联通网优门户网站软件架构设计文档

|  |  |
| --- | --- |
| 项目名称：联通网优门户网站 | 版本：V1.0 |
| 文档类型：软件架构设计文档 | 日期：2014-01-20 |
|  | |

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2014-01-20 | V1.0 | 创建 | 宫文博 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 5

1.1 目的 5

1.2 范围 5

1.3 定义、首字母缩写词和缩略语 5

1.4 参考资料 5

1.5 概述 5

2. 整体说明 5

2.1 简介 5

2.2 构架表示方式 5

2.3 构架目标和约束 5

3. 用例视图 6

3.1 核心用例 6

3.2 用例实现 6

4. 逻辑视图 6

4.1 逻辑视图 6

4.2 分层 6

4.2.1 应用层 6

4.2.2 业务层 6

4.2.3 中间层 6

4.2.4 系统层 7

4.3 架构模式 7

4.4 设计机制 7

4.5 公用元素及服务 7

5. 进程视图 7

6. 部署视图 7

7. 实施视图 7

7.1 概述 7

7.2 层 8

7.3 部署 8

8. 数据视图 8

9. 大小和性能 8

10. 质量 8

11. 其它说明 8

12. 附录A 指南 8

13. 附录B 规范 8

14. 附录C 模版 8

15. 附录D 示例 9

软件架构设计文档

# 简介

联通网优门户网站是联通网优推出的下一代BS架构的系统，主要包括门户首页、个人工作台、报表分析、工作流程、信息交流、系统管理等几大功能模块，每个功能模块又分为很多子功能模块。此软件架构设计文档的目的在于为上述需求提供灵活、可扩展的解决方案。

## 目的

本文主要阐述联通网优门户网站的主要技术架构，以及各个技术架构之间的关系。通过多种不同的架构视图，描述整个系统的基本架构。

此文档框定了系统的主要技术架构，适用于所有开发和需求人员。

## 范围

此软件架构文档主要是针对联通网优门户网站的需求撰写的，后续一些和该网站类似的BS架构的网站也同样适用。

## 定义、首字母缩写词和缩略语

|  |  |  |  |
| --- | --- | --- | --- |
| 缩略语 | 说明定义 | 英文全称 | 中文译名 |
| IOC | 控制反转 | Inversion Of Control | 控制反转 |
| DI | 依赖注入 | Dependency injection | 依赖注入 |
| MVC | 一种软件设计典范 | Model-View-Controller | 模型-视图-控制器 |
| MVVM | 一种软件设计模式 | Model-View-ViewModel | 模型-视图-视图模型 |
| WCF | 微软最新的通信开发技术 | Windows Communication Foundation | 微软通信库 |
| OOP | 面向对象编程 | Object Oriented Program | 面向对象编程 |
| AOP | 面向方面编程[面向切面]，专注应用点，使用织入等手段切入功能 | Aspect Oriented Program | 面向方面编程 |
| SOA | 面向服务的应用开发，适用于系统交互 | Service Oriented Application | 面向服务应用 |
| REST | 全新的针对web应用的开发风格 | Representational State Transfer | 表述性状态转移 |

## 参考资料

## 概述

# 整体说明

本系统采用自中间向外的设计方法：第一步先根据系统的总体要求，包括系统的业务特征、系统负载、总体吞吐、关键业务场景及其性能要求、资源预算，来制定总体系统架构；第二步即根据业务场景建立中间业务抽象，再由此抽象向上推导UI设计和向下推导数据访问和数据库设计。

本系统采用OOA和OOD的设计技术：即面向对象的分析和设计方法，所有领域模型包括接口和业务对象，都根据OO原则和OO模式来组织以及实现。

遵循设计原则：低耦合、高内聚原则，面向接口和抽象进行设计开发，提高模块和包内的聚合度，降低模块和包间的耦合度

遵循基本的OOA和OOD原则（单一职责、开放-封闭、接口隔离、依赖倒置）；使用AOP、IOC组件和中等粒度的设计模式。

## 简介

整体技术架构：



OSGI.Net结构：

插件式开发目录结构：



|  |  |
| --- | --- |
| 操作系统 | Windows Server X |
| 运行平台 | IIS \ Oracle11.1G \ .Net Framework 4.0 |
| 开发语言 | C# \ ASP.Net MVC 4.0 |
| 第三方技术 | OSGI.Net \ IBatis.Net \ ServiceStack.Net |

## 构架表示方式

此架构整体采用OSGI.Net插件式开发框架，使用MVC4.0开发模式，数据库持久层采用IBatis.Net,上层采用KendoUI（一种Html/JavaScript的UI框架）。

## 构架目标和约束

1. 可扩展、可复用性。
2. 插件式开发方式。

# 用例视图

## 核心用例

## 用例实现

# 逻辑视图

## 逻辑视图



## 分层

### 表现层

使用HTML5 + CSS3 + JavaScript + Kendo.UI第三方客户端UI控件，用Ajax调用Controller中的Action方法获取数据，然后绑定到上层控件上显示。

### 应用层

各个功能模块采用ASP.NET MVC4.0开发模式，通过OSGI.Net将各个功能模块组织起来。

又可细分为：实体类层、数据访问层、领域模型层。

通过AOP面向切面对业务逻辑和日志记录、异常捕获等和业务不相关的功能进行分离。

### 服务层

用ServiceStack.NET将所有的操作方法封装成web服务，通过http协议的get和post方式调用。

使其他系统可以很容易的使用我们的服务，并且提供可分布式部署的能力。

### 数据层

采用IBatis.Net作为数据库持久层框架，提供和数据库的交互接口。

## 架构模式

整体采用插件式开发模式，一个宿主项目和多个插件项目，各个插件项目可单独进行开发、调试，插件项目开发完成之后组装到宿主项目中即可使用。这种开发模式使项目之间的耦合度很低，并且各个项目保持自身独立，并且功能的扩展性很好。

## 设计机制

## 公用元素及服务

通用方法、公共控件、基本功能、报表服务、工作流引擎。

# 进程视图

# 部署视图



# 实施视图

## 概述

## 层

## 部署

部署包主要分为：网优门户、报表服务、数据预生成程序。

网优门户是我们的B/S系统，对所有的功能模块进行组织呈现。

报表服务为网优门户提供报表分析功能，可自己定制需要的报表功能。

数据预生成程序是另外一个单独的C/S系统，主要功能是每天固定时间和频率进行数据汇总，为我们的业务功能模块提供数据，将一些需要大量计算和比较耗时的任务进行预先汇总，业务模块只需要查询对应的汇总数据表即可，可大大提高用户感知度。

# 数据视图



整体采用IBatis.Net数据持久层框架，从数据库中直接得到实体类对象，然后再将实体类对象转换为JSON串，传递到UI解析显示。

# 大小和性能

各个功能模块采用分页显示数据，保证每次读取的数据不能太大，提高用户使用感知。

# 质量

通过OSGI.Net插件式开发，使增加或者删除功能模块变得很简单，实现系统的可扩展性。

通过AOP面向方面编程，对系统的日志记录、异常捕获进行处理，实现系统的可靠性。

通过ServiceStack.Net，Web服务框架，对所有的方法调用进行封装，使用通用的http协议的get或者post方式调用，实现系统的可移植性。另外，前端UI使用纯客户端的UI框架，使UI层具有可移植性。

# 其它说明

# 附录A 指南

# 附录B 规范

# 附录C 模版

# 附录D 示例