

mixi アプリ VIEWER の情報表示

[[mixi アプリ](#)]

閲覧者情報の表示サンプル

The screenshot shows the mixi API Viewer interface. At the top, there are buttons for 'アプリトップ' (App Top), 'アプリを開く' (Open App), and 'マイアプリ解除' (Unlink My App). Below this, the title bar displays a yellow icon and the text 'ひろとさんのtest'. The main content area is a hierarchical table of properties and their values.

Property	Type	Value
getId	function	mixi.jp:25[REDACTED]
getDisplayName	function	ひろと
getField	function	undefined
isViewer	function	true
isOwner	function	true

Property	Type	Value
thumbnailUrl	string	http://profile.img.mixi.jp/photo/member/97/61/2599761_4254363890s.jpg
nickname	string	ひろと
lastLogin	string	2009-05-20T16:49:19Z

name	object	Property	Type	Value						
		getField	function	undefined						
		fields_	object	<table border="1"><thead><tr><th>Property</th><th>Type</th><th>Value</th></tr></thead><tbody><tr><td>unstructured</td><td>string</td><td>[REDACTED]</td></tr></tbody></table>	Property	Type	Value	unstructured	string	[REDACTED]
Property	Type	Value								
unstructured	string	[REDACTED]								

dateOfBirth	object	Property	Type	Value
		toJSON	function	[REDACTED]

isViewer	string	true
hasApp	string	true
aqe	number	[REDACTED]

ソース

重要 API

API	概要
newDataRequest	コンテナから友だちのデータ、プロフィール、アプリケーションデータ、アクティビティなどのソーシャル情報を要求するために使用
newFetchPersonRequest	指定した個人 ID のプロフィールを要求するアイテムを作成。Person オブジェクトを返す

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="hello world example">
    <Require feature="opensocial-0.8"/>
  </ModulePrefs>
  <Content type="html">
    <![CDATA[
      <script type="text/javascript">
        // 閲覧者情報取得要求
        function request() {
          var req = opensocial.newDataRequest();
          req.add(req.newFetchPersonRequest("VIEWER"), "viewer_data");
          req.send(response);
        }
        // 閲覧者情報取得応答
        function response(dataResponse) {
          var viewer = dataResponse.get("viewer_data").getData();
    
```

```

        document.getElementById("viewer_inf").innerHTML
            = printPropertiesAsTable(viewer, 0);
    }
    // 閲覧者情報表示
    function printPropertiesAsTable(obj, dpth) {
        if (dpth > 5) {
            return "<b>parse limit over</b>";
        }
        var tbl = "<table border='1'>";
        tbl += "<tr><th>Property</th><th>Type</th><th>Value</th></tr>";
        for (prop in obj) {
            var type = typeof(eval("obj." + prop));
            var blnkt = (type == "function")?"()":"";
            tbl += "<tr>";
            tbl += "<td>" + prop + "</td>";
            tbl += "<td>" + type + "</td>";
            if (type == 'object') {
                tbl += "<td>" + printPropertiesAsTable(eval("obj." + prop), dpth++) +
                "</td>";
            } else {
               tbl += "<td>" + eval("obj." + prop + blnkt) + "</td>";
            }
           tbl += "</tr>";
        }
       tbl += "</table>";
        return tbl;
    }
    gadgets.util.registerOnLoadHandler(request);
</script>
<div id="viewer_inf"/>
]]>
</Content>
</Module>

```