处理PushClient发送过来的Push推送请求，转发给客户端；

初始化推送服务

```
chain.boot()
        .setNext(new CacheManagerBoot())//1.初始化缓存模块
        .setNext(new ServiceRegistryBoot())//2.启动服务注册与发现模块
        .setNext(new ServiceDiscoveryBoot())//2.启动服务注册与发现模块
        .setNext(new ServerBoot(mPushServer.getConnectionServer(), mPushServer.getConnServerNode()))//3.启动接入服务
        .setNext(() -> new ServerBoot(mPushServer.getWebsocketServer(), mPushServer.getWebsocketServerNode()), wsEnabled(
        .setNext(() -> new ServerBoot(mPushServer.getUdpGatewayServer(), mPushServer.getGatewayServerNode()), udpGateway(
        .setNext(() -> new ServerBoot(mPushServer.getGatewayServer(), mPushServer.getGatewayServerNode()), tcpGateway())/
        .setNext(new ServerBoot(mPushServer.getAdminServer(), null))//7.启动控制台服务
        .setNext(new RouterCenterBoot(mPushServer))//8.启动路由中心组件
        .setNext(new PushCenterBoot(mPushServer))//9.启动推送中心组件
        .setNext(() -> new HttpProxyBoot(mPushServer), CC.mp.http.proxy_enabled)//10.启动http代理服务，dns解析服务
        .setNext(new MonitorBoot(mPushServer))//11.启动监控服务
        .end();
```

启动服务

```
public final class PushCenterBoot extends BootJob {
    private final MPushServer mPushServer;

    public PushCenterBoot(MPushServer mPushServer) {
        this.mPushServer = mPushServer;
    }

    @Override
    protected void start() {
        mPushServer.getPushCenter().start();
        startNext();
    }

    @Override
    protected void stop() {
        stopNext();
        mPushServer.getPushCenter().stop();
    }
}
```

调用PushCenter->BaseService#start()，然后start()最终调用子类PushCenter#dostar()

PushCenter doStart()

```
    @Override
    protected void doStart(Listener listener) throws Throwable {
        this.pushListener = PushListenerFactory.create();1
        this.pushListener.init(mPushServer);2

        if (CC.mp.net.udpGateway() || CC.mp.thread.pool.push_task > 0) {  3
            executor = new CustomJDKExecutor(mPushServer.getMonitor().getThreadPoolManager().getPushTaskTimer());
        } else {//实际情况使用EventLoo并没有更快，还有待测试
            executor = new NettyEventLoopExecutor();4
        }

        MBeanRegistry.getInstance().register(new PushCenterBean(taskNum), null);5
        ackTaskQueue.start();6
        logger.info("push center start success");
        listener.onSuccess();
    }

    @Override
    protected void doStop(Listener listener) throws Throwable {
        executor.shutdown();
        ackTaskQueue.stop();
        logger.info("push center stop success");
        listener.onSuccess();
    }
```

1、通过SPI，找到mpush-core模块中PushListenerFactory接口的实现类GatewayPushListener，得到GatewayPushListener实例；
2、获取并设置PushCenter对象到GatewayPushListener中
3、如果是UDP模式，使用自定义线程池推送消息
4、如果是TCP模式，使用GatewayServer work 线程池
5、注册MBEAN对象PushCenterBean，用于JMX监控推送任务数量
6、启动ACK任务，获取ACK连接池对象，用于异步处理ACK任务