

UMC 概要

UMC 把移动的端请求协议化了，把移动界面组件化了，所以只要服务器返回协议支持JSON格式协议就能驱动移动开发

现在我们己经用java和.net完成此协议的高标准封装，我们给到后端开发者是一个极简代码，他只要用自己的后端知识体系就能快速开发起来。

所以现在用UMC开发APP不但优化了IOS和Android开发人员投入，同时简化服务端开发逻辑，与传统服务端架构相比最少减少50%代码量。

UMC简化的技术难度，明显提高开发效率、有效的提高代码质量，来源于UMC让大家用一个标准模式来组织大家开发生产，沟通就有了准线，团队都听懂，团队成员也有能力落实到位，非常适合大团队的合作开发，创造出1+1大于2的效果。

我们来说UMC服务端架构，看了其他文件会发现UMC的是采用请求总线、模块、指令来完成路由分配的。在开发实现上团队采用同一开发语言，我们可以采用分模块来组织开发。在请求总线这层也可以配置UMC API转移，转移后对开发语言没有要求，从而达成跨语言协作能最大化利用现在的开发资源。

UMC 服务端实例，

我们在java和.net不但封装好了UMC服务协议，我们还封装好了一套实体对象访问数据库API，帮助开发者快而有效的访问操作数据库，同时他与采用的数据库无关，现在只要采用此套数据库API，您的应用就可以跨数据库部署。

UMC Android前端

用java实现Android UMC前端协议和界面组件。

UMC IOS前端

用object-c实现IOS UMC前端协议和界面组件。

UMC H5移动端实例

用JS和H5实现UMC前端协议和界面组件。

UMC H5 PC后端实例

UMC PC后端是移动端扩展，采用单页应用界面架构，用Hash来完成界面路由，采用懒加载提高运行和加载效率，同时在开发上他基于UMC前端JS库，让大家用极少的代码完成开发。

从上面我们可知UMC是一个面对业务实现的快速开发架构，不知你是否发现我们提高效率的核心秘诀是什么呢，现在我们要做一个覆盖各端应用，在前端上，我们需要组建IOS小组、Android小组、H5小组和小程序小组和服务端开发小组，现在市面上无一套有效的把各小组有机统一起来的架构，基本各端采用各自的架构，对服务端提出各自的要求，大部分资源投入在沟通和协调对接上，业务实现消耗的并不多

现在UMC出现了，他优化了Android、IOS、小程序和H5开发人员，赋能于后端开发者，让团队直击业务实现，这就是我们的高效率的秘诀。