WebPascal 5 产品宣传手册

让 Delphi 开发者，用熟悉的语法，优雅的进入 Web 时代！

WebPascal 5 是专为 Delphi 开发者打造的 Web 开发后端中间层框架，以 **Pascal 原生语法、纯脚本驱动、企业级安全、开箱即用**为核心特性，一站式解决 Delphi 开发者转型 Web 开发的技术痛点，无需放弃熟悉的开发习惯，无需学习陌生编程语言，让桌面开发经验无缝迁移至 Web 开发场景，轻松实现网站、Web 应用、HTTP API、RESTFul API 等各类 HTTP 应用的开发与部署。

一、直击痛点：Delphi 开发者的 Web 开发困境

深耕 Delphi 开发的你，是否曾在 Web 开发门前望而却步？

习惯了 Pascal 语法的严谨与优雅，却要被迫学习 JavaScript、PHP、Python 等陌生语言，重新搭建知识体系；依赖 Delphi 庞大的 IDE 进行开发，无法实现轻量化的 Web 开发调试，随时随地的维护需求难以满足；从零搭建 Web 服务，需兼顾跨域、安全、性能、文件处理等诸多问题，耗时耗力却难出成果；现有 Web 框架鲜有适配 Pascal 语法的，开发习惯难以兼容，效率大打折扣。

WebPascal 5 的出现，彻底打破这一僵局。框架全程围绕 Delphi 开发者的使用习惯设计，不改变、不迁就、不妥协，让你用最顺手的 Pascal 语法，直接上手 Web 开发，实现从桌面开发到 Web 开发的无缝转型。

二、四大核心优势，打造专属 Web 开发体验

零学习成本：Pascal 原生语法，拿起就用

WebPascal 5 完全兼容 Delphi 开发者熟悉的 Pascal 语法，保留所有经典写法与常用函数，真正做到**零学习成本，拿起即开发**。支持 var 变量声明、for/while 循环、if/else 条件判断等基础语法，保留 writeln、FormatDateTime、IntToStr、Trim 等 Delphi 高频使用函数，甚至连代码书写逻辑都与桌面开发高度一致。无需额外学习新框架、新语法，Delphi 开发者一看就会，直接编写业务代码，让开发效率拉满。

简单的 99 乘法表代码，在 WebPascal 5 中直接运行，无需任何适配：

```
for var i := 1 to 9 do
```

```
begin
  var s := '';
  for var j := 1 to i do
    s += Format('%d*%d=%d ', [j, i, i * j]);
  writeln(s+'<br>');
end;
```

纯脚本驱动：轻量化开发，无 IDE 依赖

WebPascal 5 采用类 PHP 的纯脚本设计，彻底摆脱庞大 IDE 的束缚，实现**开发、调试、部署全流程轻量化**。记事本、简易代码编辑器甚至手机端编辑器，均可编写脚本文件，随时随地完成开发与维护工作，满足移动化的开发需求。

脚本文件采用 UTF-8 编码规范，无需编译，直接上传至服务器即可运行，部署流程极简。框架设计了清晰的目录结构，将脚本、静态资源、上传文件、日志、临时文件分离管理，每个目录各司其职，项目维护成本大幅降低，单人即可轻松管理多个项目。

```
WebPascal/
├─── script/      # Web 脚本（支持多级子目录，按业务模块划分）
├─── task/        # 计划任务脚本（极简配置，无需额外调度服务）
├─── uploads/     # 上传文件（自动按日期分目录，避免文件混乱）
├─── www/         # 静态资源（直接映射到根目录，支持 HTML/CSS/JS 等）
├─── log/         # 日志文件（自动按日期生成，无需手动管理）
└─── cache/       # 临时文件（后台自动清理，无冗余堆积）
```

企业级安全：全栈加密防护，满足开发合规

WebPascal 5 内置**完整的企业级安全体系**，从身份认证到数据加密，从请求防护到编码规范，全方位保障 Web 应用安全，无需开发者额外开发安全逻辑，轻松满足日常开发的合规要求。

- 身份认证：支持 JWT 令牌的生成与验证，可自定义密钥、扩展信息、有效期，实现安全的接口身份校验；
- 数据加密：集成全量哈希算法（MD5/SHA1/SHA2/SHA3/SM3）与对称加密算法（AES/DES/3DES/SM4），支持 ECB/CBC 多种加密模式，满足各类数据加密、签名需求；
- 防注入：数据库操作支持参数化 SQL 查询（:p1~:p18），自动区分参数类型，从源头避免 SQL 注入风险；
- 编码安全：所有接口均以 UTF-8 编码传输数据，要求脚本文件统一采用 UTF-8 编码，彻底避免编码乱码与潜在的编码漏洞；

- 异常防护：标准化的异常提示格式，精准定位问题位置，避免敏感信息泄露，同时减少安全漏洞暴露的可能性。

开箱即用：一站式 Web 能力，无需从零搭建

WebPascal 5 集成了现代 Web 开发所需的全部核心能力，框架部署完成后即可直接使用，省去繁琐的基础开发工作，让开发者聚焦于业务逻辑本身。

- 基础 Web 能力：原生支持 HTTPS(SSL/TLS)、伪静态页面、Gzip 压缩、CORS 跨域配置，同时支持 IP 黑名单、多域名绑定、浏览器缓存设置等实用功能，适配现代 Web 服务器的所有需求；
- 文件处理：内置标准化/upload 文件上传接口，支持大文件分块上传，采用低内存占用设计，自动按日期分目录存储文件，同时完美适配 Nginx 反向代理场景；
- 数据库支持：原生适配 SQLite、MySQL、SQL Server、Oracle 等主流数据库，一套统一的 API 即可操作所有数据库，大幅降低多数据库适配的开发成本，完美适配不同项目的存储需求；
- 自动化调度：内置轻量级计划任务调度器，支持 5 种调度模式，轻松实现自动化运维，无需额外部署第三方调度服务；
- 工具集全覆盖：验证码、二维码、正则表达式、雪花算法、时间处理、编码转换等开发高频工具，均实现一键调用，无需重复造轮子。

三、全栈核心功能，覆盖 Web 开发所有场景

WebPascal 5 的功能设计围绕 Web 开发的实际需求展开，从基础语法到高级应用，从数据处理到第三方对接，全方位覆盖开发全场景，真正做到一站式解决 Web 开发问题。

基础语法与开发配套能力

支持完整的 Pascal 语法体系，内置百余种常用函数，满足基础开发需求；支持 include()脚本引入功能，实现模块化开发，提升代码复用性；提供全局变量管理功能，支持增删查、默认值读取、批量清理，满足跨脚本的数据共享需求。

同时配备全场景日志系统，分为 debug/info/warn/error/fatal 五个级别，日志支持缓存写入，自动按日期生成日志文件，且会自动清理 30 天前的日志，无需开发者手动维护，便于问题排查与系统监控。

高效数据处理能力

针对 Web 开发中的数据处理痛点，设计了轻量化、高效率的处理方案，覆盖 JSON、字符串、编码、正则等所有高频场景：

- JSON 处理：支持 JSON 字符串与对象的相互转换，支持多层 JSON 对象遍历、键值

删除，提供默认值操作符，避免空值报错，同时支持 JSON 字符串的自动转义，适配前后端数据交互；

- 正则表达式：实现匹配、是否匹配、批量匹配、替换等全功能，支持分组提取，满足各类字符串解析、校验、替换需求；
- 编码转换：支持 URL/Base64 / 十六进制的编解码，支持 Unicode 解码，且 URL 编码支持 emoji 符号解析，适配各类数据传输与存储场景；
- 字符串处理：集成去空格、大小写转换、拆分、拼接、截取、排序等全功能，同时支持字符串与数值的相互转换（含默认值兜底），覆盖字符串开发的所有需求。

现代 Web 核心能力

全面适配现代 Web 开发的 HTTP 服务需求，兼顾服务端与客户端能力，同时做好细节优化，提升开发与使用体验：

- HTTP 服务端：支持 GET/POST/PUT/DELETE 等所有常用请求方法，自动解析请求 URL、客户端 IP、请求头、URL 参数、表单参数，封装标准化的 request 请求对象，参数获取简单高效；
- HTTP 客户端：完整实现 Get/Post/Json/FormData/Put/Delete 等请求方式，支持自定义请求头、编码格式、超时时间，可实现远程接口调用、文件上传、远程文件转存等功能；
- 跨域与静态资源优化：支持 CORS 跨域的自由开启与禁用，默认强制所有接口允许跨域；支持 HTTP 多级子目录默认首页文件，新增 URL 自动重定向功能，优化浏览器对静态资源路径的识别；Gzip 压缩仅对静态资源生效，兼顾传输性能与接口响应的实时性。

多数据库支持（SQLite/MySQL/SQLServer/Oracle 等）

WebPascal 5 原生适配 SQLite、MySQL、SQL Server、Oracle 等主流数据库，一套统一的 API 即可操作所有数据库，大幅降低多数据库适配的开发成本，完美适配不同项目的存储需求：

1. 统一连接方式：通过 db.conn 函数一行代码完成任意数据库的连接配置，仅需修改数据库类型与连接字符串，核心操作逻辑完全一致；
2. 双核心操作函数：提供 db.query（查询）与 db.exec（增删改）两个通用函数，适配所有数据库的查询与执行需求；
3. 通用参数化查询：所有数据库均支持:p1~:p18 的参数化占位符，自动区分参数类型，从源头避免 SQL 注入风险，且参数传递逻辑统一，无需适配不同数据库的语法差异；
4. 标准化返回格式：所有数据库操作均返回 JSON 格式结果，包含 code 状态码与

msg 信息，查询操作额外返回 data 结果集，执行操作返回影响行数，前后端处理逻辑统一，无歧义。

主流数据库操作示例：

```
// 1. SQLite 连接与操作
db.conn('sqlite','DriverID=SQLite;Database=.\sqlite.db');
// 2. MySQL 连接与操作
db.conn('mysql','DriverID=MySQL;Database=mysql;Server=127.0.0.1;Port=3306
;User_name=root');
// 3. SQL Server 连接与操作
db.conn('mssql','DriverID=MSSQL;Server=127.0.0.1,1433;Database=test;User_
name=sa;Password');
// 4. Oracle 连接与操作
db.conn('oracle','DriverID=Ora;Database=OraSrv:1521/orcl;User_name=ADDemo
;Password=a;CharacterSet=UTF8');

// 通用查询逻辑（适配所有数据库）
var res := db.query('sqlite','SELECT * FROM users WHERE id=:p1 AND
status=:p2', 1, 1);
// 处理查询结果（通用逻辑）
if res.code = 200 then
  writeln('查询到 ' + IntToStr(res.data.length()) + ' 条数据')
else
  logger.error('查询失败：' + res.msg);

// 通用新增逻辑（适配所有数据库）
var execRes := db.exec('sqlite','INSERT INTO users(name,age)
VALUES(:p1,:p2)', '李四',25);
```

灵活的计划任务调度

内置轻量级计划任务调度器，无需额外部署第三方调度服务，5 种调度模式全覆盖，满足秒级循环、定时执行、周期执行等所有自动化运维需求，且实现**调试与生产一体化**，大幅提升开发效率。

- 模式 0：按秒循环执行，适用于高频次的定时任务，如每 10 秒执行一次；
- 模式 1：按固定年月日时分秒执行，适用于一次性的定时任务，如指定具体时间执行数据备份；
- 模式 2：每年按月日时分秒执行，适用于年度周期任务，如每年 5 月 20 日执行年度统计；

- 模式 3：每月按日时分秒执行，适用于月度周期任务，如每月 20 日执行月度对账；
- 模式 4：每日按时分秒执行，适用于日常周期任务，如每日凌晨执行数据库备份。

计划任务脚本设计简洁，去掉<task>配置块后，即可作为普通脚本在 script 目录下调试，无需修改任何业务代码，调试完成后添加配置块即可转为计划任务，开发流程无缝衔接。

一键调用的内置工具集

将 Web 开发中高频使用的工具功能封装为内置函数，一行代码即可调用，无需开发者自行开发，大幅提升开发效率：

- 验证码生成：captcha('内容', 品质)，支持中文、数字、字母组合内容，可自定义生成品质，直接返回可嵌入页面的图片资源；
- 二维码生成：QRCode('内容', 品质, 放大倍数)，支持自定义内容、生成品质与图片放大倍数，一键生成二维码图片；
- 时间处理：保留 Delphi 开发者熟悉的 FormatDateTime 函数，同时支持 Unix 时间戳与本地时间的互转、UTC 时间与本地时间的互转，满足各类时间处理需求；
- 雪花算法：Snowflake()，一键生成分布式唯一 ID，适配分布式部署场景，避免 ID 重复；
- 容量格式化：ByteSizeToStr()，自动区分 GB 与 GiB，精准格式化存储/传输容量，适配各类资源统计场景。

四、简洁高效的设计哲学，贴合 Delphi 开发习惯

WebPascal 5 的所有设计均围绕**Delphi 开发者的使用习惯**展开，以「简洁、高效、自动化、轻量化」为核心设计哲学，拒绝反人类设计，让 Web 开发像 Delphi 桌面开发一样简单、顺手。

简洁的 API 设计

所有内置 API 均遵循「简洁、直观、贴合 Pascal」的原则，调用方式统一，记忆成本低，无需开发者频繁查阅文档。无论是 JSON 处理、正则表达式，还是 HTTP 客户端调用，都能通过简单的函数实现复杂的功能，代码可读性高，维护成本低。

```
// JSON 处理：默认值操作符，避免空值报错
var obj := json.parse('{"name":"张三","age":18}');
var email := obj.email ?? '未设置';
```

```
// 正则表达式：快速提取 URL 中的域名
var domain := regex.match('https://webpascal.com',
'(?<=https://).*?(?=/)');
// HTTP 客户端：一行代码调用远程接口
var result := http.get('https://api.example.com/users');
```

严谨的错误处理

设计了标准化的 JSON 异常返回格式，脚本执行出错时，会返回固定结构的错误信息，包含 500 错误码与具体的错误描述，且精准标注错误所在的行号与列号。若脚本中使用 include()引入了其他脚本，行号会自动累加被包含脚本的行数，让开发者能够快速定位问题位置，调试效率提升 10 倍。

```
{
    "code": 500,
    "msg": "Script error: Unexpected \"=\" [line 42, column 4]"
}
```

自动化的资源管理

框架对所有日志、临时文件、上传文件等资源实现**全自动化管理**，无需开发者手动维护，大幅减少运维成本：日志文件自动按日期生成，30 天前的日志会被自动删除；临时文件存储在固定的 cache 目录，框架启动时会自动清空该目录，且后台有子线程每小时清理一次 3 小时前的临时文件；上传文件自动按日期分目录存储，避免文件覆盖与目录混乱，让开发者无需关注资源管理，聚焦于业务开发。

轻量化的部署与扩展

WebPascal 5 采用轻量化的部署方案，仅需三步即可完成服务启动，耗时不超过 5 分钟：下载框架压缩包并解压至服务器任意目录；修改配置文件，简单设置端口、上传路径、跨域配置等基础参数；运行 webpascal.exe 即可启动服务，也可通过 nssm 等第三方工具将其注册为 Windows 服务，实现后台稳定运行，默认通过 http://localhost:8833 访问。

同时框架具备良好的扩展性，脚本目录支持多级子目录，可按业务模块自由划分，便于大型项目的开发与维护；完美兼容 Nginx 反向代理，可通过 Nginx 实现 HTTPS 配置、负载均衡等高级功能，适配企业级的部署需求。

五、极简上手流程，3 分钟开启 Web 开发之旅

WebPascal 5 的部署与开发流程均做了极致简化，无需复杂的环境配置，无需专业的

Web 开发知识，Delphi 开发者按照熟悉的逻辑，即可快速开启 Web 开发之旅。

部署流程（3 步完成）

1. **下载**：获取 WebPascal 5 官方压缩包，支持中英文版本，解压至服务器或本地的任意目录；
2. **配置**：打开框架配置文件，根据需求简单设置服务端口、上传文件路径、跨域配置等基础参数，无需复杂配置；
3. **启动**：直接运行 webpascal.exe 即可启动服务，也可通过 nssm 工具注册为 Windows 服务，默认访问地址为 http://localhost:8833，直接访问即可查看演示案例。

开发流程（零适配，贴合 Delphi 逻辑）

1. **写脚本**：在框架的 script 目录下创建以.api 为后缀的脚本文件，使用熟悉的 Pascal 语法编写业务逻辑；
2. **调接口**：通过 http://域名/ask/脚本名.api 的格式访问接口，框架会自动执行对应的脚本文件；
3. **看结果**：脚本可直接输出 HTML、JSON、文本等任意格式的内容，执行日志自动记录至 log 目录，若有异常会返回标准化的错误信息，精准定位问题。

Hello World 极简示例：

在 script 目录创建 hello.api 文件，写入以下代码：

```
writeln('Hello WebPascal 5!');
writeln('Delphi 开发者的 Web 开发专属利器！');
```

直接访问 http://localhost:8833/ask/hello.api，即可看到输出结果，全程无需其他操作，真正做到开箱即用。

六、版本与服务，灵活选择适配不同需求

WebPascal 5 提供**免费版**与**商业版**两个版本，满足个人开发、学习测试、企业开发、商业项目等不同场景的需求，同时提供丰富的配套资源，让开发者开发无忧。

免费版

- 永久免费使用，无任何功能限制，可自由用于个人开发、小型项目、学习测试等场景；
- 免费获取框架的最新版本更新，紧跟最新功能与优化；

- 免费获取官方提供的教程资料、演示案例、前端模板等配套资源。

商业版

- 一次性购买源码，即可终身使用，无后续续费成本；
- 包含 Delphi 完整源码，支持开发者进行二次开发，适配企业个性化需求；
- 享受永久免费的版本升级服务，及时获取最新的功能与问题修复；
- 提供永久的邮件技术支持，官方会及时响应并解决开发过程中遇到的问题；
- 适用于企业开发、商业项目、生产环境等正式场景。

丰富的官方配套资源

- 完整开发文档：覆盖框架的所有功能、语法、技术规范，讲解详细易懂，是开发者的必备参考；
- 15+核心演示案例：包含基础语法、多数据库操作、文件上传、加密安全、计划任务、微信公众号对接等所有核心模块的演示脚本，可直接参考修改，快速落地业务；
- 前端模板：配套 AdminLTE 前端模板、Vue 模板，与框架无缝对接，快速实现前后端一体化开发。

七、总结

WebPascal 5 是 Delphi 开发者的 Web 开发专属利器，它以 Pascal 语法为核心，以纯脚本驱动为载体，以企业级安全为保障，以多数据库适配、开箱即用为目标，让 Delphi 开发者无需改变开发习惯，无需学习陌生技术，即可轻松踏入 Web 开发的殿堂。

从基础的语法支持到复杂的企业级应用，从简单的接口开发到完整的网站搭建，WebPascal 5 都能完美适配，是 Delphi 开发者转型 Web 开发、实现技术拓展的最佳选择。

立即下载 WebPascal 5，用熟悉的 Pascal 语法，开启属于你的 Web 开发新旅程！

官方地址：https://webpascal.com

咨询邮箱：593106@qq.com

Copyright © 2026 webpascal.com 版权所有 | 专为 Delphi 开发者打造