Google Developer Day 2009

OpenSocial
应用开发技巧

吴伊自
06/05/2009

Google
Developer
Day2009

如何开发OpenSocial小应用

- 需要获取用户的信息
  - 通过Gadget模式或者RESTful模式的API
- 需要跟自己的后台进行交互
  - 通过makeRequest接口
- 需要一些开发和调试工具
  - OSDA，OSDE，Firebug，YSlow，Fiddler……
  - RESTful客户端开发包，如果要开发Flash，可以有AS3开发包
- 需要赚钱
  - 通过虚拟货币接口

主要内容

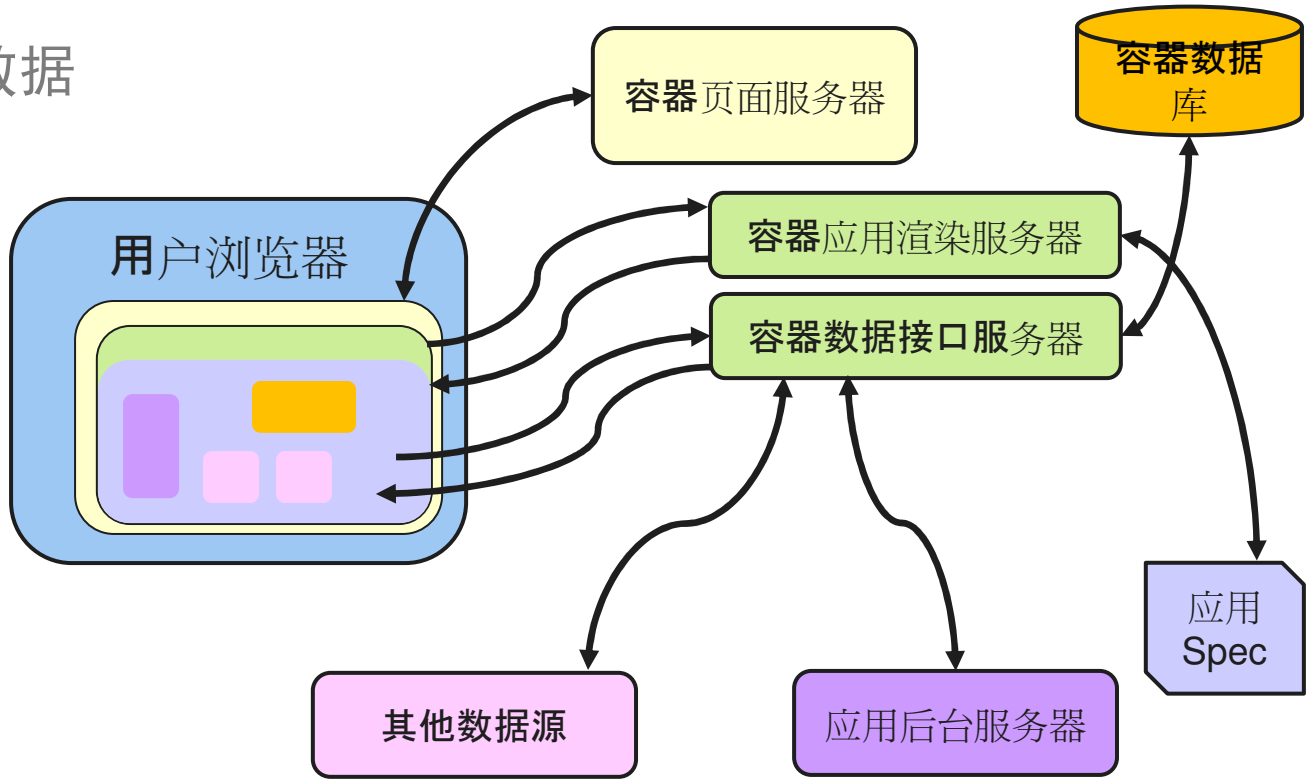
- **两种模式：Gadget与RESTful**
- **Gadget小应用**
  - OSDA与Javascript调试技巧
  - OSDE简介
  - 缓存数据
  - 签名的数据请求
- **RESTful小应用**
  - 认证与授权
  - 客户端开发包 Client Libraries
- **Flash应用开发包：AS3 Client Library**
  - 架构与使用说明
- **Virtual Currency API 虚拟货币接口简介**

Gadget模式与RESTful模式

Google
Developer
Day2009

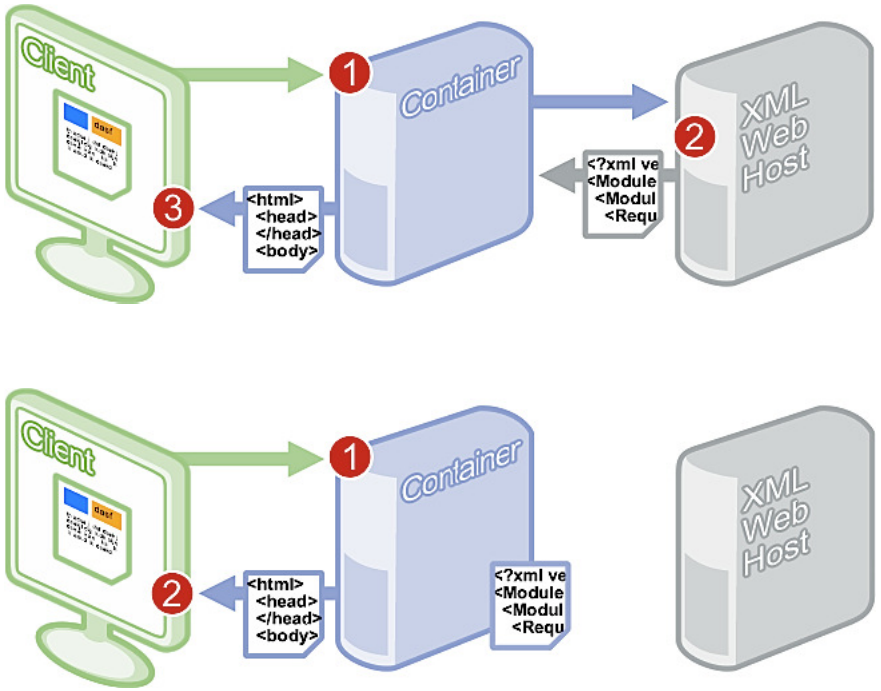
Gadget模式

- Gadget Spec XML
- 容器 Container
- 浏览器
- 其他远程数据



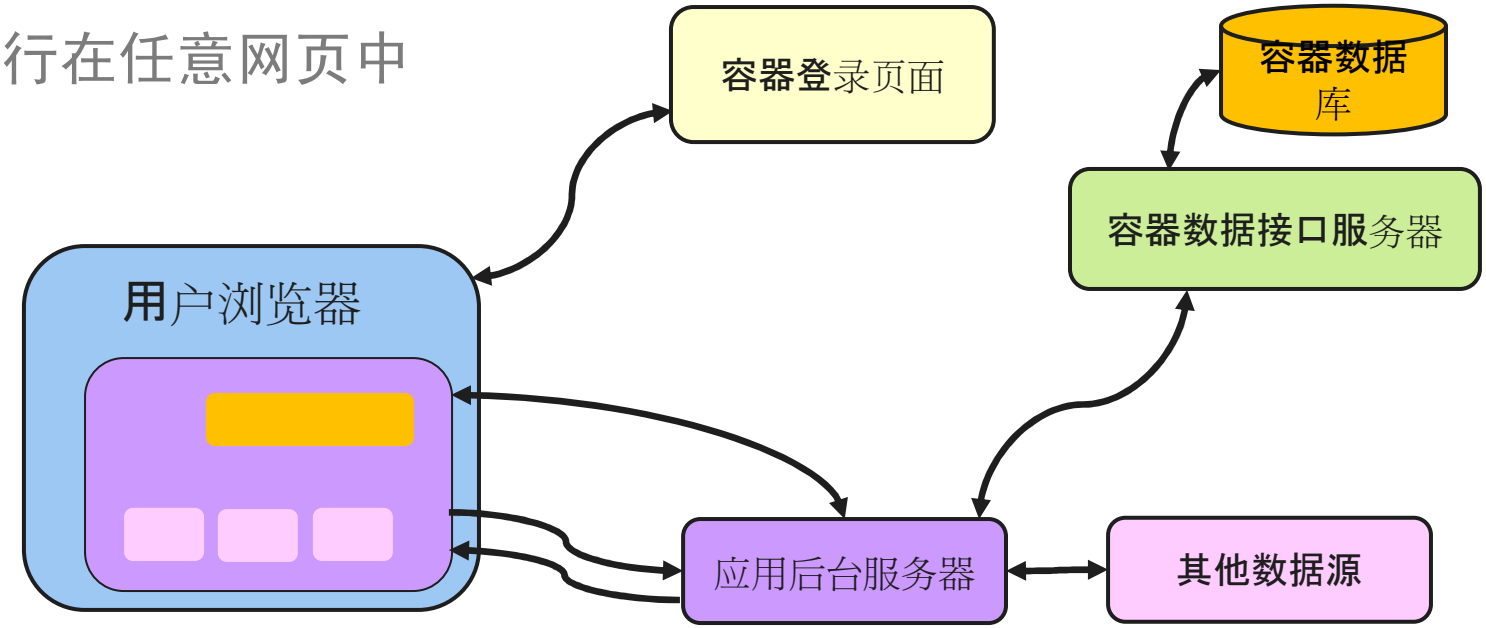
Gadget模式

- Gadget是一个HTML页面
- Gadget Spec是一个有特殊结构的XML，逻辑由Javascript实现
- Gadget Spec的渲染



RESTful模式

- 应用的RESTful客户端 与 容器的RESTful服务器
- CRUD数据请求
- 运行在容器的iframe环境中
- 运行在任意网页中



Gadget小应用

Google
Developer
Day2009

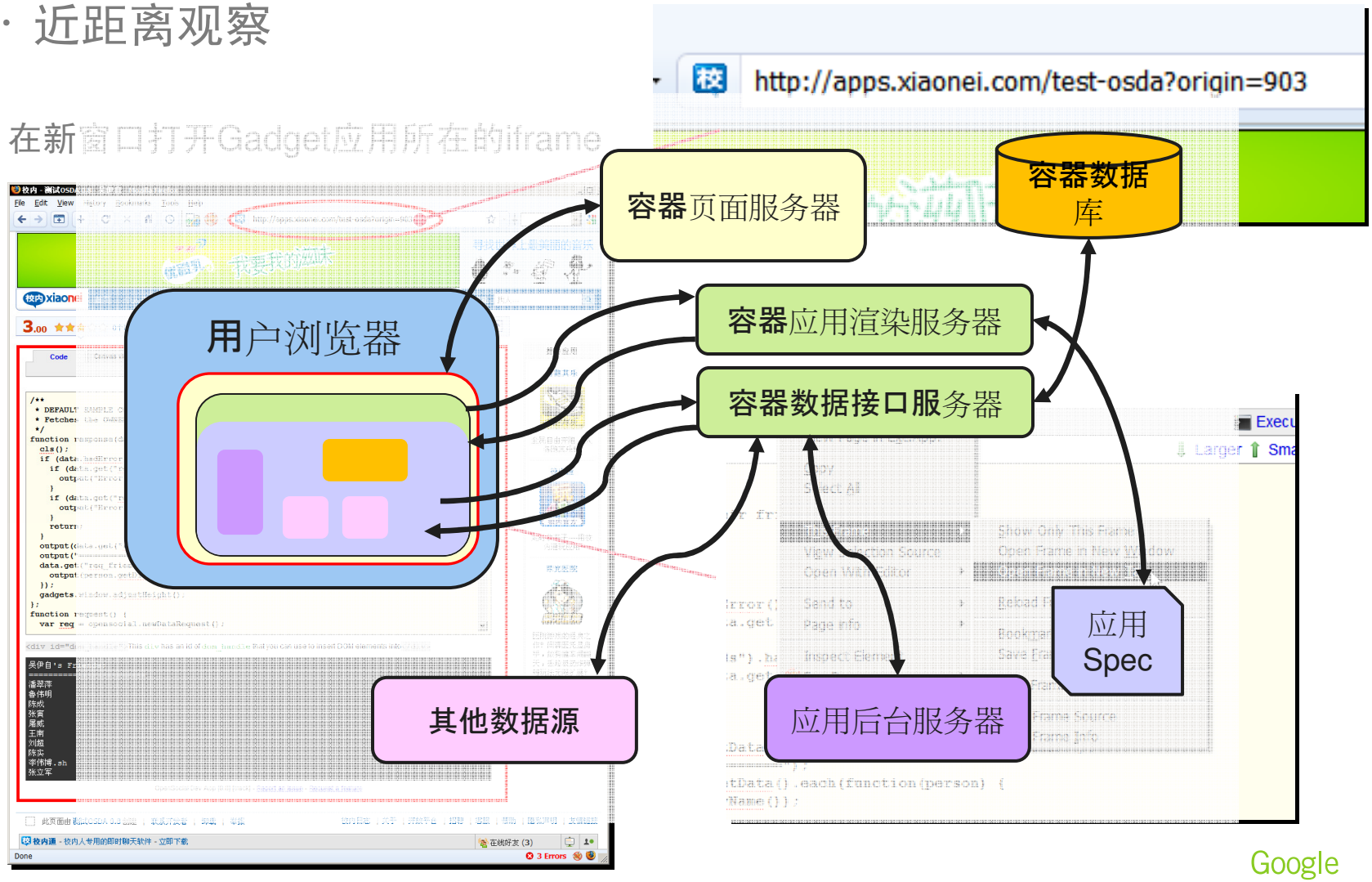
Gadget小应用开发

- 第一步：先看看数据
  - OSDA - Ope
  - 各

Google Developer Day2009

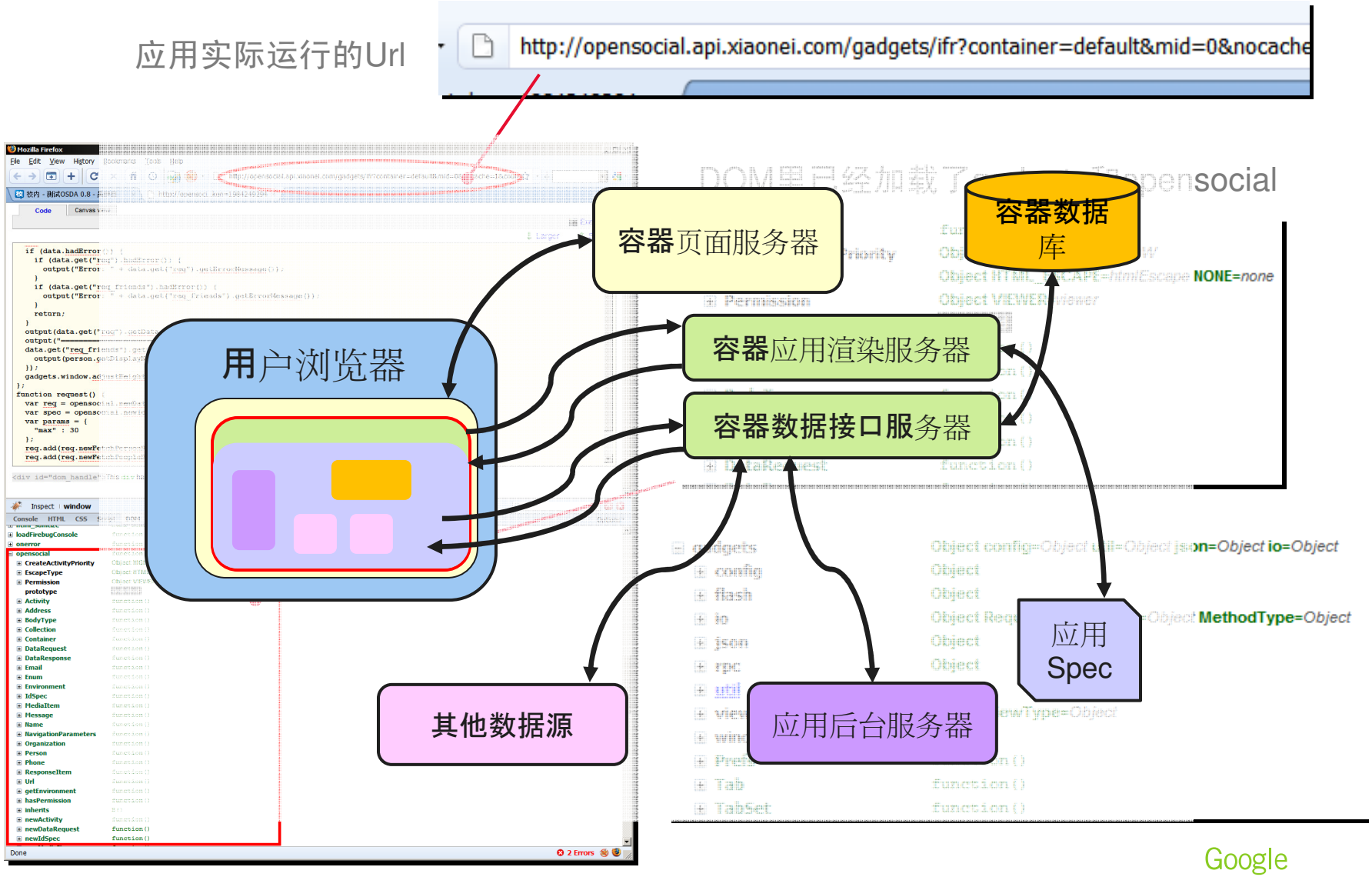
Gadget小应用开发

- 近距离观察

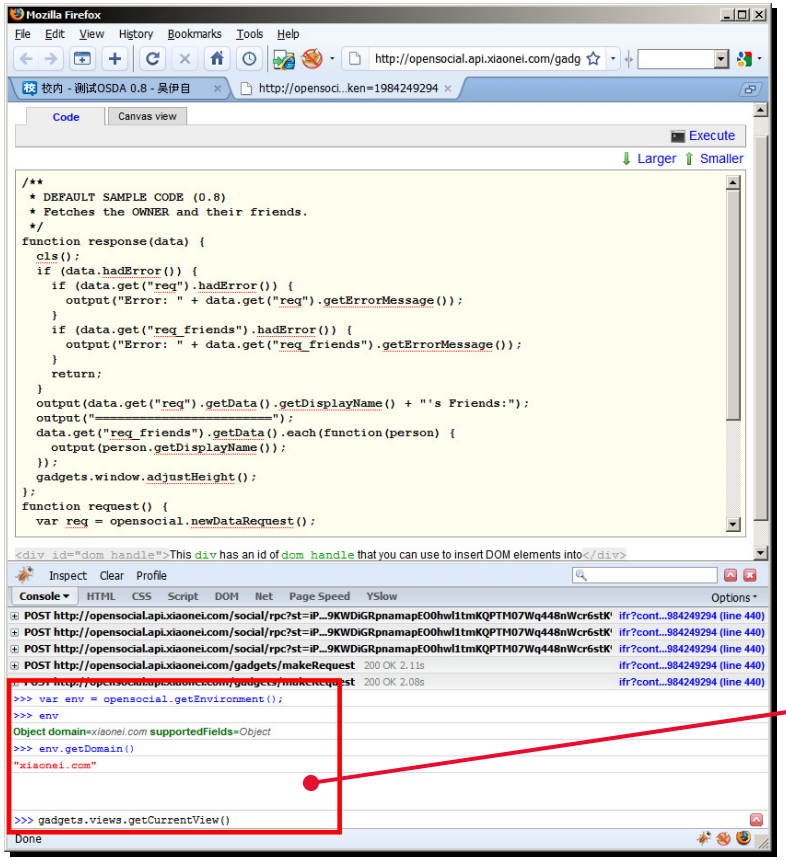


Gadget小应用开发

应用实际运行的Url



Gadget小应用开发

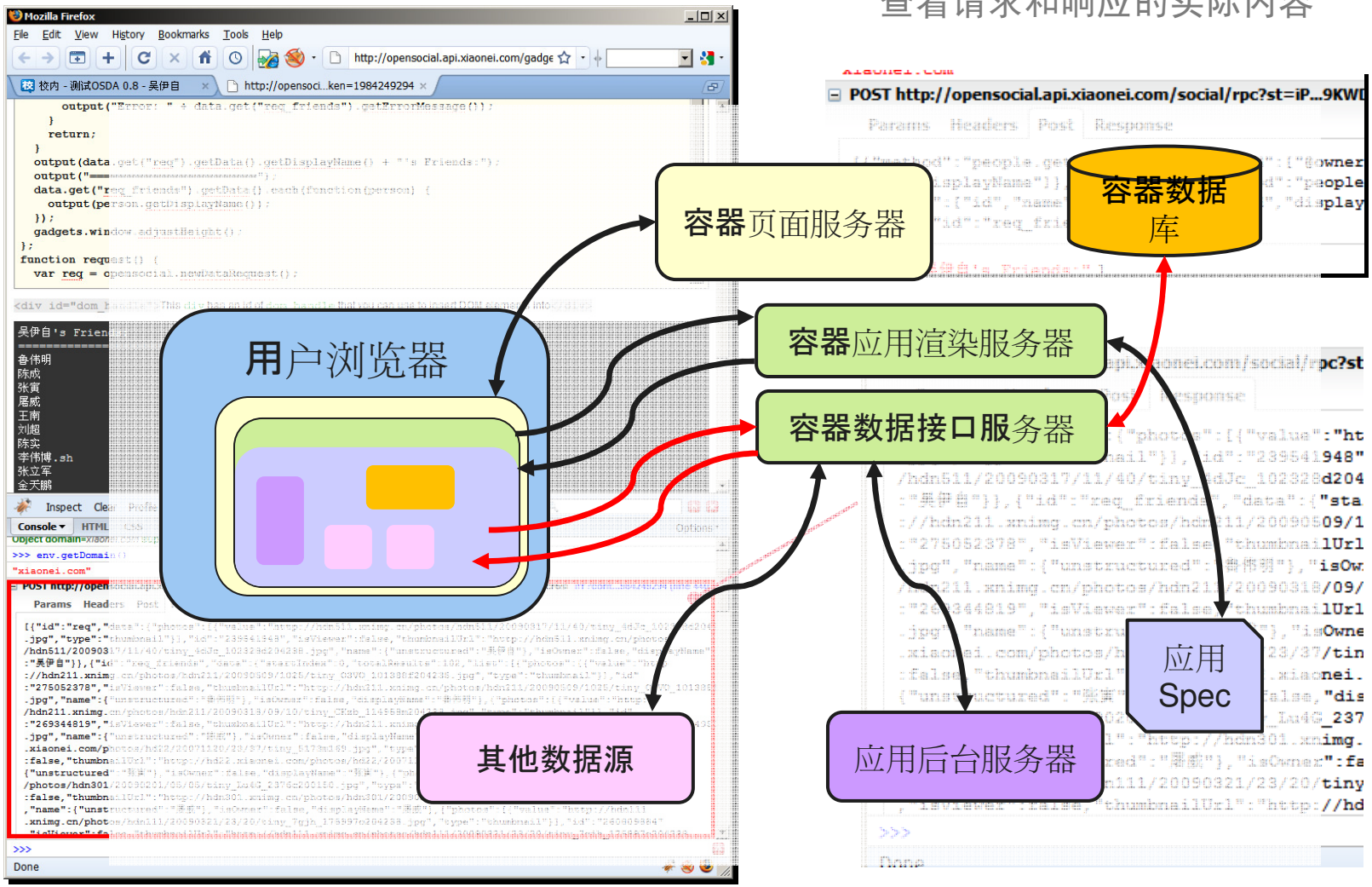


Console下直接可以执行Javascript语句和API

```
>>> var env = opensocial.getEnvironment();
>>> env
Object domain=xiaonei.com supportedFields=Object
>>> env.getDomain()
"xiaonei.com"

>>> gadgets.views.getCurrentView()
Done
```

Gadget小应用开发

发送数据请求后，在Console下查看请求和响应的实际内容



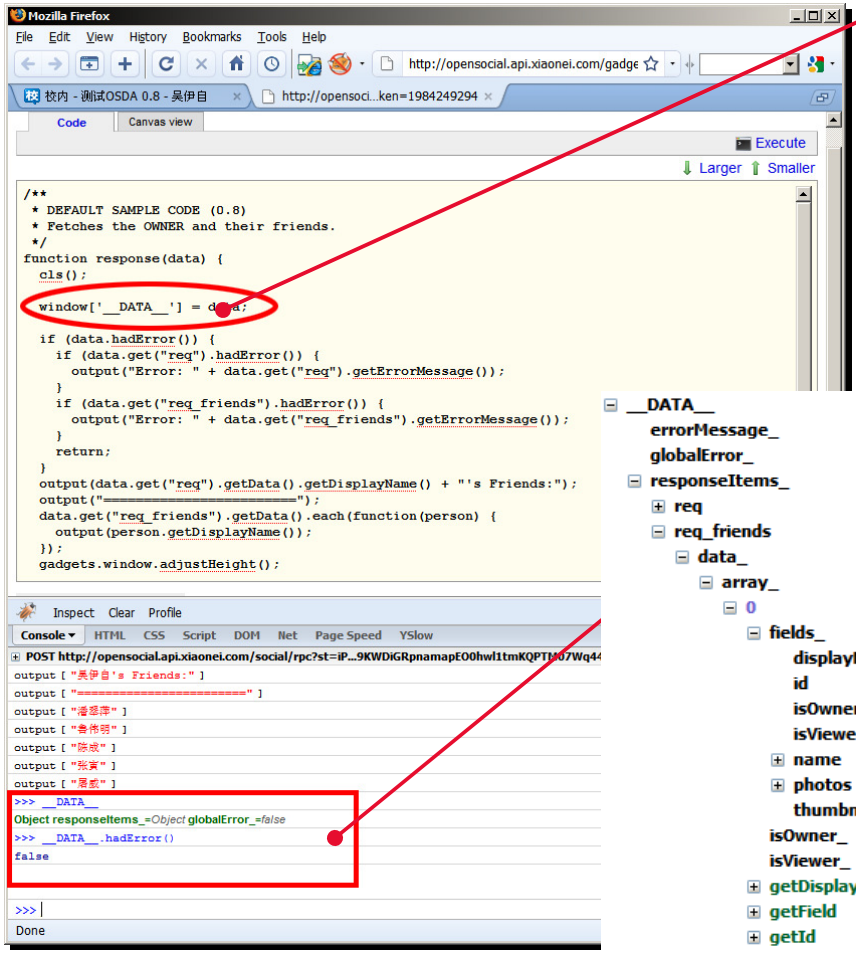
Gadget小应用开发

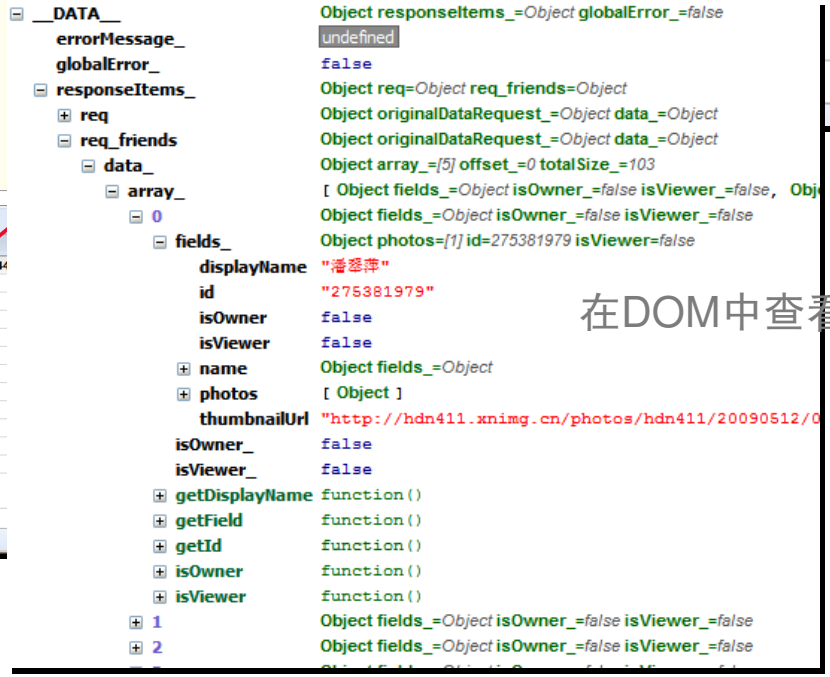
添加输出中间变量语句

```
cls();

window['__DATA__'] = data;

if (data.hadError()) {
```

在Console中操作输出的临时中间变量

```
>>> __DATA__
Object responseItems_=Object globalError_=false
>>> __DATA__.hadError()
false
```

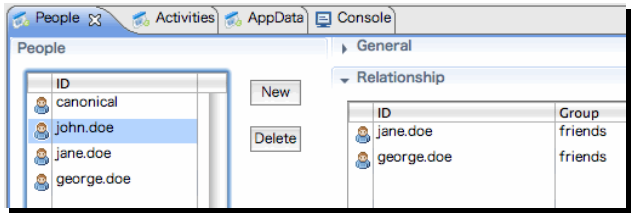


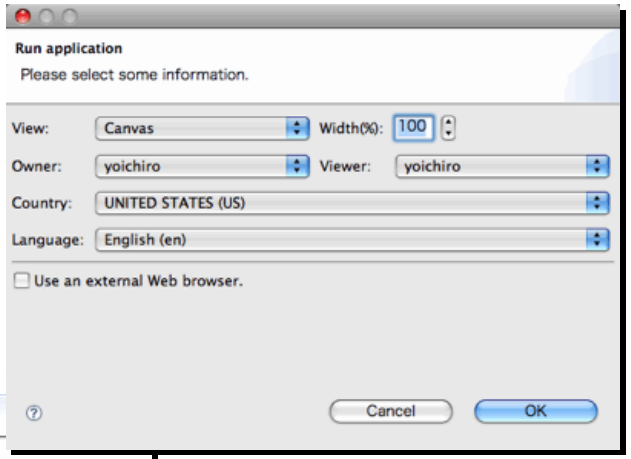


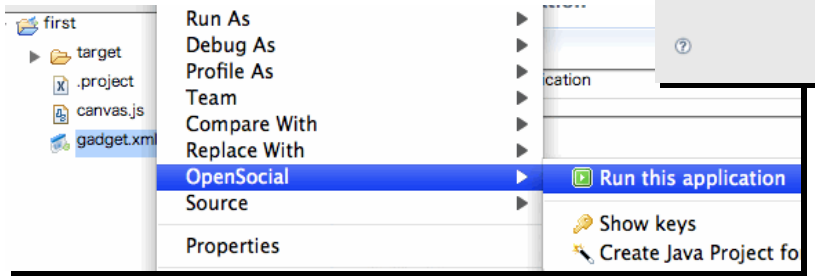
在DOM中查看临时中间变量

Gadget小应用开发

- OSDE - OpenSocial Development Environment
  - http://code.google.com/p/opensocial-development-environment/
  - OSDE是一个Eclipse插件，专门用于本地开发OpenSocial应用，不需要上传添加到容器
  - 快速入门：http://bit.ly/kiMcC







Google Developer Day2009

Gadget小应用开发

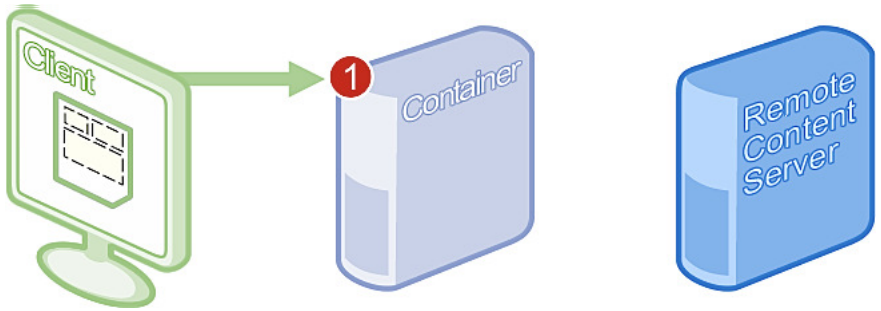
- 利用容器代理
  - refresh = 3600即一个小时，86400即一天

```
// In Javascript
var params = {“refreshInterval” : 3600 };
var proxiedImg = gadgets.io.getProxyUrl(“http://somepic.jpg”,
                                        params);

/* proxiedImg的值为:
 *   http://<container-app-domain>/gadgets/proxy?refresh=3600
 *   &url= http://somepic.jpg
 */
```

```
// In HTML
<link rel=“stylesheet”
  href=“/gadgets/proxy?refresh=86400&url=http://somestyle.css”>
<img src=“/gadgets/proxy?refresh=86400&url=http://somepic.jpg”>
```

Gadget小应用开发

- 远程数据请求
  - gadgets.io.makeRequest
  - 向任意的网页进行请求
    - 新闻类小应用 Feed阅读小挂件
  - 向自己的应用后台发送数据
    - 各种游戏资料与数据
  - 浏览器先向容器发送makeRequest请求，由容器向真正的目标数据进行请求并解析返回



Gadget小应用开发

- 保护应用的后台与数据
  - 用签过名的请求来访问自己的应用后台
    - 应用后台对接到的请求进行签名验证之后，可以确保请求来自特定的容器和特定的用户
    - 签名验证失败的请求可能经过恶意篡改
- 在Gadget中发起一个SIGNED makeRequest

```
// 这是要请求的自己服务器上的数据，只有该当前容器中的这个应用才可以访问
var server_url = “http://myserver/data”;
var params = {};                                    //要发送的参数
params[gadgets.io.RequestParameters.AUTHORIZATION] =
        gadgets.io.AuthorizationType.SIGNED; // 指定容器签名再发送
//发送请求
gadgets.io.makeRequest(server_url,
    function(data){
      //得到数据
    }, params);
```

Gadget小应用开发

- 在应用后台验证收到的签名请求
  - 使用OAuth.php包 （http://oauth.net/ 也有其他语言的包）

```
<?php
 require_once(“OAuth.php”);
 class MySignatureMethod extends OAuthSignatureMethod_RSA_SHA1 {
   protected function fetch_public_cert(&$request) {
        return “...”;  // 返回容器的公钥
   }   //以51为例，用此公钥 http://os.51.com/public.cer 验证
 }

 $request = OAuthRequest::from_request(null, null,
            array_merge($_GET, $_POST));
 $signature_method = new MySignatureMethod();

 // 验证签名
 @$signature_valid = $signature_method->check_signature(
      $request, null, null, $_GET[“oauth_signature”]);
```

Gadget小应用开发

```
// 验证结束，判断结果
$payload = array();
if ($signature_valid == true) {
  //成功，可以继续处理请求
  $payload[“validated”] = “Success! The data was validated”;
} else {
  //失败，不继续运行程序，直接返回401验证错误或403无权访问
  $payload[“validated”] = “This request was spoofed”;
}

$payload["get"] = $_GET;
$payload["post"] = $_POST;
echo var_export($payload);
?>
```

Gadget小应用开发

- 一次签名请求的各个参数

```
'validated' => 'Success! The data was validated',
'get' =>
 array (
  'oauth_nonce' => 'dbe8e68265d6313587921ecaf2d4964d',
  'oauth_timestamp' => '1242945860',
  'oauth_consumer_key' => '51.com',
  'container' => 'fo',
  'opensocial_owner_id' => 'dogcaptain',
  'opensocial_viewer_id' => 'dogcaptain',
  'opensocial_app_id' => '4263b53d89eaf36ecd43448412ef9fcd',
  'oauth_token' => '',
  'xoauth_signature_publickey' => 'http://os.51.com/public.cer',
  'oauth_signature_method' => 'RSA-SHA1',
  ‘oauth_signature’ => ‘aiaJsQvRXSeRD8dOC...Smtreg35U=',
 ),
'post' =>
 array (
 )
```

RESTful 小应用

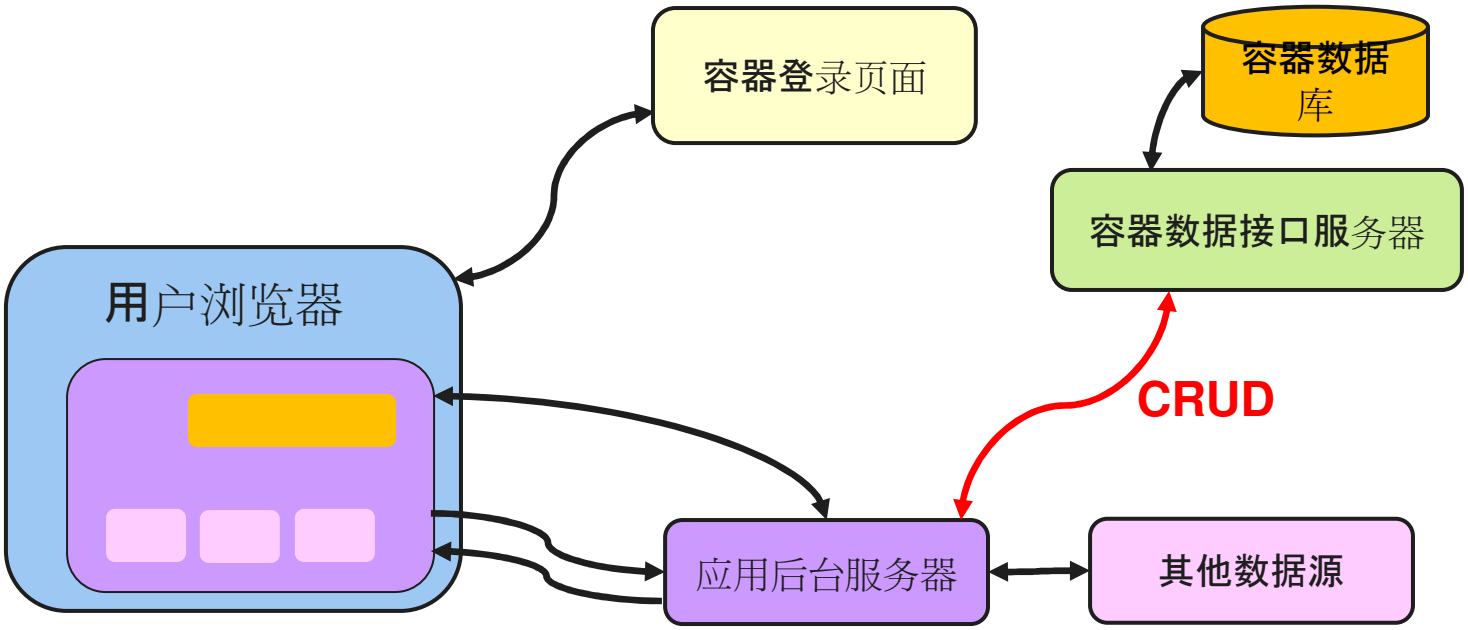
Google
Developer
Day2009

RESTful小应用

- 应用后台服务器朝容器发出请求，得到数据或者操作完毕之后应用后台渲染出界面，发送到用户的浏览器
- 数据资源都是URL，用HTTP请求进行交互，不同的HTTP方法表示不同的操作（CRUD，即创建，获取，更新以及删除）
- 返回格式常用的有JSON和XML

- 资源URL例如：
  - /people/@me/@self // 自己
  - /people/{guid}/@self // 某一个用户
  - /people/{guid}/@friends // 某一个用户的朋友

RESTful小应用



RESTful模式

- 认证与授权
  - 不论是应用后台还是容器，都应该验证收到的请求是否合法
  - OAuth 开放标准
  - 两腿OAuth
    - 认证容器与应用后台两者的身份
    - 应用可以获取任何指定用户的部分不需授权的公开信息
  - 三腿OAuth
    - 用户授权容器中用户信息给应用后台
    - 应用通过用户授权可以发布更新，获取好友列表等需授权的信息

RESTful小应用

- RESTful客户端开发包
  - OpenSocial **Java** Client
    - http://code.google.com/p/opensocial-java-client/
  - OpenSocial **PHP** Client
    - http://code.google.com/p/opensocial-php-client/
  - OpenSocial **Python** Client
    - http://code.google.com/p/opensocial-python-client/
  - OpenSocial **Ruby** Client
    - http://code.google.com/p/opensocial-ruby-client/
  - .NET, Objective C, …

RESTful小应用

- 以PHP Client Library为例
  - /osapi/*
  - /examples/*
  - 下载并放在PHP的路径底下
  - 通过PHP访问examples/listFriends?test=<container>来测试
    - examples/listFriends?test=partuza（三腿）
      - 需要跳转到Partuza页面由用户自行登录并授权
    - examples/listFriends?test=myspace（两腿）
      - 无需用户授权，用户在应用后台hardcoded指定

RESTful小应用

Myspace.com (2-legged)

Back to the index. Run this sample using data from: **MySpace**, orkut, orkut (REST), Partuza, Plaxo

List Friends Example

Request:

This sample fetched the current viewer and 2 friends, asking for the fields: *aboutMe, bodyType, currentLocation, drinker, happiestWhen, lookingFor*

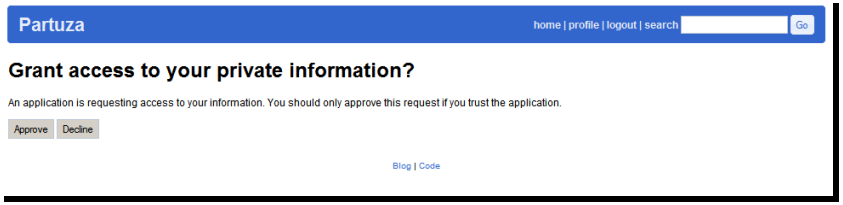
Response for the *self* request:

```
osapiPerson Object
(
    [aboutMe] =>
    [currentLocation] => Array
        (
            [country] => US
            [postalCode] => 43221
            [region] => Ohio
        )

    [displayName] => Resource
    [id] => myspace.com:439607992
    [lookingFor] => Array
        (
            [0] =>
        )

    [name] => Resource
    [isOwner] =>
    [isViewer] =>
)
```

Response for the *friends* request:

```
osapiCollection Object
(
    [list] => Array
        (
            [0] => osapiPerson Object
                (
                    [id] => myspace.com:6221
                    [nickname] => Tom
                    [profileUrl] => http://www.myspace.com/tom
                    [thumbnailUrl] => http://b2.ac-images.myspacecdn.com/00000/20/52/2502_s.jpg
                    [hasAppInstalled] =>
                    [isOwner] =>
                    [isViewer] =>
                )

            [1] => osapiPerson Object
                (
                    [id] => myspace.com:411981843
                    [nickname] => Matt
                    [profileUrl] => http://www.myspace.com/legider
                    [thumbnailUrl] => http://x.myspacecdn.com/images/no_pic.gif
                    [hasAppInstalled] => 1
                    [isOwner] =>
                    [isViewer] =>
                )

        )

    [startIndex] => 1
    [totalResults] => 5
    [itemsPerPage] => 2
    [filtered] =>
    [sorted] =>
    [updatedSince] =>
)
```

Partuza.nl (3-legged)



Back to the index. Run this sample using data from: MySpace, orkut, orkut (REST), **Partuza**, Plaxo

List Friends Example

Request:

This sample fetched the current viewer and 2 friends, asking for the fields: *aboutMe, bodyType, currentLocation, dr*

Response for the *self* request:

```
osapiPerson Object
(
    [displayName] => Yizi Wu
    [id] => 1510
    [lookingFor] => Array
        (
            [displayValue] =>
        )

    [name] => Yizi Wu
    [profileUrl] => http://www.partuza.nl/profile/1510
    [isOwner] => 1
    [isViewer] => 1
)
```

Response for the *friends* request:

```
osapiCollection Object
(
    [list] => Array
        (
        )

    [startIndex] => 0
    [totalResults] => 0
    [itemsPerPage] => 2
    [filtered] =>
    [sorted] =>
    [updatedSince] =>
)
```

RESTful小应用

- 为PHP Client Library添加一个容器的配置
  - 以51.com为例
  - 在 osapi/provides里新建一个 osapi51Provider.php，代码：

```
class osapi51Provider extends osapiProvider {
  public function __construct(
      osapiHttpProvider $httpProvider = null) {
    parent::__construct(
        null,
        null,
        null,
       “http://os.51.com/social/rest”, // 51的restful的路径地址
       “http://os.51.com/social/rpc”,  // 51的rpc的路径地址
       "51", true, $httpProvider);
  }
}
```

RESTful小应用

- 在同一目录下的osapiProvider.php里添加一句：

```
require_once "osapi51Provider.php";
```

- 然后去examples目录里修改__init__.php文件，增加一个case的选项：

```
case '51':
  $userId = 'dogcaptain';
  $osapi = new osapi(
    new osapi51Provider(),
    new osapiOAuth2Legged(
        "a16ef11ac341aa9da83ccc5453554ab0", //填写app public key
        "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx", //填写 app secret key
        $userId));
  break;
```

Flash / Flex开发包：AS3 Client Library

Google
Developer
Day2009

Flash/Flex开发包：AS3 Client Library

- OpenSocial **Java** Client
  - http://code.google.com/p/opensocial-java-client/
- OpenSocial **PHP** Client
  - http://code.google.com/p/opensocial-php-client/
- OpenSocial **Python** Client
  - http://code.google.com/p/opensocial-python-client/
- OpenSocial **Ruby** Client
  - http://code.google.com/p/opensocial-ruby-client/

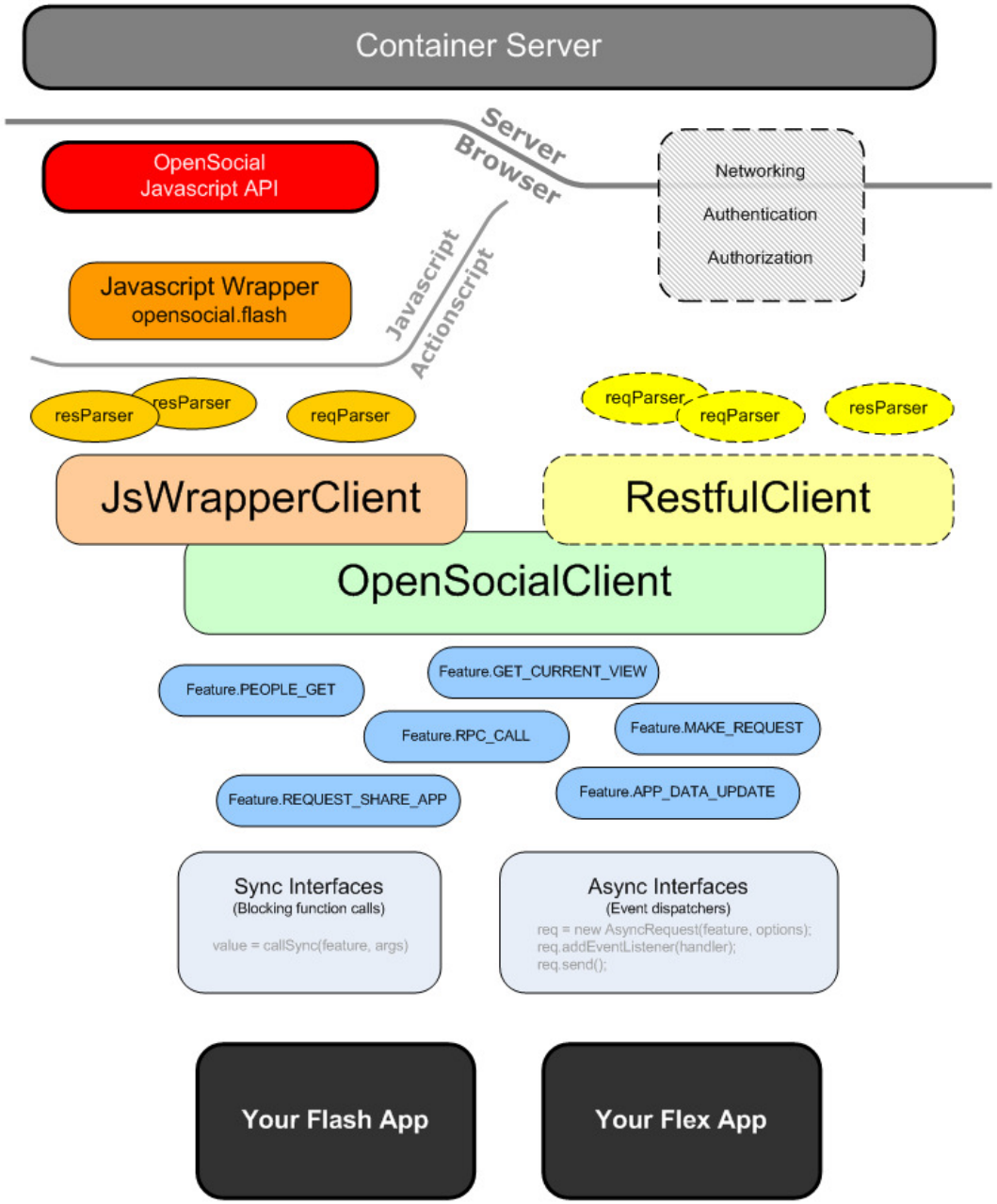
- **OpenSocial AS3 Client**
  - **http://code.google.com/p/opensocial-as3-client/**

Flash/Flex开发包：AS3 Client Library

- Flash小应用
  - Flash小应用既可以是Gadget模式的，也可以是RESTful模式的
  - 这个Client Library帮助开发者专注于Flash应用本身，而将OpenSocial各种调用模式，封装到简单的事件模型上
  - Gadget模式是通过ExternalInterface对OpenSocial Javascript API进行封装，因此几乎所有容器都适用
  - RESTful模式是直接从swf中发出符合OpenSocial格式的RESTful形式的数据请求，无需经过Javascript

Flash/Flex开发包：AS3 Client Library

- 关于开发包
  - 一个较完整的OpenSocial APIs，囊括了除批量发送外OpenSocial的所有编程接口(APIs)
  - 一个完整的事件驱动开发模型
  - 已经有一个基于FlexUnit和JsUnit的测试框架，以方便这个项目开源协作开发
  - 两类例子，分别对应于Flash和Flex的开发环境
  - 目前的版本还不支持RESTful模式



Flash/Flex小应用与AS3 Client Library

- JsWrapperClient使用说明
  - 下载整个包
  - 看看 /sample/ 下面的示例文件的代码
  - 把 /src/ 下面 org.opensocial.client.* 各源文件都加入项目中
  - 用 Flex 或者 Adobe Flash 编译出swf 文件
  - 将swf文件，附带的一个javascript文件，以及Gadget Spec Xml文件上传到任意可被访问的地方，修正xml中链接的改动
  - 在容器中开发者页面添加gadget spec xml文件
  - 如果开发已经完成，则尽可能的缓存所需的swf文件和js文件

```
var client:JsWrapperClient;

function init():void {
  // 初始化Client对象
  client = new JsWrapperClient();
  client.addEventListener(OpenSocialClientEvent.READY,
                          onReady);
  client.start();  // 等待Client初始化结束
}

function onReady(event:OpenSocialEvent):void {
  // 初始化结束，开始同步数据请求
  var helper:SyncHelper = new SyncHelper(client);
  var domain:String = helper.getDomain();
  var view:String = helper.getCurrentView();

  // 开始异步数据请求，请求获得本人信息，并侦听请求对象
  var req:AsyncDataRequest = new AsyncDataRequest(
          Feature.PEOPLE_GET,
          new PeopleRequestOptions()
              .setUserId("@me")
              .setGroupId("@self"));
  req.addEventListener(ResponseItemEvent.COMPLETE, handler);
  req.send(client);
}
```

Flash/Flex开发包：AS3 Client Library

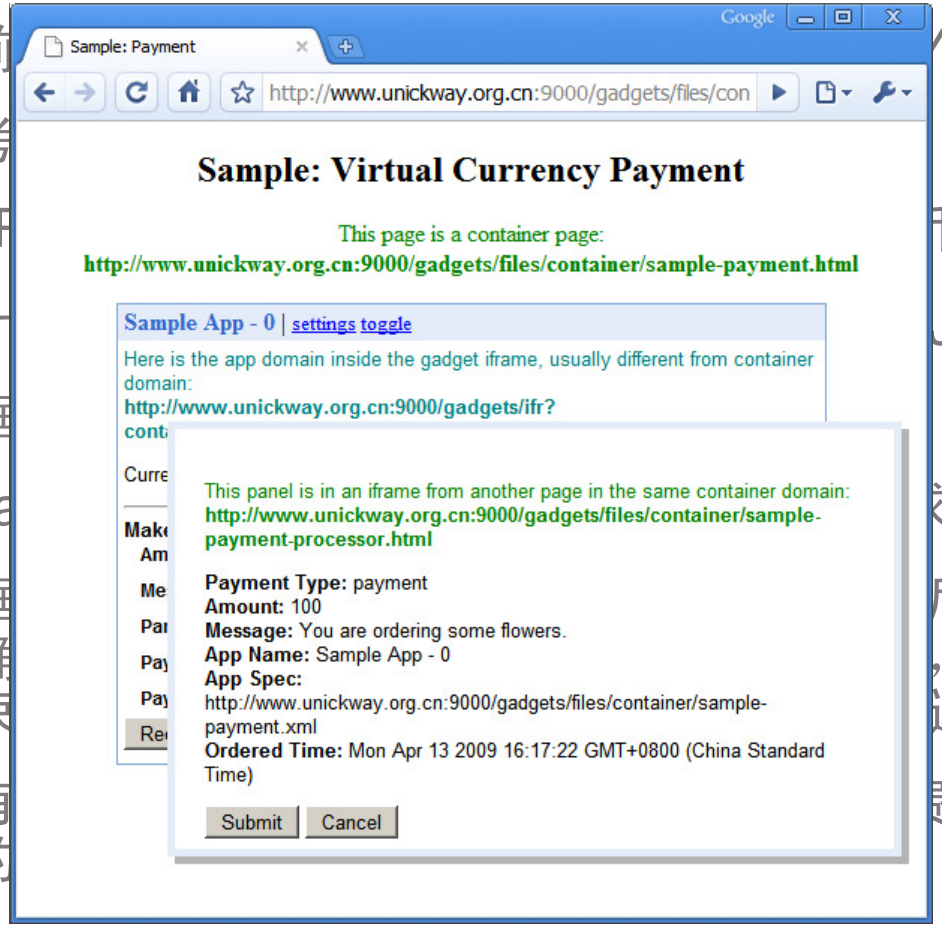
Google Developer Day2009

Virtual Currency API 虚拟货币接口

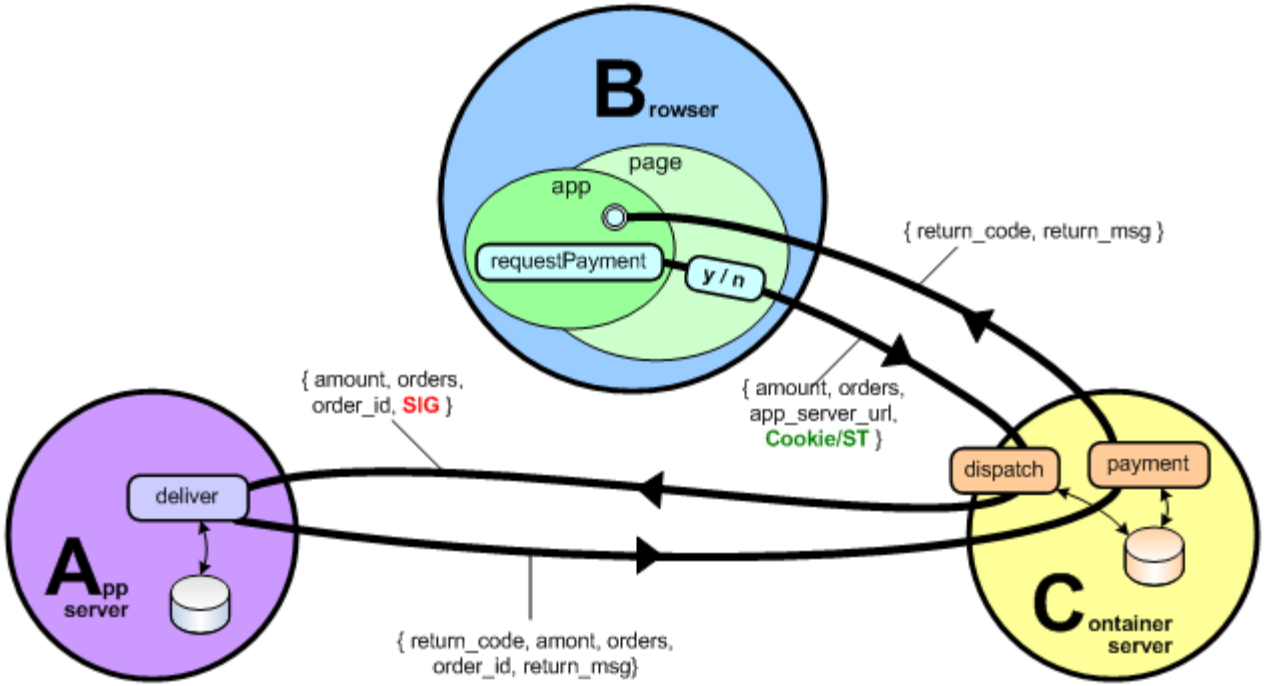
Google
Developer
Day2009

Virtual Currency API 虚拟货币接口

- OpenSocial虚拟货币是OpenSocial Spec的一个扩展
  - 草案当前 YbWtW)
  - 同时参考
  - Hi5上的P 币接口的应用
- Spec主要针 Javascript的
  - 采用弹框
  - 应用在Ja 求
  - 经过弹框 后会向应用的后台发出确 ，同时响应容器支持结束 返回给浏览器
  - 允许没有 要需求，比如按次计费的



Virtual Currency API 虚拟货币接口



Virtual Currency API 虚拟货币接口

- Gadget中的Javascript的接口

```
function requestPayment() {
  var paymentHandlerUrl = 'http: //my-app-backend';
  var params = {};
  params[opensocial.Payment.Field.AMOUNT] = 50;
  params[opensocial.Payment.Field.MESSAGE] = 'Buy some food';
  params[opensocial.Payment.Field.PARAMETERS] = {'bread', '5'};
  var payment = opensocial.newPayment(params);
  opensocial.requestPayment(payment, paymentHandlerUrl,
      function(responseItem) {
        if (responseItem.hadError()) {
          // 出错处理
          return;
        }
        var respPayment = responseItem.getData();
        var id = data.getField(
            opensocial.Payment.Field.ORDER_ID);
        var msg = data.getField(
            opensocial.Payment.Field.RESPONSE_MESSAGE);
        // 付款成功，更新界面
    });
};
```

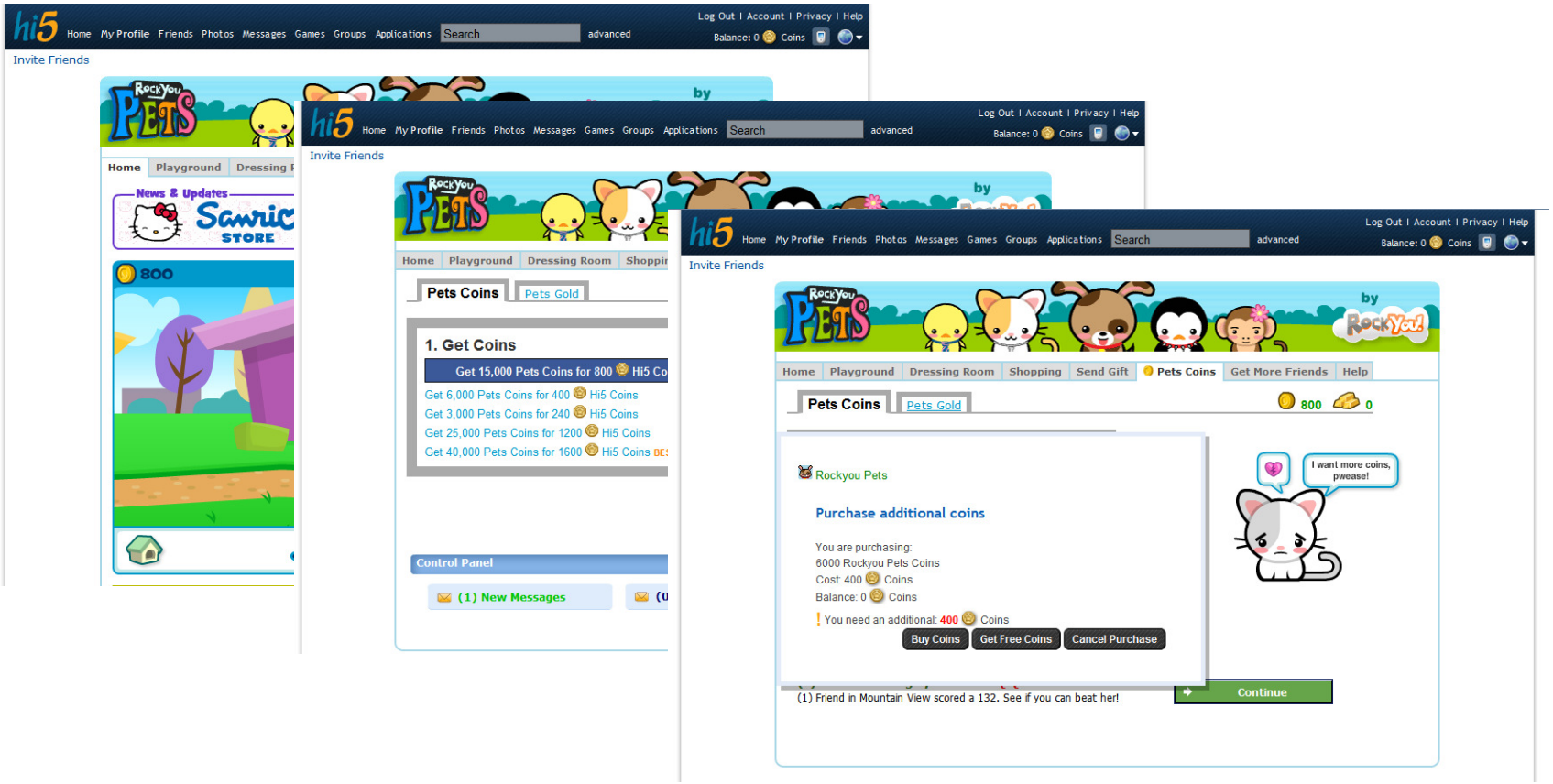
Virtual Currency API 虚拟货币接口

- 应用后台服务器的接口

```
<?php
 require_once("../oauth-handler.php");
 // 收到的是一个Oauth请求，需要对其签名进行验证
 if ($signature_valid == true) {
   $res = array();
   $amount = $payload["post"]["amount"];
  if (isOK($amount)) {
     // 收款和发货逻辑
     update_my_backend($payload["post"]["viewerId"],
                       $payload["post"]["parameters"]);
     $res[“responseMsg”] = “交易成功”
     $res["responseCode"] = "OK";
     echo json_encode($res);
   } else {
     // 处理出错
   }
 } else {
   header ("HTTP/1.0 400 Bad Request", true, 400);
   echo “<h1>Invalid OAuth Signature – 认证失败 </h1>";
 }
?>
```

Virtual Currency API 虚拟货币接口

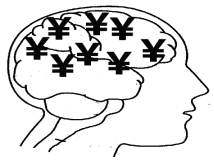
- RockYou Pets on hi5



Google Developer Day2009

总结一下

总结一下

	Javascript Gadget 小应用	RESTful 小应用	Flash/Flex Gadget 小应用	虚拟货币
Javascript / HTML / DHTML / AJAX	✓	✓		
PHP / Java Servlet	✓	✓	✓	
Actionscript / Flash UI			✓	
				✓

相关资源

- http://www.opensocial.org
- http://wiki.opensocial.org
- http://code.google.com/apis/opensocial
- http://groups.google.com/group/opensocial
- http://groups.google.com/group/opensocial-china

- 更多信息请参阅 http://code.google.com

谢谢

Google
Developer
Day2009

Google Developer Day 2009