Why TMS XData Is My Go-To Choice for Building Web Services in Delphi

By Dr. Holger Flick

After years of working with various frameworks and tools for building web services, I've come to a clear conclusion: **TMS XData is the best tool for the job—if you're a Delphi developer**. Now, I'll be upfront—I've been doing extensive work with the company that distributes TMS XData, so there's certainly a personal connection here. But my preference isn't just based on familiarity or convenience. There are solid, practical reasons why I keep reaching for XData whenever I need to build robust, professional web services in the Delphi ecosystem.

Let me be clear from the start: I'm talking specifically about **the best solution for Delphi developers**. Other languages have their own excellent options—Node.js has Express, Python has Flask and FastAPI, .NET has ASP.NET Core. Each language ecosystem has evolved its own tools optimized for its strengths. But if your world is Delphi, and you need to build web services that leverage your existing Delphi skills and codebase, TMS XData stands above the rest.

What Is TMS XData?

For those unfamiliar with it, TMS XData is a Delphi framework that enables developers to create complete REST/JSON HTTP/HTTPS web services with minimal effort. It's part of the TMS Business Core suite and is designed to handle everything from simple API endpoints to complex, enterprise-grade web services.

What sets XData apart is its approach: rather than forcing you to write boilerplate code for every endpoint, XData lets you focus on your business logic while it handles the heavy lifting of request routing, JSON serialization, authentication, and response formatting. You define your service operations as regular Delphi methods, and XData transforms them into fully functional REST endpoints.

The framework supports both simple REST services and complex OData v4 implementations, making it incredibly versatile whether you're building a straightforward API or a sophisticated data service that clients can query with complex filters and operations.

Why I Consider It the Best for Delphi: My Key Reasons

1. Automatic API Documentation That Actually Works

Here's a pain point every developer knows: keeping API documentation synchronized with your actual implementation. You make a change to an endpoint, and suddenly your documentation is outdated. With many frameworks, documentation becomes a maintenance burden—another task on an already full plate.

TMS XData solves this problem elegantly by automatically documenting your web services.

Out of the box, XData generates complete API documentation using either **Swagger UI** or **Redoc**. And I'm not talking about some basic list of endpoints—I mean full, interactive documentation that includes:

- All available endpoints and their HTTP methods
- Request and response schemas
- Parameter definitions and types

- Example requests and responses
- The ability to test endpoints directly from the documentation interface

This happens automatically. You write your service code, and the documentation is generated from your actual implementation. There's no separate documentation file to maintain, no risk of drift between docs and code, and no need to remember to update documentation after every change.

For anyone who has spent hours manually updating API documentation or dealt with the embarrassment of handing outdated docs to a client, this feature alone is transformative. Your documentation is always current because it's derived directly from your code.

2. Comprehensive Database Connectivity Options

A web service is only as good as its ability to work with your data. This is where many frameworks stumble—they either lock you into a specific database or make multi-database support unnecessarily complicated.

TMS XData provides remarkable flexibility in database connectivity, supporting a wide range of options to match whatever your project requires:

Native Support for Critical Databases:

- **Microsoft SQL Server**: Full native support for one of the most common enterprise databases
- **SQLite**: Perfect for embedded scenarios, testing, or lightweight applications

FireDAC Integration: When you need to connect to other databases, XData integrates seamlessly with Delphi's FireDAC framework, opening up access to virtually any database system: PostgreSQL, MySQL, Oracle, InterBase, Firebird, DB2, and many others. FireDAC's powerful features—like connection pooling, batch operations, and cached updates—are all available to your XData services.

ORM Through TMS Aurelius: For those who prefer working at a higher abstraction level, XData integrates beautifully with TMS Aurelius, a powerful Object-Relational Mapping framework. With Aurelius, you can:

- Work with objects instead of SQL
- Let the ORM handle CRUD operations
- Benefit from automatic query generation
- Use LINQ-style queries for data access
- Support multiple databases without changing your code

This means you're not boxed in. Whether you're maintaining a legacy system with a specific database, building a new greenfield project, or need to support multiple database backends for different clients, XData has you covered. You can choose the data access strategy that fits your architecture rather than adapting your architecture to fit the framework's limitations.

3. Cross-Platform Deployment: Windows AND Linux

One of the most powerful aspects of modern Delphi development is often overlooked: **the ability to compile for multiple platforms from the same codebase**. TMS XData fully embraces this capability.

You can deploy your XData web services to both Windows and Linux servers using the Delphi Linux compiler. This is a game-changer for several reasons:

Deployment Flexibility: Your clients or hosting environment prefers Linux? No problem. Need to run on Windows Server? That works too. You write your code once and deploy wherever makes the most sense.

Cost Optimization: Linux servers often come with lower licensing costs than Windows Server. Being able to deploy to Linux can significantly reduce infrastructure expenses, especially in cloud environments where every dollar counts.

Modern DevOps Practices: Many container orchestration platforms and cloud-native deployments favor Linux. With XData's Linux support, you can containerize your services, deploy to Kubernetes clusters, or leverage modern infrastructure without being locked into Windows-only hosting.

Development to Production Consistency: Develop on Windows (where most Delphi developers are comfortable) and deploy to Linux production servers without code changes or compatibility concerns.

This cross-platform capability means your Delphi web services can run anywhere, giving you the deployment freedom that modern applications demand. For Delphi developers, this is a significant advantage—you're not sacrificing deployment options by staying in the Delphi ecosystem.

4. Web Frontend Integration with TMS WEB Core

XData can be used with any frontend technology, but one Web framework offers specific components: **TMS WEB Core**, TMS's comprehensive web frontend solution. This combination creates a complete, end-to-end Delphi web development stack.

TMS XData serves as the ideal backend for TMS WEB Core, TMS's comprehensive web frontend solution. This combination creates a complete, end-to-end Delphi web development stack.

TMS WEB Core is the web frontend framework from TMS Software that enables Delphi developers to create modern web applications using their existing Delphi skills. When paired with XData as the backend, you get a powerful, cohesive development experience.

When TMS WEB Core Makes Sense: This approach is particularly valuable if you want to stay entirely within the Delphi ecosystem and leverage your existing team's skills without introducing new technologies. It's an excellent choice for teams who aren't ready to adopt TypeScript, React, or modern JavaScript frameworks, or for projects where maintaining a unified technology stack is more important than adopting cutting-edge frontend technologies.

Note: I've also written extensively about migrating Delphi applications to modern TypeScript/React/Next.js frontends while keeping XData as the backend—a powerful combination for teams ready to embrace modern web development practices. Both approaches have their place depending on your team's skills, project requirements, and technology adoption timeline.

Rapid Development Cycles: The tight integration between WEB Core and XData means faster development iterations. Changes to your data model or service interface are immediately reflected across both tiers.

Seamless Component Integration: TMS WEB Core provides multiple specialized components designed specifically for connecting to XData backends:

- **TWebHttpRequest**: For direct REST API calls to your XData services
- **TWebClientDataSet**: Enables traditional dataset-based programming against XData endpoints
- **TXDataWebConnection**: Purpose-built connector that handles authentication, connection management, and data synchronization

Unified Development Experience: What makes this combination particularly powerful is that you can use the same Delphi skillset for both frontend and backend development. Your team doesn't need to learn separate languages or frameworks—the same developers who build your XData services can create the web frontend using familiar Delphi patterns and components.

5. Seamless Integration with TMS FlexCel for Reporting and PDF Generation

Web applications frequently need to generate reports, invoices, statements, or other formatted documents. This is where many web service frameworks leave you on your own, forcing you to cobble together third-party libraries or external services.

TMS XData works beautifully in conjunction with TMS FlexCel, making report generation and PDF creation straightforward and powerful.

FlexCel is a sophisticated component suite for creating and manipulating Excel files and generating PDFs from Excel templates. When combined with XData, you can:

Create Professional Reports: Design your report templates in Excel (which non-developers can do), then have your XData service populate them with data and generate PDFs on demand. No complex report designer to learn, no proprietary template formats.

Generate Documents on the Fly: Need to create invoices, statements, contracts, or any formatted document from your web service? FlexCel makes it simple. Your XData endpoints can return PDFs directly or store them for later retrieval.

Leverage Excel's Power: Excel is an incredibly capable tool for designing layouts, creating calculations, and formatting data. FlexCel lets you use all of that capability in your automated document generation.

No Client-Side Dependencies: Everything happens server-side in your XData service. Your web application requests a report, and your service returns a ready-to-use PDF. Clean, simple, and secure.

This integration is particularly valuable for Delphi developers who are building web services to support business applications. Reporting is often a critical requirement, and having a first-class solution that works seamlessly with your web service framework eliminates a major pain point.

6. No Deployment Licensing Costs

Let's talk about something that often gets overlooked until it becomes a painful surprise: **deployment licensing**.

In 2025, this is far from a given. Many commercial frameworks operate on models where you purchase development licenses and then face additional costs for every production server, every CPU core, or every concurrent user. These costs can add up quickly, especially as your application scales or when you're deploying across multiple client environments.

With TMS XData, you purchase a license to develop, and production deployment is free.

Read that again: **free production deployment**. No per-server fees. No per-core charges. No unexpected bills when you need to scale up to handle growth.

This licensing model has several significant advantages:

Predictable Costs: You know exactly what you're paying upfront. There are no surprises when your application succeeds and you need to deploy additional instances.

Freedom to Scale: Your success doesn't trigger penalties. Need to add more servers to handle increased load? Go ahead—no license fees standing in your way.

Multiple Deployments: Working with multiple clients, each needing their own server? Deploy as many instances as needed without multiplying your licensing costs. Whether you're deploying to Windows or Linux servers, the licensing remains the same: free for production.

Development and Testing: Set up staging environments, testing servers, and development instances without worrying about license compliance or additional costs.

This approach aligns the vendor's interests with yours. TMS wants you to succeed with XData, not to charge you more as your success grows. In an era where cloud costs and SaaS subscriptions can quickly spiral, having one component with straightforward, development-only licensing is refreshing.

Why This Matters for Delphi Developers Specifically

Let me emphasize this point: **I'm not claiming TMS XData is the best web service framework for all developers in all languages**. That would be absurd. If you're primarily a Python developer, you should probably be looking at FastAPI or Flask. If you're in the .NET ecosystem, ASP.NET Core is excellent. Node.js developers have wonderful options with Express and NestJS.

But **if you're a Delphi developer**, the calculus is different. You're working in an ecosystem with specific strengths, and you want tools that leverage those strengths rather than fighting against them. TMS XData is purpose-built for Delphi developers who need to create web services.

Here's what that means in practice:

You can leverage your existing Delphi skills and code. That business logic you've been maintaining for years? It can power your web services directly. Those Delphi libraries you've built up? They're available to your XData services. You're not starting from scratch or translating your expertise into a completely different paradigm.

You get the performance characteristics that Delphi is known for—compiled, native code running efficiently on your server. You're not dealing with interpreted languages or managed runtimes if that's not your preference.

You maintain consistency across your technology stack. If your desktop applications, services, and backend systems are all in Delphi, adding XData web services keeps everything in one ecosystem, simplifying maintenance, deployment, and knowledge management.

The Complete Package for Delphi Shops

These factors—automatic documentation, flexible database connectivity, cross-platform deployment, seamless reporting capabilities, and deployment-friendly licensing—combine to create a framework that respects both your development time and your business constraints within the Delphi ecosystem.

But beyond these specific features, there's something else worth mentioning: TMS Software has built a reputation for quality, responsiveness, and continuous improvement. The XData framework receives regular updates, the documentation is thorough, and the support forum is active. When you encounter an issue or need guidance, there's a community and a responsive team behind the product.

Is It Right for You?

If you're a Delphi developer building web services, TMS XData deserves serious consideration. It's built for your language, your ecosystem, and your way of working. It doesn't try to force Delphi into patterns that work better

in other languages—it embraces what makes Delphi development productive and extends those principles into the world of web services.

Other languages have other excellent options, and that's as it should be. Each ecosystem evolves tools optimized for its particular strengths. For Delphi developers, TMS XData is that optimized tool. It lets you build modern, documented, cross-platform web services while leveraging your existing skills, code, and knowledge.

Yes, I work closely with the company behind XData, and yes, that relationship has deepened my familiarity with the product. But my recommendation is based on the framework's merits: it saves development time, reduces maintenance burden, provides deployment flexibility across Windows and Linux, integrates with powerful reporting tools, and doesn't punish you financially for success.

For Delphi developers building web services in 2025, TMS XData is simply the best tool available. It's purpose-built for our ecosystem, and it shows in every aspect of its design and functionality.

Are you a Delphi developer working with web services? Have you tried TMS XData or have questions about implementing web services in Delphi? I'd love to hear about your experiences or help answer any questions.

INDEPENDENT · AD-FREE · READER-SUPPORTED

Free to read. Not free to make.

Every tutorial, article, and line of example code on Holger's Code is published without a paywall and without ads. This page is about what keeps it that way.

Professional work, published free

The content here is held to the same standard as professional client work: every tutorial is built as a real project, tested against current versions, documented step by step, and revisited when the frameworks move on. A single in-depth tutorial routinely takes days — and that's before the bills behind it: servers, storage, build infrastructure, licenses, and the hardware everything is verified on.

If an article or tutorial here has saved you an afternoon of debugging, taught you a technique, or helped you ship, it created real value for you. Supporting it isn't charity — it's paying for professional work you found useful, at whatever price you choose. And if you can't, keep reading anyway. That's what the free part means.

Ways to support

KO-FI

Buy me a coffee

A one-time tip — no account, no subscription, any amount.

BOOKS & COURSES

Buy the paid work

The paid tier of everything published free here — and you get something lasting in return.

SPREAD THE WORD

Share an article

Send a tutorial to a colleague, cite an article, subscribe to the RSS feed.

Nothing here goes behind a paywall, whether you chip in or not. But if something on this site earned its keep today, this is where to say so.

Support on Ko-fi —