## 服务器启动

* Web.xml中配置了服务器启动时进行**游戏初始化**操作的servlet：**com.manu.dynasty.core.servlet.InitServlet**
  1. 加载**Hibernate**
  2. 加载游戏各种配置：GameServerInit.init()
     1. SessionManager管理、注册协议
     2. 聊天模块
     3. 连接memcached缓存（DataManager）
     4. 加载配置文件模版数据（DataLoader）
     5. 初始化各个模块（BigSwitch）
     6. 启动网络通讯监听（**mina框架,** TXSocketMgr）
* 底层通讯框架用的是mina
* 消息的收发回复用的是google protocol buffer
* 日志记录：log4j
* 数据库：MySQL
* 数据库操作：hibernate框架
* 服务器缓存：memcached
* 项目布署：ant

## 游戏登录

* **相关项目：qxrouter（未特殊表明相关项目，都是指qxmobile）**
* 公告、注册、登录都是由前端直接访问对应的URL，后端在URL对应的jsp页面上进行操作。**相关页面：WebContent下的注册accountReg.jsp、登录accountLogin.jsp、公告sysNotice.jsp**
* 数据库表：accounts、ServerConfig
* 未注册用户，注册访问accountReg.jsp，服务器做处理
* 已注册用户，登录访问accountLogin.jsp，服务器做处理。会返回上次登录服务器（accounts表），与服务器列表（ServerConfig表）
* 选择服务器，进入游戏，即连接到**项目qxmobile**

## Account模块

相关包：com.qx.account

协议文件：Account.proto

数据库表：accounts

## 定时任务功能模块

相关包：com.qx.timeworker

协议文件：TimeWorker.propo

数据库表：TimeWorker

* **定时请求体力操作：**
  1. 实现方式：
     1. 玩家刚上线时：服务器会更新玩家离线时间累计增加的体力。
     2. 客户端每次请求加体力操作。服务器处理增加体力操作，返回给客户端消息（带有剩余时间）
* **定时加免费洗练次数：**
  1. 实现方式：
     1. 每当客户端请求XiLianReq(UserEquip.proto)进行装备洗练查看操作时。服务器会调用TimeWorker.calcOfflineXilian()方法，计算玩家离线时间（或者是本次操作距离上次增加洗练的时间）累计增加的洗练次数，并返回次数、剩余时间。

## 玩家背包

相关包：com.qx.bag

BagGrid：背包格子，Bag：背包，EquipGrid：穿在君主身上的装备格子。

协议文件：BagOper.proto

数据库表：BagGrid

* **BagGrid表（君主背包的格子）**
  1. db\_id：junzhuId\*1000 + gridNum（db\_id没有重用，需重新编写）
  2. instId：指的是表UserEquip的db\_id

## 战斗

相关包：

## 成就系统

相关包：com.qx.achievement

协议文件：Achievement.proto

数据库表：Achievement

* **查询**：客户端请求成就列表，服务端判断是否有该玩家的成就列表，没有则将所有成就初始信息插入数据库，返回。 有的话，查出返回。
* **修改**：基于事件驱动模式。当玩家完成某一成就相关的任务时，将完成该成就任务的信息封装成事件。添加到事件列表，由事件管理类执行成就事件。
* **消息推送：**当某一成就信息发生改变（进度增加或完成）就会主动给客户端推送消息
* **现在有些成就任务还没添加上**

## 商城系统

相关包：

* **购买体力、铜币**

相关包：com.qx.purchase

协议指令：junZhu.proto(请求剩余次数)

数据库表：TiLi，TongBi

* **购买宝箱抽奖**

相关包：com.qx.card

协议指令：LuckyCard.proto

## PVE战斗系统

## 装备成长

相关包：com.qx.equip

协议文件：UserEquip.proto

数据库表：BagGrid，EquipGrid，EquipXiLian（记录上次洗练并且没有进行确认或 取消操作的装备，洗练次数存在TimeWorker表中），UserEquip

* **强化**
* **洗练**

先请求装备列表PD. C\_EQUIP\_LIST

1. 洗练指令PD.C\_EQUIP\_XiLian，根据action值来执行对应的操作

* **进阶**

## 武将

* **武将**

相关包：com.qx.hero

协议文件：WuJiang.proto

数据库表：WuJiangs，WjKeJi

1. 武将获得
   1. 君主创建成功后，会自动带一名武将，武将id：31010。在[JunZhuMgr](eclipse-javadoc:%E2%98%82=qxmobile/src%3Ccom.qx.junzhu%7BJunZhuMgr.java%E2%98%83JunZhuMgr).fixCreateJunZhu()中添加

* **武将科技**

相关包：com.qx.hero

协议文件：WuJiang.proto

数据库表：WuJiangs，WjKeJi

* **获得武将科技**
  1. 现在没有在直接往数据库里插入君主的武将科技信息。
  2. 玩家在请求武将科技信息时
     1. 若数据库里找不到对应的武将科技信息**:**会临时创建（[WuJiangKeJiMgr](eclipse-javadoc:%E2%98%82=qxmobile/src%3Ccom.qx.hero%7BWuJiangKeJiMgr.java%E2%98%83WuJiangKeJiMgr).createDefaultBean()）一个默认的武将科技信息发给前端
     2. 若数据库里能找到，直接将数据返回
  3. 玩家在升级武将科技时
     1. 若数据库里找不到对应的武将科技信息同**b)**，完成科技升级，**保存数据库**
     2. 若数据库里能找到，直接将数据返回，并完成科技升级，修改数据库
* **科技升级**
  1. 前端发送需要升级的 -科技id，配置表里的id
  2. 后端判断科技类型。分别进行低级科技升级或者高级科技升级。升级所需的物品及其数量从**KeJi.xml表**中读取
     1. 低级科技：需要判断玩家铜币是否足够
     2. 高级科技：需要判断武将的金色精魄数量是否足够
     3. 低级、高级科技升级，都要判断当前要升级的科技对应的KeJi.xml条目的preId配置的科技类型当前的等级是否达到（>=）表里配的值