**定义实体**

什么时候需要定义实体:

需要进行数据存储。

能够方便的远程访问。

需要引擎管理和监控, 例如: AOI、Trap、等等。

当灾难发生后服务端可以自动进行灾难的恢复。

什么时候需要定义实体的属性:

需要进行数据存储。

实体被迁移后数据仍然有效(仅cellapp会迁移实体，比如跳转场景)。

当灾难发生后服务端可以自动进行灾难的恢复。

什么时候需要定义实体的方法:

能够方便的远程访问。

**您需要执行以下步骤：**

1. 注册实体

目录地址 : demo/scripts/entities.xml

例子:

<root>

<Account/>

<Avatar/>

<Spaces/>

<Space/>

<Monster/>

<NPC/>

<Gate/>

</root>

1. 在demo/scripts/entity\_defs目录下创建一个定义文件

例子: Account.def

1. 你可能需要定义一些属性和方法
   1. 在demo/scripts/目录下有三个子目录(base、cell、client)，你可以根据需要添加Account.py。
   2. 不是每个实体都存在(client, base, cell)这三个部分, 你可以根据需要做出一些选择。

**定义文件的格式**

<root>

<Properties>

// 属性名称

<accountName>

// 属性类型

<Type> UNICODE </Type>

// (可选)

// 属性的自定义协议ID，如果客户端不使用kbe配套的SDK来开发，客户端需要开发跟kbe对接的协议,

// 开发者可以定义属性的ID便于识别，c++协议层使用一个uint16来描述，如果不定义ID则引擎会使用

// 自身规则所生成的协议ID, 这个ID必须所有def文件中唯一

<Utype> 1000 </Utype>

// 属性的作用域 (参考下方:属性作用域章节)

<Flags> BASE </Flags>

// (可选)

// 是否存储到数据库

<Persistent> true </Persistent>

// (可选)

// 存储到数据库中的最大长度

<DatabaseLength> 100 </DatabaseLength>

// (可选， 不清楚最好不要设置)

// 默认值

<Default> kbengine </Default>

// (可选)

// 数据库索引， 支持UNIQUE与INDEX

<Index> UNIQUE </Index>

</accountName>

...

...

</Properties>

<ClientMethods>

// 客户端暴露的远程方法名称

<onReqAvatarList>

// 远程方法的参数

<Arg> AVATAR\_INFOS\_LIST </Arg>

<Utype> 1001 </Utype>

</onReqAvatarList>

...

...

</ClientMethods>

<BaseMethods>

// Baseapp暴露的远程方法名称

<reqAvatarList>

// cell暴露方法必须存在这个标记

<Exposed/>

<Utype> 1002 </Utype>

</reqAvatarList>

...

...

</BaseMethods>

<CellMethods>

// Cellapp暴露的远程方法名称

<hello>

<Utype> 1003 </Utype>

</hello>

</CellMethods>

</root>

例子: 在客户端中调用base方法获得角色列表(Account.py):

self.base.reqAvatarList()

**属性作用域**

[类型] [client] [base] [cell]

BASE - \* -

BASE\_AND\_CLIENT \* \* -

CELL\_PRIVATE - - \*(cell)

CELL\_PUBLIC - - \*(cells)

CELL\_PUBLIC\_AND\_OWN \*(client) - \*(cells)

ALL\_CLIENTS \*(clients) - \*(cells)

OWN\_CLIENT \*(client) - \*(cell)

OTHER\_CLIENTS \*(other clients) - \*(cells)