SYNERGEX

Unlocking the Power of Modern Web Services

Revolutionizing DBL solutions through RESTful architecture

WHITE PAPER

Contents

Executive summary

RESTful web services have become an increasingly important focus for business application developers as they strive to integrate with other solutions, enhance their UI/UX, and protect their software investment. This white paper explains how developers can easily build RESTful web services to expose Synergy data and business logic. In this paper, we provide a high-level overview of the basic principles of REST and RESTful web services, discuss the benefits of adopting REST architecture, define how to get started building your own REST services for your DBL application, and examine the extensive benefits that Synergex's Harmony Core open-source framework has to offer in making this transition.

What are web services?

Web services are a mechanism to securely expose APIs (functionality and data) to other applications via web technologies, such as HTTP. Essentially, web services are remote subroutines you can interact with to access a piece of data or accomplish tasks. Called by client software (desktop or server apps, websites, other web services, mobile apps, the Internet of Things, etc.), they expose data and business logic via the web. All modern applications use client-server topology, and web services are the most common way to do it.

Web services are usually protected by authentication and authorization processes. They're generally well documented and easy to use. Although there have been several iterations over the years, the current gold standard for web services is REST. REST is the de facto standard for building web services, and most public APIs are now based on REST principles. Following REST guidelines results in web services that are fast, scalable, simple, modifiable, portable, reliable, and testable.

Why create web services?

Web services enable you to integrate different, disparate applications, whether in-house or third-party, using industry-standard transports, protocols, and payloads.

Web services are an important component of application modernization strategies. They

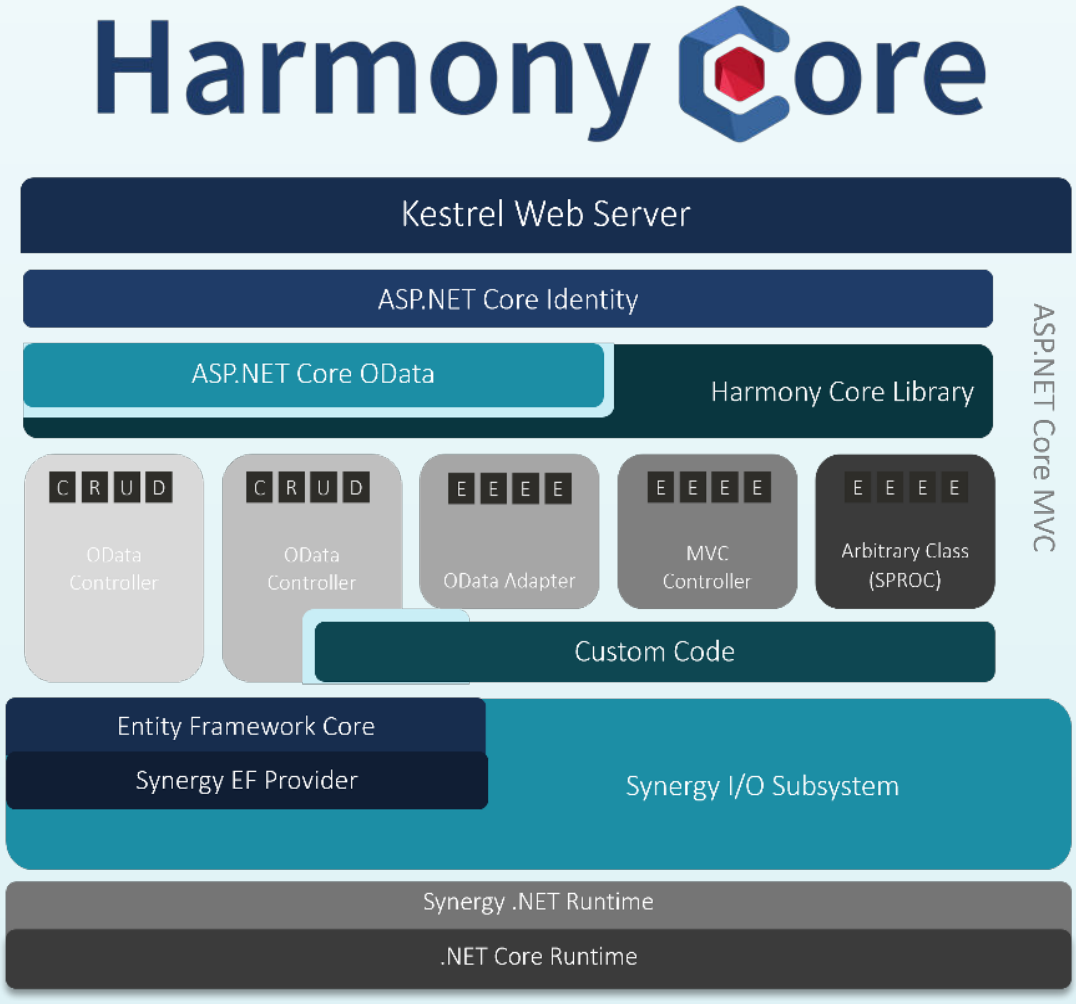
- **Simplify** the process of updating your front-end UIs, allowing you to update more frequently. Client apps can be thin, presentation-only layers.
- **Future-proof** the core of your application by enabling it to integrate with modern technologies.
- **Protect** your investment in your existing code and data, offering an opportunity for evolution instead of revolution.
- **Take advantage** of the latest and greatest UI technologies.

Harmony Core web services framework

Our solution for helping Synergy developers implement web services is Harmony Core. Harmony Core is a modular software development framework developed by Synergex that is open sourced, standards based, and developed and supported on GitHub. A series of libraries distributed in binary format as NuGet packages, Harmony Core runs anywhere .NET runs and offers full integration with all Synergy platforms.

Harmony Core helps you use your existing Synergy code in completely new ways. It enables you to build RESTful web services and OData APIs with minimal effort using repository-defined structures (or other sources of Synergy metadata, such as xfServerPlus methods in a Synergy Method Catalog). By leveraging information in your data structure definitions to generate code, it allows you to expose data directly without writing any custom code.

If you do want to add custom code, the environment is completely extensible, with extensibility points throughout the software. Custom code can be written in Synergy .NET or traditional Synergy. Harmony Core provides support for adapters, which are a way to put OData front ends on custom code, and it can also expose custom code in a variety of ways.



Features and capabilities

Harmony Core has multiple components, and you can pick and choose the parts that make sense for your environment. The primary capabilities of Harmony Core are

- **Code-generated data-centric web services.** Harmony Core exposes the data structures in your repository, including information about relations (aka joins), and provides code-generated services for your data files.
- **Code-generated code-centric web services.** This is the migration path for modernizing xfServerPlus environments. The code generator reads your method catalog and creates the necessary web service wrappers for all that code.
- **Custom code-centric web services.** You can extend web services through custom code-centric endpoints, where you provide 100% of the code. This can be achieved using .NET or traditional Synergy through a Harmony Core mechanism called Traditional Bridge.

Harmony Core also supports

- Multi-tenancy (multiple data sets with runtime-defined file specs and locations)
- Allocation of primary keys for new records (if necessary, via your custom code)
- Custom field types
- Calculated fields
- Custom endpoints based on .NET or traditional Synergy code on all systems (Windows, Unix, Linux, OpenVMS)
- OAuth2 or custom code authentication using JSON web tokens (JWTs)
- Controller, endpoint, and field-level authorization

Traditional Bridge

Traditional Bridge is a technology that helps you incorporate legacy code into a Harmony Core solution. By enabling you to call traditional Synergy routines from .NET services, Traditional Bridge allows your existing functionality to take full advantage of modern software architectures. Depending on your requirements, your traditional Synergy routine can provide all the business logic required for an entire web service endpoint, or it can merely contribute supporting functionality for operations you're performing in the web service you need to interact with.

The underpinnings of Harmony Core

Harmony Core enables you to create RESTful web services built on top of OData, ASP.NET Core, and Entity Framework Core. Harmony Core's APIs are self-documenting using Swagger UI.

OData

OData (Open Data Protocol) provides powerful query tools for your REST APIs (like SQL, but for web services). When using OData, Harmony Core generates plain JSON data, which is easily consumable by most clients. It lets you expose entities as operations your code supports and then manages the filtering and relationships, so your code doesn't have to. Using OData significantly reduces the amount of work required to expose Synergy logic and data via a web service.

ASP.NET Core

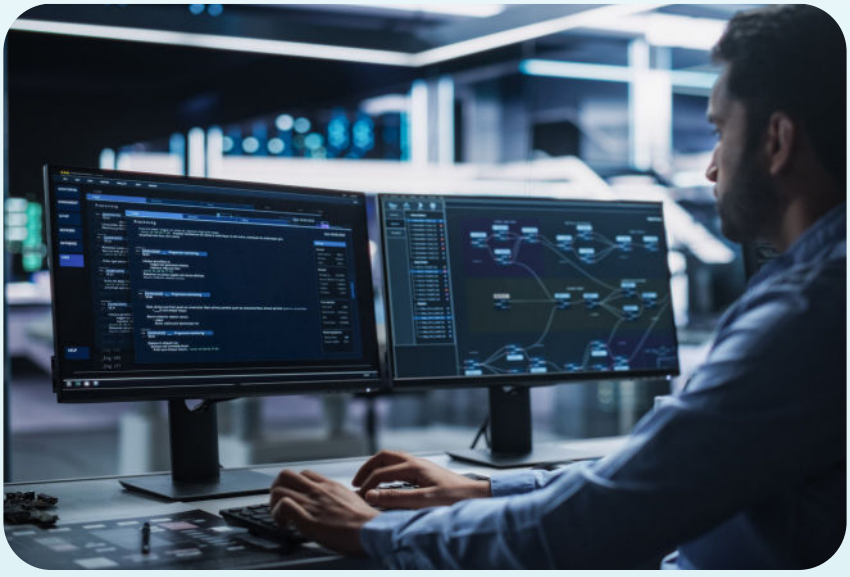
Microsoft describes ASP.NET Core as "a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps." Developed openly on GitHub to foster community involvement, ASP.NET Core offers modularity, scalability, and robust security.

Entity Framework Core

EF Core emphasizes portability, performance, and extensibility. A key feature is support for non-relational databases, including Synergy DBMS. It integrates well with DBL's Select class, which can perform all the underlying read operations that EF Core supports, and therefore is ideal for accessing DataObjects. Leveraging EF Core's optimizations with Select's efficient data access improves the performance of data retrieval operations.

Swagger UI

Swagger UI is another open-source, industry-standard tool that provides a way of documenting and testing APIs. It has a user-friendly interface that enables you to easily see available endpoints, required parameters, and response formats.



Benefits of using Harmony Core

Harmony Core framework

Easy to learn. Well documented and quick ramp-up time.

Protects your investment. Preserves your investment in existing code and data.

Minimal code disruption. Enables modern web services with minimal code changes.

Built on industry standards. Leverages standards for security, end-to-end encryption, etc.

Interoperable. Provides a consistent way to interact with third-party software.

Simplified data access. Eliminates the need for most data access code.

Reliable and well-supported. Actively maintained, stable, and backed by expert support resources.

Harmony Core–based web services

Flexible. Easily customizable to match your unique business needs.

Fast. Adherence to performance best practices ensures optimal performance is built in.

Scalable. Designed to grow with your business, handling more users, data, and workloads without sacrificing performance.

Extensible. Future-ready—add new features or integrate with new tools and frameworks as needed.

Accessible. Reachable from any client, thanks to standards-based RESTful APIs.

Easy. Simple to create, build, maintain, and debug, with developer-friendly tools.

Secure. Built with modern security best practices to protect your data and services.

Steps to developing web services

It's easy to start your web services journey with Harmony Core. It works almost out of the box, is well-documented, and is backed by Synergex's expert Developer Support team. The basic steps are

1. Describe your Synergy repository. If you already use xfODBC or ReportWriter, your repository is probably already in great shape.
2. Create a new Harmony Core development solution in Visual Studio. Project templates to give you a starter environment are provided.
3. Generate code based on your repository structures. List the data structures you want to generate services for and execute CodeGen.
4. Add the generated code into the solution.
5. Build, run, and test.

For more detailed information and tutorials, visit the Harmony Core wiki.

Requirements

Licensing

Harmony Core uses the BSD-2 Clause license: You're free to use it however you'd like. Regular Synergy Runtime licenses will be consumed when you run your DBL code under Synergy .NET or the traditional Synergy runtime.

Required tools

For Harmony Core development, you'll need the following components installed on a Windows development machine:

- Visual Studio 2022 or later, with the ".NET desktop development" and "ASP.NET and web development" workloads (at a minimum) installed and all updates applied
- .NET SDK 8.0
- Synergy/DE 12.1 or higher (preferably the latest version of all Synergy products at all times), including Synergy/DE 32-bit and 64-bit and Synergy DBL Integration for Visual Studio
- CodeGen (a tool that developers working in a Synergy development environment can use to generate source code in a variety of languages)
- Harmony Core Solution Templates (to help you get up and running quickly within a recommended pre-configured development environment)
- Harmony Core Solution Upgrade Tool (a command-line tool that upgrades projects created with earlier versions of Harmony Core)
- SSL certificate (because Harmony Core web services are exposed via the HTTPS protocol, you need an SSL certificate to install on your web server)

For more details, refer to the Required Tools section of the Harmony Core wiki.

CodeGen

CodeGen is an open-source tool that automatically generates source code written in DBL or other languages. It interprets metadata (stored in a Synergy repository) that defines data structures along with rules (stored in template files) that define how a given piece of code should be created and merges that information to produce source code that addresses a specific requirement.

CASE STUDIES

Learn how a few of our customers have implemented web services.

Synergex Helps UNFI Bring Harmony to Supplier Portal

When United Natural Foods, Inc. (UNFI) faced challenges with multi-user data access and system scaling in its supplier portal, Synergex's Professional Services Group helped them leverage the Harmony Core framework to enhance multi-threaded access and allow faster SQL-style queries. The changes reduced data retrieval times, allowed real-time updates for orders, and simplified data access for developers, significantly improving system efficiency.

Read more

Forward Solutions Fuels Its ERP Refresh with Harmony Core

Forward Solutions partnered with Synergex to modernize their ERP system's user interface. To update the UI without a full application rewrite, Synergex proposed using Harmony Core with Traditional Bridge. The new architecture simplified cloud deployment and improved performance, giving the ERP a modern, scalable web-based interface and setting it up for future growth while preserving its existing business logic and functionality.

Read more

You can find additional success stories on the Synergex website.

Conclusion

Creating web services for DBL applications with Harmony Core enables you to…

- Expose application functionality and data to other apps (whether building new applications yourself or exposing APIs for third-party venders and customers)
- Build dashboards and enable ad hoc queries through tools like Excel and PowerBI Desktop
- Provide observability by integrating with platforms like Datadog, Grafana, or Prometheus
- Protect your software investment by enabling secure, industry-standard integration with modern technologies, with minimal effort
- Deliver lightweight, modern UIs

To learn more about Harmony Core and building RESTful web services, visit the Harmony Core wiki or contact Synergex today.

Get expert guidance from Synergex Professional Services Group

If you'd like assistance with your web services project, our experienced team can help you get started.

Learn About PSG

Synergex provides software development tools, application integration technologies, and expert consulting services to help enterprise application developers retain their software investment, keep up with advancing technologies, and bring their applications into the future. Since 1976, Synergex technologies have been the foundation of applications that drive commerce around the world. Every day, millions of users interact with these systems in e-commerce, global logistics, manufacturing, healthcare, and other industries.