

Plaza Protocol Specification

Open Connected Gaming (version 0.1)

Introduction

Gaming knows many services meant for supporting developers with features like *Achievements*, matchmaking, rich presence and other connectivity features like *Steam*, *Xbox Live*®, and *PlayStation*® Network. However, no fully open platform for these features is available. Plaza aims to be an open protocol and platform for the implementation of such features.

Protocol basics

The Plaza protocol is mostly based on HTTP for transport and JSON and XML for data containers. Also, it is meant to be implementable without needing more than a simple web server and a basic configuration of (for example) PHP.

Automatically generated data uses the JSON format, and some server-to-server/client-to-server connections use the DBNRPC serialized format (see the specific chapter for details). Data that is meant to be able to be written by humans uses (non-validated, freeform) XML.

[... to be written ...]

Definitions

[... to be written ...]

Protocol documentation

Authentication and discovery

Discovering the service

For discovering the base request URI format for a Plaza request based on an index URI, an HTTP request (GET or HEAD) is sent to the specific index URI setting the **X-Plaza-Discovery** header to **server**. The response will then contain the **X-Plaza-Discovery** header set to the request URI format *as an absolute URI*.

```
GET /service/ HTTP/1.1
```

```
Host: plaza.example
```

```
X-Plaza-Discovery: server
```

```
User-Agent: libplaza/0.1
```

```
HTTP/1.1 200 OK
```

```
[..]
```

```
X-Plaza-Discovery: http://plaza.example/service/plaza/$call/
```

[..]

In this URL format, the following variables exist:

\$call: the call name

Obtaining the server manifest

Firstly, the server manifest XML should be obtained. This file contains important information such as server capabilities and other possibly required data. This is done by simply **GET**ing the data obtained by a service request using the call 'get_manifest'.

The file format for a server manifest is as in the following example:

```
<plaza-manifest version="1.0">
  <friendly-name>Plaza example server</friendly-name> <!-- the friendly
name, as configured by the server owner -->

  <supported-capabilities>
    <capability id="net.dotbas.plaza.remote-storage" /> <!-- id is a global
namespace ID of the capability -->
    <capability id="net.dotbas.plaza.user-statistics" />
    <!-- the delegate should be treated as if this server supports the
feature itself -- a remote auth ticket should be opened -->
    <capability id="example.provider.update-service"
delegate="http://cdn.example/" />
  </supported-capabilities>
</plaza-manifest>
```