

## <AppML> 架构

### MVC 架构

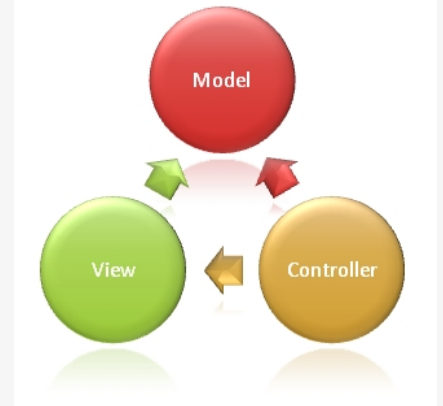
<AppML> 采用了 MVC 架构。

MVC 全名是 Model View Controller，是模型（Model）- 视图（View）- 控制器（Controller）的缩写，一种软件设计典范。

**模型(Model)** 描述你的应用。

**视图(View)** 显示你的数据。

**制器(Controller)** 控制你的应用。



Wikipedia: [Model](#), [View](#), [Controller](#)

### 模型(MODEL) - 仅仅是一个简单的XML文件

模型描述了您的应用程序，并且可在不同的硬件和软件平台（PC、iPhone、Tablets 等）重复使用。它不关心用户界面（UI）或表现形式。

模型采用xml编写，存储于web服务器中。

```
<appml>

<datasource>
  <database>
    <connection>Northwind</connection>
    <sql>SELECT CustomerName,ContactName,City,Country FROM Customers</sql>
  </database>
</datasource>

<filters>
<query>
  <field label="Customer">CustomerName</field>
  <field>City</field>
  <field>Country</field>
</query>
<order>
  <field label="Customer">CustomerName</field>
  <field>City</field>
  <field>Country</field>
</order>
</filters>

</appml>
```

以上实例定义了数据源来自 Northwind 数据库。

该模型允许使用预定义的 SQL 获取数据。它还允许通过 Customer、City 和 Country 查询数据和排序。

## 视图 ( VIEW ) - 仅仅是一个普通的 HTML 文件

视图即是 UI ( User Interface : 用户界面 )。它通常是一个显示和输入数据 ( 可选 ) HTML 页面：

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="appml.css" />
</head>
<body>
<h1>My First Web Application</h1>
<div id="Place01"></div>
<script src="appml.js"></script>
<script>
customers=new AppML("appml.html","Customers.xml");
customers.run("Place01");
</script>
</body>
</html>
```

以上 HTML 页面使用通过执行脚本语言创建一个 AppML 对象，并将数据显示在 id="Place01" 的 div 中。

采用了 "appml.js" 脚本文件。

## CONTROLLER(控制器) - 仅仅是一个浏览器很服务端的脚本

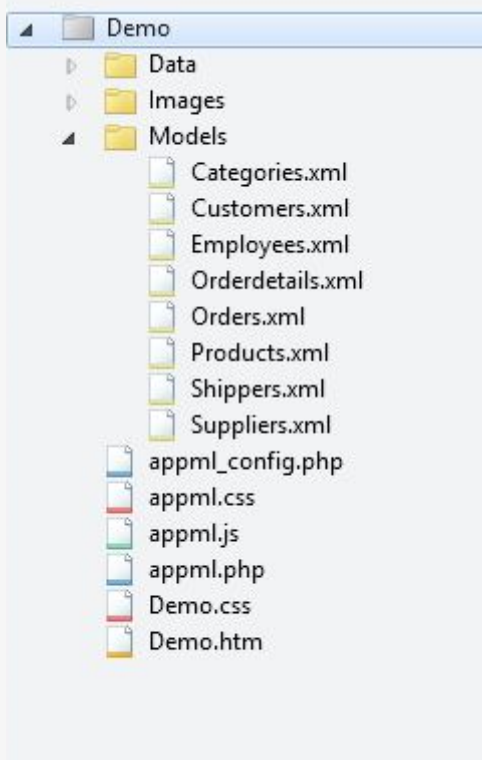
服务端脚本通过以下方式控制应用：

- 从浏览器中接收请求数据
- 将模型和数据返回给浏览器
- 从浏览器中接收更新数据
- 在服务器上更新数据
- 数据通信过程，请进行数据安全验证。

浏览器脚本通过以下方式控制应用：

- 当页面加载时，你可以加载<AppML> 控制器到页面上。
- 使用控制器，你可以在页面创建 <AppML> 对象。
- 当执行 <AppML> 对象时，它会向服务器请求数据。
- <AppML> 对象从服务器接受数据（使用数据模型）。
- <AppML> 对象（或者你的代码）在页面中显示你的数据。
- （可选）web用户修改数据。
- （可选）<AppML> 可以向服务器发送修改请求。

## 典型的 Web 文件和文件夹：



web文件夹：**Demo**

数据文件夹：**Data**

图片文件夹：**Images**

模型文件夹：**Models**

应用：**Demo.htm**

样式：**Demo.css**

<AppML> 配置文件：**appml\_config.php** (或者 .htmlx)

<AppML> 样式文件：**appml.css**

<AppML> 浏览器控制器：**appml.js**

<AppML> 服务器控制器：**appml.php** (或者 .htmlx)

## 快速、灵活的应用开发

快速应用开发（Rapid Application Development、RAD）不仅是一种需求抽取方法，它还是软件开发为一体的方法。快速应用开发目的是快速发布系统方案，而技术上的优美相对发布的速度来说是次要的。

<AppML> 提供超快速的原型设计，比传统的软件开发方法高100倍的速度。

应用程序原型可以直接从应用程序模型运行，无需任何编码。

[Wikipedia: Rapid Application Development](#)

敏捷软件开发是基于用户和开发者相互协作的基础上一步一步的发展而来的方法。

<AppML> 应用从原型到完整的应用可以通过递增的方式一步步来编写实现。

[Wikipedia: Agile Software Development](#)

## 声明式编程

软件开发往往无法按照预期的时间和预算完成。软件编码错误也是经常出现。这是因为计算机代码是很难开发、测试、维护。编码已经**过时了**。你应该更多描述做什么，而不是如何实现它。

使用 <AppML> 你需要在模型中**声明**你的应用。

使用 <AppML> 可以**少写或者不用编写代码**。

**Wikipedia:** [Declarative Programming](#)

## 代码先行（Code First）

Web应用程序开发可以使用以下两种不同的方式：

1. 代码先行 ( Code First ) : 使用预编程, 预先测试的代码, 只增加新的应用程序说明。
  2. 契约优先 ( Contract First ) : 从头开始使用完整的应用程序的说明要求编写应用程序。
- <AppML> 采用最合理的概念: **代码先行 ( Code First )**。

## 面向服务的体系结构 ( service-oriented architecture , SOA )

**Web Service** 是一个数据接口, 通过URL指定, 就像一个web页面。但它有别于web页面, 它只是一种传达信息的方式。一个典型的 **Web Service** 为页面提供了数据。

使用 <AppML>, HTML 显示为用户界面, <AppML> 提供数据。

**Original Web Services** 设计使用了 XML 标志如 SOAP、WSDL 和 UDDI。

**Modern Web Services** 比如 <AppML> 应用更加简单。

- 更容易理解 - 可以被我们阅读
- 轻量级 - 没有不必要的代码或标记
- 易于实施 - 没有所需的开发工具

面向服务的体系结构 ( service-oriented architecture , SOA ) 是一个组件模型, 它将应用程序的不同功能单元 ( 称为服务 ) 通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的, 它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

## Web Services的优势

- Web services 只需要少量的代码
- Web services 被设计来处理一组有限的任务
- Web services 使用基于HTTP的通信协议
- Web services 独立于操作系统
- Web services 独立于编程语言
- Web services 可以连接不同的应用程序, 系统和设备
- Web Services 可以很容易地发布信息
- Web Services 有利于快速应用程序开发

例如一个Web services, 可以设计一个小程序, 提供其他最新的股票的交易价格的应用程序。

Web services使用HTTP协议与其他系统进行通信, Web服务是独立于操作系统和编程语言。

调用Web services的应用程序将始终使用HTTP协议发送请求。调用应用程序将永远不会关心其他计算机运行的操作系统或编程语言。

Web services可以为更多的企业创造新的可能性, 因为它提供了一种简单的分发大量信息的方式。

比如: 航班时刻表和机票预定系统。

## 云计算(Cloud Computing)

云计算（ Cloud Computing ）是SOA的扩展：应用即服务（ Application-as-a service ），存储即服务（ Storage-as-a-service ），数据即服务（ Data-as-a-service ）。

对于大多数人，云计算是在web上存储数据：

- 邮寄很日历
- 文档和电子表格
- 书籍、笔记、待办事项列表
- 音乐、图片和电影
- 数据库和应用程序

原因很明显：

- 有来自世界各地的访问数据
- 与他人分享我的数据
- 硬件升级或者崩溃

<AppML> 可以很轻松地将数据库和应用程序放在云中。

[← AppML 下载](#)

[AppML 参考手册 →](#)

[📝 点我分享笔记](#)