The content of this guide is updated for PR #493

How to Use

Documentation Generation (generate-md-docs)

Add the following to the pom file of the module containing the extension classes. (Please note that all configuration parameters are not required. Check Configuration).

The filename of the generated documentation file is inferred based on the project version and this file will be generated inside the "api/" directory inside the "`docGenBaseDirectory`". The home page would be updated accordingly as well with a list of versions available in the "api/" directory.

```
<plugin>
   <groupId>org.wso2.siddhi</groupId>
   <artifactId>siddhi-doc-gen</artifactId>
   <version>${siddhi.version}</version>
   <executions>
       <execution>
           <goals>
               <goal>generate-md-docs</goal>
           </goals>
       </execution>
   </executions>
   <configuration>
       <moduleTargetDirectory>../target/</moduleTargetDirectory>
       <homePageTemplateFile>../../../README.md</homePageTemplateFile>
       <mkdocsConfigFile>../../../mkdocs.yml</mkdocsConfigFile>
       <docGenBaseDirectory>../../../docs/</docGenBaseDirectory>
       <homePageFileName>index</homePageFileName>
       <readMeFile>../../../README.md</readMeFile>
   </configuration>
</plugin>
```

Configuration

Configuration	Optional	Default Value	Description
`moduleTargetDirectory`	Yes	{current-module-build-directory}/	The build directory ("target/" directory) from which the extension metadata will be retrieved. The class files in the classes directory in the build directory will be

			loaded and from them the annotated metadata will be retrieved.
`homePageTemplateFile`	Yes	{root-maven-module-directory}/README.md	The path to the hoem page template file. The content of this file will be copied to the top of the mkdocs homepage with a list of “API Docs” at the bottom.
`mkdocsConfigFile`	Yes	{root-maven-module-directory}/mkdocs.yml	The path to the mkdocs configuration file
`docGenBaseDirectory`	Yes	{root-maven-module-directory}/docs/	The base directory in which the documentation files will be generated.
`homePageFileName`	Yes	index	The name of the home page file that will be used by mkdocs. This is relative to the `docGenBaseDirectory`. Markdown file extension (.md) is automatically added.
`readmeFile`	Yes	{root-maven-module-directory}/README.md	The README file of the project

Extensions Index Generation (generate-extensions-index)

Add the following to the pom file of the siddhi main repository’s siddhi-core-doc-gen module. (Please note that all configuration parameters are not required. Check Configuration)

```
<plugin>
   <groupId>org.wso2.siddhi</groupId>
   <artifactId>siddhi-doc-gen</artifactId>
   <version>${siddhi.version}</version>
   <executions>
       <execution>
           <goals>
               <goal>generate-extensions-index</goal>
           </goals>
       </execution>
   </executions>
   <configuration>
       <docGenBaseDirectory>../../../docs/</docGenBaseDirectory>
       <indexGenFileName>extensions</indexGenFileName>
```

```
        <extensionRepositoryOwner>wso2-extensions</extensionRepositoryOwner>
        <extensionRepositories>
            <extensionRepository>siddhi-map-xml</extensionRepository>
            <extensionRepository>siddhi-execution-extrema</extensionRepository>
            <extensionRepository>siddhi-execution-math</extensionRepository>
            <extensionRepository>siddhi-gpl-execution-geo</extensionRepository>
        </extensionRepositories>
    </configuration>
</plugin>
```

Configuration

Configuration	Optional	Default Value	Description
`extensionRepositories`	No	None	The extension repositories names list.
`extensionRepositoryOwner`	Yes	wso2-extensions	The owner (On GitHub) of the extension repository.
`docGenBaseDirectory`	Yes	{root-maven-module-directory}/docs/	The directory in which the index file will be generated.
`indexGenFileName`	Yes	extensions	The name of the output extensions index file that will be generated. This is relative to the `docGenBaseDirectory`. Markdown file extension (.md) is automatically added.

MkDocs site Deployment on GitHub Pages (deploy-mkdocs-github-pages)

Add the following to the pom file. (Please note that all configuration parameters are not required. Check Configuration)

```
<plugin>
    <groupId>org.wso2.siddhi</groupId>
    <artifactId>siddhi-doc-gen</artifactId>
    <version>${siddhi.version}</version>
    <executions>
        <execution>
            <goals>
                <goal>deploy-mkdocs-github-pages</goal>
            </goals>
```

```
        </execution>
    </executions>
    <configuration>
        <mkdocsConfigFile>../../../mkdocs.yml</mkdocsConfigFile>
        <docGenBaseDirectory>../../../docs/</docGenBaseDirectory>
        <readMeFile>../../../README.md</readMeFile>
    </configuration>
</plugin>
```

Automating the goal execution with maven release plugin

This goal is intended to be run as a preparation step of the “prepare” goal of the maven release plugin. This can be achieved using maven profiles. To do so use the following steps.

1. Add a profile and add the plugin goal to the profile

```
<profile>
	<id>documentation-deploy</id>
	<build>
		<plugins>
			<plugin>
				<groupId>org.wso2.siddhi</groupId>
				<artifactId>siddhi-doc-gen</artifactId>
				<executions>
					<execution>
						<phase>compile</phase>
						<goals>
							<goal>deploy-mkdocs-github-pages</goal>
						</goals>
					</execution>
				</executions>
			</plugin>
		</plugins>
	</build>
</profile>
```

2. Activate the profile in the preparation goal of the release:prepare goal

```
<plugin>
	<groupId>org.apache.maven.plugins</groupId>
	<artifactId>maven-release-plugin</artifactId>
	<configuration>
		<preparationGoals>clean install -Pdocumentation-deploy</preparationGoals>
		<autoVersionSubmodules>true</autoVersionSubmodules>
	</configuration>
</plugin>
```

Configuration

Configuration	Optional	Default Value	Description
`mkdocsConfigFile`	Yes	{root-maven-module-directory}/mkdocs.yml	The path to the mkdocs configuration file
`docGenBaseDirectory`	Yes	{root-maven-module-directory}/docs/	The directory in which the index file will be generated.
`readmeFile`	Yes	{root-maven-module-directory}/README.md	The README file of the project

Special Notations in homePageTemplate (README.md file by default)

By adding the following headings in the home page template file, sections containing useful information can be generated automatically in the home page. If the heading names need to be changed, a source code level change is required in the Constants class in the siddhi-doc-gen module.

- ## Features
- ## Latest API Docs

Features

The heading "## Features" can be used for adding a section in the homePageTemplate (README.md file by default). Adding this heading will replace the content below it until a heading of equal level or higher is encountered, with a list of extension processors and their descriptions.

Latest API Docs

The heading "## Features" can be used for adding a section in the homePageTemplate (README.md file by default). Adding this heading will replace the content below it until a heading of equal level or higher is encountered, with a link to the latest API Docs version.

How to add a new Extension (Source code level changes)

1. Add the new extension to ExtensionType enum class and add the superclass of the extension type to the superClassMap hash map. (The value of the enums should be similar to the actual

names used throughout all the documentation. This will affect the names displayed in the documentation as well as the hyperlinks)

2. Update the "`resources/documentation.md.ftl`" to support the new extension type