用户操作

[留言] [发消息] [加为好友]

订阅我的博客

0 位读者
POWERED BY FEEDSKY

订阅

订阅到鲜果

订阅到Google

订阅到抓虾

leo_fanaq的公告

口味较杂，将就看吧

文章分类

C/C++
JAVA
MASM32
Spring
设计模式
数据库

好友

原 AXIS2/C 客户端调用Xfire WebServices服务 收藏

用Axis2自带的wsdl2c生成的客户端，复杂 呵呵，但是很OOP。这里用一个简单的来实现

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements.  See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License.  You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <stdio.h>
#include <axiom.h>
#include <axis2_util.h>
#include <axiom_soap.h>
#include <axis2_client.h>

axiom_node_t *
build_om_payload_for_echo_svc(const axutil_env_t *env);
```

存档

- 2009年06月(1)
- 2008年12月(1)
- 2008年11月(4)
- 2008年10月(3)
- 2008年09月(4)
- 2008年08月(2)
- 2008年07月(5)
- 2008年04月(1)
- 2008年03月(7)
- 2008年02月(7)
- 2008年01月(10)
- 2007年05月(1)

```
int main(int argc, char** argv)
{
    const axutil_env_t *env = NULL;
    const axis2_char_t *address = NULL;
    axis2_endpoint_ref_t* endpoint_ref = NULL;
    axis2_options_t *options = NULL;
    const axis2_char_t *client_home = NULL;
    axis2_svc_client_t* svc_client = NULL;
    axiom_node_t *payload = NULL;
    axiom_node_t *ret_node = NULL;
    axiom_node_t *payload2 = NULL;
    axiom_node_t *ret_node2 = NULL;
    const axis2_char_t *un = NULL;
    const axis2_char_t *pw = NULL;
    /*axutil_allocator_t *allocator = NULL;*/

    /* Set up the environment */
    env = axutil_env_create_all("echo.log", AXIS2_LOG_LEVEL_TRACE);

    /* Set end point reference of echo service */
    address = "http://xxxx.xxxx.xxx/xxxx";
    if (argc > 1)
    {
        if (axutil_strcmp(argv[1], "-h") == 0)
        {
            printf("Usage : %s [endpoint_url] (-auth [username] [password]) ",
                   argv[0]);
            printf("use -auth option for HTTP Authentication ");
            printf("use -h for help ");
            return 0;
        }
        else
        {
            address = argv[1];
        }

    }
    printf("Using endpoint : %s ", address);

    /* Create EPR with given address */
    endpoint_ref = axis2_endpoint_ref_create(env, address);

    /* Setup options */
    options = axis2_options_create(env);
    axis2_options_set_to(options, env, endpoint_ref);
    // 这是与Xfire服务兼容的关键
    axis2_options_set_soap_version(options, env, AXIOM_SOAP11 );
    //axis2_options_set_action(options, env, http://xxx.xxx.xxx);

    /* Set up deploy folder. It is from the deploy folder, the configuration is picked up
     * using the axis2.xml file.
     * In this sample client_home points to the Axis2/C default deploy folder. The client_home can
     * be different from this folder on your system. For example, you may have a different folder
     * (say, my_client_folder) with its own axis2.xml file. my_client_folder/modules will have the
     * modules that the client uses
     */
    client_home = AXIS2_GETENV("AXIS2C_HOME");
    if (!client_home || !strcmp (client_home, ""))
        client_home = "../..";

    /* Create service client */
    svc_client = axis2_svc_client_create(env, client_home);
    if (!svc_client)
    {
        printf("Error creating service client, Please check AXIS2C_HOME again ");
        AXIS2_LOG_ERROR(env->log, AXIS2_LOG_SI, "Stub invoke FAILED: Error code:"
                " %d :: %s", env->error->error_number,
                AXIS2_ERROR_GET_MESSAGE(env->error));
          return -1;
    }

    /* Set service client options */
    axis2_svc_client_set_options(svc_client, env, options);

    /* Engage addressing module */
    //axis2_svc_client_engage_module(svc_client, env, AXIS2_MODULE_ADDRE
SSING);



    /* Build the SOAP request message payload using OM API.*/
    payload = build_om_payload_for_echo_svc(env);

    /* Send request */
    ret_node = axis2_svc_client_send_receive(svc_client, env, payload);

    if (ret_node)
    {
        axis2_char_t *om_str = NULL;
        om_str = axiom_node_to_string(ret_node, env);

        if (om_str)
            printf(" Received OM : %s ", om_str);
        printf(" echo client invoke SUCCESSFUL! ");

        AXIS2_FREE(env->allocator, om_str);
        ret_node = NULL;
    }
    else
    {
        AXIS2_LOG_ERROR(env->log, AXIS2_LOG_SI, "Stub invoke FAILED: Err
or code:"
                " %d :: %s", env->error->error_number,
                AXIS2_ERROR_GET_MESSAGE(env->error));
        printf("echo client invoke FAILED! ");
    }

    payload2 = build_om_payload_for_echo_svc(env);
    ret_node2 = axis2_svc_client_send_receive(svc_client, env, payload2);
    if (ret_node2)
    {
        axis2_char_t *om_str = NULL;
        om_str = axiom_node_to_string(ret_node2, env);

        if (om_str)
            printf(" Received OM : %s ", om_str);
        printf(" echo client invoke SUCCESSFUL! ");

        AXIS2_FREE(env->allocator, om_str);
        ret_node2 = NULL;
    }
    else
    {
        AXIS2_LOG_ERROR(env->log, AXIS2_LOG_SI, "Stub invoke FAILED: Err
or code:"
                " %d :: %s", env->error->error_number,
                AXIS2_ERROR_GET_MESSAGE(env->error));
        printf("echo client invoke FAILED! ");
    }

    if (svc_client)
    {
        axis2_svc_client_free(svc_client, env);
        svc_client = NULL;
    }

    if (env)
    {
        axutil_env_free((axutil_env_t *) env);
        env = NULL;
    }

    return 0;
}

/* build SOAP request message content using OM */
```

```
axiom_node_t *
build_om_payload_for_echo_svc(const axutil_env_t *env)
{
    axiom_node_t *echo_om_node = NULL;
    axiom_element_t* echo_om_ele = NULL;
    axiom_node_t* text_om_node = NULL;
    axiom_element_t * text_om_ele = NULL;
    axiom_namespace_t *ns1 = NULL;
    axis2_char_t *om_str = NULL;

    ns1 = axiom_namespace_create(env, "http://xxx.xxx.xxx", "ns1");
    echo_om_ele = axiom_element_create(env, NULL, "CheckServPWD", ns1, &echo_om_node);
    // 绑定参数
    text_om_ele = axiom_element_create(env, echo_om_node, "in0", ns1, &text_om_node);
    axiom_element_set_text(text_om_ele, env, "0351", text_om_node);

    text_om_ele = axiom_element_create(env, echo_om_node, "in1", ns1, &text_om_node);
    axiom_element_set_text(text_om_ele, env, "000", text_om_node);

    text_om_ele = axiom_element_create(env, echo_om_node, "in2", ns1, &text_om_node);
    axiom_element_set_text(text_om_ele, env, "0000", text_om_node);

    text_om_ele = axiom_element_create(env, echo_om_node, "in3", ns1, &text_om_node);
    axiom_element_set_text(text_om_ele, env, "123456", text_om_node);

    om_str = axiom_node_to_string(echo_om_node, env);

    if (om_str)
    {
        printf(" Sending OM : %s ", om_str);
        AXIS2_FREE(env->allocator, om_str);
        om_str =  NULL;
    }
    return echo_om_node;
└}
```

发表于 @ 2008年01月24日 18:18:00 | 评论(0) | 举报 | 收藏

旧一篇:Oracle存储过程实现多线程对表数据的抽取 | 新一篇:Retroweaver 项目演进中的助手

发表评论

表 情:

评论内容:

用 户 名: 匿名用户 登录 注册

验 证 码: MNQW 重新获得验证码

发表评论