U9-WS功能说明

文档信息：

|  |  |
| --- | --- |
| 作者： |  |
| 创建日期（yyyy-mm-dd）： | yyyy-mm-dd |
| 审核者： |  |
| 审核日期（yyyy-mm-dd）： |  |
| 最后修订者： |  |
| 最后修订日期（yyyy-mm-dd）： | yyyy-mm-dd |

文档修订历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订者 | 修订内容 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[**1** **整体说明** 4](#_Toc29072258)

[**2** **项目文件说明** 4](#_Toc29072259)

[**3** **配置说明** 11](#_Toc29072260)

[**4** **服务开发** 19](#_Toc29072261)

[**5** **项目布署** 19](#_Toc29072262)

### **整体说明**

#### **开发目的**

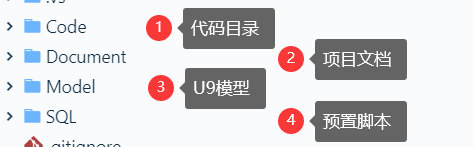
1. 简化外部系统对U9服务的调用。
2. 增强U9服务的管理及维护功能。

#### **功能介绍**

1. 融合于U9系统中，不需要独立站点部署,可根据需求配置所需的功能。
2. 采用Newtonsoft.Json替换默认的DataContractJsonSerializer的Json转换器，解决Json日期格式、循环引用等问题
3. 统一对调用中出现的异常的进行处理
4. 包含两种初始化U9上下文环境的方式:固定上下文、Token上下文
5. 包含两种记录服务的调用过程的日志方式：Log4Net日志、数据库日志
6. 包含获取Token认证及Token校验功能，包含两种存储Token的方式:内存存储、数据库存储
7. 提供通用CommService服务调用方式，支持调用BP,SV,RestSV

### **项目文件说明**

#### **项目目录**



图表 目录顶层结构

##### **代码目录**



图表 代码目录结构

包含三种代码:

* + - 1. 基础服务代码：对外发布服务、基础架构及一些附加功能的代码。
      2. 数据库日志：基于U9的数据库日志查询及管理BE、BP、SV、UI代码。
      3. 服务管理：基于U9的服务管理的BE、BP、SV、UI代码。

##### **项目文档**

1. 代码及配置的说明文档
2. 项目开发中使用的一些规范性文档

##### **U9模型**

1. 数据库日志功能模型
2. 服务管理功能模型:现只包含Token管理

##### **预置脚本**

1. 项目中部分功能使前需要在系统数据库中预置的数据脚本

#### **基础服务代码详细说明**

##### **发布目录:WCFService**

存放项目的web.config配置文件及对外开放服务的\*.svc文件。

##### **基础模块:WS.Base**

项目的基础功能模块，主要实现:

* 采用Newtonsoft.Json替换默认的DataContractJsonSerializer的Json转换器(Behavior/WebHttp/NewtonsoftJson\*.cs)
* IIS WCF服务宿主重写，主要用于设置默认配置(Services/ IISHost\*.cs)
* 统一调用异常处理(Behavior/WebHttp/ExtendErrorHandler.cs)
* Log4Net日志终结点行为(Behavior/ Endpoint/ Log4Net\*.cs)，用于记录日志到portal/log/目录下
* 返回Json格式定义（Interfaces/ IReturnMessage.cs）
* 其它辅助功能:配置帮助类、Json帮助类等

##### **Json模块: WS.Json**

由于U9标准产口自带了Newtonsoft.Json,考虑到兼容性及部分功能的需要，针对Newtonsoft.Json-6.0.1进行了部分修改后的版本，修改内容如下:

* Newtonsoft.Json的JsonSerializerSettings中添加IsAutoCreateMemberValue属性，默认为false,当IsAutoCreateMemberValue=true时，在对象转换为Json时，当对象引用的对象是null时，会自动创建一个对象，当数组或IList列表为null时，自动创建数组或IList对象
* Newtonsoft.Json的JsonSerializerSettings中添加MaxWritingDepth属性,如果MaxWritingDepth大于0，当一个对象转换为Json时，如果扩开路径阶层大于MaxWritingDepth,将会停止展开

##### **上下文基础: WS.U9Context**

* U9上下文处理的基础代码

上下文对象(ContextObject.cs)

* 采用固定上下文时，上下文的配置及管理

配置处理(Configuration/\*.cs)

设置固定上下文终结点行为(Behavior/Endpoint/\*.cs)

固定上下文管理(ContextInfoManager.cs)

##### **Services目录**

用来存放对外开放服务的代码，与WCFService下的\*.svc相对应，服务的开发方式与标准wcf服务的开发方式相同，项目中自带两个服务:

###### **通用服务: CommService**

提供针对U9的BP、SV的通用调用方法,方法如下:

* 获取代理请求对象

传入BP、SV的对应的Agent.dll、Agent类及展开阶层(int)，则可以获取调用代理的对象Json串

* 执行代理请求

根据传入BP、SV的对应的Agent.dll、Agent类及代码对象的Json串，通过代理调用BP、SV

* Job执行代理请求
* 根据传入BP、SV的对应的Agent.dll、Agent类及代码对象的Json串，采用Job通过代理调用BP、SV

###### **测试服务: TestService**

主要用来作为Json数据类型的测试及服务开发的模板。

##### **Token目录**

提供当外部系统调用接口之前进行身份认证及调用接口时Token是否有效等功能。

###### **Token基础模块:WS.Token**

Token相关的基础代码，主要实现的功能如下:

* Token管理(TokenManagement.cs),现提供2种不同存储方式的Token管理方式：内存管理方式、数据库管理方式
* Token配置（Configuration/\*.cs）
* 认证服务(Services/ AuthTokenService.cs)
* Token校验终结点行为(Behavior/Endpoint/TokenValidation \*.cs)
* 通过Token设置上下文终结点行为(Behavior/Endpoint/ SetU9ContextByToken\*.cs)

###### **Token内存管理:MemoryProvider**

采用内存存储Token,由Token管理启用Timer调度清理失效的Token。

###### **Token数据库管理:DBProvider**

采用数据库存储Token,需U9系统中设置调度进行清理失效的Token。

##### **U9动作目录**

###### **U9动作基础模块: WS.U9Action**

由于WCF的Endpoint Behavior执行顺序问题，如下例:

A Behavior:进入时初始化上下文，返回时清理上下文

B Behavior:进入时操作U9,返回时操作U9,

WCF的Endpoint Behavior执行顺序:A->B->A->B,这就会造成B Behavior在操作U9时，U9上下文已被清理掉。所以定义了一个U9Action Behavior,在内部统一执行U9的动作，动作的执行顺序改为:A->B->B->A。

U9动作基础模块实现的功能如下:

* 针对配置节点U9Actions代码（Configuration/\*.cs）
* U9Action终结点行为(Behavior/Endpoint/ U9ActionEndpointBehavior\*.cs)，作为U9Action动作的入口点。
* 设置固定上下文动作(Action/\*.cs)

###### **通过Token设置上下文动作U9Action.Token**

通过Token设置上下文动作

###### **数据库日志动作: U9Action.DBLog**

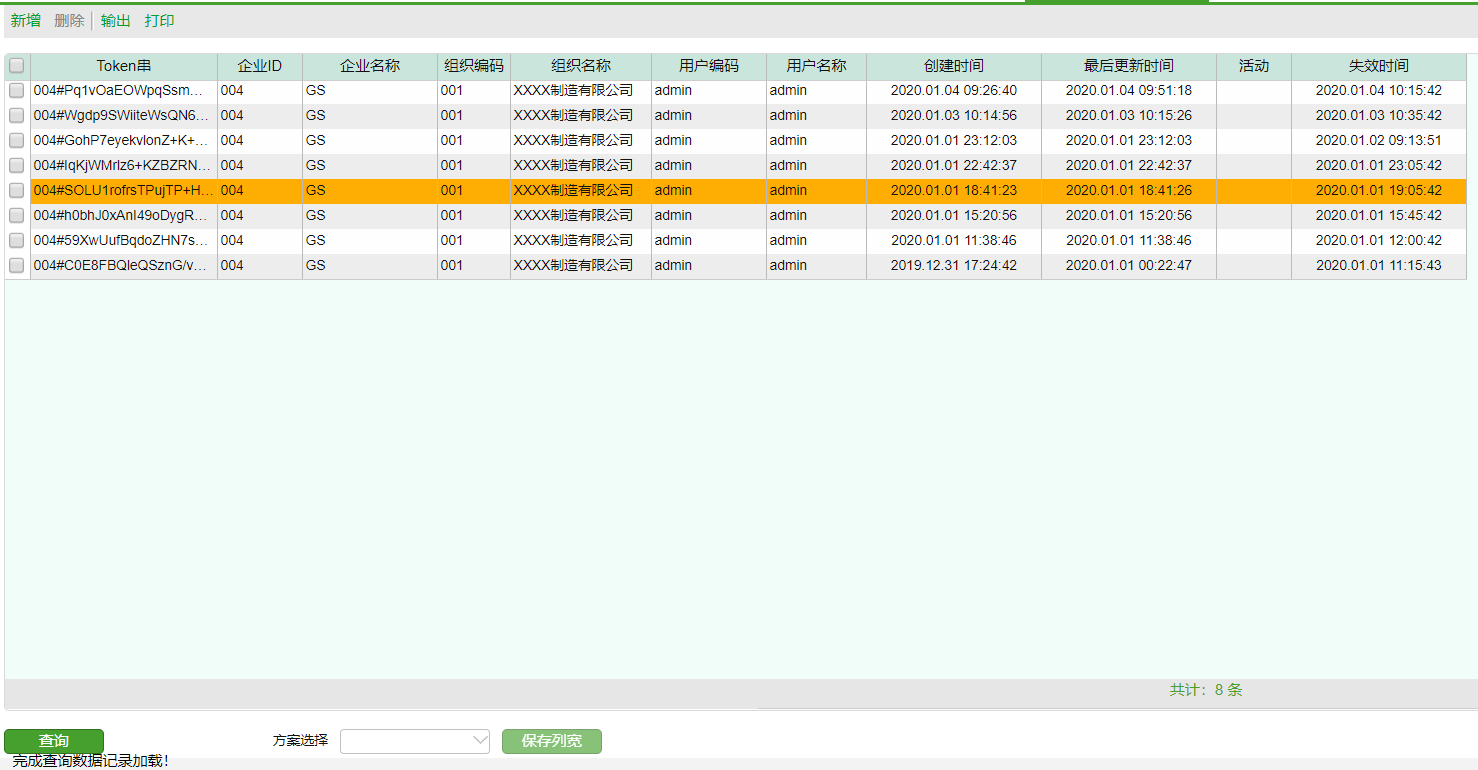
将服务的调用日志记录在数据库中，记录日志方式采用两步式的方法，当服务接口被调用请求传入时(接口还没有有被调用)，记录一份日志，当调用请求已完成，准备返回调用结果时，再去补全调用前生成的日志记录。

#### **U9相关功能说明**

具体实现比较简单，此处不再针对每个代码进行解释。

##### **服务管理**

###### **Token管理**

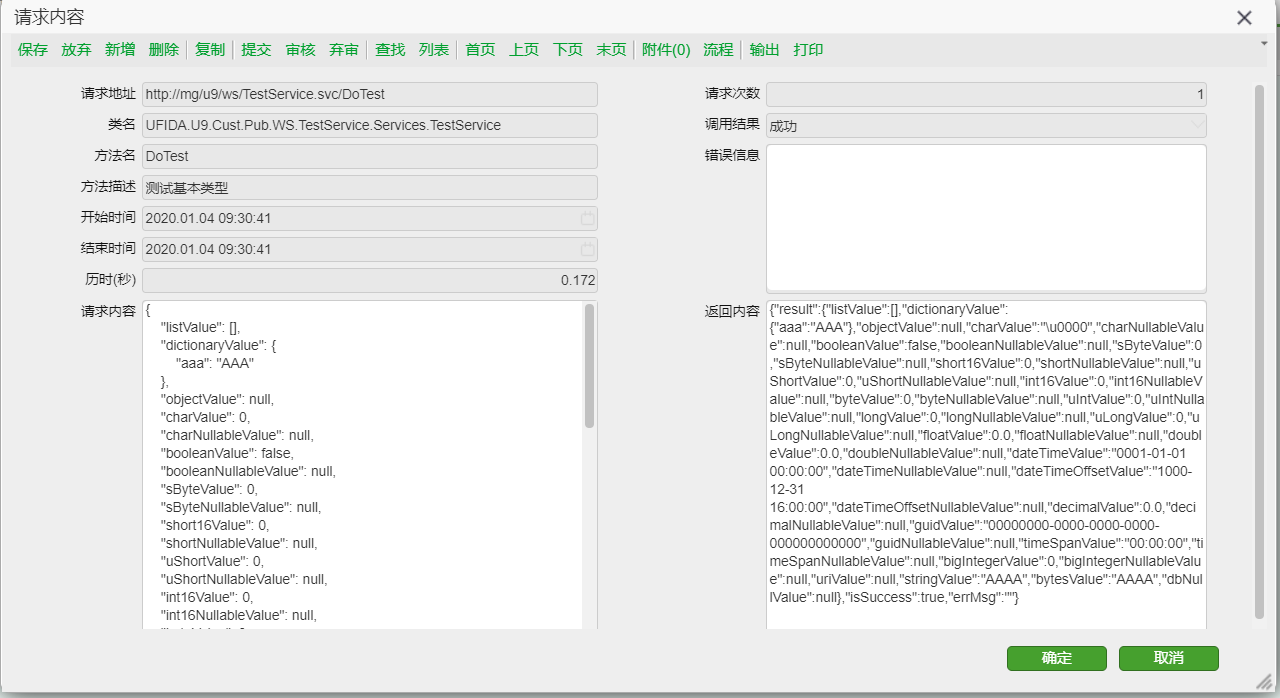


图表 Token管理列表

##### **服务日志**



图表 服务日志列表



图表 单条日志记录

### **配置说明**

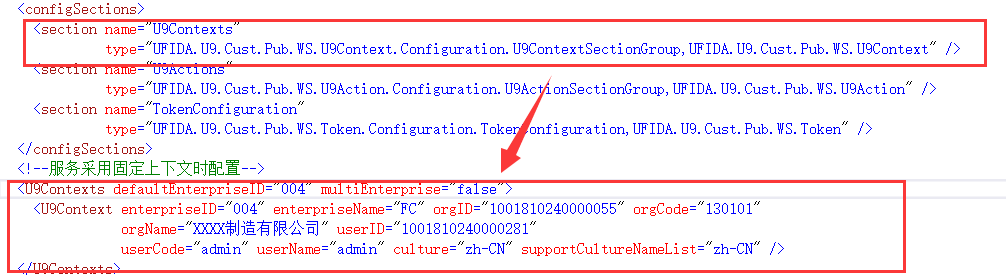
以WCFService目录下中web-All.config内容进行讲解。

#### **配置节点ConfigSection**

暂支持三个可配置节点

##### **固定单一用户上下文配置**

当采用固定用户作为服务调用上下文时，需配置此节点，不采用固定用户作为上下文，此节点可删除。



图表 固定单一用户上下文节点配置

|  |  |  |
| --- | --- | --- |
| **U9Contexts节点配置** | | |
| 属性英文名 | 属性中文名 | 值说明 |
| defaultEnterpriseID | 默认企业ID | U9的企业编码，当multiEnterprise=false时，不允许为空 |
| multiEnterprise | 是否多企业 | 如果服务涉及多个企业，指定为true,  *当值=true时，请求headers需指定EnterpriseID* |
| **U9Context节点配置** | | |
| enterpriseID | 企业ID | U9的企业编码，不允许为空 |
| enterpriseName | 企业名称 | U9的企业名称 |
| orgID | 组织ID | 不能为空 |
| orgCode | 组织编码 | 不能为空 |
| orgName | 组织名称 | 不能为空 |
| userID | 用户ID | 不能为空 |
| userCode | 用户编码 | 不能为空 |
| userName | 用户名称 | 不能为空 |
| culture | 语言 | 可空，空时自动设为”zh-CN” |
| supportCultureNameList | 支持语言列表 | 可空，空时自动等于culture |

##### **U9动作节点配置**

用于在U9ActionEndpointBehavior行为中执行的动作。如果不采用U9ActionEndpointBehavior，则不需要配置此节点。现提供的U9Action主要包括：

* + 设置单一用户上下文动作
  + 通过Token认证设置上下文动作
  + 两步式日志动作



图表 U9动作节点配置

配置中首先要选择一种设置上下文的方式，后续再根据需要配置其它的动作节点。现有两种组合方式:

* + 设置单一用户上下文动作 + 两步式日志动作
  + 通过Token认证设置上下文动作 + 两步式日志动作

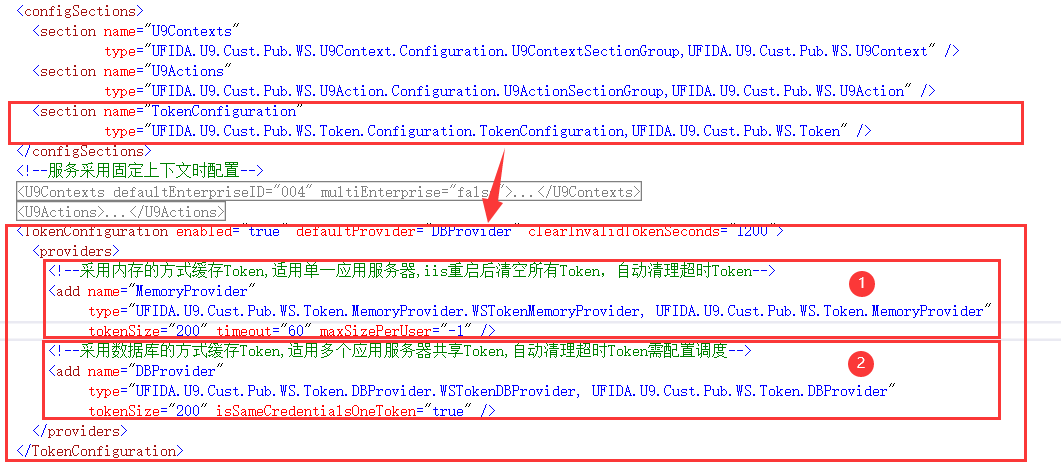
U9Action的执行顺序,根据配置执行顺序:

BeforeDo:设置上下文->两步式日志前事件->AfterDo: 两步式日志后事件->清除上下文。

##### **Token配置节点**

当需要进行Token认证时，需配置此节点。现提供两种Token提供者进行Token管理:

* MemoryProvider：采用内存的方式缓存Token,适用单一应用服务器,iis重启后清空所有Token，自动清理超时Token
* DBProvider: 采用数据库的方式缓存Token,适用多个应用服务器共享Token,自动清理超时Token需在U9中配置调度。



图表 Token配置节点

|  |  |  |
| --- | --- | --- |
| **TokenConfiguration配置节点** | | |
| 属性英文名 | 属性中文名 | 值说明 |
| enabled | 是否启用 | true/false |
| defaultProvider | 默认提供者 | 需指定使用的Provider的name |
| clearInvalidTokenSeconds | 清除失效Token调度秒数 | 清除失效Token调度秒数 |
| **MemoryProvider节点配置** | | |
| tokenSize | Token字符长度 | Token字符长度 |
| timeout | 超时时间 | 当最后一次调用时间与当前时间间隔大于超时时间时，则失效token |
| maxSizePerUser | 最大使用用户数 | 小于或等于0时，不限制  大于0时，如果在线有效token数量超过这个设定值，则自动清掉最后更新时间最早的并大于最大使用用户数的token |
| **DBProvider节点配置** | | |
| tokenSize | Token字符长度 | Token字符长度 |
| isSameCredentialsOneToken | 相同认证采用同一个token | =true时，相同企业ID,组织Code,用户Code的采用同一个token |

DBProvider在U9系统中配置：

* 基础模块(集团)->参数设置-> 设置【Token超时时间(秒)】参数
* 增加调度：



图表 清除Token调度方案

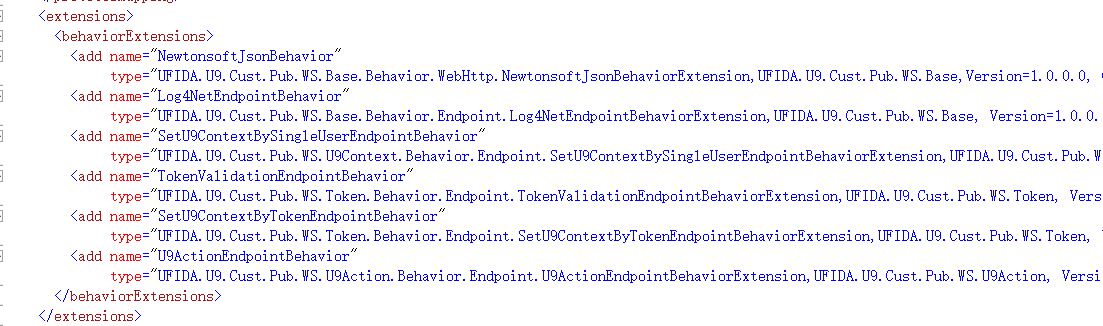


图表 清除Token请求管理方案

#### **Behavior配置**

##### **Behavior扩展配置**

在针对服务进行Behavior配置前，需定义Behavior扩展，可根据需要配置，现支持的全部配置如下:



图表 behavior扩展定义

##### **服务Behavior配置**

采用多种Endpoint Behavior组合的方式为每个服务定制相应的终结点行为。



图表 服务Behavior配置

* Token认证服务

NewtonsoftJsonBehavior+Log4NetEndpointBehavior

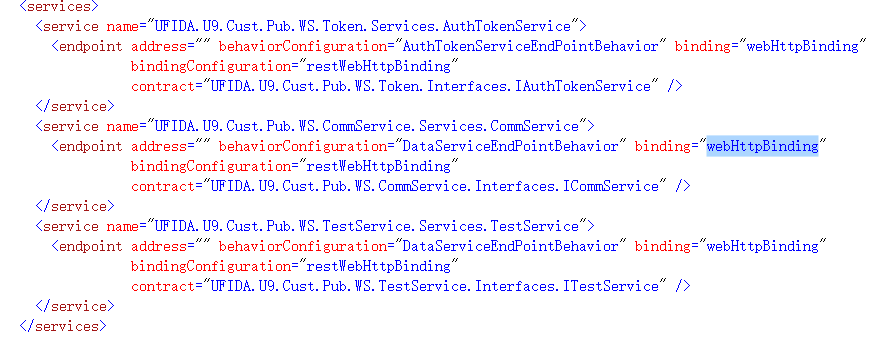
* 数据服务

可采用多种方式进行组合。各Behavior、U9Action之间的关系如下表：

|  |  |  |
| --- | --- | --- |
| **Behavior** | **U9Action** | **功能说明** |
| NewtonsoftJsonBehavior |  | 必选项，替换Json转换器 |
| Log4NetEndpointBehavior |  | 可选，记录日志到Portal/log目录下 |
| SetU9ContextBySingleUserEndpointBehavior |  | 当采用固定上下文且不采用U9Action方式时，必选 |
| SetU9ContextByTokenEndpointBehavior |  | 采用Token认证时且不采用U9Action方式时，必选 |
| TokenValidationEndpointBehavior |  | 采用Token认证时必选，否则不选 |
| U9ActionEndpointBehavior | SetU9ContextBySingleUser | 当采用固定上下文且采用U9Action方式时，必选 |
| SetU9ContextByToken | 采用Token认证时且采用U9Action方式时，必选 |
| TwoStepDBLog | 使用两步式数据库日志时，必须使用U9Action Behavior |

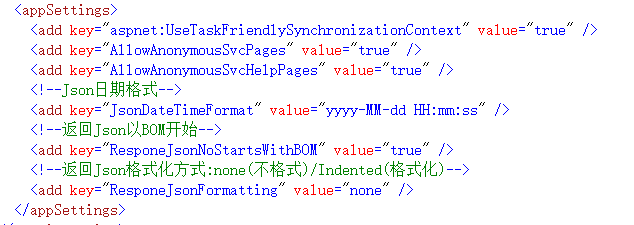
#### **服务定义配置**

针对每个提供的服务，需在此处定义服务，其中behaviorConfiguration需指定对应的服务behavior配置。



图表 服务定义配置

#### **其它配置**



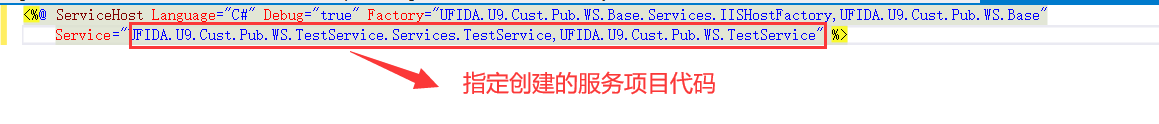
图表 appSetting配置

|  |  |
| --- | --- |
| **Key名称** | **说明** |
| JsonDateTimeFormat | 可根据需要设置Json中的日志格式 |
| ResponeJsonNoStartsWithBOM | 返回Json以BOM开始 |
| ResponeJsonFormatting | 返回Json格式化方式:none(不格式)/Indented(格式化) |

### **服务开发**

建议的开发步骤：

* 1. 在U9中创建轻量级SV,SV必须指定一个DTO作为传入参数。
  2. 依据Services中TestService为模板，创建新的服务项目，服务项目中引用已开发的U9 SV并创建方法对SV进行调用。
  3. 在web.config中对服务进行定义配置，在WCFService网站下创建\*.svc文件



图表 SVC文件配置

* 1. 部署文件
  2. 测试(postman)

### **项目布署**

1. 如果使用了Token管理、DBLog功能，需要先以U9 补丁的形式打上这些功能的补丁。
2. 将基础服务代码复制到portal/bin目录下
3. 在U9 portal下创建一个目录，当前以portal创建ws目录为例
4. WCFService网站目录下的web.confg、\*.svc文件复制到ws目录下
5. 访问[http://xx.xx.xx.xx/u9/ws/\*.svc](http://xx.xx.xx.xx/u9/ws/*.svc)尝试发布是否成功

WCFService网站目录下提供了两个批处理来进行快速部署：

* AutoBuild-Simple：简单功能部署

copy .\CommService.svc D:\yonyou\U9V60\Portal\ws

copy .\TestService.svc D:\yonyou\U9V60\Portal\ws

copy .\Web-Simple.config D:\yonyou\U9V60\Portal\ws\Web.config

copy .\bin\UFIDA.U9.Cust.Pub.WS.Base.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Base.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Json.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Json.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Context.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Context.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.CommService.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.CommService.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.TestService.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.TestService.pdb D:\yonyou\U9V60\Portal\bin

* AutoBuild-All：所有功能部署

copy .\AuthTokenService.svc D:\yonyou\U9V60\Portal\ws

copy .\CommService.svc D:\yonyou\U9V60\Portal\ws

copy .\TestService.svc D:\yonyou\U9V60\Portal\ws

copy .\Web-All.config D:\yonyou\U9V60\Portal\ws\Web.config

copy .\bin\UFIDA.U9.Cust.Pub.WS.Base.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Base.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Json.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Json.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Context.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Context.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.DBProvider.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.DBProvider.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.MemoryProvider.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.Token.MemoryProvider.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.DBLog.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.DBLog.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.Token.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.U9Action.Token.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.CommService.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.CommService.pdb D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.TestService.dll D:\yonyou\U9V60\Portal\bin

copy .\bin\UFIDA.U9.Cust.Pub.WS.TestService.pdb D:\yonyou\U9V60\Portal\bin