# Gunslinger Documentation

Get it right the first time, every time with Gunslinger!

## What is it?

Gunslinger combines the concept of data providers and templates, making it easy to use configuration to generate whole new sections of your code base.

## What can it do?

Gunslinger uses a combination of configuration and templates to create file output based on metadata provided by any number of Data Providers.

**Example:** A SQL Data Provider would return metadata for all of the tables in a given database.

## The case for code generation

Imagine starting a project and being 80% done on the first day! Generated code can give us stable, working, testable code with the press of a button. In a scenario where a code base has very predictable elements and a template has been refined, generated code can absolutely provide 100% complete and working code. Generated code may not provide all of the bells and whistles that we might want for a finished product, but it can provide a working starting point that can then be customized to deliver value more quickly and reliably.

## Templating

Gunslinger uses [Handlebars.Net](https://github.com/rexm/Handlebars.Net) to render templates. Handlebars source is available on Github and the package is available on NuGet. Handlebars has a good pedigree, being an enhancement on the mustache template engine written for .Net. It is blisteringly fast and quite easy to use.

## Data Providers

Gunslinger comes with three Data Providers that provide metadata for the following sources:

1. SQL Server Database – the most common use case
2. Swagger Endpoint – connect to an API with greater ease
3. Class Library (DLL) – the sky is the limit with this new *alpha* *feature*

It is relatively easy to wire up a new Data Provider, though you may never need to do so.

A Data Provider class can be created for pretty much any data source that has a consistent rule set. Here are some examples of data sources for which a provider could be easily created:

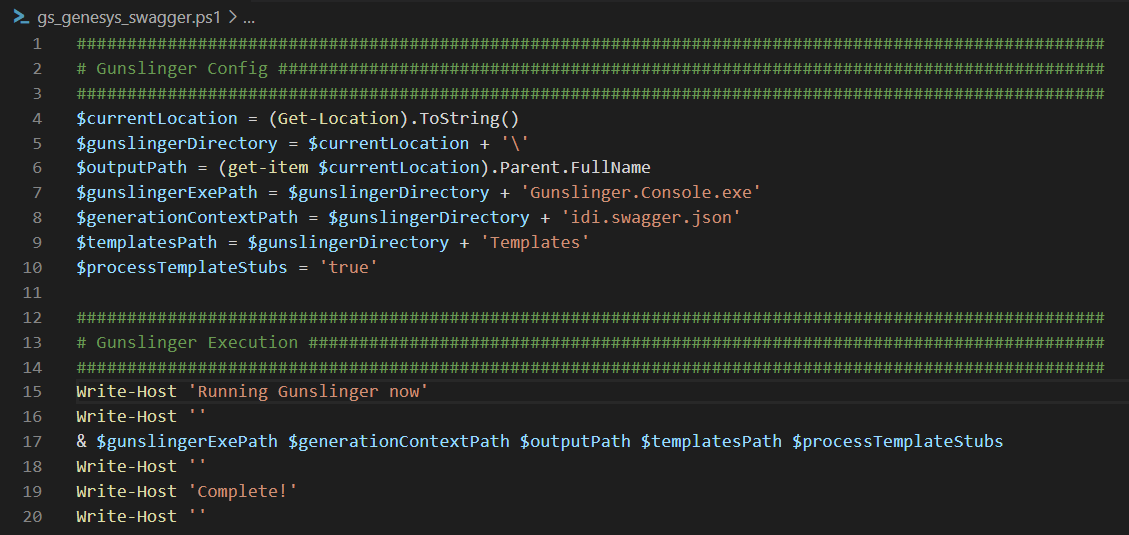
* JSON file source
* XML file source

## Operate with a scalpel

You don't have to run every generator that is used in your project at once because each time that Gunslinger is run it references a single JSON Configuration file that can configure as many or as few Data Providers and Templates as needed. In this way, we can group related processes or isolate distinct processes.

## Run it with PowerShell!

Run Gunslinger however you want by creating PowerShell scripts that run Gunslinger once, or several times in succession.

Running Gunslinger 

Any errors that come up will display in the PowerShell window. When errors occur, they are often issues with accessibility to the requested data source.

## JSON Configuration

##### OutputDirectory

Data Type: string

Indicates the root path for the output for the current context.

**Example:** C:\Projects\GunslingerOutput\

##### ProcessTemplateStubs

Data Type: boolean

Indicates whether or not to run templates that have been designated as “stubs” via the IsStub property.

##### TemplateDirectory

Data Type: string

Indicates the root location of the templates that will be used for this context.

##### ExcludeTheseEntities

Data Type: string array

Indicates entities that should be ignored in this context. This is useful when there are entities that are unwanted in the output.

##### ExcludeTheseTemplates

Data Type: string array

Indicates templates that should be ignored in this context. This is useful when you don’t want to create a new configuration file, but also want to skip some templates.

##### IncludeTheseEntitiesOnly

Data Type: string array

Indicates that all entities other than the ones listed here should be ignored in this context. This is useful when you know that changes have been made to an entity or a small group of entities and you don’t want to process output for any others.

##### IncludeTheseTemplatesOnly

Data Type: string array

Indicates that all templates other than the ones listed here should be ignored in this context. This is useful when you don’t want to create a new configuration file, but also want to skip some templates.

##### Resources

Data Type: Resource Array

##### Resource

##### Source

Data Type: string

##### Destination

Data Type: string

Used for copying files or folders from a source to a destination. Useful for making sure that certain artifacts that might change are included in the output.

##### DataProviders

Data Type: DataProvider array

##### DataProvider

##### Name

Data Type: string

This value needs to be unique within the current configuration document. It is used to reference the data provider from within the template configuration.

##### TypeName

Data Type: string

Used to reference the actual C# type in the compiled source. Currently available values are:

* SQLModelDataProvider
* SwaggerDataProvider
* ReflectionDataProvider (alpha)
* ReflectionMethodInfoDataProvider (alpha)

##### UseLocalDataSource

Data Type: boolean

Indicates to use the “LocalDataSource” property.

##### LocalDataSource

Data Type: string

An alternate string used to connect to a local resource. Only active when the “UseLocalDataSource” property is set to true. Use this when there is limited connectivity to a production data source or when you want to manually modify a test version of a data source.

##### DataSource

Data Type: string

The string used to connect to the resource. When using SQLModelDataProvider this will be a connection string, with the SwaggerDataProvider it will be a URL.

##### OpenDataSourceUrlInDefaultBrowser

Data Type: boolean

This allows us to mitigate issues with authorization on some endpoints by opening the data source URL in a browser window. A particular endpoint that was used early on while developing Gunslinger required a user to authorize by visiting the page in a browser before the endpoint will allow us to pull data from that application context. This helps speed up the resolution of that issue.

##### NonSpecifiedPropertiesAreNullable

Data Type: boolean

Indicates whether or not to treat all properties that aren’t specified as required to be treated as if they were nullable. This is only relevant within the SwaggerDataProvider and is not the standard way that Swagger treats properties. This option is here to provide the ability to allow the SwaggerDataProvider to behave in a non-standard way because, for better or worse, some endpoints are configured this way and Gunslinger needed to accommodate that. Templates

Data Type: Template array

##### Template

##### Type

Data Type: TemplateType enum

##### TemplateType

**Setup** – indicates that only one file will be output. Information about every model will be provided to the template.

**Model** – indicates that one file per model will be output.

##### Name

Data Type: string

Mostly just an identifier for the user who configures the template, but also used in reporting an error finding the template file.

##### Namespace

Data Type: string

Available for use inside the template. The intended use is to designate the output file’s namespace.

##### IsStub

Data Type: boolean

Designates a template as a “stub”, or a file that should be used by the developer to extend a generated base class. When the “ProcessTemplateStubs” value is true a template with this value will get run, when the “ProcessTemplateStubs” value is false it will not.

##### InputRelativePath

Data Type: string

The file path of the template. This is relative to the “TemplateDirectory” value provided.

**Example**: SwaggerModel.txt

**Example**: SwaggerTemplates\\SwaggerModel.txt

##### OutputRelativePath

Data Type: string

The file path for the files output by this template. This is relative to the “OutputDirectory” value provided.   
There are two special template strings can be used inside this string, they are {schema} and {entityName}. Schema is only relevant for the SQLModelDataProvider.

**Example**: \\{entityName}.cs

**Example**: Data\\{schema}\\Models\\{entityName}.cs

##### Language

Data Type: Language enum

##### Language

csharp  
sql  
javascript  
html

This value helps the metadata property factories determine how to translate data types from the data source type system to the type system in the destination language. Set this value to reflect the language of the output file type.

##### RunOnce

Data Type: Boolean

No longer in use. Ignore this or remove the reference. It is being removed in favor of the “Type” designation.

##### DataProviderName

Data Type: string

Indicates the source Data Provider that the metadata for running the template will come from.

##### Imports

Data Type: string array

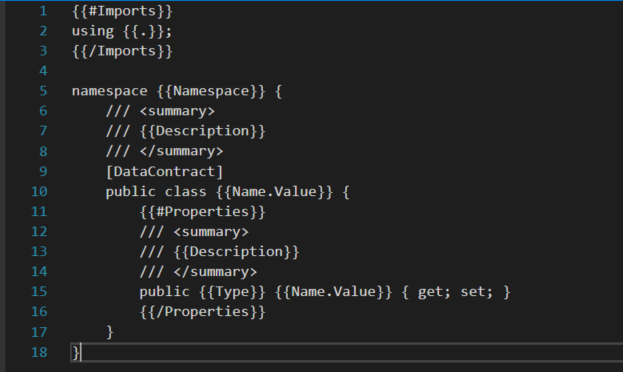
Available for use inside the template. The intended use is to assign the imports used in the file that is output. It is not uncommon for some models to need imports that other models don’t need. Every import for the entire batch of files will need to be included here for the output code to compile. This may result in some files having more imports than needed.

## Template Example

There are some good template examples in the source code in the Tests project.

Here are some for quick reference.

Simple Model



Slightly more complex  
