



<?php

$API\_URL = 'http://localhost:85/d6s/api/';

// Server REST - user.login

$user\_data = array(

'username' => 'xmlrpc',

'password' => '5340940',

);

$curl = curl\_init($API\_URL . 'user/login.json');

curl\_setopt($curl, CURLOPT\_POST, 1); // Do a regular HTTP POST

curl\_setopt($curl, CURLOPT\_POSTFIELDS, http\_build\_query($user\_data)); // Set POST data

curl\_setopt($curl, CURLOPT\_HEADER, FALSE); // Ask to not return Header

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, TRUE);

curl\_setopt($curl, CURLOPT\_FAILONERROR, TRUE);

$response = curl\_exec($curl);

$http\_code = curl\_getinfo($curl, CURLINFO\_HTTP\_CODE);

if ($http\_code == 200) { $logged\_user = json\_decode($response); }

else {

$http\_message = curl\_error($curl);

die($http\_message);

}

// Server REST - node.create

$node\_data = array(

'title' => 'car 1',

'type' => 'car',

'field\_code[0][value]' => 'C1',

'field\_year[0][value]' => '2010',

'field\_manufacturer[0][value]' => 'VAZ',

'taxonomy[1]' => '1',

'language' => 'ru',

);

// Define cookie session

$cookie\_session = $logged\_user->session\_name . '=' . $logged\_user->sessid;

// cURL

$curl = curl\_init($API\_URL . 'node.json');

curl\_setopt($curl, CURLOPT\_POST, 1); // Do a regular HTTP POST

curl\_setopt($curl, CURLOPT\_POSTFIELDS, http\_build\_query($node\_data)); // Set POST data

curl\_setopt($curl, CURLOPT\_HEADER, FALSE); // Ask to not return Header

curl\_setopt($curl, CURLOPT\_COOKIE, "$cookie\_session"); // use the previously saved session

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, TRUE);

curl\_setopt($curl, CURLOPT\_FAILONERROR, TRUE);

$response = curl\_exec($curl);

$http\_code = curl\_getinfo($curl, CURLINFO\_HTTP\_CODE);

if ($http\_code == 200) {

$node = json\_decode($response);

}

else {

$http\_message = curl\_error($curl);

die($http\_message);

}

print\_r($node);

<http://drupal.org/node/783460>

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Net;

using System.IO;

using System.Collections.Specialized;

using RestSharp;

namespace d6s

{

public partial class Form1 : Form

{

public string nl = Environment.NewLine;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

d6sApi a = new d6sApi("xmlrpc", "5340940");

/\*d6sApiSession s = a.Login();

textBox1.Text += "Login" + nl + nl;

textBox1.Text += "sessid: " + s.sessid + nl;

textBox1.Text += "session\_name: " + s.session\_name + nl + nl;

d6sApiNodeCreate n = a.CreateCar("car from c#", ".net", 2005, "ZAZ");

textBox1.Text += "Node creation" + nl + nl;

textBox1.Text += "nid: " + n.nid + nl;

textBox1.Text += "uri: " + n.uri + nl + nl;\*/

string ci = a.CarIndex();

textBox1.Text += "Car index: " + ci + nl + nl;

string cc = a.CarCode("aaa");

textBox1.Text += "Car code: " + cc + nl + nl;

string ce = a.CarEcho();

textBox1.Text += "Car echo: " + ce + nl + nl;

}

}

//https://github.com/restsharp/RestSharp/wiki

public class d6sApi

{

const string BaseUrl = "http://localhost:85/d6s/api";

readonly string \_username;

readonly string \_password;

string \_sessid;

string \_session\_name;

public d6sApi(string username, string password)

{

\_username = username;

\_password = password;

}

public T Execute<T>(RestRequest request) where T : new()

{

var client = new RestClient();

client.BaseUrl = BaseUrl;

var response = client.Execute<T>(request);

//TODO: if response is ok

return response.Data;

}

public d6sApiSession Login()

{

RestRequest request = new RestRequest();

request.Resource = "user/login.json";

request.Method = Method.POST;

request.AddParameter("username", \_username);

request.AddParameter("password", \_password);

d6sApiSession session = Execute<d6sApiSession>(request);

\_sessid = session.sessid;

\_session\_name = session.session\_name;

return Execute<d6sApiSession>(request);

}

public d6sApiNodeCreate CreateCar(string title, string code, int year, string manufacturer) {

RestRequest request = new RestRequest();

request.Resource = "node.json";

request.AddCookie(\_session\_name, \_sessid);

request.Method = Method.POST;

request.AddParameter("title", title);

request.AddParameter("type", "car");

request.AddParameter("field\_code[0][value]", code);

request.AddParameter("field\_year[0][value]", year);

request.AddParameter("field\_manufacturer[0][value]", manufacturer);

request.AddParameter("taxonomy[1]", "1");

request.AddParameter("language", "ru");

return Execute<d6sApiNodeCreate>(request);

}

public string CarIndex()

{

RestRequest request = new RestRequest();

request.Resource = "car.json";

RestClient client = new RestClient();

client.BaseUrl = BaseUrl;

RestResponse response = client.Execute(request);

return response.Content;

}

public string CarCode(string code)

{

RestRequest request = new RestRequest();

request.Resource = "car/{code}.json";

request.AddParameter("code", code, ParameterType.UrlSegment);

RestClient client = new RestClient();

client.BaseUrl = BaseUrl;

RestResponse response = client.Execute(request);

return response.Content;

}

public string CarEcho()

{

RestRequest request = new RestRequest();

request.Resource = "car/echo.json";

RestClient client = new RestClient();

client.BaseUrl = BaseUrl;

RestResponse response = client.Execute(request);

return response.Content;

}

}

public class d6sApiSession {

public string sessid { get; set; }

public string session\_name { get; set; }

}

public class d6sApiNodeCreate {

public string nid { get; set; }

public string uri { get; set; }

}

}

<?php

function car\_service\_services\_resources() {

return array(

'car' => array(

'retrieve' => array(

'callback' => '\_car\_service\_retrieve',

'access callback' => 'user\_access',

'access arguments' => array('access content'),

'access arguments append' => FALSE,

'args' => array(

array(

'name' => 'code',

'type' => 'string',

'description' => 'The code of the car to get',

'source' => array('path' => '0'),

'optional' => FALSE,

),

),

),

'index' => array(

'callback' => '\_car\_service\_index',

'access callback' => 'user\_access',

'access arguments' => array('access content'),

'access arguments append' => FALSE,

),

'actions' => array(

'echo' => array(

'callback' => '\_car\_service\_echo',

'access callback' => 'user\_access',

'access arguments' => array('access content'),

'access arguments append' => FALSE,

),

),

),

);

}

function \_car\_service\_retrieve($code) {

return array('\_car\_service\_retrieve: ' . $code);

}

function \_car\_service\_index() {

return '\_car\_service\_index';

}

function \_car\_service\_echo() {

return '\_car\_service\_echo';

}

/\*

function noteresource\_perm() {

return array(

'note resource create',

'note resource view any note',

'note resource view own notes',

'note resource edit any note',

'note resource edit own notes',

'note resource delete any note',

'note resource delete own notes',

);

}

function car\_service\_services\_resources() {

return array(

'car' => array(

'retrieve' => array(

'help' => 'Simple retrieve',

'callback' => '\_tmp\_service\_retrieve',

'access callback' => '\_tmp\_service\_access',

'access arguments' => array('view'),

'access arguments append' => TRUE,

'args' => array(

array(

'name' => 'code',

'type' => 'string',

'description' => 'The code of the car to get',

'source' => array('path' => '0'),

'optional' => FALSE,

),

),

),

'create' => array(

'help' => 'Simple create',

'callback' => '\_tmp\_service\_create',

'access arguments' => array('note resource create'),

'access arguments append' => FALSE,

'args' => array(

array(

'name' => 'data',

'type' => 'struct',

'description' => 'The car object',

'source' => 'data',

'optional' => FALSE,

),

),

),

'update' => array(

'help' => 'Simple update',

'callback' => '\_tmp\_service\_update',

'access callback' => '\_tmp\_service\_access',

'access arguments' => array('update'),

'access arguments append' => TRUE,

'args' => array(

array(

'name' => 'id',

'type' => 'int',

'description' => 'The id of car to update',

'source' => array('path' => 0),

'optional' => FALSE,

),

array(

'name' => 'data',

'type' => 'struct',

'description' => 'The car object',

'source' => 'data',

'optional' => FALSE,

),

),

),

'delete' => array(

'help' => 'Simple delete',

'callback' => '\_tmp\_service\_delete',

'access callback' => '\_tmp\_service\_access',

'access arguments' => array('delete'),

'access arguments append' => TRUE,

'args' => array(

array(

'name' => 'id',

'type' => 'int',

'description' => 'The id of car to delete',

'source' => array('path' => 0),

'optional' => FALSE,

),

),

),

'index' => array(

'help' => 'Simple index',

'callback' => '\_car\_service\_index',

'args' => array(

array(

'name' => 'page',

'type' => 'int',

'description' => '',

'source' => array('param' => 'page'),

'optional' => TRUE,

'default value' => 0,

),

array(

'name' => 'parameters',

'type' => 'array',

'description' => '',

'source' => 'param',

'optional' => TRUE,

'default value' => array(),

),

),

),

),

);

}

function \_car\_service\_retrieve($code) {

return '\_car\_service\_retrieve\_' . $code;

}

function \_car\_service\_create($data) {

return '\_car\_service\_create\_' . print\_r($data, true);

}

function \_car\_service\_update($id, $data) {

return '\_car\_service\_update\_' . $id . '\_' . print\_r($data, true);

}

function \_car\_service\_delete($id) {

return '\_car\_service\_delete\_' . $code;

}

function \_car\_service\_index($page, $parameters) {

return array(

'hello' => 'world'

);

}\*/