

# 游戏玩家匹配

## 开发指南

### 产品文档



腾讯云

**【 版权声明 】**

© 2013–2022 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 商标声明 】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 服务声明 】**

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

**【 联系我们 】**

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

## 文档目录

### 开发指南

#### 匹配实现机制

- 匹配票据参数说明

- 多人组队匹配

- 队伍分配

#### 匹配使用流程说明

- 需要请求对战服资源的匹配

- 独立使用的匹配

#### 匹配规则

- 规则脚本设计指南

- 规则类型介绍

- 规则示例

#### 事件推送

- 推送机制介绍

- 事件推送安全

## 开发指南

## 匹配实现机制

## 匹配票据参数说明

最近更新时间：2022-03-29 15:28:48

### 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍游戏玩家匹配 GPM 的匹配票据参数，关于匹配票据的名词解释，详情可参见 [词汇表](#)。

### 状态说明

在 GPM 执行匹配请求的过程中，匹配票据将会经历状态的流转，状态会立即被更新在 **Status** 参数中。

#### 匹配票据状态的取值

- **SEARCHING**：匹配搜索中。此状态的票据，正在被 GPM 执行匹配搜索，GPM 将在匹配池中搜索满足规则约束的其他票据，并在超时时间内撮合出一个完整的匹配。如果在设定的超时时间内没有找到符合条件的其他票据，当前票据的状态将会流转为 **TIMEDOUT**。
- **PLACING**：匹配放置中。此状态的票据，已经搜索到完整的匹配，正在被 GPM 将匹配结果放置到游戏服务器 GSE 中，即在 GSE 中启动一个游戏服务器会话。并非所有票据都会经历此状态，只有配置了为匹配“请求 GSE 资源”的 MatchCode，其发起匹配的票据才会经历这个状态。
- **COMPLETED**：匹配已完成。此状态的票据，已经成功搜索到符合规则条件的完整匹配，并结束在 GPM 的生命周期。
  - 对于不通过 GPM 请求对战服资源的票据，此状态表示当前票据被撮合到一个完整的匹配。
  - 对于需要由 GPM 自动请求 GSE 对战服务资源的票据，此状态表示当前票据已经被撮合到一个完整的匹配，且已成功放置到 GSE 的指定队列上。
- **CANCELLED**：匹配已取消。此状态的票据，表示在 **SEARCHING** 阶段被用户主动请求取消。
- **TIMEDOUT**：匹配已超时。此状态的票据，表示在 **SEARCHING** 阶段未能成功，在设置匹配超时时间 TimeOut 时间内找到符合规则约束的其他票据形成完整的匹配。TimeOut 时间是 MatchCode 的参数，在 [创建匹配](#) 时定义。
- **FAILED**：匹配已失败。此状态的票据，表示在 **SEARCHING**、**PLACING** 阶段因为系统内部错误而失败。

### MatchResult 字段值解析

- MatchType 取值为 NORMAL 时，MatchResult 字段值请按如下协议进行解析：

#### 参数说明

字段名	类型	描述
MatchedPlayers	Array of MatchedPlayer	被匹配到同一对局的游戏玩家信息列表

#### 示例代码

```
"MatchResult": "{ \"MatchedPlayers\": [ { \"PlayerId\": \"xxxx\", \"PlayerSessionId\": \"\", \"MatchTicketId\": \"xxxxxxx\" }, { \"PlayerId\": \"xxxxx\", \"PlayerSessionId\": \"\", \"MatchTicketId\": \"xxxxxxx\" } ] }
```

- MatchType 取值为 GSE 时，MatchResult 字段值请按如下协议进行解析

#### 参数说明

字段名	类型	描述
MatchedPlayers	Array of MatchedPlayer	被匹配到同一对局的游戏玩家信息列表
DnsName	string	由 GSE 返回的 DNS 标志, 此字段可能为空
GameServerSessionId	string	由GSE返回的游戏服务器会话 ID
IpAddress	string	由 GSE 返回的 IP 地址
Port	number	由 GSE 返回的端口号

◦ 示例代码

```
"MatchResult": "{ \"DnsName\": \"\", \"GameServerSessionId\": \"gameserversession/fleet-xxx\", \"IpAddress\": \"xx.xx.x
x.xx\", \"Port\": xxx, \"MatchedPlayers\": [ { \"PlayerId\": \"xxx\", \"PlayerSessionId\": \"psess-xxxxxx\", \"MatchTicketId
\": \"xxx\" }, { \"PlayerId\": \"xxx\", \"PlayerSessionId\": \"psess-xxxxxx\", \"MatchTicketId\": \"xxxxxx\" } ] }"
```

• MatchedPlayer

字段名	类型	描述
PlayerId	string	玩家 ID, 用户发起匹配时传入
PlayerSessionId	string	玩家会话 ID, 由 GSE 返回。当 MatchType 取值为 NORMAL 时, 该字段为空
MatchTicketId	string	当前玩家所在的匹配票据 ID

# 多人组队匹配

最近更新时间：2022-03-29 15:29:00

## 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文用于说明玩家组队发起匹配时，GPM 的匹配实现机制。

## 发起组队匹配

在同一个 [发起匹配](#) 接口中传入的玩家，即为组队发起匹配的玩家。组队匹配的玩家将处于同一个匹配票据中。

### GPM 如何为组队的票据进行规则运算

#### • partyAggregation

在创建规则脚本时，可以通过“partyAggregation”参数的配置来定义组队玩家属性的计算方式。

有效取值	含义	作用的参数类型
avg	在计算当前规则时，用组队玩家该属性的平均值为票据中的每个玩家进行属性值替换	number 属性、地域延迟
min	在计算当前规则时，用组队玩家该属性的最小值为票据中的每个玩家进行属性值替换	number 属性、地域延迟
max	在计算当前规则时，用组队玩家该属性的最大值为票据中的每个玩家进行属性值替换	number 属性、地域延迟
each	在计算当前规则时，保留每个玩家各自的属性值	string 属性、number 属性、地域延迟
any	在计算当前规则时，随机选择组队玩家该属性的任意值，为票据中的每个玩家进行属性值替换	string 属性、number 属性、地域延迟

## 注意：

当组队的玩家选择不同的队伍时，描述玩家属性的规则中定义的 partyAggregation，仅在同一个票据内选择相同队伍的玩家之间生效。

## 示例

以如下规则配置为例：

```
{
  "version": "v1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number"
  }],
  "teams": [{
    "name": "red",
    "maxPlayers": 4,
    "minPlayers": 4
  }],
  "rules": [ {
```

```
"name": "TeamSkill",
"measurements": [
  "avg(teams[red].players.playerAttributes[skill])"
],
"referenceValue": 30,
"maxDistance": 10,
"partyAggregation": "max"
}]
}
```

以上规则中，一个成功的匹配需要找到4个玩家组成 red 队。要求 red 队玩家 skill 属性的平均值与参考值30相差10以内。TeamSkill 规则的 partyAggregation 配置为“max”，表示在计算 TeamSkill 规则时，用组队玩家 skill 的最大值为票据中的每个玩家进行属性值替换。示例如下：

- 假设 MatchTicket1 中包含3个组队匹配的玩家：player\_A、player\_B、player\_C。
- 假设 MatchTicket2 中包含1个组队匹配的玩家：player\_D。
- 4个玩家的 skill 属性分别传入 A (20)、B (30)、C (35)、D (30)。

可以看出  $\max(A, B, C) = 35$ ， $\max(D) = 30$ 。则计算 MatchTicket1、MatchTicket2 是否满足 TeamSkill 规则的匹配条件时，GPM 会先进行如下替换：

- A'(35)、B'(35)、C'(35)、D'(30)

因为  $\text{avg}(A', B', C', D') = 33.75$ ，满足 TeamSkill 规则的条件设定。所以 MatchTicket1、MatchTicket2 可以被匹配成功。

# 队伍分配

最近更新时间：2022-03-29 15:29:06

## 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文以如下规则为例，说明 GPM 如何进行队伍 team 分配。

## 示例规则

以下规则表示一个完整的匹配包含1个 red 队，2个 blue 队。两种类型队伍的人数都为3，即 3v3v3 的对局结构。

```
{
  "version": "v1.0",
  "teams": [{
    "name": "red",
    "maxPlayers": 3,
    "minPlayers": 3
  }, {
    "name": "blue",
    "maxPlayers": 3,
    "minPlayers": 3,
    "number": [2,2]
  }]
}
```

### 场景一：玩家不指定 team

#### • 单人发起匹配

假如玩家 A 单独发起匹配，匹配请求中的 team 参数传入空值，则 GPM 有可能将其分入红1队、蓝1队、蓝2队任意之一。匹配完成后，玩家 A 的 team 参数将被更新为"red\_000"或"blue\_000"或"blue\_001"。

#### • 多人组队匹配

假如玩家 A、B 组队发起匹配，匹配请求中的 team 参数均传入空值，则 GPM 会将 A、B 匹配进同一个队伍。玩家 A、B 的 team 参数将被同时更新为"red\_000"或"blue\_000"或"blue\_001"。

### 场景二：玩家指定 team

#### • 单人发起匹配

假如玩家 A 单独发起匹配，匹配请求中的 team 参数传入 red，则 GPM 只会将其分入红1队。匹配完成后，玩家 A 的 team 参数将被更新为"red\_000"。

#### • 多人组队匹配

假如玩家 A、B 组队发起匹配，匹配请求中的 team 参数均传入 blue，则 GPM 只会将玩家 A、B 同时分入蓝1队或蓝2队。玩家 A、B 的 team 参数将被同时更新为"blue\_000"或"blue\_001"。



 注意:

GPM 暂不支持组队匹配的玩家指定不同的队伍类型。

# 匹配使用流程说明

## 需要请求对战服资源的匹配

最近更新时间：2022-03-29 15:29:16

### ⚠ 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

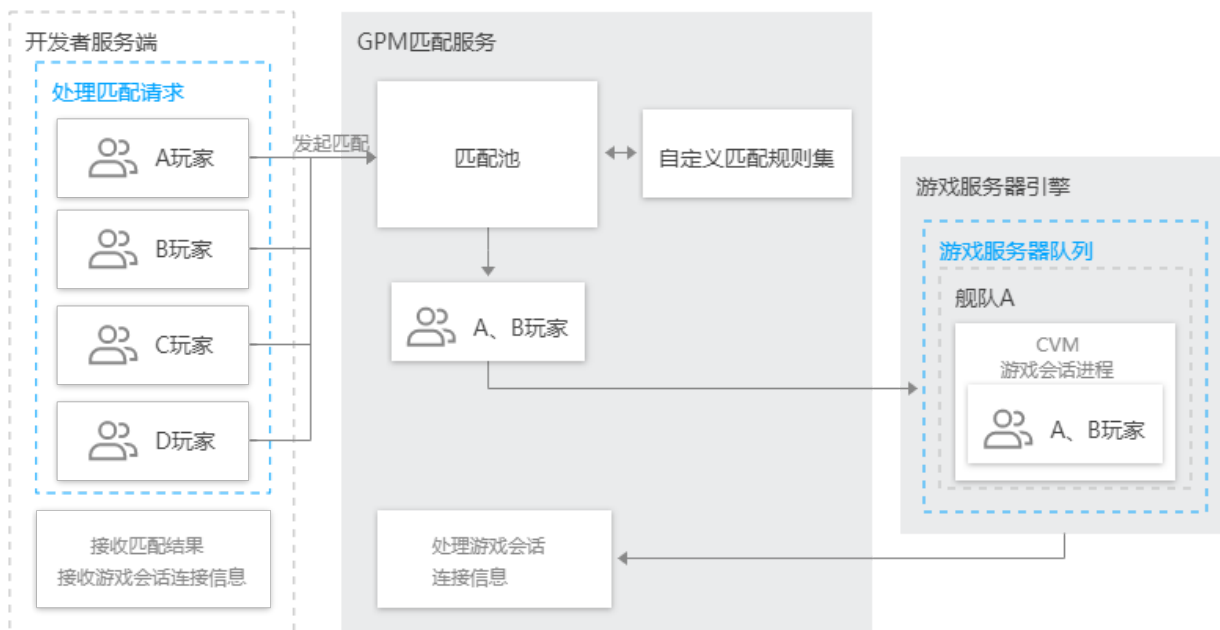
本文向您介绍如何搭配游戏服务器伸缩 GSE 使用游戏玩家匹配 GPM 服务，使您拥有更便捷的游戏开发体验。

## 前提条件

- 已在游戏服务器伸缩 GSE 完成 [游戏服务器队列创建](#)。
- 已完成 [匹配创建](#)，并选择“请求对战服资源”类型进行匹配创建。

## 流程架构

搭配游戏服务器伸缩 GSE 使用游戏玩家匹配 GPM 流程架构如下所示：

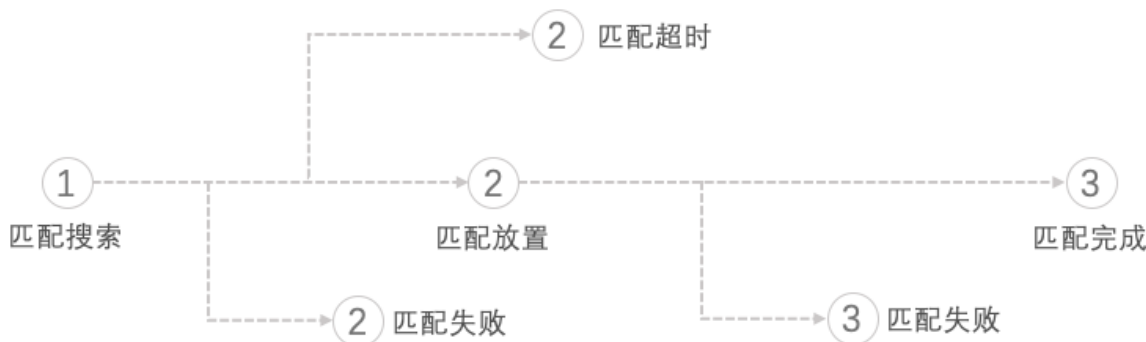


## 匹配状态流转

为 GPM 匹配结果请求 GSE 资源，您的匹配票据状态将会经历如下过程：

### 🔍 说明：

关于匹配票据的详细说明，详情请参见 [匹配票据参数说明](#)。



## 操作步骤

### 1. 在客户端服务器发起匹配请求

您需要准备一个后端服务来处理来自客户端的玩家请求，并通过此客户端服务器调用 GPM 的 [StartMatching](#) 接口，向 GPM 请求匹配服务。一次成功的 [StartMatching](#) 调用，将会生成一个唯一的匹配票据，您需要跟进这个匹配票据的状态，来完成您的匹配流程。

您也可以通过客户端服务器调用 GPM 的 [CancelMatching](#) 接口，用于取消正在匹配中的匹配票据。

### 2. 通过匹配事件通知，跟进匹配状态

成功发起匹配后，您配置过接收事件通知的地址，将会收到来自 GPM 的匹配事件推送。您可以设置一个定时器，周期性的处理事件推送。关于不同类型匹配事件推送的详细介绍，详情请参见 [事件推送](#)。

### 3. 处理已完成的匹配票据

- 对于匹配完成的票据，GPM 已为当前票据在游戏服务器队列上启动一个游戏服务器会话，并将游戏连接信息填充至匹配票据的对应字段中。
- 您需要解析 [推送协议](#) 里的匹配票据信息，以获得游戏连接信息，以及被匹配到同一个对战的玩家 ID、玩家会话 ID。
- 您的客户端服务将解析出的游戏连接信息和玩家会话 ID 信息通知到玩家客户端，玩家客户端根据这些信息接入已启动的 GSE 游戏服务器会话。如何将玩家从客户端连接到 GSE 的游戏服务器会话，详情请参见 [客户端连接 GSE 的 gRPC 服务端](#)。
- 对于 [匹配取消](#)/[匹配超时](#)/[匹配失败](#) 的票据，您可以根据实际的业务逻辑自行对玩家进行通知，或在客户端服务器上为票据内的玩家重新发起匹配。

### 4. (建议) 利用查询匹配进度接口，作为匹配事件推送的补充机制

如果您没有配置接收匹配事件进度的服务器地址，或因为某些原因无法接受到匹配事件通知，您可以通过 GPM 的 [DescribeMatchingProgress](#) 接口来定向跟进指定匹配票据的状态。关于不同匹配票据的不同状态说明，详情请参见 [匹配票据状态说明](#)。

## 相关问题

### 1. 玩家的地域延迟数据如何作用于 GPM 和 GSE?

- GPM 使用地域延迟**
  - 如果 GPM 的规则脚本中没有配置延迟策略，则 GPM 不会使用到玩家地域延迟数据。
  - 如果 GPM 的规则脚本中配置了延迟策略，则 GPM 匹配出的对局中的所有玩家，将满足至少有一个共同地域能符合规则中描述的延迟策略。

关于 GPM 使用地域延迟数据的详细说明，详情请参见 [GPM 延迟规则](#)。

- GSE 使用地域延迟**

- 如果用户发起匹配时没有传入玩家延迟数据，则 GSE 将按照 queue 中 fleet 的优先级来选择放置的 fleet。
- 如果用户发起匹配时传入了玩家延迟数据，则 GSE 对于玩家延迟的用法，取决于用户是否给 queue 配置了延迟策略。
  - 未配置延迟策略：GSE 将在玩家共有的区域中，按 fleet 优先级放置。
  - 已配置延迟策略：GSE 将在玩家共有的区域中，按延迟策略进行放置；当出现同时符合延迟策略的多个区域，GSE 将按 fleet 优先级选择。

关于 GSE 使用地域延迟数据的详细说明，详情请参见 [GSE 就近调度](#)。

2. GPM 中配置的匹配超时时间，和 GSE 的 queue 中配置的超时时间，有什么区别？

• GPM中配置的匹配超时

← 创建匹配

The screenshot shows the '创建匹配' (Create Match) configuration page. It includes fields for '名称' (Name) set to 'Test', '描述 (选填)' (Description), and '关联规则' (Associated Rules). The '匹配超时时间' (Match Timeout) field is highlighted with a red box, showing a value of 30 seconds. Below it, a note states '可配置范围1-600秒' (Configurable range 1-600 seconds). There is also a link to '创建规则' (Create Rule).

在 GPM 中配置的匹配超时时间，表示 GPM 执行匹配搜索的最长时间，即 MatchTicket 的状态为 SEARCHING 的最长时间；在这个时间内，如果找不到其他玩家组成对局，此 MatchTicket 的状态将会流转为 TimedOut。

• GSE 的 queue 中配置的分配超时时间

← 新建游戏服务器队列

The screenshot shows the '新建游戏服务器队列' (New Game Server Queue) configuration page. Under the '基本信息' (Basic Information) section, the '标识符' (Identifier) is 'TestQueue'. The '分配超时时间' (Allocation Timeout) field is highlighted with a red box, showing a value of 600 seconds. Below it, there are fields for '标签 (选填)' (Tag) with a dropdown for '标签键' (Tag Key) and a dropdown for '标签值' (Tag Value).

在 GSE 的 queue 中配置的分配超时时间，表示 GSE 收到匹配放置请求后，在 queue 中搜索符合条件的 fleet 的最长耗时。如果超过这个超时时间，GSE 没有搜索到可以放置的 fleet，GPM 会将此次放置中涉及的所有 MatchTicket 的状态流转为 Failed。

🔍 说明：

匹配全流程的实际耗时，主要包括 GPM 的 searching 时间，加上 GSE 的放置时间。

# 独立使用的匹配

最近更新时间：2022-03-29 15:29:22

**注意：**

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

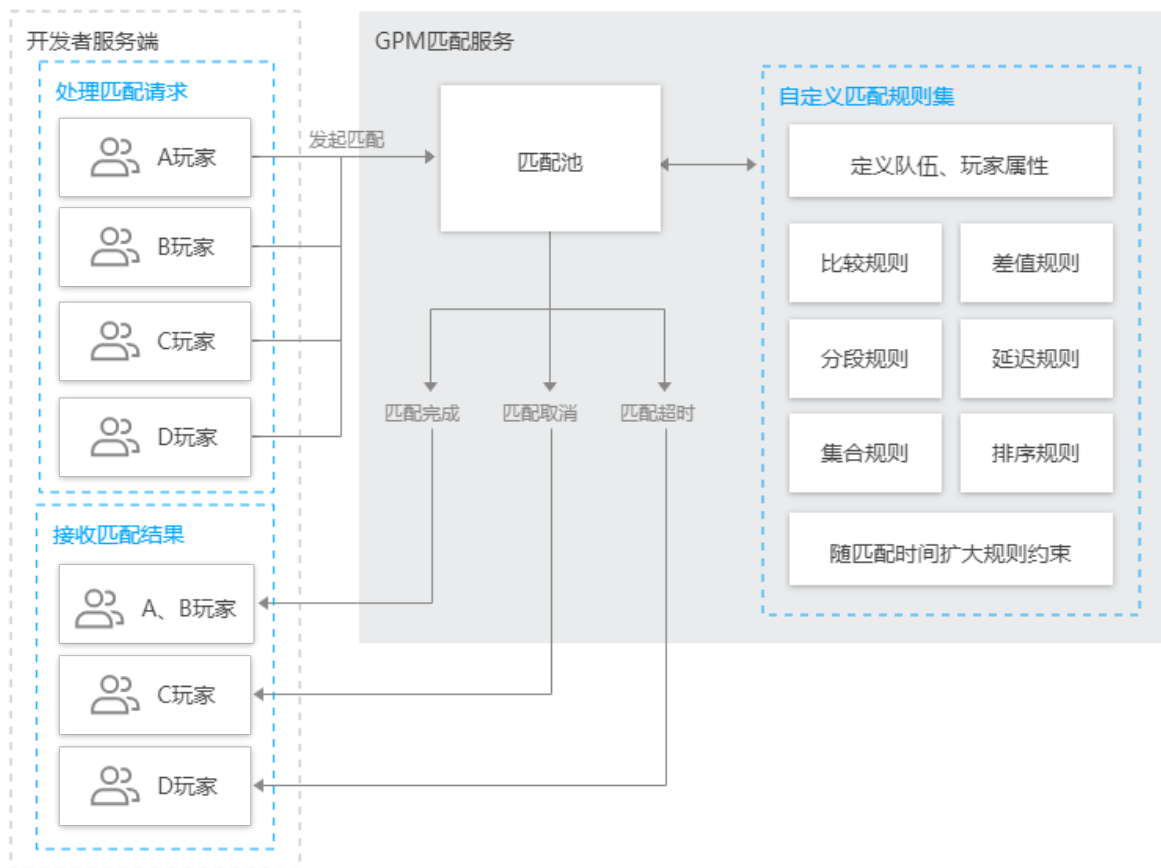
本文向您介绍如何独立使用 GPM 匹配服务。

## 前提条件

已完成 [匹配创建](#)，并选择“**无需自动连接对战的匹配**”类型进行匹配创建。

## 流程架构

独立使用 GPM 匹配服务架构图如下：

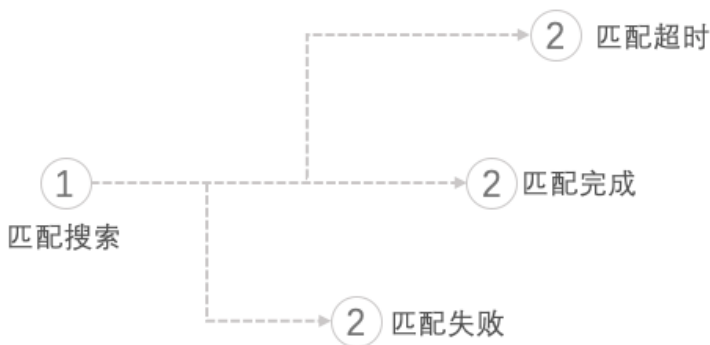


## 匹配状态流转

独立使用 GPM 匹配服务，您的匹配票据状态将会经历如下过程：

**说明：**

关于匹配票据的详细说明，详情请参见 [匹配票据参数说明](#)。



## 操作步骤

### 1. 在客户端服务器发起匹配请求

您需要准备一个后端服务来处理来自客户端的玩家请求，并通过此客户端服务器调用 GPM 的 [StartMatching](#) 接口，向 GPM 请求匹配服务。一次成功的 [StartMatching](#) 调用，将会生成一个唯一的匹配票据，您需要跟进这个匹配票据的状态，来完成您的匹配流程。

您也可以通过客户端服务器调用 GPM 的 [CancelMatching](#) 接口，用于取消正在匹配中的匹配票据。

### 2. 通过匹配事件通知，跟进匹配状态

成功发起匹配后，您配置过接收事件通知的地址，将会收到来自 GPM 的匹配事件推送。您可以设置一个定时器，周期性的处理事件推送。关于不同类型匹配事件推送的详细介绍，详情请参见 [事件推送](#)。

### 3. 处理已完成的匹配票据

- 匹配搜索到潜在的对战组合票据后，您需要解析 [推送协议](#) 里的匹配票据信息，以获得被匹配到同一个对战的玩家 ID。
- 对于匹配到符合对战要求的票据，您需要为这些票据内的玩家请求对战服务器，并将这些玩家连接到同一个对战进程里。由于您独立使用 GPM 服务，匹配完成后的玩家连接、对战逻辑需要您自行处理。
- 对于被取消/匹配超时/匹配失败的票据，您可以根据实际的业务逻辑自行对玩家进行通知，或在客户端服务器上为票据内的玩家重新发起匹配。

### 4. (可选) 利用查询匹配进度接口，作为匹配事件推送的补充机制

如果您没有配置接收匹配事件进度的服务器地址，或因为某些原因无法接受到匹配事件通知，您可以通过 GPM 的 [DescribeMatchingProgress](#) 接口来定向跟进指定匹配票据的状态。关于不同匹配票据的不同状态说明，详情请参见 [匹配票据状态说明](#)。

# 匹配规则

## 规则脚本设计指南

最近更新时间：2022-03-29 15:29:30

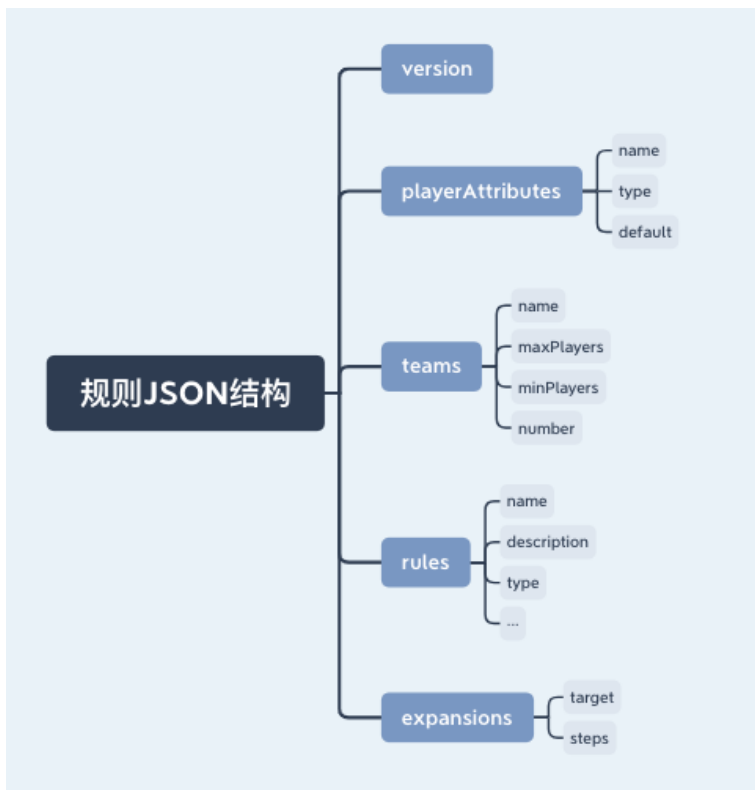
### 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍如何进行匹配规则脚本设计。

## 规则 JSON 结构

匹配规则脚本结构如下：



## 参数详解

### version

版本说明，必填，当前版本仅支持"v1.0"。示例代码如下：

```
"version": "v1.0"
```

### playerAttributes

玩家属性数组，包含0 - 5组属性值，非必填。

#### • 参数说明

参数	类型	必填	说明
name	string	是	属性名称，与 <a href="#">发起匹配</a> 接口中传入的属性名对应。数组内 name 不能重复，支持字母、数字。
type	string	是	属性类型。合法取值："number", "string"
default	number/string	否	属性默认值，必须与 type 定义的类型保持一致。 <ul style="list-style-type: none"> <li>当玩家在发起匹配请求中没有传入该值时，将用默认值为玩家进行匹配运算</li> <li>若没有定义默认值，则玩家必须在发起匹配时传入该属性的值</li> </ul>

#### • 示例代码

```

"playerAttributes": [
  {
    "name": "skill",
    "type": "number",
    "default": 10
  },
  {
    "name": "gameMode",
    "type": "string",
    "default": "classic"
  }
]
    
```

## teams

一个完整匹配中的所有队伍类型，必填。当前版本中，所有队伍的玩家最大数量不超过40。

#### • 参数说明

参数	类型	必填	说明
name	string	是	队伍类型名称。数组内 name 不能重复，支持字母、数字。
maxPlayers	number	是	一个该类型队伍中的最大玩家数量
minPlayers	number	是	一个该类型队伍中的最小玩家数量
number	number_list	否	<ul style="list-style-type: none"> <li>表示该类型的队伍数量</li> <li>number_list 只能包含两个元素，合法格式为：[ MinTeamCount, MaxTeamCount ]</li> <li>若不填，默认为[1,1]，表示该类型队伍在一个完整的匹配中有且仅有1个</li> <li>当前版本中，MinTeamCount 与 MaxTeamCount 必须相等，即暂不支持动态队伍数量</li> </ul>

#### • 示例代码

```

"teams": [
  {
    "name": "soldier",
    "maxPlayers": 4,
    "minPlayers": 4,
    "number": [2,2]
  }
]
    
```



```

    },
    {
      "name": "king",
      "maxPlayers": 4,
      "minPlayers": 4
    }
  ]

```

## rules

匹配规则数组，包含0 - 10条规则，非必填。当前版本匹配规则类型支持差值规则 `distanceRule`、比较规则 `comparisonRule`、延迟规则 `latencyRule`。关于不同类型规则的详细介绍，详情请参见 [规则类型说明](#)。

### • 参数说明

参数	类型	必填	适用规则类型
name	string	是	所有类型
type	string	是	所有类型
description	string	否	所有类型
measurements	-	-	distanceRule, comparisonRule
referenceValue	-	-	distanceRule, comparisonRule
operation	-	-	comparisonRule
maxDistance	-	-	distanceRule, latencyRule
minDistance	-	-	distanceRule
maxLatency	-	-	latencyRule
distanceReference	-	-	latencyRule
partyAggregation	string	否	所有类型

### ❓ 说明：

`measurements`、`referenceValue`、`operation`、`maxDistance`、`minDistance`、`maxLatency` 和 `distanceReference` 的参数类型和是否必填与规则类型有关，详情可参见 [规则类型说明](#)。

### • 示例代码

```

"rules": [{
  "name": "FairTeamSkill",
  "type": "distanceRule",
  "measurements": [ "avg(teams[*].players.playerAttributes[skill])" ],
  "referenceValue": "avg(flatten(teams[*].players.playerAttributes[skill]))",
  "maxDistance": 10
}, {
  "name": "RedTeamSelection",
  "type": "comparisonRule",

```

```

"operation": "=",
"measurements": ["flatten(teams[red].players.playerAttributes[colour])"],
"referenceValue": "red"
}, {
"name": "FastConnection",
"type": "latencyRule",
"maxLatency": 50
}]
    
```

## expansions

匹配规则扩展。可以配置按等待时间调整匹配规则中的限制。包含0 - 5组扩展目标，每个目标下包含1 - 10组 steps 取值。

### 参数说明

参数	类型	必填	说明	有效取值	
target	string	是	需要进行扩展的目标	<ul style="list-style-type: none"> <li>队伍的 minPlayers、maxPlayers</li> <li>distanceRule的minDistance、maxDistance</li> <li>latencyRule的maxLatency、maxDistance</li> </ul>	
steps	waitTimeSeconds	number	是	等待的时间跨度，单位秒	number
	value	number	是	target 等待 waitTimeSeconds 后的取值	number

### 示例代码

```

"expansions": [{
"target": "teams[soldier].minPlayers",
"steps": [{
"waitTimeSeconds": 5,
"value": 6
}, {
"waitTimeSeconds": 15,
"value": 4
}]
}, {
"target": "rules[FastConnection].maxLatency",
"steps": [{
"waitTimeSeconds": 10,
"value": 100
}, {
"waitTimeSeconds": 20,
"value": 150
}]
}]
    
```

## 表达式说明

**运算符**

运算符	输入	输出	含义
avg	List< number >、List< List< number >>	number、List< number >	获取列表中数值的平均值
flatten	List< List< ? >>	List< ? >	将二维列表变成包含所有元素的一维列表

**表达式**

表达式	含义	结果类型	表达式用法
teams[soldier].maxPlayers	soldier 类型队伍的最大玩家数参数	number	<ul style="list-style-type: none"> <li>用于配置 expansions 的 target</li> </ul>
teams[soldier].players.playerAttributes[xxx]	soldier 类型队伍的玩家 xxx 属性列表	List< List< ? >>	<ul style="list-style-type: none"> <li>用于配置 rules 的 measurements</li> <li>用于配置 rules 的 referenceValue</li> </ul>
teams[*].players.playerAttributes[xxx]	所有类型队伍的玩家 xxx 属性列表	List< List< ? >>	<ul style="list-style-type: none"> <li>用于配置 rules 的 measurements</li> <li>用于配置 rules 的 referenceValue</li> </ul>
teams[soldier].players[playerId]	soldier 类型队伍的玩家 ID 列表	List< List< string >>	<ul style="list-style-type: none"> <li>用于配置 comparisonRule 类型 rules 的 measurements</li> </ul>
rules[FairTeamSkill].maxDistance	FairTeamSkill 规则的 maxDistanc 参数	number	<ul style="list-style-type: none"> <li>用于配置 expansions 的 target</li> </ul>

# 规则类型介绍

最近更新时间：2022-03-29 15:29:36

## 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍游戏玩家匹配 GPM 不同类型的规则。

## 差值规则

差值规则可用于限定测量值和参考值之间的差距范围。

### 规则属性说明

规则属性	属性描述	必填	合法取值
name	规则名称	是	在同一个 RuleScript 内不能重复，支持字母、数字
type	规则类型	是	"distanceRule"
description	规则描述	否	字符串
measurements	规则测量值，用于与 referenceValue 比较差距	是	可以解析为数值或数值列表的运算表达式。例如： ◦ "avg(teams[*].players.playerAttributes[skill])" ◦ "avg(flatten((teams[*].players.playerAttributes[skill]))"
referenceValue	比较参考值	是	数值，或可以解析为数值的运算表达式。例如： ◦ 60 ◦ "avg(flatten((teams[*].players.playerAttributes[skill]))"
minDistance	最小差距	minDistance、maxDistance 不同时为空	number。取值范围0 - 99999
maxDistance	最大差距	minDistance、maxDistance 不同时为空	number。取值范围0 - 99999
partyAggregation	当一个票据含有多个玩家时的处理方式	否	◦ 合法取值："each", "min", "max", "avg", "any" ◦ 不传默认取"each" ◦ 关于 partyAggregation 的实现机制和取值说明，详情请参见 <a href="#">多人组队匹配</a>

### 规则实现

GPM 将搜索合适的匹配票据，使 measurements 与 referenceValue 之间差距的绝对值，不超过 maxDistance 且不小于 minDistance。

- 当 measurements 中的表达式计算结果为一个数值时，用这个数值与 referenceValue 进行比较。
- 当 measurements 中的表达式计算结果为一个数值列表时，用列表中的每一个元素与 referenceValue 进行比较。

### 规则样例

每一个队伍的技能平均值，与一局游戏对战中所有玩家的技能平均值相差不超过10。示例代码如下：

```

"rules": [{
  "name": "FairTeamSkill",
  "type": "distanceRule",
  "measurements": [ "avg(teams[*].players.playerAttributes[skill])" ],
  "referenceValue": "avg(flatten(teams[*].players.playerAttributes[skill]))",
  "maxDistance": 10
}]
    
```

## 比较规则

比较规则用于比较字符串、数值类型属性的相对关系。如字符串属性之间是否相等、数值属性之间大于或小于。

### 规则属性说明

```

<tr>
  <td>partyAggregation</td>
  <td>当一个票据含有多个玩家时的处理方式</td>
  <td>否</td>
  <td><ul><li>合法取值: "each", "min", "max", "avg", "any"。(当用于配置比较规则的属性是字符串类型时, 此处合法取值为 "=", "!=") </li><li>不传默认取"each" </li><li>关于 partyAggregation 的实现机制和取值说明, 详情请参见 <a href="https://cloud.tencent.com/document/product/1294/49599">多人组队匹配</a></li></ul></td>
    
```

规则属性	属性描述	必填	合法取值
name	规则名称	是	在同一个 RuleScript 内不能重复, 支持字母、数字
type	规则类型	是	"comparisonRule"
description	规则描述	否	字符串
measurements	规则测量值, 用于与 referenceValue 比较差距	是	可以解析为数值、数值列表、字符串、字符串列表的运算表达式。例如: <ul style="list-style-type: none"> <li>"flatten(teams[*].players.playerAttributes[gender])"</li> <li>"avg(flatten(teams[*].players.playerAttributes[skill]))"</li> </ul>
referenceValue	比较参考值	否	数值、字符串, 或可以解析为数值的运算表达式。例如: "avg(flatten((teams[*].players.playerAttributes[skill]))"
operation	比较判断符	是	合法取值: "=", "!=", "<", "<=", ">", ">="。(当用于比较的元素是字符串类型时, 此处合法取值为 "=", "!=")

### 规则实现

GPM 将搜索合适的匹配票据, 使 measurements 与 referenceValue 之间相对大小, 满足 operation 定义的表判断符。

- 当 measurements 中的表达式计算结果为一个数值时, 用这个数值与 referenceValue 进行相对大小比较。
- 当 measurements 中的表达式计算结果为一个数值列表时, 用列表中的每一个元素与 referenceValue 进行相对大小比较。
- 当 measurements 中的表达式计算结果为一个字符串时, 判断这个字符串与 referenceValue 是否一致。
- 当 measurements 中的表达式计算结果为一个字符串列表时, 判断列表中的每一个元素与 referenceValue 是否一致。
- 当 referenceValue 未定义时, 判读 measurements 中每一个元素是否两两相等或不等。

### 规则样例

- 样例1: 一个成功匹配中的玩家需要选择同样的 gameMode。

```

"rules": [{
  "name": "SameGameMode",
  "type": "comparisonRule",
  "operation": "=",
  "measurements": ["flatten(teams[*].players.playerAttributes[gameMode])"]
}]
    
```

- 样例2: 红队玩家的颜色属性必须都是 red。

```

"rules": [{
  "name": "RedTeamSelection",
  "type": "comparisonRule",
  "operation": "=",
  "measurements": ["flatten(teams[red].players.playerAttributes[colour])"],
  "referenceValue": "red"
}]
    
```

## 延迟规则

延迟规则用于根据玩家的地域延迟进行匹配，使对局内玩家的延迟体验最佳。

### 说明:

GPM 仅按地域延迟进行玩家撮合，不更新玩家传入的地域信息，也不为玩家进行地域选择。按延迟规则匹配完成后，对局中的玩家至少有一个共同地域满足规则条件。GSE 将为玩家选择对战服地域，详情请参见 GSE 的 [就近调度策略](#)。

### 规则属性说明

规则属性	属性描述	必填	合法取值
name	规则名称	是	在同一个 RuleScript 内不能重复，支持字母、数字
type	规则类型	是	"latencyRule"
description	规则描述	否	字符串
maxLatency	玩家到区域可接受的最高延迟	是	number。取值范围为0 - 999999
maxDistance	每个玩家到一个区域的延迟与 distanceReference 之间的最大差值。	否。必须与 distanceReference 配套使用	number。取值范围为0 - 999999
distanceReference	参考值	否。必须与 maxDistance 配套使用	合法取值: "min", "avg"。 ◦ "min"表示一个匹配中所有玩家到一个地域的最小值 ◦ "avg"表示一个匹配中所有玩家到一个地域的平均值
partyAggregation	当一个票据含有多个玩家时的处理方式	否	◦ 合法取值: "each", "min", "max", "avg", "any" ◦ 不传默认取"each"

。关于 partyAggregation 的实现机制和取值说明，详情请参见 [多人组队匹配](#)

#### • 规则实现

玩家发起匹配时传入自己到各个地域的延迟数据。GPM 将搜索合适的匹配票据，使匹配到一起的玩家，至少有一个共同地域能满足延迟规则中的限制条件。

##### ② 说明：

GPM 不提供测速能力，您需要自行测速，或使用 GSE 提供的 [测速工具](#)。

#### • 规则样例

- 样例1：被匹配到同一局游戏的玩家，至少有一个地域，满足所有玩家到该地域的延迟不大于150ms。

```
"rules": [{
  "name": "lowLatency",
  "type": "latencyRule",
  "maxLatency": 150
}]
```

- 样例2：被匹配到同一局游戏的玩家，至少有一个地域，满足所有玩家到该地域的延迟不大于150ms。且匹配中所有玩家到该地域的延迟最小值，与其他玩家到该地域的延迟差距在80ms以内。

```
"rules": [{
  "name": "lowLatency",
  "type": "latencyRule",
  "maxLatency": 150,
  "maxDistance": 80,
  "distanceReference": "min"
}]
```

# 规则示例

最近更新时间：2022-03-29 15:29:48

## ⚠ 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍如下几种经典规则示例。

## 8V8V8 差值+延迟规则

使用场景：发起匹配时，每个队伍技能 team skill 均值和一个匹配内所有 team skill 均值偏差不超过10；且一个至少有一个地域满足匹配内的所有玩家到该地域的延迟不超过50ms。增加扩展规则，等待一段时间后，放宽规则限制。

```
{
  "version": "v1.0",
  "playerAttributes": [
    {
      "name": "skill",
      "type": "number",
      "default": 10
    }
  ],
  "teams": [
    {
      "name": "red",
      "maxPlayers": 8,
      "minPlayers": 8
    },
    {
      "name": "green",
      "maxPlayers": 8,
      "minPlayers": 8
    },
    {
      "name": "blue",
      "maxPlayers": 8,
      "minPlayers": 8
    }
  ],
  "rules": [
    {
      "name": "FairTeamSkill",
      "type": "distanceRule",
      "measurements": [
        "avg(teams[*].players.playerAttributes[skill])"
      ],
      "referenceValue": "avg(flatten(teams[*].players.playerAttributes[skill]))",
    }
  ]
}
```



```
"maxDistance": 10
},
{
  "name": "FastConnection",
  "type": "latencyRule",
  "maxLatency": 50
}
],
"expansions": [
  {
    "target": "rules[FairTeamSkill].maxDistance",
    "steps": [
      {
        "waitTimeSeconds": 5,
        "value": 50
      },
      {
        "waitTimeSeconds": 15,
        "value": 100
      }
    ]
  },
  {
    "target": "rules[FastConnection].maxLatency",
    "steps": [
      {
        "waitTimeSeconds": 10,
        "value": 100
      },
      {
        "waitTimeSeconds": 20,
        "value": 150
      }
    ]
  }
]
}
```

## XV8 非对称匹配+比较规则

使用场景：红队不固定人数，蓝队固定人数。人数不对等的对局，并且要求玩家选择相同的游戏模式。

```
{
  "version": "v1.0",
  "playerAttributes": [{
    "name": "gameMode",
    "type": "string",
    "default": "turn-based"
  }
]
```

```
}],
"teams": [{
  "name": "red",
  "maxPlayers": 7,
  "minPlayers": 2
}, {
  "name": "blue",
  "maxPlayers": 8,
  "minPlayers": 8
}],
"rules": [ {
  "name": "SameGameMode",
  "type": "comparisonRule",
  "operation": "=",
  "measurements": ["flatten(teams[*].players.playerAttributes[gameMode])"]
}]
}
```

## OV5人机大战

使用场景：凑5人组一队，与机器人对战(机器人由用户实现)

```
{
  "version": "v1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number"
  }],
  "teams": [{
    "name": "Marauders",
    "maxPlayers": 5,
    "minPlayers": 5
  }],
  "rules": [{
    "name": "lowLatency",
    "description": "Sets maximum acceptable latency",
    "type": "latencyRule",
    "maxLatency": 150
  }],
  "expansions": [{
    "target": "rules[lowLatency].maxLatency",
    "steps": [{
      "waitTimeSeconds": 12,
      "value": 200
    }]
  }]
}
```

## 2V2，两种队伍，其中一种没有规则约束

```
{
  "version": "v1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number"
  }],
  "teams": [{
    "name": "red",
    "maxPlayers": 2,
    "minPlayers": 2
  }, {
    "name": "blue",
    "maxPlayers": 2,
    "minPlayers": 2
  }],
  "rules": [{
    "name": "redTeamRule",
    "type": "distanceRule",
    "measurements": ["teams[red].players.playerAttributes[skill]"],
    "referenceValue": 10,
    "maxDistance": 5
  }]
}
```

# 事件推送

## 推送机制介绍

最近更新时间：2022-03-29 15:30:01

### 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍 GPM 的事件消息推送，帮助您利用 GPM 的事件推送机制跟进匹配状态。

## 前提条件

- 准备一个用于接收匹配事件推送的服务地址。
- 已完成 [匹配创建](#)，为匹配配置服务地址的 URL，用于接收事件消息。

## 事件推送协议

字段名	类型/值	描述
Time	string	推送事件时间戳。例如 "1970-01-01T00:00:00Z"
Account	string	腾讯云账户 Uin
Region	string	当前匹配所在地域
Source	string	取值为"TencentCloud.GPM"
Type	string	事件消息类型
Id	string	事件唯一 ID，由 GPM 生成
MatchTickets	Array of <a href="#">MatchTicket</a>	事件中的匹配票据列表。匹配票据在发起匹配时由开发者传入，或由 GPM 生成
CustomEventPushData	string	自定义推送数据。在 <a href="#">创建匹配</a> 时定义
MatchTokens	Array of string	当前有效的匹配 Token 列表，用于校验当前消息是否可信。关于 MatchToken 的用法说明，详情请参见 <a href="#">事件推送安全</a> 。

## 事件消息类型

- **MatchmakingSearching**  
表示当前事件中的匹配票据正在被 GPM 执行搜索。
- **MatchCreated**  
表示当前事件中的匹配票据可组成一个符合规则条件的潜在对局。
- **MatchmakingSucceeded**  
表示当前事件中的匹配票据已在 GSE 指定队列中放置成功，GSE 已经为即将开始的对战准备好游戏服务器会话和玩家会话，等待玩家从客户端接入。
- **MatchmakingTimedOut**  
表示当前事件中的匹配票据已超时，未能搜索到符合条件的玩家。

- **MatchmakingCancelled**  
表示当前事件中的匹配票据已被取消。
- **MatchmakingFailed**  
表示当前事件中的匹配票据执行失败。

## 事件推送时机

事件推送时机取决于事件推送类型，不同的事件类型将在不同的时机被推送至您配置的 URL。

- 一个匹配票据开始被搜索时，将触发一个类型为 **MatchmakingSearching** 的事件推送。
- 一个匹配票据被成功取消时，将触发一个类型为 **MatchmakingCancelled** 的事件推送，当前票据在 GPM 中的生命周期结束。
- 一个匹配票据在配置的超时时间内没有搜索到潜在对局的票据组合，将触发一个类型为 **MatchmakingTimedOut** 的事件推送，当前票据在 GPM 中的生命周期结束。
- 一个匹配票据在任意阶段发生错误，将触发一个类型为 **MatchmakingFailed** 的事件推送，当前票据在 GPM 中的生命周期结束。
- 搜索到潜在对局的票据组合后，将形成一个 **match**，一个 **match** 将触发一个类型为 **MatchCreated** 的事件推送；
  - 对于不请求 GSE 资源的匹配，此 **match** 中包含的所有票据在 GPM 中的生命周期结束。
  - 对于请求 GSE 资源的匹配，此 **match** 中包含的所有票据状态流转至下一个阶段：**PLACING**。
- 对于需要请求 GSE 资源的匹配，会为成功的 **match** 在 GSE 中的游戏服务器队列上进行放置。放置成功后，将触发一个类型为 **MatchmakingSucceeded** 的事件推送。此 **match** 中包含的所有票据在 GPM 中的生命周期完成。

# 事件推送安全

最近更新时间：2022-03-29 15:30:05

## 注意：

由于产品逻辑已无法满足游戏行业技术发展，游戏玩家匹配 GPM 将于2022年6月1日下线，请您在2022年5月31日前完成服务迁移。

本文为您介绍 GPM 的事件消息推送的安全机制，帮助您识别可靠的事件消息，提高业务安全性。

## 前提条件

- 已完成 [匹配创建](#)，获取一个 MatchCode。
- 已为 MatchCode 配置接收事件消息的 URL 地址。您可以在创建匹配或编辑匹配时进行配置。

## MatchToken 简介

MatchToken 相当于为您的 MatchCode 设定一把钥匙，用于验证此 MatchCode 推送出的事件消息是否可信。在您使用 GPM 的事件消息推送功能时，验证 MatchToken 并不是必要的，但通过实现对 MatchCode 的校验，可以增加您的业务安全性，防止推送请求被恶意伪造。

## MatchToken 生成与更新机制

您需要通过调用 [ModifyToken](#) 接口为匹配生成或更新 MatchToken，GPM 不会自动为您生成或更新。您可以自定义 MatchToken，或由 GPM 为当前匹配随机生成一个 MatchToken。您需要指定当前 MatchToken 的兼容时间，表示本次设置的 MatchToken 被更新后，GPM 将同时推送原 MatchToken 和新 MatchToken 的时间跨度。

对于同一个 MatchCode，每次变更 MatchToken 需要间隔三分钟以上。

## MatchToken 的查询

通过 [DescribeToken](#) 接口查询指定 MatchCode 的最新 MatchToken。

## MatchToken 校验指南

您需要自行对 MatchToken 进行校验。每次设置或更新 MatchToken 后，您需要记录下最新的 MatchToken，用于验证是否与事件消息中的 MatchToken 相符。

为 MatchCode 设置 MatchToken 完成后，该 MatchCode 的每一条事件消息都将在推送体的 **MatchTokens** 字段里包含当前生效中的所有 MatchToken，即最新的 MatchToken 和所有未超过兼容时间的旧 MatchToken。