入口函数：

## ShopWorker.cpp

#### WebServiceEntryProcess()

调用CShopWorker::Initialize() 以及 CShopWorker::run()

#### CShopWorker::Initialize()

初始化过程

m\_oConfiguration 初始化

C2C\_WWLOG->QuickInitForFastCGI() 日志初始化

CCGI\_STUB\_CNTL->Initialize()

CONFIG\_SET->Initialize()

REGION\_INFO->Initialize()

m\_oViewManager.Initialize() 初始化

CMEM\_OPT->Initialize() 消息组件初始化

m\_strPPFileContent 成员变量初始化

#### CShopWorker::run()

CShopWorker::CheckBadRequest() 检查连接的合法性

文件名为空？？？？？？？ 出现情况

CShopWorker::GetStaticHtmlFile（）

文件名不为空????????? Search shop这些情况 index 这些情况

m\_oUrlParser.Parser() 解析，店铺搜索，店铺分类以及 一部分过滤条件

（返回值： dPageType CUrlParser::CUrlParser() 匹配返回返回值）

CShopWorker::GetHost() 返回是网购店铺还是拍拍店铺

CShopWorker::CheckMobileRedirect() 手机refer 检查如果是手机访问，直接返回失败界面？？？？

URL\_SHOP\_COMMON

有二种读取数据的方式

m\_oConfiguration.IsInPageCacheGray(oRequestFilter.GetUin())

灰度卖家，从cache中取数据，如果cache失败，GetPage去页面数据；

非灰度卖家，GetPage获取数据；CShopWorker::GetPage（）

#### CUrlParser::Parser()

oRequestFilter.SetPatternId(URL\_ITEM);

oRequestFilter.SetPageType(HTML\_PAGE);

oRequestFilter.SetSearchType(ITEM);

oRequestFilter.SetPage(SHOP\_ITEM\_PAGE);

oRequestFilter.SetItemId(SubString(strUrl,2));

oRequestFilter.SetUin(CCommodityIdTool::GetSellerUinFromCommodityId(oRequestFilter.GetItemId()));

break;

CRequestFilter& oRequestFilter 过滤填充的过程；

sss

店铺首页填入的数据

case URL\_SHOP:

oRequestFilter.SetPatternId(URL\_SHOP);

oRequestFilter.SetPageType(HTML\_PAGE);

oRequestFilter.SetSearchType(SHOP);

oRequestFilter.SetPage(SHOP\_HOME\_PAGE);

oRequestFilter.SetUin(strtoull(SubString(strUrl,1).c\_str(),NULL,10));

break;

#### CShopWorker::GetPage（）

if(m\_oConfiguration.GetFuncProperty() & FUNC\_PROPERTY\_MQ) 判断是否发送消息（发送MQ消息）

MQ消息 部门消息吗？？？？？？

// begin: 支持商详、店祥页面内部链接保持与URL一致 by yokozhang @ 2013-06-23

uint32\_t dwDomain = GetDomainByUrl(pRequest->GetUrl());

oRequestFilter.SetDomain(dwDomain); 保持一致性用途？？？？？？？

m\_oViewManager.PageBuilder() 创建页面

iRetCode=m\_oViewManager.PageBuilder() 返回值 登陆页面，首页等；

## ViewManager.cpp

#### m\_oViewManager.Initialize()

shopWorker init 初始化函数会调用

#define BASE\_CONFIG\_FILE "../conf/Config4.xml"

#define MVM\_CONFIG\_FILE "ModuleViewMapping.xml"

#define VTM\_CONFIG\_FILE "ViewTemplateMapping.xml"

#define SVM\_CONFIG\_FILE "SearchTypeViewMapping.xml"

#define SMM\_CONFIG\_FILE "SearchTypeModuleMapping.xml"

#define MODULE\_CONFIG\_FILE "../conf/ShopModuleConfig.xml"

#define AMACM\_CONFIG\_FILE "AuctionMainAreaControlMapping.xml"

#define AMASM\_CONFIG\_FILE "AuctionMainAreaStuffMappinp.xml"

uint32\_t dwRoleNum=m\_oConfiguration.GetRoleConfig().size(); 得出dwRoleNum 等于2

配置文件<Item Name="RoleConfig" Value="../conf/base/|../conf/mall/"/>

**../conf/mall/目录 以及../conf/base 目录**

m\_vecMVMManager 对应于 #define MVM\_CONFIG\_FILE "ModuleViewMapping.xml"

m\_vecVTMManager 对应于 #define VTM\_CONFIG\_FILE "ViewTemplateMapping.xml"

m\_vecSVMManager 对应于 #define SVM\_CONFIG\_FILE "SearchTypeViewMapping.xml"

m\_vecSMMManager 对应于 #define SMM\_CONFIG\_FILE "SearchTypeModuleMapping.xml"

**../conf/base 目录**

m\_oCAMACMManager 对应于#define AMACM\_CONFIG\_FILE "AuctionMainAreaControlMapping.xml"

m\_oCAMASMManager 对应于#define AMASM\_CONFIG\_FILE "AuctionMainAreaStuffMappinp.xml"

CViewManager::Initialize() 初始化二次

#### CMVMManager::Initialize()

**其初始化的过程**

TotalConfigNum 获取到个数

取配置文件Config

CStringHelper::SplitStringIntoVector(vecMap[1].c\_str(),":",vecTerm);

for(uint32\_t dwPos=0;dwPos<MVM\_TERM\_NUM;dwPos++)

{

获取中间的数值；即 ddwKey

通过移位 \*1024 （十进制数值）

dwValue=strtoull(vecMap[0].c\_str(),NULL,10); 字符串 转换成数字；

#### CViewManager::PageBuilder()

/\*

<Item Name="TotalConfigNum" Value="56"/>

<Item Name="Config0" Value="0|10|html"/>

<Item Name="Config1" Value="0:0|11|html"/>

<Item Name="Config2" Value="0:0:0:1023:1:0|12|html"/>

PageType:FrameID:AreaID:ModuleFrameID:ModuleID:ModuleStyle|ViewID|Type

\*/

Config 配置文件获取html等页面配置；

1.调用CDataManager oDataManager()初始化

2. // page no need layout, build with one viewId

if(IsWithoutLayoutPage(oRequestFilter.GetPage()))

{

const\_cast<uint32\_t&>(dwViewID) = m\_vecMVMManager[0].GetPageView(oRequestFilter.GetPage(), dwPageType);

}

3. oShopInfoDaoStub4Web.GetModuleType()

uint32\_t dwRet = oShopInfoDaoStub4Web.GetModuleType

(oCntlInfo, strMachineKey,strSource,dwUin, strInReserve,dwModuleType,

strErrmsg,strOutReserve);

1. 三种根据条件选择一种生成页面；

根据 url parse 获取获取到的信息；

CViewManager::ViewPageMaker ()

dwPageType!=HTML\_PAGE||dwViewID!=0||dwTssID!=0

URL\_SHOP\_VIEW oRequestFilter.SetViewId(dwViewID);

URL\_SHOP\_DESCRIPTION oRequestFilter.SetTssId(strtoull(SubString(strUrl,3).c\_str(),NULL,10));

URL\_SHOP\_COMMON

CViewManager::ModulePageMaker()

else if(ddwSequenceID!=0)

URL\_SHOP\_MODULE oRequestFilter.SetSequenceId(strtoull(SubString(strUrl,2).c\_str(),NULL,10));

其他条件都是这个页面生成的；

CViewManager::WholePageMaker()

URL\_SHOPDYNAMIC

#### CViewManager::WholePageMaker()

iRetCode=oDataManager.Run(); 一共有二次

第一次拉取//取用户/布局/店铺信息； 第二次拉取商品数据；

1.首先获取部分参数

获取用户店铺id 以及

const uint32\_t& dwUin =oRequestFilter.GetUin();

const uint32\_t& dwPageID =oRequestFilter.GetPageId();

const uint32\_t& dwPatternID =oRequestFilter.GetPatternId();

const uint32\_t& dwPageType =oRequestFilter.GetPageType();

const std::string& strItemId =oRequestFilter.GetItemId();

c2cent::po::shop::CDetailFilter& oDetailFilter=oDataManager.GetFilter();

c2cent::po::shop::CDetailPo& oDetailPo=oDataManager.GetPo();

2. int32\_t CViewManager::CheckJumpLimit() 检查跳转？？？？？

3. //取用户/布局/店铺信息

c2cent::po::shop::CDetailFilter& oDetailFilter=oDataManager.GetFilter();

m\_oDetailFilter 参数的填充

oDetailFilter.SetCommand(m\_oConfiguration.GetPrimWholePageCommand());

oDetailFilter.SetPage(oRequestFilter.GetPage());

oDetailFilter.SetPageId(dwPageID);

oDetailFilter.SetUin(dwUin);

if(!strItemId.empty())

{

oDetailFilter.SetItemId(strItemId);

}

oDetailFilter.SetScene(ConvRedPageViewId(oRequestFilter.GetViewId()));

iRetCode=oDataManager.Run(); //通过page pageid通过过滤从ao dao中获取需要的数据

4

int32\_t PrimCheckHost()

//检查主要的跳转逻辑

if(pPrimCheckHostFun!=NULL)

{

iRetCode=(\*pPrimCheckHostFun)(m\_oConfiguration,oDetailFilter,oDetailPo,iRetCode,oRequestFilter,dwHost);

if(iRetCode!=0)

{

if(iRetCode==RETCODE\_TOVIEWRENDER)

{

ViewPageMaker(oDataManager,oRequestFilter,strContent,dwHost,pPrimCheckHostFun,pOtherCheckHostFun,oDetailFilter.GetCommand());

return RETCODE\_NOTFOUND405;

}

else

{

C2C\_WW\_LOG\_ERR(iRetCode,"PrimCheckHost Error PatternID:%u Uin:%u CommodityID:%s dwHost:%u",dwPatternID,dwUin,strItemId.c\_str(),dwHost);

return iRetCode;

}

}

}

5．//填写商品列表过滤器

FillFilter(oDetailPo,oDetailFilter.GetFilterList());

//构造搜索页布局

iRetCode=FillSearchModule(oRequestFilter,oDataManager);

if(iRetCode!=0)

{

m\_oConfiguration.m\_oAuctionPageGroup.RequstReport(TWS\_AUCTION\_404\_SEARCHMODULE\_FAIL);

iRetCode=RETCODE\_NOTFOUND;

C2C\_WW\_LOG\_ERR(iRetCode,"FillSearchModule Failed PatternID:%u Uin:%u CommodityID:%s",dwPatternID,dwUin,strItemId.c\_str());

return iRetCode;

}

6.打印配置文件的店铺基本信息

if(m\_oConfiguration.Dump())

{

std::stringstream ssDump;

C2C\_WW\_LOG\_DEBUG("Dump Module PatternID:%u Uin:%u CommodityID:%s ShopModule:%s",

dwPatternID,dwUin,strItemId.c\_str(),PassiveDump(ssDump,vecLayout).str().c\_str());

ssDump.str("");

C2C\_WW\_LOG\_DEBUG("Dump Filter PatternID:%u Uin:%u CommodityID:%s Filter:%s",

dwPatternID,dwUin,strItemId.c\_str(),oDetailFilter.DumpDataU(ssDump).str().c\_str());

}

7. //合并请求命令； 获取页面信息；

uint32\_t dwRole=vecLayout[0].GetRole();

std::set<uint32\_t> setViewList;

iRetCode=m\_vecMVMManager[dwRole%m\_vecMVMManager.size()].GetViewList(vecLayout,oRequestFilter,dwPageType,setViewList);

if(iRetCode!=0)

{

C2C\_WW\_LOG\_ERR(iRetCode,"GetViewList Error PatternID:%u Uin:%u CommodityID:%s ViewListSize:%lu",dwPatternID,dwUin,strItemId.c\_str(),setViewList.size());

}

uint32\_t dwCarrageID=oDetailPo.GetItem().GetTransportPriceType();

if(dwCarrageID>=MIN\_TEMPLATE\_INNER\_ID)

{

oDetailFilter.SetFreightId(dwCarrageID);

}

8.//剔除已请求的命令

oDetailFilter.SetCommand(m\_vecVTMManager[dwRole%m\_vecVTMManager.size()].GetTotalCommand(setViewList)&(~oDetailFilter.GetCommand()));

oDetailFilter.SetScene(ConvRedPageViewId(oRequestFilter.GetViewId()));

9.//获取数据

iRetCode=oDataManager.Run();

10. //检查额外的跳转逻辑

if(pOtherCheckHostFun!=NULL)

{

iRetCode=(\*pOtherCheckHostFun)(m\_oConfiguration,oDetailFilter,oDetailPo,iRetCode,oRequestFilter,dwHost);

if(iRetCode!=0)

{

C2C\_WW\_LOG\_ERR(iRetCode,"OtherCheckHost Error PatternID:%u Uin:%u CommodityID:%s dwHost:%u",dwPatternID,dwUin,strItemId.c\_str(),dwHost);

return iRetCode;

}

}

11 // 套餐详情获取失败 返回404页面

if(oDetailFilter.GetPage() == SHOP\_ITEMCOMBO\_PAGE)

{

if(oDetailFilter.HasCommand(DETAIL\_ITEMCOMBO\_KEY) &&

(!oDetailPo.HasResult(DETAIL\_ITEMCOMBO\_KEY) || !oDetailPo.HasResult(DETAIL\_ITEM\_KEY)))

{

m\_oConfiguration.m\_oAuctionPageGroup.RequstReport(TWS\_AUCTION\_404\_COMBOITEM\_FAIL);

iRetCode = RETCODE\_NOTFOUND;

return iRetCode;

}

}

C2C\_TIME\_ELF\_ITIL oRenderWTimeElf(C2C\_WWLOG,"Render",&iRetCode,&m\_oConfiguration.m\_oRenderGroup);

全部使用拍拍模板，网购模板停止使用 ？？？？？？？？？

//20140514

C2C\_WW\_LOG\_DEBUG("g\_iModuleType=%d",m\_iModuleType);

if( this->m\_iModuleType < 0 )

{

if( 0 == dwRole )

this->m\_iModuleType=TEMPLATE\_PAIPAI;

else

this->m\_iModuleType=TEMPLATE\_WANGGOU;

}

**12． //渲染**

CRender oRender(oRequestFilter,oDataManager,m\_vecMVMManager,m\_vecVTMManager,m\_oCAMACMManager,m\_oCAMASMManager,m\_oTemplate,m\_oConfiguration.GetHost(), m\_strTaSid, m\_mapTuanID,this->m\_iModuleType);

uint32\_t dwIndex=0;

iRetCode=oRender.Render(LEVEL\_PAGE,strContent,dwIndex);

if(iRetCode!=0)

{

iRetCode=RETCODE\_NOTFOUND;

C2C\_WW\_LOG\_ERR(iRetCode,"Render Failed PatternID:%u Uin:%u CommodityID:%s dwHost:%u",dwPatternID,dwUin,strItemId.c\_str(),dwHost);

}

#### int32\_t CViewManager::FillFilter()

最开始oDetailPo 来源（datamanage里面来的） m\_oDetailFilter,

if(m\_oConfiguration.GetRole()==ROLE\_SHOP)

{

iRetCode=m\_oStub.GetShopDetailInfo(

m\_oCntlInfo,

m\_strVisitKey,

\_\_FILE\_\_,

m\_oDetailFilter,

strInReserve,

oDetailPo,

strOutReserve);

}

else

{

iRetCode=m\_oStub.GetItemDetailInfo(

m\_oCntlInfo,

m\_strVisitKey,

\_\_FILE\_\_,

m\_oDetailFilter,

strInReserve,

oDetailPo,

strOutReserve);

}

1.获取布局；segment frame； 整个页面的布局信息；

/\*\*

\* 获取布局信息

\*

\* @return vecLayoutList value 类型为:std::vector<c2cent::po::shop::CDetailLayout>

\*

\*/

const std::vector<c2cent::po::shop::CDetailLayout> & GetLayoutList() const {

return vecLayoutList;

}

const std::vector<c2cent::po::shop::CDetailLayout>& vecLayout = oDetailPo.GetLayoutList();

uint32\_t dwSize=vecLayout.size();



2. 填充过滤map mapFilterMap

c2cent::po::shop::CDetailItemFilter oItemFilter;

oItemFilter.SetFilterMapVal(ITEM\_FILTER\_SHOPCATEGORY,getcategory(strCategoryID));

3. 填充vecFilter

int32\_t CViewManager::FillFilter(

const c2cent::po::shop::CDetailPo& oDetailPo,

std::vector<c2cent::po::shop::CDetailItemFilter>& vecFilter)

vecFilter.push\_back(oItemFilter);

4. 获取商品信息

const c2cent::po::shop::CDetailItem& oItem = oDetailPo.GetItem();

/\*\*

\* 获取商品信息

\*

\* @return oItem value 类型为:c2cent::po::shop::CDetailItem

\*

\*/

const c2cent::po::shop::CDetailItem& GetItem() const {

return oItem;

}

//架下商品:

//1、快照

//2、售完

//3、用户下架,发布到仓库中

//4、BOSS下架

//5、历史

// 下架商品优化 增加获取同品类商品列表Filter ， Sessionid 置为0;

c2cent::po::shop::CDetailItemFilter oItemFilter;

oItemFilter.SetSellerUin(oItem.GetQqUin());

oItemFilter.SetOrderType(C2C\_WW\_COMDY\_FIND\_ORDERY\_BY\_SOLD\_DESC);

oItemFilter.SetStartIndex(0);

oItemFilter.SetPageSize(SHOP\_ITEMDESC\_DOWNITEMLIST\_CNT);

oItemFilter.SetFilterMapVal(ITEM\_FILTER\_STATE,ITEM\_FILTER\_STATE\_SELLING);

oItemFilter.SetFilterMapVal(ITEM\_FILTER\_SESSION\_ID,ITEM\_FILTER\_MAPVAL\_DEFAULT\_SESSION\_ID);

char sLeafClassId[16] = {0};

snprintf(sLeafClassId,sizeof(sLeafClassId),"%u",oItem.GetLeafClassId());

oItemFilter.SetFilterMapVal(ITEM\_FILTER\_LEAFCLASSID,sLeafClassId);

vecFilter.push\_back(oItemFilter);

#### CViewManager::FillSearchModule（）

1.搜索商品列表参数是这几个，返回的是搜索结果

// 搜索商品列表页

if(SHOP\_CATEGORY == dwSearchType || SHOP\_RECOMMEND == dwSearchType || SHOP\_SEARCH == dwSearchType)

{

2. 返回的是店铺介绍页面，以及新手页面等

CSMMManager& oSMMManager=m\_vecSMMManager[dwRole%m\_vecSMMManager.size()];

填充页面通过段，框架等页面；

c2cent::po::shop::CDetailLayout oLayout;

oLayout.SetRole(dwRole);

oLayout.SetPage(dwPage);

oLayout.SetSegmentSequence(vecLayout[vecLayout.size()-1].GetSegmentSequence());

oLayout.SetSegmentId(vecLayout[vecLayout.size()-1].GetSegmentId());

oLayout.SetSegmentNo(vecLayout[vecLayout.size()-1].GetSegmentNo());

oLayout.SetFrameId(oSMMManager.GetFrameID(oRequestFilter.GetSearchType()));

oLayout.SetAreaNo(oSMMManager.GetAreaID(oRequestFilter.GetSearchType()));

oLayout.SetModuleFrame(oSMMManager.GetModuleFrameID(oRequestFilter.GetSearchType()));

oLayout.SetModuleId(oSMMManager.GetModuleID(oRequestFilter.GetSearchType()));

oLayout.SetModuleStyle(oRequestFilter.GetDisStyle());

#### int32\_t CMVMManager::GetViewList()

**setViewList 1000 这种数据填入到set中**

**模块化的主要元素**：页page、框frame、区域area、段segment、模块module、模块外框moduleframe。

int32\_t CMVMManager::GetViewList(

const std::vector<c2cent::po::shop::CDetailLayout>& vecLayoutList,

const CRequestFilter& oRequestFilter,

const uint32\_t dwPageType,

std::set<uint32\_t>& setViewList) const

{}

步骤：

三个函数GetModuleView（）模块 GetFrameView（）框 GetPageView（） 页面

都是从m\_mapModuleView 中map中取出



1. 调用 GetModuleView（）

<Item Name="Config0" Value="1000|0|html"/> <!-- 店招页[保留] -->

移位获取到页面

int32\_t CMVMManager::GetModuleView(

const uint32\_t dwFrameID,

const uint32\_t dwPageType,

const uint32\_t dwAreaID,

const uint32\_t dwModuleID,

const uint32\_t dwModuleStyle) const

{

if(dwFrameID>=IGNORE\_ID||dwAreaID>IGNORE\_ID||dwModuleID>IGNORE\_ID||dwPageType>XML\_PAGE)

{

C2C\_WW\_LOG\_ERR(-1,"CMVMManager GetModuleView Failed FrameID:%u PageType:%u AreaID:%u ModuleID:%u ModuleStyle:%u",

dwFrameID,dwPageType,dwAreaID,dwModuleID,dwModuleStyle);

return -1;

}

uint64\_t ddwKey=IGNORE\_ID;

ddwKey<<=ID\_BITLENGTH;

ddwKey+=dwFrameID;

ddwKey<<=ID\_BITLENGTH;

ddwKey+=dwAreaID;

ddwKey<<=ID\_BITLENGTH;

ddwKey+=IGNORE\_ID;

ddwKey<<=ID\_BITLENGTH;

ddwKey+=dwModuleID;

ddwKey<<=ID\_BITLENGTH;

ddwKey+=dwModuleStyle;

ddwKey<<=TYPE\_BITLENGTH;

ddwKey+=dwPageType;

std::map<uint64\_t,uint32\_t>::const\_iterator it=m\_mapModuleView.find(ddwKey);

if(it!=m\_mapModuleView.end())

{

return (it->second);

}

C2C\_WW\_LOG\_ERR(-2,"CMVMManager GetModuleView Failed FrameID:%u PageType:%u AreaID:%u ModuleID:%u ModuleStyle:%u",

dwFrameID,dwPageType,dwAreaID,dwModuleID,dwModuleStyle);

return -2;

}

1. 调用GetFrameView（）
2. 调用GetPageView（）

#### int32\_t CDataManager::Run()

1. 获取tssid以及设置tssid

CDataManager::\_SetProtocolInfo()

enum SpecialTssID

{

TSS\_SHOP2DEFINE=100,

TSS\_NEWBUYER=999,

TSS\_SHOPPINGENSURE=998,

TSS\_SHOPINTRO=997,

TSS\_SHOPTITLE=996,

};

## CUrlParse.cpp

/\*\*

\* 获取商品列表过滤器

\*

\* @return oItemFilter value 类型为:c2cent::po::shop::CDetailItemFilter

\*

\*/

const c2cent::po::shop::CDetailItemFilter& GetItemFilter() const {

return oItemFilter;

}s

### enum PatternID

{

URL\_ITEM,

URL\_SHOP,

URL\_ITEM\_SEARCH,

URL\_ITEM\_VIEW,

URL\_SHOP\_COMMON,

URL\_SHOP\_LEAVEMSG,

URL\_SHOP\_VIEW,

URL\_SHOP\_EDIT,

URL\_SHOP\_MODULE,

URL\_SHOP\_DESCRIPTION,

URL\_SHOP\_JS,

URL\_ITEM\_DESCRIPTION,

URL\_SWF\_FILE,

URL\_HTML\_FILE,

URL\_ITEMCOMBO,

URL\_SHOPDYNAMIC,

};

### enum SearchType

{

SHOP,

SHOP\_SEARCH,

SHOP\_ACTION,

SHOP\_CATEGORY,

SHOP\_LEAVEMSG,

SHOP\_RECOMMEND,

SHOP\_DESCRIPTION,

ITEM,

ITEM\_DESCRIPTION,

SHOP\_ANNOUNCE,

SHOP\_FEEDBACK,

};

### enum PageType

{

HTML\_PAGE,

JSON\_PAGE,

XML\_PAGE,

};

#define SHOP\_HOMEPAGE 0

#### CUrlParser::CUrlParser()

url\_shop 以及url\_common 取得值是一致的；

### //店铺首页URL\_SHOP

店铺首页url http://shop.paipai.com/2201189840s

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})(/){0,1}(\\?.\*){0,1}$");

oRequestFilter.SetPatternId(URL\_SHOP); 模式id

oRequestFilter.SetPageType(HTML\_PAGE); 返回页面类型

oRequestFilter.SetSearchType(SHOP); 搜索类型ss

oRequestFilter.SetPage(SHOP\_HOME\_PAGE); 页面类型

oRequestFilter.SetUin(strtoull(SubString(strUrl,1).c\_str(),NULL,10)); 用户uid

### URL\_SHOP\_COMMON

页面？？？？？？

url：

http://shop.paipai.com/103650649/0-0000000000000000000400000000000000000080000000000000000000000000-0-1-1-0-3-0-0-0/index.shtml?PTAG=40042.1.5

所有宝贝：<http://shop.paipai.com/1641624992/0-0000000000-0-1-1-0-3-0-0-0/index.shtml>

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})/([0-9]{1,4})(-([0-9a-fA-F]{1,64})-([0-9]{1,2})-([0-9]{1,2})-([0-9]{1,5})-([0-9]{1,2})-([0-9])-([0-9])-([0-9]{1,3})-([0-9]{1,3})(/(.{1,255})){0,1}){0,1}/index.(shtml|html|js|xml)(\\?.\*){0,1}$");

URL\_SHOP\_LEAVEMSG

URL\_SHOP\_VIEW:

URL\_SHOP\_EDIT:

URL\_SHOP\_MODULE:

URL\_SHOP\_DESCRIPTION:

URL\_SHOP\_JS:

URL\_SHOPDYNAMIC:

### URL\_SHOP\_VIEW

店铺商品列表 http://shop.paipai.com/1641624992/514/index.shtml

新手付款帮助 http://shop.paipai.com/2780428972/800/index.shtml

购物保障 http://shop.paipai.com/2780428972/1100/index.shtml

信用评价 http://shop.paipai.com/2780428972/10/index.shtml

url：http://shop.paipai.com/855005227/513/index.shtml

？？？？页面

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})\\.([0-9]{1,4})(.\*)$");

### URL\_SHOP\_LEAVEMSG

留言板url： http://shop.paipai.com/2457768010/leavemsg/index.shtml

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})(/shop){0,1}(/leavemsg)(/[1-9][0-9]{4,9}){0,1}(/index.(shtml|html))(\\?.\*){0,1}$");

### URL\_SHOP\_EDIT

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})/edit/([0-9])/([0-9]{1,11})(/index.[shtml|html]){0,1}([\\?|/].\*){0,1}$");

### URL\_SHOP\_MODULE

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})/module/([0-9]{1,20})(/index.(shtml|html|js|xml)){0,1}(\\?.\*){0,1}$");

### URL\_SHOPDYNAMIC

sm\_vecPatternList.push\_back("^([1-9][0-9]{4,9})/dynamic(/([0-9])){0,1}(/([0-9]{1,3})-([0-9]{1,3})){0,1}(/index.[shtml|html]){0,1}([\\?|/].\*){0,1}$");

### URL\_SHOP\_DESCRIPTION

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9})/description/([0-9])-([0-9]{1,3})(/index.(shtml|html|js|xml)){0,1}(\\?.\*){0,1}");

### URL\_SHOP\_JS

m\_vecPatternList.push\_back("^([1-9][0-9]{4,9}).(js|xml)(\\?.\*){0,1}$");

### URL\_SWF\_FILE

m\_vecPatternList.push\_back("^([a-z0-9A-Z\_-]+.swf)(\\?.\*){0,1}$");

### URL\_HTML\_FILE

m\_vecPatternList.push\_back("^((ping.htm)|(index.(s){0,1}html)(\\?.\*){0,1}|((\\?.\*){0,1})){0,1}$");

### //URL\_ITEM

m\_vecPatternList.push\_back("^(([a-zA-Z0-9]{32})([\\?|/].\*){0,1})$");

### URL\_ITEM\_SEARCH

m\_vecPatternList.push\_back("^search/([0-9]\*(-[0-9]\*){0,4})/([a-zA-Z0-9]{32})-([0-9a-zA-Z]{4})\\.html([\\?|/].\*){0,1}$");

### URL\_ITEM\_VIEW

m\_vecPatternList.push\_back("^([a-zA-Z0-9]{32})\\.([0-9]{1,4})(.\*)$");

### URL\_ITEMCOMBO

m\_vecPatternList.push\_back("^combo/(([a-zA-Z0-9]{32})([\\?|/].\*){0,1})$");

### URL\_ITEM\_DESCRIPTION

m\_vecPatternList.push\_back("^commodity/getdescription/([0-9]{1,10})/([a-zA-Z0-9]{32}).(js|xml)([\\?|/].\*){0,1}$");

## Render.cpp

#### int32\_t CRender::Render()

渲染页面的过程（shtml 页面合在一起以及参数填充到shtml页面） 还需要细化；

const c2cent::po::shop::CDetailFilter& oDetailFilter=m\_oDataManager.GetFilter(); 获取过滤信息

const std::vector<c2cent::po::shop::CDetailLayout>& vecLayout=m\_oDataManager.GetPo().GetLayoutList(); 获取布局信息

const uint32\_t& dwPageType=m\_oRequestFilter.GetPageType();

\_GetPage \_GetFrameID \_GetAreaID \_GetModuleID \_GetModuleFrameID

1. 递归调用 int32\_t CRender::RenderView() 以及render

RenderView(strCon,dwViewID,dwLevel,dwMyIndex,strTemp,0,dwRole);

strContent+=strTemp;

RenderView 主要作用从shop4 以及shop5 页面获取shtml静态页面；

#### int32\_t CRender::\_GetPage()

## DataManager.cpp

店铺默认是ROLE\_SHOP

if(m\_oConfiguration.GetRole()==ROLE\_SHOP)

{

iRetCode=m\_oStub.GetShopDetailInfo(

m\_oCntlInfo,

m\_strVisitKey,

\_\_FILE\_\_,

m\_oDetailFilter,

strInReserve,

oDetailPo,

strOutReserve);

}

#### CDataManager::\_SetShopTssID()

if(m\_oDetailFilter.HasCommand(DETAIL\_SHOPTSS\_KEY))

？？？？？

#### CDataManager::\_SetProtocolInfo()

根据 m\_oDetailFilter 获取商品信息；

if(m\_oDetailFilter.HasCommand(DETAIL\_ALL\_KEY))

iRetCode=m\_oStub.GetShopDetailInfo(

m\_oCntlInfo,

m\_strVisitKey,

\_\_FILE\_\_,

m\_oDetailFilter,

strInReserve,

oDetailPo,

strOutReserve);

#### CDataManager::\_SetCacheForDes()

？？？？？

#### CDataManager::\_GetCacheForDes()

？？？？？ss

#### CDataManager::\_SetClassAttribute()

设定类目导航

m\_oDetailPo.SetNav\_u();

设置类目属性

m\_oDetailPo.SetAttrList\_u();

#### CDataManager::\_SetShopClass()

设置店铺类别名称；

std::string strShopClass = oNav.GetNavNode().GetName();

m\_oDetailPo.GetShop().SetShopClassName(strShopClass);

#### CDataManager::\_ReviseAuc2()

if(m\_oConfiguration.GetRole()!=ROLE\_AUCTION)

{

return;

}

商品详情使用

sss

