java:29)
2018-05-30 14:47:56 - [INFO] - load event handlers start - OThinker.H3.Engine.Daemons.Handlers.ExtEngineEventHandler.<in
it>(ExtEngineEventHandler.java:33)
2018-05-30 14:47:56 - [INFO] - get event handlers root -> /C:/Users/lenovo/Desktop/发包目录/测试一下/h3bpm-java-win-rele
ase-10.5/h3bpm-engine/OThinker\H3\EventHandlers - OThinker.H3.Engine.Daemons.Handlers.ExtEngineEventHandler.GetEventHand
lers(ExtEngineEventHandler.java:43)
2018-05-30 14:47:56 - [INFO] - jar file length -> 0 - OThinker.H3.Engine.Daemons.Handlers.ExtEngineEventHandler.GetEvent
Handlers(ExtEngineEventHandler.java:55)
2018-05-30 14:47:56 - [INFO] - load event handlers end - OThinker.H3.Engine.Daemons.Handlers.ExtEngineEventHandler.<init
>(ExtEngineEventHandler.java:35)
```

2.2.3 启动 Portal

进入 Portal 所在目录，运行 ./startup.bat，即可启动 Portal，当日志出现 started 字样时，表明 Portal 启动成功，如下图：

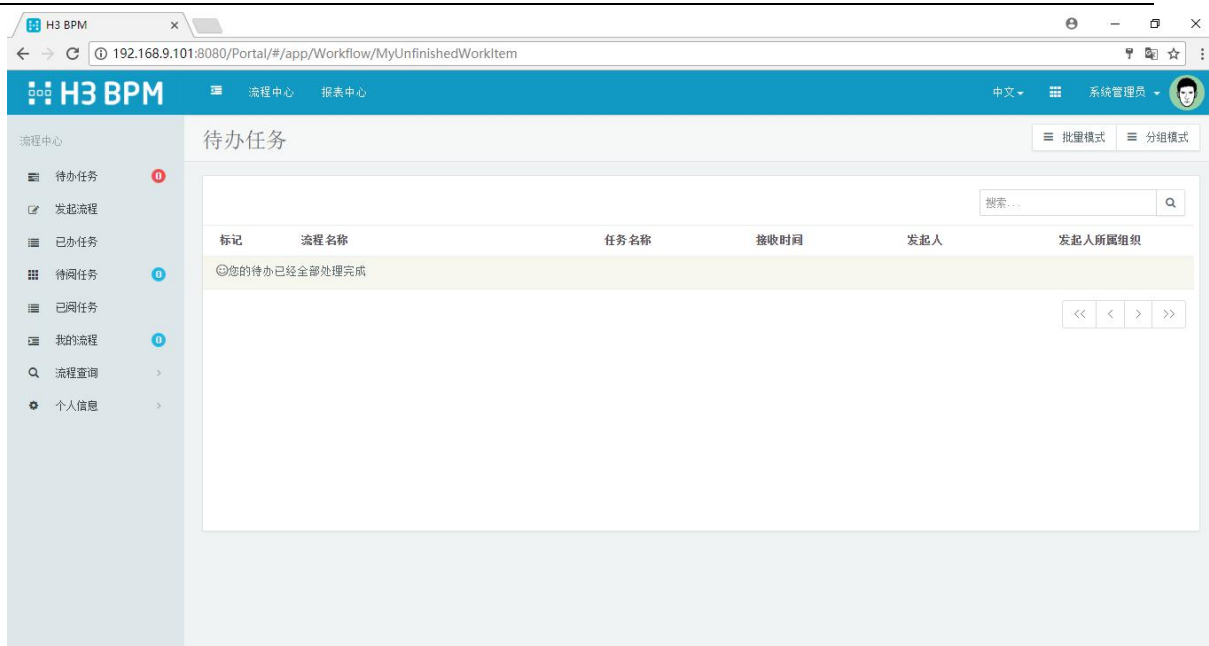


```
2019-08-27 18:56:29.483 [main] INFO com.h3bpm.web.WebApiBootStartupApplicat
ion Line:59 - Started WebApiBootStartupApplication in 6.481 seconds (JVM ru
nning for 7.199)
```

2.2.4 启动 Nginx

1、启动 nginx，进入 nginx 安装包解压目录，运行 Nginx.bat 文件，按照操作提示启动 nginx

2、用户打开网页浏览器，输入 `http://{Nginx 所在机器 ip}:{Nginx 监听端口号}/`，打开系统登陆页后，输入初始用户名：Administrator，初始密码：000000；如下图，若登陆成功并进入系统门户页面，则说明系统 PC 端安装并启动成功。



3、用户使用手机打开网页浏览器，输入 `http://{Nginx 所在机器 ip}:{Nginx 监听端口号}/Mobile`，打开系统移动端登陆页后，输入初始用户名：Administrator，初始密码：000000；如下图，若登陆成功并进入移动端首页，则说明系统移动端安装并启动成功。

⚙️
H3 BPM
🔍 筛选

待办 138
待阅 49
已办
已阅

数据项.9

接收时间: 2019-01-18 14:27

短文本: High

长文本:

逻辑: false

整数: 0

长整数: 0

数值: 0.0

日期:

单选:

时间段: 0天0小时0分钟0秒

链接: <https://www.baidu.com/>

[审批](#)

表单控件.17

接收时间: 2019-01-18 14:09

[审批](#)

表单控件.17

接收时间: 2019-01-10 11:05

[审批](#)

表单控件.17

接收时间: 2019-01-10 09:04

[审批](#)

表单控件.17

接收时间: 2019-01-09 21:20

[审批](#)

首页中心

发起流程

我的流程

应用中心

2.3 CAS 单点登录集成

H3BPM 提供了 CAS 单点登录服务，第三方系统可以集成 H3 CAS，使用 H3BPM 的组织机构进行单点登录

修改 application.yml 文件配置，启用 cas

```
h3:
  cas:
    enabled: true
    server_url: http://127.0.0.1:8089
    localhost: http://127.0.0.1:8081
    success_target_url: /Portal/#/app/Workflow/MyUnfinishedWorkItem
    failure_target_url: /Portal/#/app/Workflow/MyUnfinishedWorkItem
```

Enabled 设为 true，启动 cas 登录

Server_url 为 cas 服务地址

Localhost 为 portal 服务地址

Success_target_url 是登录成功后需要跳转的地址

failure_target_url 是登录失败后需要跳转的地址

2.4 关闭系统

先关闭 Portal 的命令窗口，再关闭 Engine 的命令窗口，再按照 Nginx.bat 命令窗口提示关闭 nginx、退出，即可结束系统的运行。

2.5 激活系统

注意：系统默认安装后为开发版，若要切换为正式版，需要输入相应的激活码进行激活，系统激活一定要在系统关闭的情况下进行，激活成功后再次启动系统后即可生效。

1、进入 Engine 所在的目录，运行“register.bat”，根据提示，输入相应的激活码，点击注册按钮即可，如图：

H3 BPM注册

欢迎使用H3 BPM

机器码 2DC9-F10D-1ACE-D7D5-AA3B-F8B2

注册码

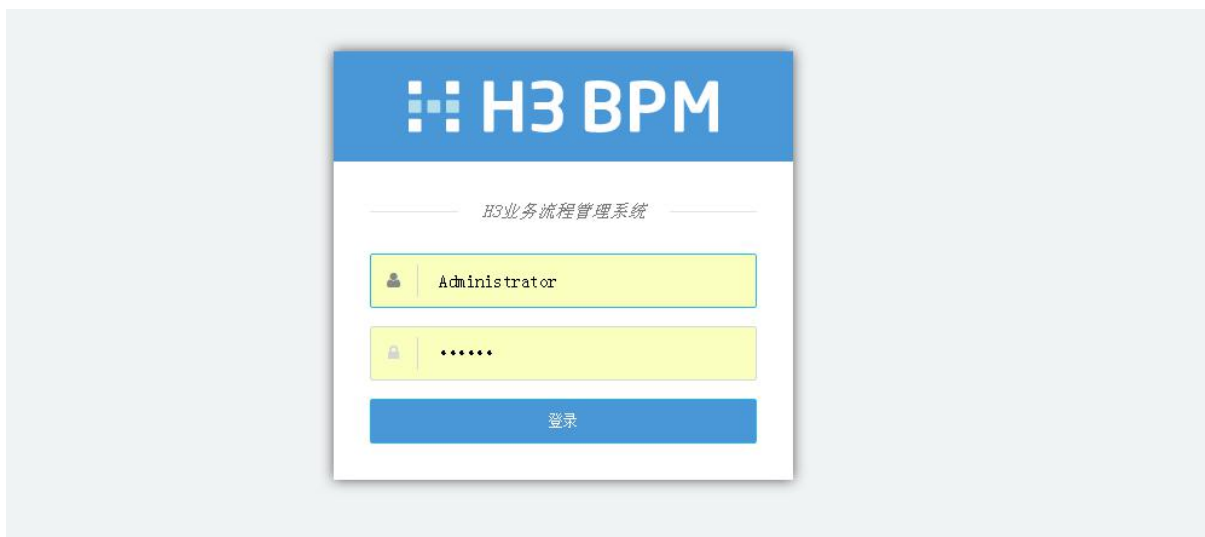
注册

3 系统登录

3.1 系统首页

用户提供给终端用户进行使用，包含流程中心、报表中心和个人设置修改等功能。

输入 URL 地址：`http://{Portal 所在机器 ip}:{Tomcat 端口号}/`即可访问系统首页



输入用户名和密码，点击登录，进入门户管理；

初始用户名：Administrator，初始密码：000000；

登录后“流程中心”界面如下：

标记	流程名称	任务名称	接收时间	发起人	发起人所属组织
🟢		主管审批	2017-09-29 09:05	系统管理员	集团公司
🟢	BPM发起流程并新增数据至业务系统.2	审批	2017-09-28 21:09	系统管理员	集团公司
🟢	BPM发起流程并新增数据至业务系统.2	审批	2017-09-28 18:16	系统管理员	集团公司
🟢	子流程: 0928	子流程填单	2017-09-28 16:37	系统管理员	集团公司
🟢	子流程: 0928	子流程填单	2017-09-28 16:37	系统管理员	集团公司
🟢	测试	表单选人	2017-09-28 10:59	系统管理员	集团公司
🟢	十月支持	主管审核	2017-09-25 14:51	Lydia	区域本部
🟢	带附件	审批	2017-09-25 11:15	系统管理员	集团公司
🟢	111	表单选人	2017-09-24 21:10	系统管理员	集团公司
🟢	表单和控件.2	手工	2017-09-22 10:33	系统管理员	集团公司

3.2 H3 管理中心

H3 管理中心包含：组织机构、业务集成、流程模型、应用中心、移动应用、系统管

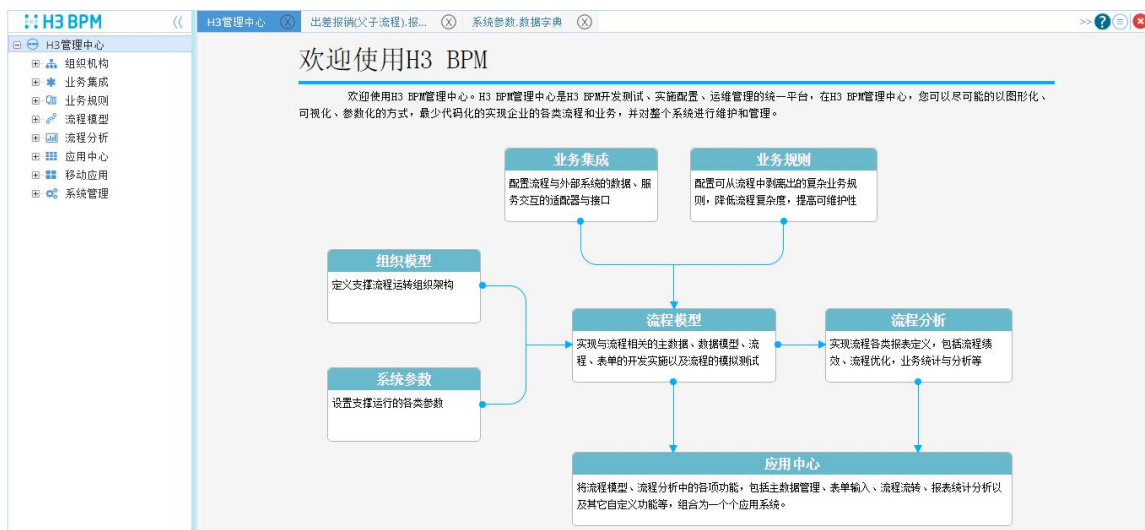
理等功能，本文档中将以一个常见的流程案例做为指引，介绍 H3 BPM 流程创建和运行的过程。



输入用户名和密码，点击登录，进入后台管理；

初始用户名：Administrator，初始密码：000000；

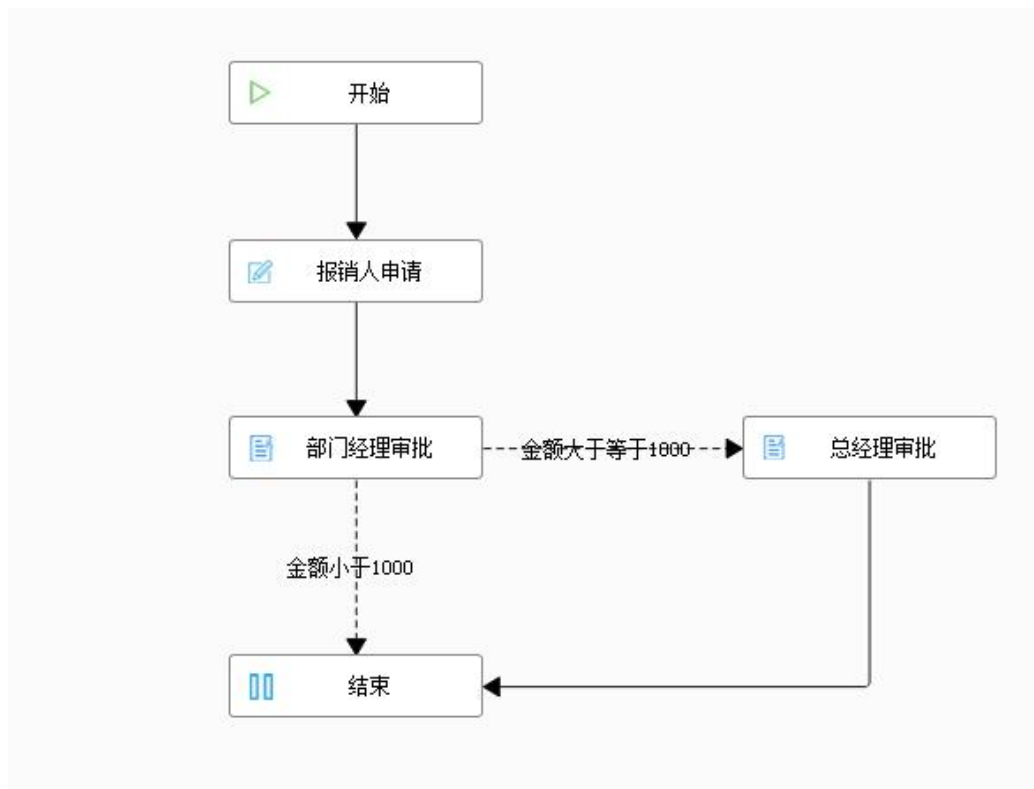
登录后，显示界面：



4 新建流程

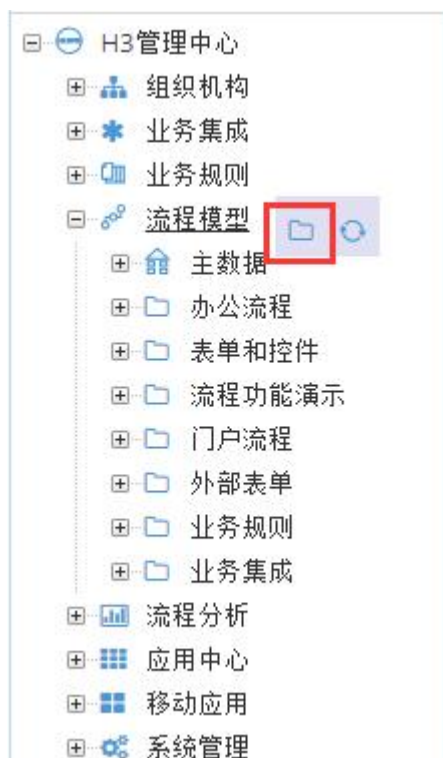
4.1 需求概述

报销人填写申请单，当申请的金额小于 1000，部门经理直接审批即可流程结束；若申请的金额大于等于 1000，则需要部门经理和总经理审批，流程方可结束。

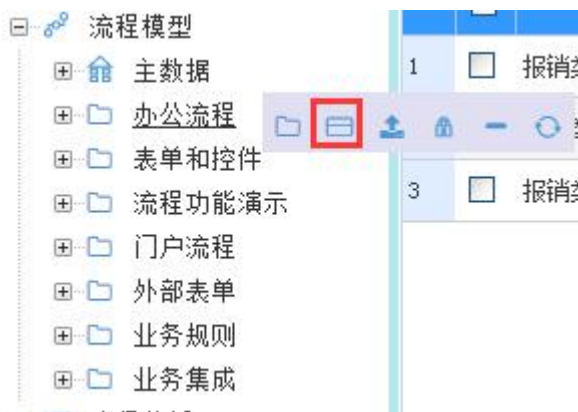


4.2 新建流程包

登录 H3 管理中心，点击左侧菜单：【流程模型】，点击新建流程目录图标



输入：【演示流程】后点击保存，系统将会添加名称为演示流程的流程目录，在该目录的右键菜单中，点击添加流程包的按钮



系统将弹出添加流程包的界面，输入流程包编码和显示名称

新建流程包

保存

所属流程目录: 办公流程

编码:

显示名称:

类型: ☒ 本地存储 ☐ 外部数据源

是否共享: ☐

引用共享包: ☐

排序键: 1

4.3 数据建模

数据建模是定义表单的数据对象，例如报销单中包含：主题、出差事由、报销金额、附件等业务数据，这些数据定义在数据模型中。

选择展开【报销流程】--【数据模型】点击【添加】，弹出数据项页面，添加完成后，点击发布

按钮，进行发布数据模型（注：只有发布后的数据模型才允许运行），发布完成后，已发布框为绿色选中状态。



4.4 设计表单

H3 BPM 提供了系统的默认表单，当默认表单不满足我们的实际需求时，我们可以使用表单设计器对表单控件、样式以及逻辑进行重新定义。

以报销类型为例，引用系统参数的数据字典做为下拉框选项。

- 1、先定义数据字典，打开【系统参数】——【数据字典】，点击添加报销类型，如下图所示：

+ 添加 - 删除 导入						
类别		报销类型				
	类别	编码	显示文本	排序号	默认值	编辑
1	<input type="checkbox"/> 报销类型	001	车费	1	<input type="checkbox"/>	编辑
2	<input type="checkbox"/> 报销类型	002	住宿费	1	<input type="checkbox"/>	编辑
3	<input type="checkbox"/> 报销类型	003	的士费	1	<input type="checkbox"/>	编辑

- 2、流程表单中设置控件，在【报销流程】——【默认表单】，修改报销类型的控件，弹出选择控件界面，如图：

选择控件

控件ID: ct1585605

数据项名称: [Title]主题

数据项类型: 短文本

控件类型: SheetLabel

确定 取消

3、选择控件类型为: SheetDropDownList, 点击确定。

4、选择报销类型的 SheetDropDownList, 右侧属性 MasterDataCategory 设置为数据字典的: 报销类型, 如图:

出差报销(父子流程)

保存 预览 还原

出差报销(父子流程)

基本信息

发起人	Originator.UserName	发起时间	OriginatedDate
所属组织	Originator.OUName	流水号	SequenceNo

表单信息

报销类型: [SheetDropDownList]

出差事由: [Text Area]

附件: 未选择任何文件浏览...

陪同人员: [Text Area]

行程: TravelItem属性

序号	出发地	目的地	离开日期	开始日期	交通工具	删除

属性

样式	ControlSelected
高度	%
宽度	%
是否可见	<input checked="" type="checkbox"/>
控件属性	
DataField	Title
DefaultItems	
DisplayEmptyItem	false
EmptyItemText	
DisplayRule	
MasterDataCatego	报销类型
Queryable	true
SchemuCode	
QueryCode	
Filter	
DataValueField	
DataTextField	

MasterDataCategory: 获取或设置绑定数据字典的类型

5、点击保存, 若流程已发布, 点击预览, 该报销类型可以读取到数据字典中数据, 如图:

报销类型	出差补助 ▼
	车费
	住宿
	出差补助

4.5 流程建模

1) 定义流程活动

报销流程的流程活动有：开始，报销人申请，部门经理审批，总经理审批，结束活动

2) 设计流程图

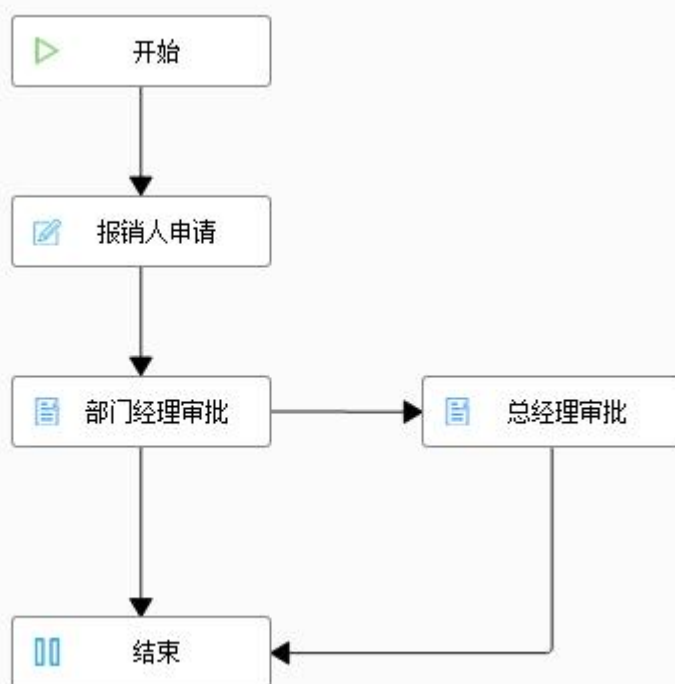
在流程图的头部，选中“审批”按钮拖拽到流程设计板中。

在流程设计板中，选中流程活动手工，右键选择属性，属性中的显示名称修改为：报销人申请，如图：

一般设置	
编码	Activity2
显示名称	手工
排序码	0
前置条件	<input checked="" type="radio"/> 任一 <input type="radio"/> 全部
任务表单	默认表单[Sheet_baoxiaoliuche ▼]
摘要	

同理：选中活动审批，右键属性，属性的显示名称分别修改为部门经理审批，总经理审批。

如图：



3) 活动路径规则设计

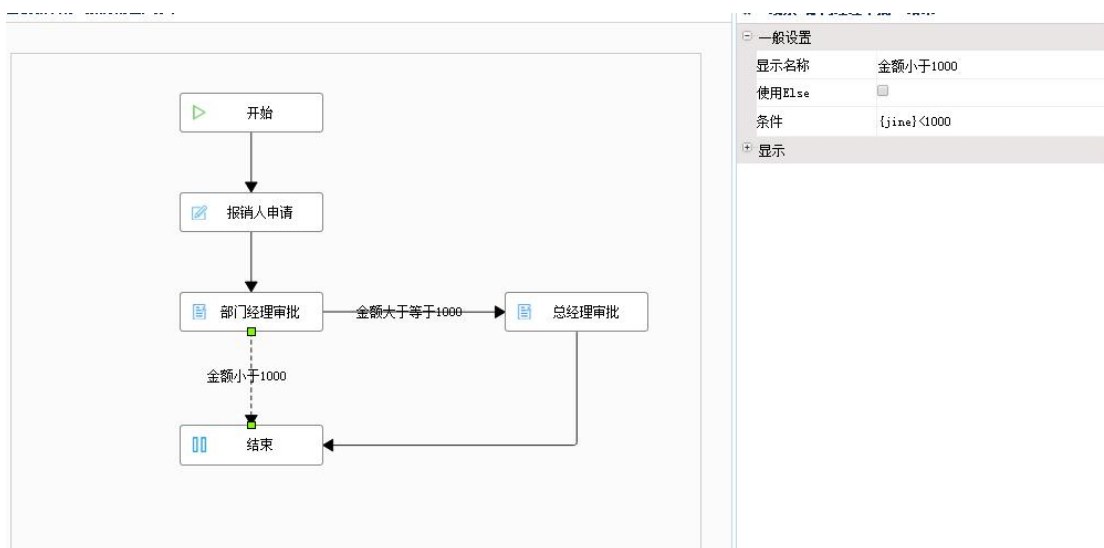
选中活动部门经理审批到结束的路径线条，右键属性，填写线条的名称为：金额小于 1000，如图：

» 线条:部门经理审批->结束	
- 一般设置	
显示名称	金额小于1000
使用Else	<input type="checkbox"/>
条件	
+ 显示	

属性中，设计条件，点击条件，弹出编辑规则的界面，填写规则，如下：

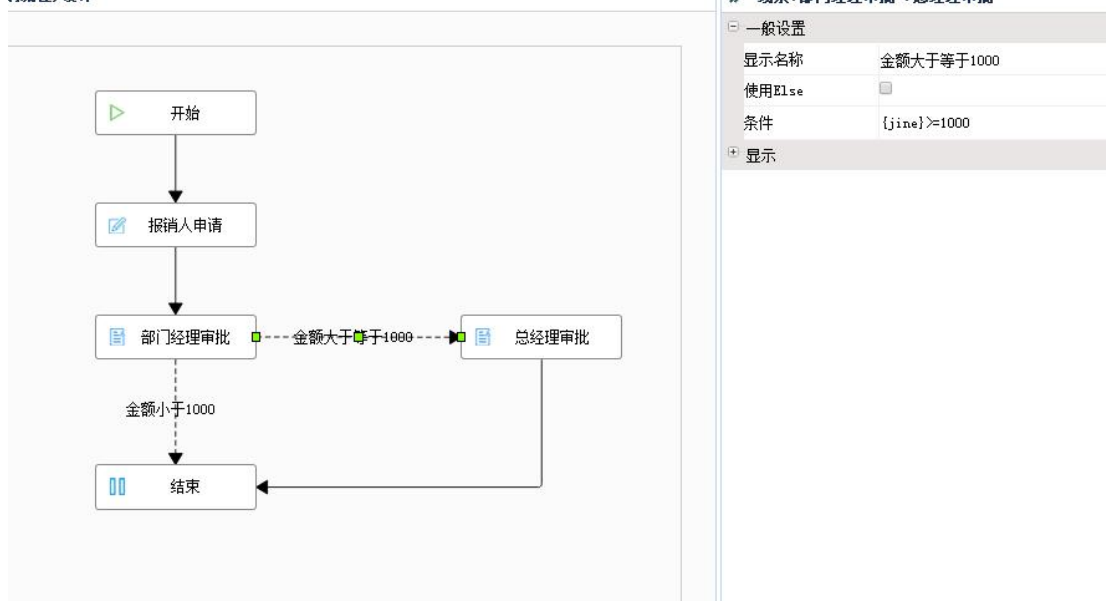


点击确定，设计路径条件成功，如图：



同理，活动部门经理审批到总经理审批的路径线条，右键属性，填写显示名称为：金额大于等于 1000，条件设计为 $jine \geq 1000$ ，如图：

流程, 设计



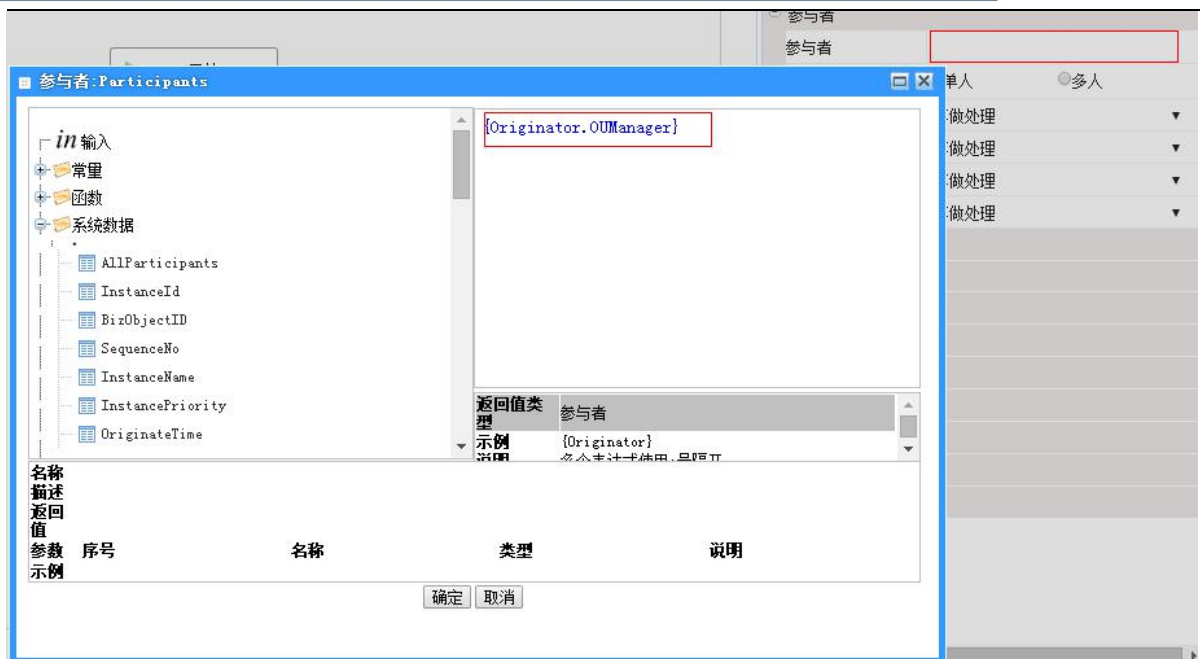
4) 设置活动节点属性的操作

a) 选中活动报销人申请, 右键属性, 数据权限中, 数据项: 报销时间、报销类型、金额、发票附件, 勾选可写/必填, 点击保存



b) 选中活动部门经理审批, 右键属性, 右侧展开参与者项, 点击参与者控件, 弹出的编辑公式界面

输入参与者为: {Originator.OManager}, 如图:

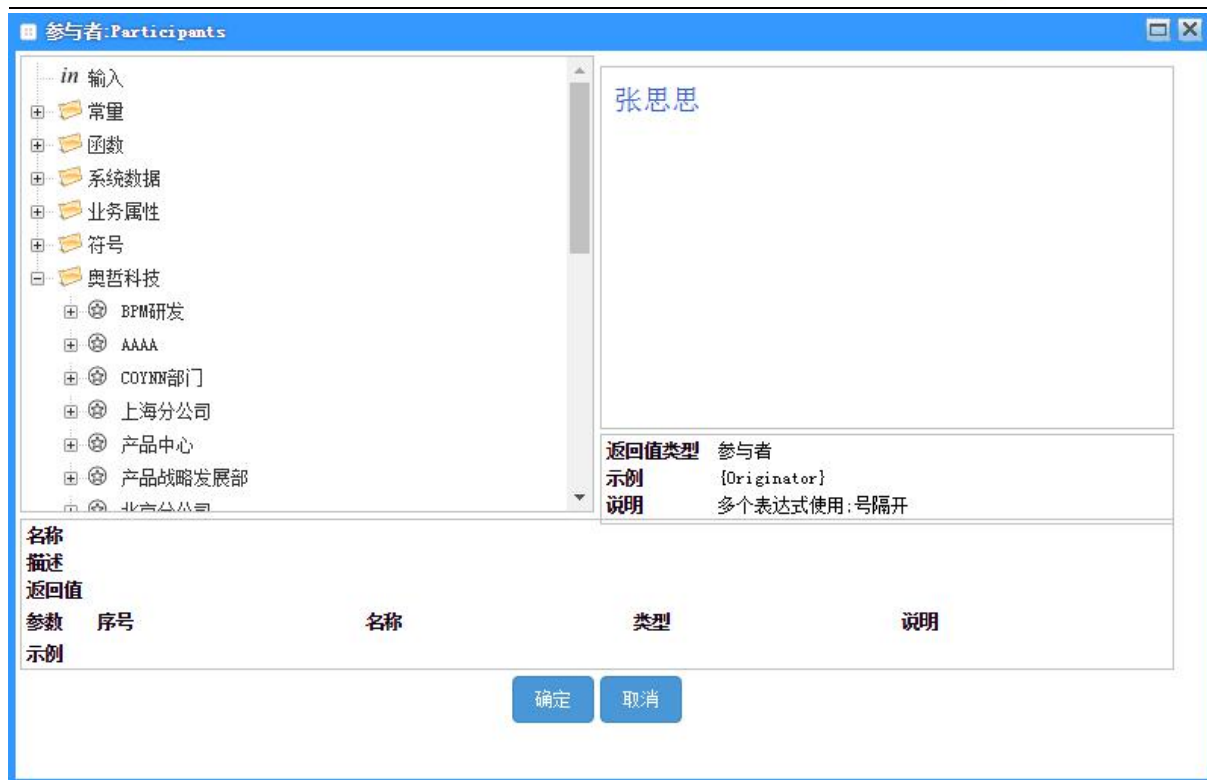


点击确定，设置成功。

选中数据权限，设置部门经理审批数据项，勾选可写/必填，如图：



C) 选中活动总经理审批，右键属性，右侧展开参与者，点击参与者，设置参与者为总经理：如张思思；如图：



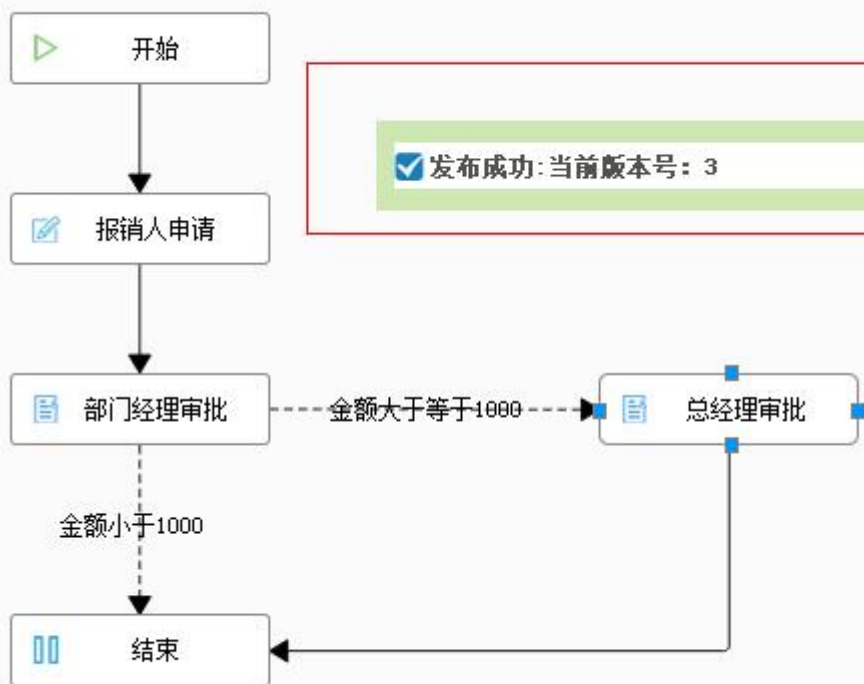
点击确定，设置成功，如图：



数据权限，数据项总经理审批，勾选可写/必填：



在流程设计器的头部，点击保存，流程活动设计保存成功；在流程设计器的头部，点击发布，流程发布成功：



上图，说明流程发布成功，可以在流程模拟运行或登录门户发起流程。

5 流程模拟

展开【H3 管理中心】--【流程模型】--【流程模拟测试验证】--【流程模拟】

5.1 新增用例

1) 点击新增用例，弹出新建模拟的界面：

新建模拟

保存

用例信息

模版名称 字段类型展示

用例名称

发起人

预设

	数据项名称	数据类型	预设值	设置
1	Title	短文本		设置
2	LTX	逻辑型		设置
3	IntType	整数		设置
4	ShortText	短文本		设置
5	LongText	长文本		设置

活动名称 预设值 是否可忽略 设置

2) 数据项设置预设值

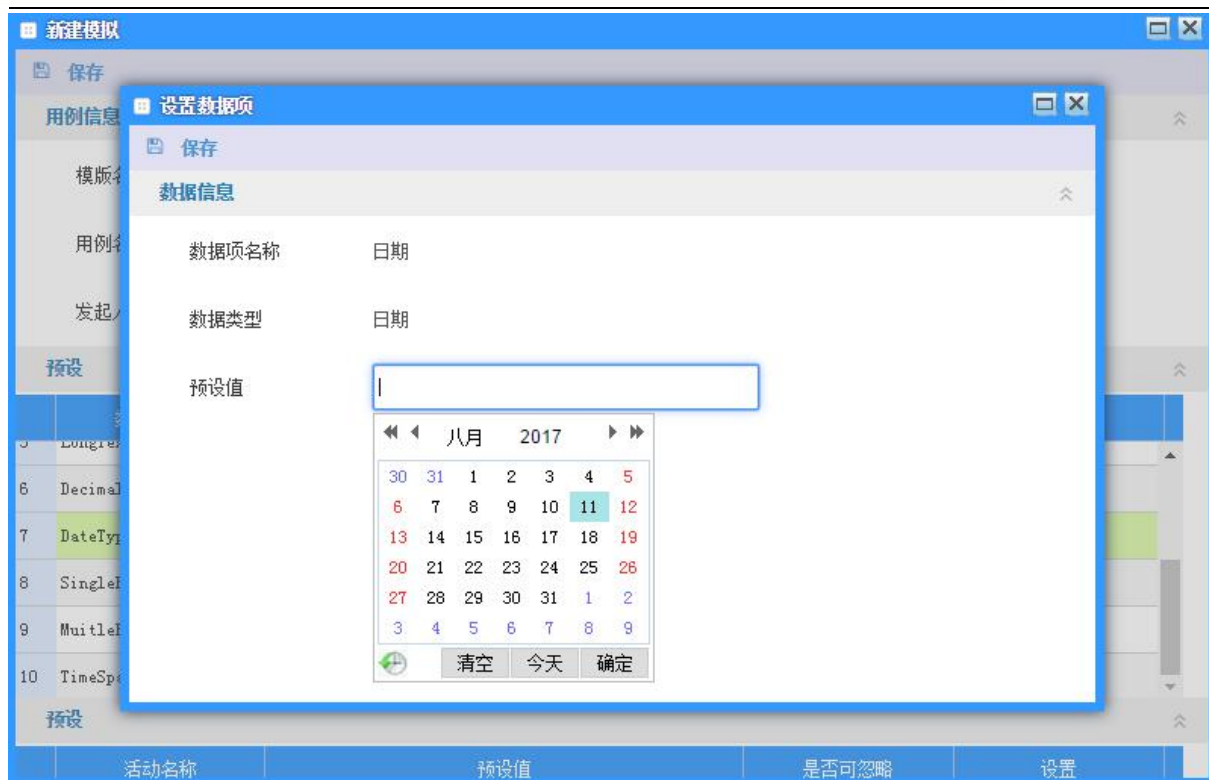
a) 数据项 Title，点击设置：



点击确定，设置成功。



b) 数据项 DateTpye 预设值，点击设置，预设值为：日期。



c) 数据项 DecimalType 预设值，点击设置，预设值为 1000。



d) 数据项 SingleParticipant 单人选人控件预设值，点击设置，预设值可设为“总经理”。



e) 数据项 MuiltParticipant 多人选人控件预设值，点击设置，预设值可设为“总经理”、“职能部门”。



5) 在新建用例的界面，点击保存，用例添加成功，如图：

测试用例		测试用例集			
+ 新增用例		▷ 运行		- 删除	
		+ 导入		○ 刷新	
<input type="checkbox"/>	案例名称	发起人	状态	运行时间	编辑
1 <input type="checkbox"/>	流程模拟测试验证	总经理	未运行		编辑

5.2 用例运行

选中用例，点击运行：

测试用例		测试用例集			
+ 新增用例		▷ 运行		- 删除	
		+ 导入		○ 刷新	
<input type="checkbox"/>	案例名称	发起人	状态	运行时间	编辑
1 <input type="checkbox"/>	xxxxx	系统管理员	查看	2017/6/23 15:57:17	编辑

状态中，点击运行中，查看运行流程的状态：

测试用例		测试用例集			
+ 新增用例		▷ 运行		- 删除	
		+ 导入		○ 刷新	
<input type="checkbox"/>	案例名称	发起人	状态	运行时间	编辑
1 <input type="checkbox"/>	流程模拟测试验证	总经理	查看	2017/8/11 10:56:44	编辑

运行历史: 流程模拟测试验证

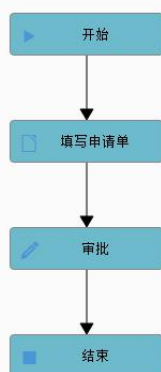
所选用例

用例名称 流程模拟测试验证

用例的所有实例情况

	流程名称	发起人	预设数据项	状态	查看
1	流程模拟测试验证	总经理	Title: 标题1; 标题2; MultiParticipant: 职能部门;	已结束	查看

点击查看详情，可以查看流程的状态：



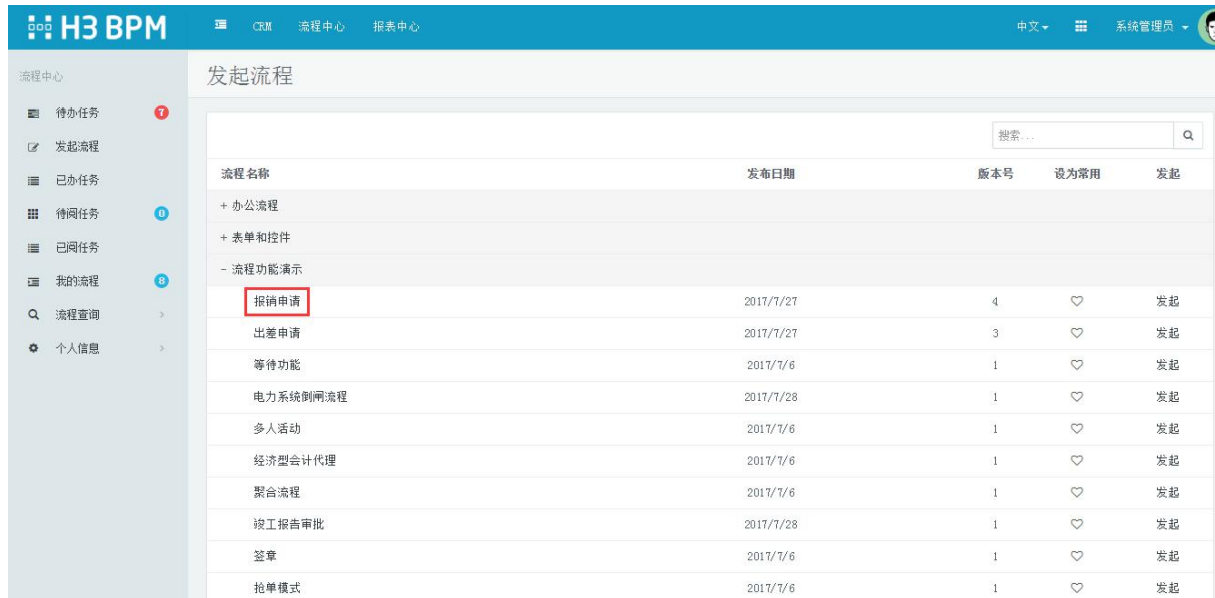
日志

序号	活动名称	接收时间	完成时间	使用时间	参与者	操作
1	填写申请单	2017-08-11 10:56:45	2017-08-11 10:56:48	3秒	总经理	通过
2	审批	2017-08-11 10:56:48	2017-08-11 10:56:51	2秒	总经理	通过
3	结束	2017-08-11 10:56:51	2017-08-11 10:56:51	-		

6 发起流程

进入首页，点击【流程中心】，进入流程中心的页面，默认打开待办任务。

点击【发起流程】项，可以查看到报销流程：



选择报销流程，点击【发起】，页面跳转到流程表单如下图：

出差报销

提交 保存 流程状态 关闭

➤ 基本信息

发起人: 系统管理员 发起时间: _____

所属组织: 我的公司 流水号: _____

➤ 表单信息

主题: _____

出差事由:

附件:

点击上传或拖拽文件到此上传

随同人员: _____

点击【提交】，流程表单提交成功，并表单关闭回到发起流程的页面。

过滤条件【进行中】可以查看到流程的当前处理人，如下图：

流程中心 待办任务 8 发起流程 已办任务 待阅任务 1 已阅任务 我的流程 9 流程查询 个人信息	我的流程				
	流程名称	流程模板	进行中 ▼	开始时间	至 结束时间
	张三的报销流程	报销申请		2017-08-15 14:55	
	系统管理员的请假申请	请假流程		2017-08-11 11:05	
	3	电力系统倒闸流程		2017-07-28 17:07	
	3434	报销申请		2017-07-27 11:22	
	1212	报销申请		2017-07-27 11:21	
	字段类型展示. 3	模板子表		2017-07-25 17:23	

当前处理人登录门户系统，首页可以查看到该报销流程，并可以进行审批。

审批完，流程结束，在【已完成】中可以查看到报销流程的后续处理状态。