devonfw guide

Updated at 2021-02-26 10:22:20 UTC

Table of Contents

Unresolved directive in master.asciidoc - include::getting-started.wiki/getting-started.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::ide.wiki/devonfw-ide.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::devon4j.wiki/devon4j.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::devon4ng.wiki/master-devon4ng.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::devon4net.wiki/master-devon4net.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::devon4node.wiki/master-devon4node.asciidoc[leveloffset=0]

Part I: Choosing your Database

1. Database

For your business application with devonfw you need to choose the right database. In devonfw we are not biased for a particular product so you have the freedom of choice.

1.1. RDBMS

The classical and well-established form of a database is a relational database management system (RDBMS). In devonfw we recommend to use an RDBMS unless you have specific need. However, in case you have the need for big data, graph-data, BLOB focus, or schema-less dynamic data you can have a look at NoSQL options but be aware that these may be experimental and are not fully supported by devonfw.

1.1.1. Options

In devonfw we are not biased for a particular RDBMS so you have the freedom of choice. Here are the most common options:

- SAP Hana (high performance in-memory, many advanced features)
- Oracle (most established, well featured for enterprise)
- PostgreSQL (great open-source RDBMS)
- MariaDB (true OSS successor of MySQL)
- MS SQL Server (best choice for Microsoft and Windows dominated IT landscapes)

Please click on any of the above choices and go to the according guide to find specific detials such as client/driver.

1.2. NoSQL

While not (yet) officially supported and recommendet there are also interesting NoSQL (Not Only SQL) databases that could be an interesting alternative. Please be aware that you will typically not be able to use JPA (and hibernate). Further before choosing a NoSQL database you should check the following aspects:

- Is the database of choice reliable and mature enough for your project?
- Can the operators of your product support the database of choice properly (provisioning, administration, backup, scaling & clustering, monitoring, etc.)?
- Does the database of choice meet the requirements of your project (ACID vs. eventual consistencey, CAP theorem)?

There are good reasons to choose a particular NoSQL database in specific cases (e.g. extreme demand for big-data, throughput or scaling). But as indicated by the questions above you need to be fully aware of what you are doing. NoSQL databases can be *schemaless* (untyped, dynamic & flexible) and/or *schemaful* (typed, structured & strict). Furhter, there are different types of NoSQL databases that are discussed in the following sub-sections:

1.2.1. Column DB

Column NoSQL databases are more related to a regular RDBMS with their tables and columns. However, they typically do not offer relational support with joins to the same level as you expect from an RDBMS. Therefore, you have to carefully design your data-model upfront with the all the knowledge how you later want to query your data.

The most prominent options are:

- Cassandra (high-performance, schema-based DB)
- HBase (distributed, big-data Hadoop database)

1.2.2. Key-Value DB

As indicated by the name, a key-value database stores objects as key/value pairs similar to `Properties` or `Map` in Java.

The most prominent options are:

- Redis (in-memory key/value store, especially used as cache or message broker)
- aerospike

1.2.3. Document DB

A document database is similar to a key-value database, but it stores objects in standard structured formats such as XML, JSON, or BSON. Therefore not only flat key/value pairs but even trees of hierarchical data can be stored, retrieved and queried.

The most prominent options are:

- MongoDB
- CouchDB
- RavenDB

1.2.4. Graph DB

If the connections (links/relations) between your data is key and an RDBMS is just not flexible or fast enough for your plans, then a graph database can help you. They are very strong on storing and querying complex connections between entities. For queries there are even specific standards and languages like Gremlin.

The most prominent options are:

- neo4j
- blazegraph

1.2.5. Hybrid DB

In addition to the above types there are some NoSQL databases that are hybrid and combine the features and aspects of these types. While as an architect and developer you might love the idea to get all in one, you have to be careful with your choice. If you do not exactly know your problem, you are not ready to make the right choice for your database. Further, you might still be best-off with an good old RDBMS if you need to address multiple aspects together. Anyhow, for experiments, PoCs, or small microservices with little risk it might be a great idea to choose a hybrid NoSQL database. If you have collected very positive, profound and productive experience with such product you can grow on it.

The most prominent options are:

- OrientDB (object-oriented, hyper-flexible, column- and graph-based)

2. SAP HANA

This section contains hints for those who use SAP HANA, a very powerful and fast RDBMS. Besides general hints about the driver there are tips for more tight integration with other SAP features or products.

2.1. Driver

SAP Hana is a commercial and professional product. However, the hana JDBC driver is available in Maven Central what makes it easy to integrate. All you need is the following maven dependency:

```
<dependency>
  <groupId>com.sap.cloud.db.jdbc</groupId>
  <artifactId>ngdbc</artifactId>
  <version>${hana.driver.version}</version>
</dependency>
```

Of course the version (`${hana.driver.version}`) needs to be adopted to your needs (Hana installtion in production, e.g. `2.4.64`). For an overview of available driver versions see here.

2.2. Developer Usage

For your local development environment you will love the free SAP HANA, Express Edition.

You can run HANA in several ways:

- On-premise
  - Via a Docker image (Linux only)
  - Via a pre-configured virtual machine (Windows, Linux, OS X)
  - Installed natively on your local machine (Linux only)
- In the cloud
  - Via a pre-configured machine on the Google Cloud Platform
  - Via a pre-configured machine in the Microsoft Azure Cloud
  - Via a pre-configured machine on Amazon Web Services

To get started with SAP HANA, Express Edition you can check out the tutorials at the SAP Developer Center.

2.3. Pooling

See Overriding the JDBC Connection Pool Settings.

2.4. Fuzzy Search

See https://blogs.sap.com/2015/08/28/dynamism-of-fuzzy-search-in-sap-hana/ or the SAP HANA Search Developer Guide

3. Oracle RDBMS

This section contains hints for those who use Oracle RDBMS. Besides general hints about the driver there are tips for more tight integration with other Oracle features or products. However, if you work for a project where Oracle RDBMS is settled and not going to be replaced (you are in a vendor lock-in anyway), you might want to use even more from Oracle technology to take advantage from a closer integration.

3.1. XE

For local development you should setup Oracle XE (eXpress Edition). You need an oracle account, then you can download it from here.

The most comfortable way to run it as needed is using docker. You can build your own docker image from the downloaded RPM using the instructions and dockerfile from oracle. The following commands will build and start Oracle XE `18.4.0` on your machine:

```
git clone https://github.com/oracle/docker-images.git
cd docker-images/OracleDatabase/SingleInstance/dockerfiles
./buildDockerImage.sh -x -v 18.4.0
docker run -d -p 1521:1521 --name=oracle-xe --restart=always -e ORACLE_PWD=«my-sys-
pwd» oracle/database:18.4.0-xe
```

Please note that the `buildDockerImage.sh` will take a long time. Further after `docker run` has passed you need to give time for your new container to startup and setup the Oracle XE DB. So be patient and give it some time. (In case the build of the docker-image fails reproducibly and you want to give up with the Dockerfiles from Oracle you can also try this inofficial docker-oracle-xe solution. However, this is not recommended and may lead to other problems.).

Starting with XE 18c you need to be aware that oracle introduced a multi-tenant architecture. Hence `xe` refers to the root `CDB` while you typically want to connect to the `PDB` (pluggable database) and XE ships with exactly one of this called `xepdb1`. To connect to your local XE database you need to use `xepdb1` as the `Service Name` (typically in `SQL Developer`). The `hostname` should be `localhost` and the `port` is by default `1521` if you did not remap it with docker to something else. In order to create schema users, use `sys` as `Username` and change `Role` to `SYSDBA`.

Hint: If you happen to end up connected to `xe` instead of `xepdb1` in some case (e.g. in sqlplus), you may switch using this statement:

```
ALTER SESSION SET CONTAINER = XEPDB1;
```

The JDBC URL for your Oracle XE Database is:

```
jdbc:oracle:thin:@//localhost:1521/xepdb1
```


To locally connect as sysdba without password use the following command (`connect / as sysdba` is not working anymore):

```
sqlplus sys/Oracle18@localhost/XE as sysdba
```

3.2. Driver

The oracle JDBC driver is available in maven central. Oracle JDBC drivers usually are backward and forward compatible so you should be able to use an older driver with a newer Oracle DB, etc. Your dependency for the oracle driver should look as follows:

```
<dependency>
  <groupId>com.oracle.database.jdbc</groupId>
  <artifactId>ojdbc10</artifactId>
  <version>${oracle.driver.version}</version>
</dependency>
```

For the most recent Oracle DB 19 the property `oracle.driver.version` should be `19.8.0.0`. The number in the `artifactId` correlates to the minimum Java Version so for Java8 `artifactId` should be `ojdbc8` instead. It is fine to use `ojdbc10` with Java11 or higher.

3.3. Pooling

In order to boost performance JDBC connections should be pooled and reused. If you are using Oracle RDBMS and do not plan to change that you can use the Oracle specific connection pool "Universal Connection Pool (UCP)" that is perfectly integrated with the Oracle driver. According to the documentation, UCP can even be used to manage third party data sources. Like the JDBC driver also the UCP is available in maven central. The dependency should look like this:

```
<dependency>
  <groupId>com.oracle.database.jdbc</groupId>
  <artifactId>ucp</artifactId>
  <version>${oracle.ucp.version}</version>
</dependency>
```

with property `oracle.ucp.version` analogue to `oracle.driver.version`.

Configuration is done via application.properties like this (example):


```
#Oracle UCP
# Datasource for accessing the database
spring.datasource.url=jdbc:oracle:thin:@192.168.58.2:1521:xe
spring.jpa.database-platform=org.hibernate.dialect.Oracle12cDialect
spring.datasource.user=MyUser
spring.datasource.password=ThisIsMyPassword
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
spring.datasource.schema=MySchema

spring.datasource.type=oracle.ucp.jdbc.PoolDataSourceImpl
spring.datasource.factory=oracle.ucp.jdbc.PoolDataSourceFactory
spring.datasource.factory-method=getPoolDataSource
spring.datasource.connectionFactoryClassName=oracle.jdbc.pool.OracleDataSource
spring.datasource.validateConnectionOnBorrow=true
spring.datasource.connectionPoolName=MyPool
spring.datasource.jmx-enabled=true

# Optional: Set the log level to INTERNAL_ERROR, SEVERE, WARNING, INFO, CONFIG, FINE,
TRACE_10, FINER, TRACE_20, TRACE_30, or FINEST
# logging.level.oracle.ucp=INTERNAL_ERROR
# Optional: activate tracing
# logging.level.oracle.ucp.jdbc.oracle.OracleUniversalPooledConnection=TRACE

#Optional: Configures pool size manually
#spring.datasource.minPoolSize=10
#spring.datasource.maxPoolSize=40
#spring.datasource.initialPoolSize=20
```

Resources: FAQ, developer's guide, Java API Reference. For an in-depth discussion on how to use JDBC and UCP, see the Oracle documentation Connection Management Strategies for Java Applications using JDBC and UCP.

Note: there is a bug in UCP 12.1.0.2 that results in the creation of thousands of java.lang.Timer threads over hours or days of system uptime (see article on stackoverflow). Also, Oracle has a strange bug fixing / patching policy: instead of producing a fixed version 12.1.0.3 or 12.1.0.2.x, Oracle publishes collections of *.class files that must be manually patched into the ucp.jar! Therefore, use the newest versions only.

3.4. Messaging

In case you want to do messaging based on JMS you might consider the Oracle JMS also called Oracle Streams Advanced Queuing, or Oracle Advanced Queuing, or OAQ or AQ for short. OAQ is a JMS provider based on the Oracle RDBMS and included in the DB product for no extra fee. OAQ has some features that exceed the JMS standard like a retention time (i.e. a built-in backup mechanism that allows to make messages "unread" within a configurable period of time so that these messages do not have to be resent by the sending application). Also, OAQ messages are stored in relational tables so they can easily be observed by a test driver in a system test scenario. Capgemini has used the Spring Data JDBC Extension in order to process OAQ messages within **the same technical**

transaction as the resulting Oracle RDBMS data changes **without** using 2PC and an XA-compliant transaction manager - which is not available out of the box in Tomcat. This is possible only due to the fact that OAQ queues and RDBMS tables actually reside in the same database. However, this is higher magic and should only be tried if high transaction rates must be achieved by avoiding 2PC.

3.5. General Notes on the use of Oracle products

Oracle sells commercial products and receives licence fees for them. This includes access to a support organization. Therefore, at an early stage of your project, prepare for contacting oracle support in case of technical problems. You will need the Oracle support ID **of your customer** [i.e. the legal entity who pays the licence fee and runs the RDBMS] and your customer must grant you permission to use it in a service request - it is not legal to use a your own support ID in a customer-related project. Your customer pays for that service anyway, so use it in case of a problem!

Software components like the JDBC driver or the UCP may be available without a registration or fee but they are protected by the Oracle Technology Network (OTN) License Agreement. The most important aspect of this licence agreement is the fact that an IT service provider is not allowed to simply download the Oracle software component, bundle it in a software artefact and deliver it to the customer. Instead, the Oracle software component must be (from a legal point of view) provided by the owner of the Oracle DB licence (i.e. your customer). This can be achieved in two ways: Advise your customer to install the Oracle software component in the application server as a library that can be used by your custom built system. Or, in cases where this is not feasible, e.g. in a OpenShift environment where the IT service provider delivers complete Docker images, you must advise your customer to (legally, i.e. documented in a written form) provide the Oracle software component to you, i.e. you don't download the software component from the Oracle site but receive it from your customer.

3.6. Fix for TNS-Listener issues

When switching networks (e.g. due to VPN) you might end up that your local Oracle XE stopps working with this error:

```
Listener refused the connection with the following error:
ORA-12505, TNS:listener does not currently know of SID given in connect descriptor
```

While a reboot resolves this problem, it is a huge pain to reboot every time this error occurs as this wastes a lot of time. Therefore we suggest the following fix:

- Go to your oracle installation and open the folder `product/«version»/dbhomeXE/network/admin`.
- Edit the file `listener.ora` and change the value of the property `HOST` from your qualified hostname to `localhost` (`HOST = localhost`).
- Edit the file `tnsnames.ora` and change the value of the `HOST` properties (two occurences) from your qualified hostname to `localhost` (`HOST = localhost`).
- Reboot your machine or (on windows) restart the service `OracleServiceXE` via `services.msc`.
- Now this problem should be gone forever and you can continue your work.

On older XE versions until 11g you could run the following SQL (`sqlplus / as sysdba @reset_tns_listener.sql`):

```
WHENEVER SQLERROR EXIT;
ALTER SYSTEM SET local_listener = '(ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT
= 1521))';
ALTER SYSTEM REGISTER;
EXIT;
```

4. MS-SQL-Server

This section gives guidance and hints for those who use Microsoft SQL Server as RDBMS.

4.1. Driver

Microsoft SQL Server is a commercial and professional product. However, the JDBC driver is MIT licensed and available in Maven Central what makes it easy to integrate. Your dependency for the driver should look as following:

```
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>${mssqlserver.driver.version}</version>
</dependency>
```

Of course the version (`${mssqlserver.driver.version}`) needs to be adopted to your needs (SQL Server installtion in production and JDK version, e.g. `7.4.1.jre8`). For an overview of available driver versions see here.

5. PostgreSQL

This section gives guidance and hints for those who use PostgreSQL as RDBMS.

5.1. Driver

PostgreSQL is fully open-source. The driver is therefore available in maven central. Your dependency for the driver should look as following:

```
<dependency>
    <groupId>postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>${postgresql.driver.version}</version>
</dependency>
```

Of course the version (`${postgresql.driver.version}`) needs to be adopted to your needs (PostgreSQL installtion in production and JDBC level suitable for your JDK, e.g. `9.1-901-1.jdbc4`). For an overview of available driver versions see here.

6. MariaDB

This section gives guidance and hints for those who use MariaDB as RDBMS.

6.1. Driver

MariaDB is fully open-source. The driver is therefore available in maven central. Your dependency for the driver should look as following:

```
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>${mariadb.driver.version}</version>
</dependency>
```

Of course the version (`${mariadb.driver.version}`) needs to be adopted to your needs (MariaDB installtion in production and JDK version, e.g. `2.5.1`). For an overview of available driver versions see here.

7. Database

For your business application with devonfw you need to choose the right database. In devonfw we are not biased for a particular product so you have the freedom of choice.

7.1. RDBMS

The classical and well-established form of a database is a relational database management system (RDBMS). In devonfw we recommend to use an RDBMS unless you have specific need. However, in case you have the need for big data, graph-data, BLOB focus, or schema-less dynamic data you can have a look at NoSQL options but be aware that these may be experimental and are not fully supported by devonfw.

7.1.1. Options

In devonfw we are not biased for a particular RDBMS so you have the freedom of choice. Here are the most common options:

- SAP Hana (high performance in-memory, many advanced features)
- Oracle (most established, well featured for enterprise)
- PostgreSQL (great open-source RDBMS)
- MariaDB (true OSS successor of MySQL)
- MS SQL Server (best choice for Microsoft and Windows dominated IT landscapes)

Please click on any of the above choices and go to the according guide to find specific detials such as client/driver.

7.2. NoSQL

While not (yet) officially supported and recommendet there are also interesting NoSQL (Not Only SQL) databases that could be an interesting alternative. Please be aware that you will typically not be able to use JPA (and hibernate). Further before choosing a NoSQL database you should check the following aspects:

- Is the database of choice reliable and mature enough for your project?
- Can the operators of your product support the database of choice properly (provisioning, administration, backup, scaling & clustering, monitoring, etc.)?
- Does the database of choice meet the requirements of your project (ACID vs. eventual consistencey, CAP theorem)?

There are good reasons to choose a particular NoSQL database in specific cases (e.g. extreme demand for big-data, throughput or scaling). But as indicated by the questions above you need to be fully aware of what you are doing. NoSQL databases can be *schemaless* (untyped, dynamic & flexible) and/or *schemaful* (typed, structured & strict). Furhter, there are different types of NoSQL databases that are discussed in the following sub-sections:

7.2.1. Column DB

Column NoSQL databases are more related to a regular RDBMS with their tables and columns. However, they typically do not offer relational support with joins to the same level as you expect from an RDBMS. Therefore, you have to carefully design your data-model upfront with the all the knowledge how you later want to query your data.

The most prominent options are:

- Cassandra (high-performance, schema-based DB)
- HBase (distributed, big-data Hadoop database)

7.2.2. Key-Value DB

As indicated by the name, a key-value database stores objects as key/value pairs similar to `Properties` or `Map` in Java.

The most prominent options are:

- Redis (in-memory key/value store, especially used as cache or message broker)
- aerospike

7.2.3. Document DB

A document database is similar to a key-value database, but it stores objects in standard structured formats such as XML, JSON, or BSON. Therefore not only flat key/value pairs but even trees of hierarchical data can be stored, retrieved and queried.

The most prominent options are:

- MongoDB
- CouchDB
- RavenDB

7.2.4. Graph DB

If the connections (links/relations) between your data is key and an RDBMS is just not flexible or fast enough for your plans, then a graph database can help you. They are very strong on storing and querying complex connections between entities. For queries there are even specific standards and languages like Gremlin.

The most prominent options are:

- neo4j
- blazegraph

7.2.5. Hybrid DB

In addition to the above types there are some NoSQL databases that are hybrid and combine the features and aspects of these types. While as an architect and developer you might love the idea to get all in one, you have to be careful with your choice. If you do not exactly know your problem, you are not ready to make the right choice for your database. Further, you might still be best-off with an good old RDBMS if you need to address multiple aspects together. Anyhow, for experiments, PoCs, or small microservices with little risk it might be a great idea to choose a hybrid NoSQL database. If you have collected very positive, profound and productive experience with such product you can grow on it.

The most prominent options are:

- OrientDB (object-oriented, hyper-flexible, column- and graph-based)

8. Cassandra

This section is the place to share experience for those who use Cassandra as NoSQL database.

8.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

8.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

8.3. Spring-Data

There is spring-data support available for cassandra via spring-data-cassandra.

Please note that some time ago we had feedback from projects that had issues with spring-data-cassandra and switched back to using the driver natively. We assume the issues are meanwhile resolved. TODO: collect more feedback and update this guide.

9. neo4j

This section is the place to share experience for those who use neo4j as NoSQL database.

9.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

9.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

9.3. Spring-Data

There is spring-data integration available via spring-data-neo4j.

10. MongoDB

This section is the place to share experience for those who use MongoDB as NoSQL database.

10.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

10.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

11. CouchDB

This section is the place to share experience for those who use CouchDB as NoSQL database.

11.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

11.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

12. Redis

This section is the place to share experience for those who use Redis as NoSQL database.

12.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

12.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

13. OrientDB

This section is the place to share experience for those who use OrientDB (see also Open-Source community edition) as NoSQL database.

13.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

13.2. Driver

For driver options see here.

13.3. Administration

OrientDB comes with a powerful, impressive admin interface for your web-browser called Studio.

14. Blazegraph

This section is the place to share experience for those who use Blazegraph as NoSQL database.

14.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

14.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here.

15. HBase

This section is the place to share experience for those who use HBase as NoSQL database.

15.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

15.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see here and hbase-java-api tutorial.

16. RavenDB

This section is the place to share experience for those who use RavenDB as NoSQL database.

16.1. Attention

In devonfw we do not properly support this. You are free to use this database but do not expect advanced integration, CobiGen templates, etc.

16.2. Driver

Please be aware that there is not a regular JDBC driver in case you are using Java (devon4j). For driver options see ravendb-jvm-client and Java Client Features.

Unresolved directive in master.asciidoc - include::shop-floor.wiki/master-devonfw-shop-floor.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::cicdgen.wiki/master-cicdgen.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::production-line.wiki/master-production-line.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::cobigen.wiki/master-cobigen.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::mrchecker.wiki/master-mrchecker.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::my-thai-star.wiki/master-my-thai-star.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::dashboard.wiki/master-dashboard.asciidoc[leveloffset=0]

Unresolved directive in master.asciidoc - include::solicitor.wiki/master-solicitor.asciidoc[leveloffset=0]

Part II: Contributing

Unresolved directive in general/master-contributing.asciidoc - include::../.github/CONTRIBUTING.asciidoc[leveloffset=1]

Unresolved directive in general/master-contributing.asciidoc - include::../.github/CODE_OF_CONDUCT.asciidoc[leveloffset=1]

17. OSS Compliance

This chapter helps you to gain transparency on OSS usage and reach OSS compliance in your project.

17.1. Preface

devonfw, as most Java software, makes strong use of Open Source Software (OSS). It is using about 150 OSS products on the server only and on the client even more. Using a platform like devonfw to develop your own custom solution requires handling contained OSS correctly, i.e acting *OSS-compliant*.

Please read the Open Source policy of your company first, e.g. the Capgemini OSS Policy which contains a short, comprehensive and well written explanation on relevant OSS-knowledge. Make sure you:

- understand the copyleft effect and its effect in commercial projects
- understand the 3 license categories: "permissive", "weak copyleft" and "strong copyleft"
- know prominent license types as e.g. "Apache-2.0" or GPL-3.0" and what copyleft-category they are in
- are aware that some OSS offer dual/multi-licenses
- Understand that OSS libraries often come with sub-dependencies of other OSS carrying licenses themselves

To define sufficient OSS compliance measures, contact your IP officer or legal team as early as possible, especially if you develop software for clients.

17.2. Obligations when using OSS

If you create a custom solution containing OSS, this in legal sense is a "derived" work. If you distribute your derived work to your business client or any other legal entity in binary packaged form, the license obligations of contained OSS get into effect. Ignoring these leads to a license infringement which can create high damage.

To carefully handle these obligations you must:

- maintain an OSS inventory (to gain transparency on OSS usage and used licenses)
- check license conformity depending on usage/distribution in a commercial scenario
- check license compatibility between used OSS-licenses
- fulfill obligations defined by the OSS-licenses

Obligations need to be checked per license. Frequent obligations are:

- deliver the license terms of all used versions of the OSS licenses
- not to change any copyright statements or warranty exclusions contained in the used OSS

components

- deliver the source code of the OSS components (e.g. on a data carrier)
- when modifying OSS, track any source code modification (including date and name of the employee/company)
- display OSS license notice in a user frontend (if any)
- other obligations depending on individual license

17.3. Automate OSS handling

Carefully judging the OSS usage in your project is a MANUAL activity! However, collecting OSS information and fulfilling license obligations should be automated as much as possible. A prominent professional tool to automate OSS compliance is the commercial software "Black Duck". Unfortunately it is rather expensive - either purchased or used as SaaS.

The most recommended lightweight tooling is a combination of Maven plugins. We will mainly use the Mojo Maven License Plugin.

17.4. Configure the Mojo Maven License Plugin

You can use it from command line but this will limit the ability to sustainably configure it (shown later). Therefore we add it permanently as a build-plugin to the project parent-pom like this (already contained in OASP-parent-pom):


```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>license-maven-plugin</artifactId>
  <version>1.14</version>

  <configuration>
    <outputDirectory>${project.build.directory}/generated-resources</outputDirectory>
    <sortArtifactByName>true</sortArtifactByName>
    <includeTransitiveDependencies>true</includeTransitiveDependencies>
    <!-- the "missing file" declares licenses for dependencies that could not be
detected automatically -->
    <useMissingFile>true</useMissingFile>
    <!-- find the "missing files" in all child-projects at the following location -->
    <missingFile>src/license/THIRD-PARTY.properties</missingFile>
    <!-- if the "missing files" are not yet existing in child-projects they will be
created automatically -->
    <failOnMissing>false</failOnMissing>
    <overrideFile>src/license/override-THIRD-PARTY.properties</overrideFile>
    <!-- harmonize different ways of writing license names -->
    <licenseMerges>
      <licenseMerge>Apache-2.0|Apache 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|Apache License, Version 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|Apache Software License, Version 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|The Apache Software License, Version 2.0</licenseMerge>
    </licenseMerges>
    <encoding>utf-8</encoding>
  </configuration>
</plugin>
```

In the config above there are several settings that help to permanently improve the result of an automated OSS scan. We explain these now.

17.4.1. Declare additional licenses

Sometimes the licenses of used OSS cannot be resolved automatically. That is not the mistake of the maven-license-tool, but the mistake of the OSS author who didn't make the respective license-information properly available.

Declare additional licenses in a "missing file" within *each* maven-subproject: /src/license/THIRD-PARTY.properties.


```
# Generated by org.codehaus.mojo.license.AddThirdPartyMojo
#-------------------------------------------------------------------------------
# Already used licenses in project :
# - ASF 2.0
# - Apache 2
...
#-------------------------------------------------------------------------------
# Please fill the missing licenses for dependencies :
...
dom4j--dom4j--1.6.1=BSD 3-Clause
javax.servlet--jstl--1.2=CDDL
...
```

In case the use of "missing files" is activated, but the THIRD-PARTY.properties-file is not yet existing, the first run of an "aggregate-add-third-party" goal (see below) will fail. Luckily the license-plugin just helped us and created the properties-files automatically (in each maven-subproject) and prefilled it with:

- a list of all detected licenses within the maven project
- all OSS libraries where a license could not be detected automatically.

You now need to fill in missing license information and rerun the plugin.

17.4.2. Redefine wrongly detected licenses

In case automatically detected licenses proof to be wrong by closer investigation, this wrong detection can be overwritten. Add a configuration to declare alternative licenses within each maven-subproject: /src/license/override-THIRD-PARTY.properties

```
com.sun.mail--javax.mail--1.5.6=Common Development and Distribution License 1.1
```

This can be also be useful for OSS that provides a multi-license to make a decision which license to actually choose .

17.4.3. Merge licenses

You will see that many prominent licenses come in all sorts of notations, e.g. Apache-2.0 as: "Apache 2" or "ASL-2.0" or "The Apache License, Version 2.0". The Mojo Maven License Plugin allows to harmonize different forms of a license-naming like this:


```
    <!-- harmonize different ways of writing license names -->
    <licenseMerges>
      <licenseMerge>Apache-2.0|Apache 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|Apache License, Version 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|Apache Software License, Version 2.0</licenseMerge>
      <licenseMerge>Apache-2.0|The Apache Software License, Version 2.0</licenseMerge>
    </licenseMerges>
```

License-names will be harmonized in the OSS report to one common term. We propose to harmonize to short-license-IDs defined by the SPDX standard.

17.5. Retrieve licenses list

For a quick initial judgement of OSS license situation run the following maven command from command line:

```
$ mvn license:license-list
```

You receive the summary list of all used OSS licenses on the cmd-out.

17.6. Create an OSS inventory

To create an OSS inventory means to report on the overall bill of material of used OSS and corresponding licenses. Within the parent project, run the following maven goal from command line.

```
$ mvn license:aggregate-download-licenses -Dlicense.excludedScopes=test,provided
```

Running the aggregate-download-licenses goal creates two results.

1. a license.xml that contains all used OSS dependencies (even sub-dependencies) with respective license information
2. puts all used OSS-license-texts as html files into folder target/generated resources

Carefully validate and judge the outcome of the license list. It is recommended to copy the license.xml to the project documentation and hand it over to your client. You may also import it into a spreadsheet to get a better overview.

17.7. Create a THIRD PARTY file

Within Java software it is a common practice to add a "THIRD-PARTY" text file to the distribution. Contained is a summary-list of all used OSS and respective licenses. This can also be achieved with the Mojo Maven License Plugin.

Within the parent project, run the following maven goal from command line.


```
$ mvn license:aggregate-add-third-party -Dlicense.excludedScopes=test,provided
```

Find the THIRD-PARTY.txt in the folder: target\generated-resources. The goal aggregate-add-third-party also profits from configuration as outlined above.

17.8. Download and package OSS SourceCode

Some OSS licenses require handing over the OSS source code which is packaged with your custom software to the client the solution is distributed to. It is a good practice to hand over the source code of *all* used OSS to your client. Collecting all source code can be accomplished by another Maven plugin: Apache Maven Dependency Plugin.

It downloads all OSS Source Jars into the folder: \target\sources across the parent and all child maven projects.

You configure the plugin like this:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>3.0.2</version>

  <configuration>
    <classifier>sources</classifier>
    <failOnMissingClassifierArtifact>false</failOnMissingClassifierArtifact>
    <outputDirectory>${project.build.directory}/sources</outputDirectory>
  </configuration>
  <executions>
    <execution>
      <id>src-dependencies</id>
      <phase>package</phase>
      <goals>
        <!-- use unpack-dependencies instead if you want to explode the sources -->
        <goal>copy-dependencies</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

You run the plugin from command line like this:

```
$ mvn dependency:copy-dependencies -Dclassifier=sources
```

The plugin provides another goal that also unzips the jars, which is not recommended, since contents get mixed up.

Deliver the OSS source jars to your client with the release of your custom solution. This has been

done physically - e.g. on DVD.

17.9. Handle OSS within CI-process

To automate OSS handling in the regular build-process (which is not recommended to start with) you may declare the following executions and goals in your maven-configuration:

```
<plugin>
  ...

  <executions>
    <execution>
      <id>aggregate-add-third-party</id>
      <phase>generate-resources</phase>
      <goals>
        <goal>aggregate-add-third-party</goal>
      </goals>
    </execution>

    <execution>
      <id>aggregate-download-licenses</id>
      <phase>generate-resources</phase>
      <goals>
        <goal>aggregate-download-licenses</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Note that the build may fail in case the OSS information was not complete. Check the build-output to understand and resolve the issue - like e.g. add missing license information in the "missing file".

Part III: Release Notes

18. devonfw Release notes 2020.12

18.1. Introduction

We are proud to announce the release of devonfw version 2020.12.

This release includes lots of addition of new features, updates and bug fixes but it is very important to highlight the following improvements:

18.2. devonfw IDE

The consolidated list of features for this devonfw IDE release is as it follows.

18.2.1. 2020.12.001

Update with the following bugfixes and improvements:

- #495: Documentation corrections
- #491: Consider lombok support
- #489: Update node to v12.19.0 and VS Code to 1.50.1
- #470: reverse merge of workspace settings not sorting properties anymore
- #483: Error during installation when npm is already installed
- #415: documentation to customize settings
- #479: Error for vscode plugin installation
- #471: Preconfigure Project Explorer with Hierarchical Project Presentation

The full list of changes for this release can be found in milestone 2020.12.001.

18.2.2. 2020.08.001

Update with the following bugfixes and improvements:

- #454: update to eclipse 2020.06
- #442: update nodejs and vscode
- #432: vsCode settings are not updated
- #446: intellij: doConfigureEclipse: command not found
- #440: Software update may lead to inconsistent state due to windows file locks
- #427: release: keep leading zeros
- #450: update settings
- #431: devon build command not working correct for yarn or npm
- #449: update to devon4j 2020.08.001

The full list of changes for this release can be found in milestone 2020.08.001.

18.2.3. 2020.04.004

Minor update with the following bugfixes and improvements:

- #433: Windows: devon command line sets wrong environment variables (with tilde symbol)
- #435: fix variable resolution on bash

The full list of changes for this release can be found in milestone 2020.04.004.

18.2.4. 2020.04.003

Minor update with the following bugfixes and improvements:

- #395: variable from devon.properites unset if value is in double quotes
- #429: Added script to create a meta file in the users directory after setup

The full list of changes for this release can be found in milestone 2020.04.003.

18.2.5. 2020.04.002

Minor update with the following bugfixes and improvements:

- #418: Make projects optional
- #421: update devon4j to 2020.04.002
- #413: Update Eclipse to 2020-03
- #424: Strange errors on windows if devon.properties contains mixed line endings
- #399: launching of Intellij fails with No such file or directory error.
- #410: fix jsonmerge for boolean and null values

The full list of changes for this release can be found in milestone 2020.04.002.

18.3. devon4j

The consolidated list of features for this devon4j release is as it follows.

18.3.1. 2020.12.001

New release of `devon4j` with pluggable web security (CSRF starter) and `CompletableFuture` support for async REST service client as well as other improvements:

- #283: Support for CompletableFuture in async service client
- #307: Fix CSRF protection support
- #287: spring-boot update to 2.3.3
- #288: Update jackson to 2.11.2

- #293: Update owasp-dependency-check plugin version to 5.3.2
- #302: added guide for project/app structure
- #315: devon4j documentation correction
- #306: improve documentation to launch app

Documentation is available at devon4j guide 2020.12.001. The full list of changes for this release can be found in milestone devon4j 2020.12.001.

18.3.2. 2020.08.001

New release of `devon4j` with async REST service client support and other improvements:

- #279: support for async service clients
- #277: Update Security-Guide to recent OWASP Top (2017)
- #281: cleanup documentation

Documentation is available at devon4j guide 2020.08.001. The full list of changes for this release can be found in milestone devon4j 2020.08.001.

18.3.3. 2020.04.002

Minor update of `devon4j` with the following bugfixes and small improvements:

- #261: JUnit4 backward compatibility
- #267: Fix JWT permission expansion
- #254: JWT Authentication support for devon4j-kafka
- #258: archetype is still lacking a .gitignore
- #273: Update libs
- #271: Do not enable resource filtering by default
- #255: Kafka: Support different retry configuration for different topics

Documentation is available at devon4j guide 2020.04.002. The full list of changes for this release can be found in milestone devon4j 2020.04.002.

18.4. devon4node

New `devon4node` version is published, the changes are:

On this release we have deprecated devon4node cli, now we use nest cli, and we have added a GraphQL sample.

- #375: GraphQL Sample.
- #257: D4N cli remove

18.5. CobiGen

Various bugfixes were made as well as consolidating behavior of eclipse vs maven vs cli by properly sharing more code across the different clients. Also properly takes into account a files line delimiter instead of defaulting to those of the host system.

CobiGen CLI v7.1.0 CobiGen Maven Plug-in v7.1.0 CobiGen Eclipse Plug-in v7.1.0

18.5.1. Templates

- Removed environment.ts from the crud_angular_client_app/CRUD devon4ng Angular App increment since Cobigen did not make any changes in it
- Removed cross referencing between template increments since there is currently no useful use case for it and it leads to a few problems
- v2020.12.001

18.5.2. Java Plug-in

- Now properly merges using the input files line delimiters instead of defaulting to those of the host system.
- v7.1.0

18.5.3. TypeScript Plug-in

- Fixed NPE Added the option to read a path from an object input
- v7.1.0

18.5.4. Property Plug-in

- Now properly merges using the input files line delimiters instead of defaulting to those of the host system.
- v7.1.0

18.5.5. OpenAPI Plug-in

- Fixed an issue where nullable enums lead to errors
- 7.1.0

18.5.6. Textmerger

- Now properly merges using the input files line delimiters instead of defaulting to those of the host system.
- v7.1.0
- v7.1.1

18.6. Sonar devon4j plugin

With this release, we made the package structure configurable and did some other improvements and fixes:

- #117: Rule from checkstyle plugin could not be instantiated in our quality profile
- #118: NPE during project analysis
- #97: Custom configuration for architecture
- #92: Display warnings on the 'devonfw' config page in the 'Administration' section of SonarQube
- #95: Add 3rd Party rule to avoid Immutable annotation from wrong package
- #94: Add 3rd Party rule to avoid legacy date types
- #93: Improve devonfw Java quality profile
- #114: Deleted unused architecture config from SonarQube settings to avoid confusion

Changes for this release can be found in milestone 2020.12.001 and milestone 2020.12.002

18.7. devon4net

The consolidated list of features for `devon4net` is as follows:

- LiteDb: - Support for LiteDB - Provided basic repository for CRUD operations.
- RabbitMq: - Use of EasyQNet library to perform CQRS main functions between different microservices - Send commands / Subscribe queues with one C# sentence - Events management: Handled received commands to subscribed messages - Automatic messaging backup when sent and handled (Internal database via LiteDB and database backup via Entity Framework)
- MediatR: - Use of MediatR library to perform CQRS main functions in memory - Send commands / Subscribe queues with one C# sentence - Events management: Handled received commands to subscribed messages - Automatic messaging backup when sent and handled (Internal database via LiteDB and database backup via Entity Framework)
- SmaxHcm: - Component to manage Microfocus SMAX for cloud infrastructure services management
- CyberArk: - Manage safe credentials with CyberArk
- AnsibleTower: - Ansible automates the cloud infrastructure. devon4net integrates with Ansible Tower via API consumption endpoints
- gRPC+Protobuf: - Added Client + Server basic templates sample gRPC with Google's Protobuf protocol using devon4net
- Kafka: - Added Apache Kafka support for deliver/consume messages and create/delete topics as well
- AWS support
    - AWS Template to create serverless applications with auto generation of an APIGateway using AWS base template

    - AWS template to create pure Lambda functions and manage SQS Events, SNS Events, Generic Events, CloudWatch, S3 Management, AWS Secrets management as a configuration provider in .NET life cycle
    - AWS CDK integration component to create/manage AWS infrastructures (Infra As Code): Database, Database cluster, VPC, Secrets, S3 buckets, Roles…
- Minor performance and stability improvements such Entity framework migration integration
- Updated to the latest .net Core 3.1 TLS

18.8. dashboard (beta version)

We are adding dashboard beta version as part of this release. Dashboard is a tool that allows you to create and manage devonfw projects.It makes it easy to onboard a new person with devonfw.

- Dashboard list all ide available on user system or if no ide is availble it will provide option to download latest version of ide.
- Project creation and management: Project page list all projects created by user using dahboard. User will be able to create devon4j, devon4ng and devon4node projects using dashboard.
- Support for Eclipse and VSCode IDE
- Integrated devonfw-ide usage guide from the website

18.9. Solicitor

Solicitor is a tool which helps managing Open Source Software used within projects. Below is consolidated feature list of solicitor:

- Standalone Command Line Java Tool
- Importers for component/license information from
- Maven
- Gradle
- NPM
- CSV (e.g. for manual entry of data)
- Rules processing (using Drools Rule Engine) controls the the different phases:
- Normalizing / Enhancing of license information
- Handling of multilicensing (including selection of applicable licenses) and re-licensing
- Legal evaluation
- Rules to be defined as Decision Tables
- Sample Decision Tables included
- Automatic download and file based caching of license texts
- Allows manual editing / reformatting of license text
- Output processing

- Template based text (Velocity) and XLS generation
- SQL based pre-processor (e.g. for filtering, aggregation)
- Audit log which documents all applied rules for every item might be included in report
- "Diff Mode" allows to mark data which has changed as compared to a previous run of Solicitor (in Velocity and XLS reporting)
- Customization
- Project specific configuration (containing e.g. reporting templates, decision tables) allows to override/amend builtin configuration
- Builtin configuration might be overridden/extended by configuration data contained in a single extension file (ZIP format)
- This allows to safely provide organization specific rules and reporting templates to all projects of an organization (e.g. to reflect the specific OSS usage policy of the organization)

18.10. MrChecker

MrChecker Test Framework is an end to end test automation framework written in Java. It is an automated testing framework for functional testing of web applications, API web services, Service Virtualization, Security, native mobile apps and, in the near future, databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions. Below is consolidated list of updates in MrChecker:

- Migration of core module to junit5
- Extension of MrCheckers tests harness
- Migration of mrchecker-example-module to junit 5
- Migration guide https://devonfw.com/website/pages/docs/master-mrchecker.asciidoc_migration-from-junit4-to-junit5.html
- Upgrade to cucumber 6.7.0
- Release of the 3.0.1 version to maven-central

18.11. Trainings/tutorials

- Katakoda tutorials : https://katacoda.com/devonfw
- Youtube tutorials : https://www.youtube.com/channel/UCtb1p-24jus-QoXy49t9Xzg

19. devonfw Release notes 2020.08

19.1. Introduction

We are proud to announce the release of devonfw version 2020.08.

This release includes lots of addition of new features, updates and bug fixes but it is very important to highlight the following improvements:

19.2. devonfw IDE

The consolidated list of features for this devonfw IDE release is as it follows.

19.2.1. 2020.08.001

Update with the following bugfixes and improvements:

- #454: update to eclipse 2020.06
- #442: update nodejs and vscode
- #432: vsCode settings are not updated
- #446: intellij: doConfigureEclipse: command not found
- #440: Software update may lead to inconsistent state due to windows file locks
- #427: release: keep leading zeros
- #450: update settings
- #431: devon build command not working correct for yarn or npm
- #449: update to devon4j 2020.08.001

The full list of changes for this release can be found in milestone 2020.08.001.

19.2.2. 2020.04.004

Minor update with the following bugfixes and improvements:

- #433: Windows: devon command line sets wrong environment variables (with tilde symbol)
- #435: fix variable resolution on bash

The full list of changes for this release can be found in milestone 2020.04.004.

19.2.3. 2020.04.003

Minor update with the following bugfixes and improvements:

- #395: variable from devon.properites unset if value is in double quotes
- #429: Added script to create a meta file in the users directory after setup

The full list of changes for this release can be found in milestone 2020.04.003.

19.2.4. 2020.04.002

Minor update with the following bugfixes and improvements:

- #418: Make projects optional
- #421: update devon4j to 2020.04.002
- #413: Update Eclipse to 2020-03
- #424: Strange errors on windows if devon.properties contains mixed line endings
- #399: launching of Intellij fails with No such file or directory error.
- #410: fix jsonmerge for boolean and null values

The full list of changes for this release can be found in milestone 2020.04.002.

19.3. devon4j

The consolidated list of features for this devon4j release is as it follows.

19.3.1. 2020.08.001

New release of `devon4j` with async REST service client support and other improvements:

- #279: support for async service clients
- #277: Update Security-Guide to recent OWASP Top (2017)
- #281: cleanup documentation

Documentation is available at devon4j guide 2020.08.001. The full list of changes for this release can be found in milestone devon4j 2020.08.001.

19.3.2. 2020.04.002

Minor update of `devon4j` with the following bugfixes and small improvements:

- #261: JUnit4 backward compatibility
- #267: Fix JWT permission expansion
- #254: JWT Authentication support for devon4j-kafka
- #258: archetype is still lacking a .gitignore
- #273: Update libs
- #271: Do not enable resource filtering by default
- #255: Kafka: Support different retry configuration for different topics

Documentation is available at devon4j guide 2020.04.002. The full list of changes for this release can be found in milestone devon4j 2020.04.002.

19.4. devon4ng

This release is focused mainly on the **Angular 10 upgrade**:

- #176: Template submodules updated to Angular 10 and NgRx 10.
- #167, #168, #174 and #175: Updated electron (sample and documentation).
- #166: Update error handler.
- #165: Cypress sample.
- #164: Update to Angular 10 (samples and documentation).

19.5. devon4node

New `devon4node` version is published, the changes are:

- Updated dependencies.
- Solved bug when you introduce a name with dashes in new command.
- Add more options to the non-interactive new command.

19.6. CobiGen

CobiGen version numbers have been consolidated to now represent plug-in compatibility in the major release number (7.x.x).

19.6.1. CLI

- CLI increments can be referenced by name and description.
- Ability to configure logging.
- Fixed error on code formatting.
- Improved Performance by lazy plug-in loading.
- Possibility to prefer custom plug-ins over CobiGen ones.
- Fixed bug, which broke whole CobiGen execution in case a custom CobiGen Plug-in was throwing an arbitrary exception.

19.6.2. Eclipse

- Improved Performance by lazy plug-in loading.
- Possibility to prefer custom plug-ins over CobiGen ones.
- Fixed bug, which broke whole CobiGen execution in case a custom CobiGen Plug-in was throwing an arbitrary exception.

19.6.3. Maven

- Fixed bug to properly load template util classes.
- Improved Performance by lazy plug-in loading.
- Possibility to prefer custom plug-ins over CobiGen ones.
- Fixed bug, which broke whole CobiGen execution in case a custom CobiGen Plug-in was throwing an arbitrary exception.

19.6.4. XML Plug-in

- Added ability to provide custom merge schemas as part of the template folder.
- Added further merge strategies for merging including XML validation.

19.6.5. Java Plug-in

- Fixed NPE for annotated constructors.
- Fixed line separator handling to now prefer the file's one instead of the system ones.
- Fixed unwanted new lines in constructors after merging.
- Fixed annotation formatting after merge.

19.6.6. TypeScript Plug-in

- Fixed issue on automatic update of the ts-merger bundle.

19.7. Sonar devon4j plugin

The consolidated list of features for this `Sonar devon4j plugin` release is as it follows.

With this release, we added our own quality profile:

- #16: Install devon4j quality profile

Changes for this release can be found in milestone 2020.08.001

19.8. My Thai Star with Microservices and ISTIO Service Mesh Implementation

As always, our reference application, `My Thai Star` now has been implemented with Microservices and ISTIO Service Mesh features:

- devon4j - Java
  - My Thai Star now has a sample version on Microservices architecture.
  - The github repository for the microservices version of My Thai Star is hosted at My Thai Star with Microservices

- My Thai Star Microservices now has a multi stage docker build which generates the respective docker images for all the My Thai Star services.
- My Thai Star microservices has the Kubernetes artifacts available to be able to deploy into Kubernetes pods.
- My Thai Star microservices has ISTIO the service mesh implementation.
- Check out the guides to implement or configure ISTIO features such as Traffic Routing, Network Resiliency features(RequestRouting, RequestTimeouts, Fault Injection, Circuit Breaker), Canary Deployments.

20. devonfw Release notes 2020.04

20.1. Introduction

We are proud to announce the immediate release of devonfw version 2020.04. This version is the first one with the new versioning that will make easier to the community to identify when it was released since we use the year and month as many other software distributions.

This release includes lots of bug fixes and many version updates, but it is very important to highlight the following improvements:

- New devonfw IDE auto-configure project feature.
- Improved devonfw IDE plugin configuration.
- New devon4j kafka module.
- New devon4j JWT module.
- New devon4j authorization of batches feature.
- Dozer replaced with Orika in devon4j.
- Support for composite keys in devon4j and CobiGen.
- Multiple enhancements for project specific plugin development and usage of project specific template sets in CobiGen.
- Ability to adapt your own templates by making use of CobiGen CLI.
- Better responsiveness in eclipse and bugfixes in all assets in CobiGen.
- devon4ng updated to Angular 9, NgRx 9 and Ionic 5, including documentation, samples and templates.
- Yarn 2 support in devon4ng.
- devon4node updated to NestJS 7 (packages, samples and documentation)
- devon4node TSLint replaced with ESLint.
- @devon4node/config package added.
- devon4net updated to latest .NET Core 3.1.3 LTS version.
- Update of the Production Line templates for devonfw projects in devonfw shop floor.
- New merge feature included in the devonfw shop floor cicdgen tool.
- Updated sonar-devon4j-plugin:
  - Improved coloring and other visual cues to our rule descriptions to highlight good and bad code examples.
  - Improved the locations of issues thrown on method- and class-level.

Please check the detailed list below.

This would have not been possible without the commitment and hard work of the devonfw core team, German, Indian and ADCenter Valencia colleagues and collaborators as, among many others,

the Production Line team.

20.2. devonfw IDE

The consolidated list of features for this devonfw IDE release is as it follows.

20.2.1. 2020.04.001

Starting with this release we have changed the versioning schema in `devonfw` to `yyyy.mm.NNN` where `yyyy.mm` is the date of the planned milestone release and `NNN` is a running number increased with every bug- or security-fix update.

- #394 variable from devon.properties not set if not terminated with newline
- #399 launching of Intellij fails with No such file or directory error.
- #371 Eclipse plugin installation broke
- #390 maven get/set-version buggy
- #397 migration support for devon4j 2020.04.001
- #400 allow custom args for release

The full list of changes for this release can be found in milestone 2020.04.001.

20.2.2. 3.3.1

New release with bugfixes and new ide plugin feature:

- #343: Setup can't find Bash nor Git
- #369: Fix flattening of POMs
- #386: Feature/clone recursive
- #379: Use own extensions folder in devonfw-ide
- #381: Add ability to configure VS Code plugins via settings
- #376: Improve Eclipse plugin configuration
- #373: Fix project import on windows
- #374: Rework build on import

The full list of changes for this release can be found in milestone 3.3.1.

20.2.3. 3.3.0

New release with bugfixes and new project import feature:

- #343: Detect non-admin GitForWindows and Cygwin
- #175: Ability to clone projects and import into Eclipse automatically
- #346: devon eclipse add-plugin parameters swapped

- #363: devon ide update does not pull latest project settings
- #366: update java versions to latest fix releases

The full list of changes for this release can be found in milestone 3.3.0.

20.3. devon4j

The consolidated list of features for this devon4j release is as it follows.

20.3.1. 2020.04.001

Starting with this release we have changed the versioning schema in `devonfw` to `yyyy.mm.NNN` where `yyyy.mm` is the date of the planned milestone release and `NNN` is a running number increased with every bug- or security-fix update.

The following changes have been incorporated in devon4j:

- #233: Various version updates
- #241: Add module to support JWT and parts of OAuth
- #147: Switch from dozer to orika
- #180: Cleanup archtype
- #240: Add unreferenced guides
- #202: Architecture documentation needs update for components
- #145: Add a microservices article in the documentation
- #198: Deploy SNAPSHOTs to OSSRH in travis CI
- #90: Authorization of batches
- #221: Wrote monitoring guide
- #213: Document logging of custom field in json
- #138: Remove deprecated RevisionMetadata[Type]
- #211: Archetype: security config broken
- #109: LoginController not following devon4j to use JAX-RS but uses spring-webmvc instead
- #52: Improve configuration
- #39: Ability to log custom fields via SLF4J
- #204: Slf4j version
- #190: Rework of spring-batch integration
- #210: Rework documentation for blob support
- #191: Rework of devon4j-batch module
- #209: Include performance info in separate fields
- #207: Use more specific exception for not found entity

- #208: Remove unnecesary clone
- #116: Bug in JSON Mapping for ZonedDateTime
- #184: Fixed BOMs so devon4j and archetype can be used again
- #183: Error in executing the project created with devon4j
- #177: Switch to new maven-parent
- 169: Provide a reason, why unchecked exceptions are used in devon4j

Documentation is available at devon4j guide 2020.04.001. The full list of changes for this release can be found in milestone devon4j 2020.04.001.

20.4. devon4ng

The consolidated list of features for this devon4ng release is as it follows.

20.4.1. 2020.04.001

Starting with this release we have changed the versioning schema in `devonfw` to `yyyy.mm.NNN` where `yyyy.mm` is the date of the planned milestone release and `NNN` is a running number increased with every bug- or security-fix update.

- #111: Yarn 2 support included
- #96: devon4ng upgrade to Angular 9
  - Templates and samples updated to Angular 9, NgRx 9 and Ionic 5.
  - New internationalization module.
  - Documentation updates and improvements.
- #95: Added token management info in documentation

20.5. devon4net

The consolidated list of features for this devon4net release is as it follows:

- Updated to latest .NET Core 3.1.3 LTS version
- Dependency Injection Autoregistration for services and repositories
- Added multiple role managing claims in JWT
- Added custom headers to circuit breaker
- Reviewed default log configuration
- Added support to order query results from database via lambda expression
- Updated template and nuget packages

20.6. devon4node

The consolidated list of features for this devon4node release is as it follows:

- Upgrade to NestJS 7 (packages, samples and documentation)
- TSLint replaced with ESLint
- Add lerna to project to manage all the packages
- Add @devon4node/config package
- Add new schematics: Repository
- Improve WinstonLogger
- Improve documentation
- Update dependencies to latest versions

20.7. CobiGen

New release with updates and bugfixes:

- devonfw templates:
  - #1063: Upgrade devon4ng Ionic template to latest version
  - #1065: devon4ng templates for devon4node
  - #1128: update java templates for composite keys
  - #1130: Update template for devon4ng application template
  - #1131: Update template for devon4ng NgRx template
  - #1149: .NET templates
  - #1146: Dev ionic template update bug fix
- TypeScript plugin:
  - #1126: OpenApi parse/merge issues (ionic List templates)
- Eclipse plugin:
  - #412: Write UI Test for HealthCheck use
  - #867: Cobigen processbar
  - #1069: #953 dot path
  - #1099: NPE on HealthCheck
  - #1100: 1099 NPE on health check
  - #1101: #867 fix import of core and api
  - #1102: eclipse_plugin doesn’t accept folders as input
  - #1134: (Eclipse-Plugin) Resolve Template utility classes from core
  - #1142: #1102 accept all kinds of input

- CobiGen core:
  - #429: Reference external template files
  - #1143: Abort generation if external trigger does not match
  - #1125: Generation of templates from external increments does not work
  - #747: Variable assignment for external increments throws exception
  - #1133: Bugfix/1125 generation of templates from external increments does not work
  - #1127: #1119 added new TemplatesUtilsClassesUtil class to core
  - #953: NPE bug if foldername contains a dot
  - #1067: Feature/158 lat variables syntax
- CobiGen CLI:
  - #1111: Infinity loop in mmm-code (MavenDependencyCollector.collectWithReactor)
  - #1113: cobigen-cli does not seem to properly resolve classes from dependencies
  - #1120: Feature #1108 custom templates folder
  - #1115: Fixing CLI bugs related to dependencies and custom templates jar
  - #1108: CobiGen CLI: Allow easy use of user's templates
  - #1110: FileSystemNotFoundException blocking cobigen-cli
  - #1138: #1108 dev cli feature custom templates folder
  - #1136: (Cobigen-CLI) Resolve Template utility classes from core

20.8. devonfw-shop-floor

- Add documentation for deploy jenkins slaves
- Improve documentation
- Add devon4net Openshift template
- Add nginx docker image for devon4ng
- Add Openshift provisioning
- Production Line:
  - Updated MTS template: add step for dependency check and change the deployment method
  - Add template utils: initialize instance, openshift configuration, docker configuration and install sonar plugin
  - Add devon4net template
  - Add from existing template
  - Improve documentation
  - Refactor the documentation in order to follow the devonfw wiki workflow
  - Update devon4j, devon4ng, devon4net and devon4node in order to be able to choose the deployment method: none, docker or openshift.

    - Update the tools version in order to use the latest.
- Production Line Shared Lib
    - Add more fuctionality to the existing classes.
    - Add classes: DependencyCheckConfiguration, DockerConfiguration and OpenshiftConfiguration
- CICDGEN
    - Add devon4net support
    - Update tools versions in Jenkinsfiles to align with Production Line templates
    - Add merge strategies: error, keep, override, combine
    - Add lerna to the project
    - Minor improvements in the code
    - Add GitHub actions workflow to validate the new changes
    - Improve documentation
    - Breaking changes:
        - Remove the following parameters: plurl, ocurl
        - Add the following parameters: dockerurl, dockercertid, registryurl, ocname and merge

20.9. Sonar devon4j plugin

The consolidated list of features for this Sonar devon4j plugin release is as it follows.

20.9.1. 2020.04.001

This is the first version using our new versioning scheme. Here, the following issues were resolved:

- #60: Fixed a bug in the naming check for Use-Case implementation classes
- #67: Fixed a bug where the whole body of a method or a class was marked as the issue location. Now only the method / class headers will be highlighted.
- #68: Made our rule descriptions more accessible by using better readable colors as well as alternative visual cues
- #71: Fixed a bug where a NPE could be thrown
- #74: Fixed a bug where a method always returned null

Unrelated to any specific issues, there was some refactoring and cleaning up done with the following two PRs:

- PR #66: Refactored the prefixes of our rule names from 'Devon' to 'devonfw'
- PR #65: Sorted security-related test files into their own package

Changes for this release can be found in milestone 2020.04.001.

20.10. My Thai Star

As always, our reference application, My Thai Star, contains some interesting improvements that come from the new features and bug fixes from the other assets. The list is as it follows:

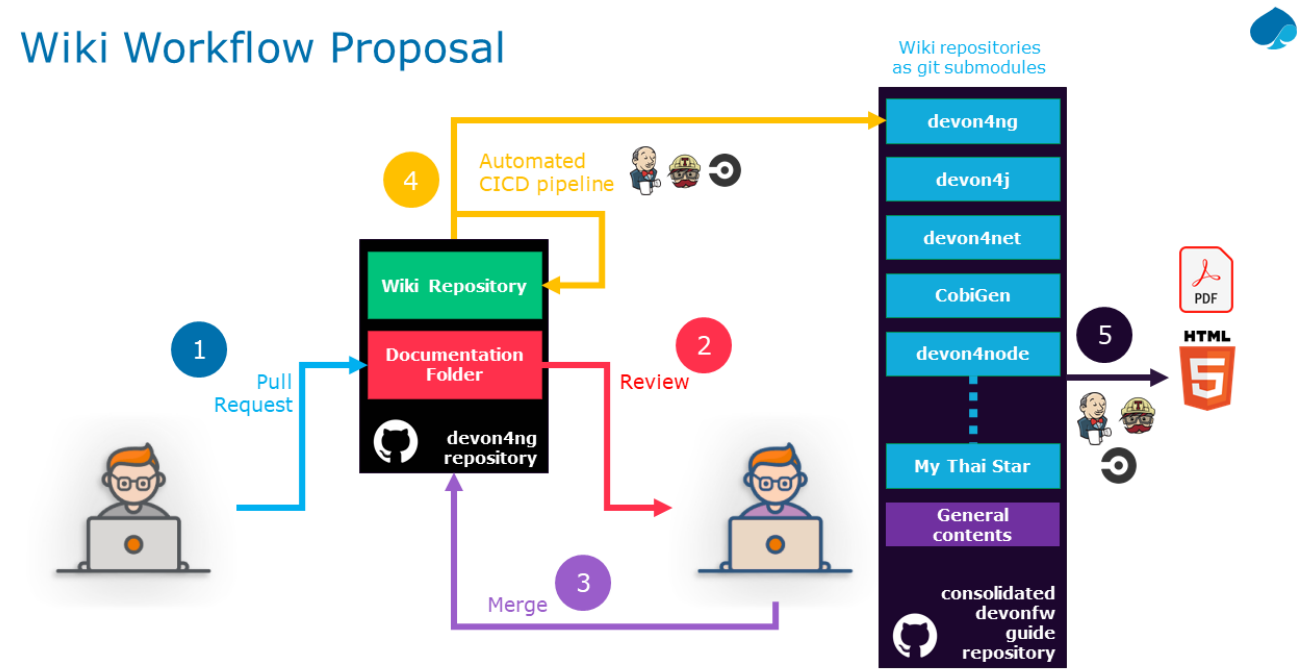
- devon4j - Java
  - Implement example batches with modified devon-batch
  - Upgrade spring boot version to 2.2.6 and devon4j 2020.004.001
  - Migrate from dozer to orika
- devon4ng - Angular
  - Move configuration to NgRx store
- devonfw shop floor - Jenkins
  - Update tools versions in order to align with Production Line templates
  - Add dependency check step (using dependency checker and yarn audit)
  - Send dependency checker reports to SonarQube
  - Changed deployment pipelines. Now pipelines are able to deploy docker containers using docker directly. No more ssh connections to execute commands in a remote machine are required.
  - Update documentation in order to reflect all changes
- devon4nde - Node.js
  - Upgrade to NestJS 7
  - Add custom repositories
  - Add exceptions and exception filters
  - Add tests (missing in the previous version)
  - Split logic into use cases in order to make the test process easier
  - Minor patches and improvemets
  - Documentation updated in order to reflect the new implementation

21. devonfw Release notes 3.2 “Homer”

21.1. Introduction

We are proud to announce the immediate release of devonfw version 3.2 (code named “Homer” during development). This version is the first one that contains the **new devonfw IDE** by default, so there is no need to download a huge ZIP with the whole distribution regardless of the use to which it will be put. The new devonfw IDE CLI will allow any user to setup a customized development environment completely configured with access to all the devonfw features, frameworks and tools. As we access to the official IDEs this is also the first version macOS compatible.

This release consolidates the documentation workflow adding the contents dynamically to the new devonfw website at the same time the PDF is generated. This have been achieved using a new GitHub action that takes the contents and builds the HTML files for the documentation section of the website. The documentation workflow proposed in the following picture is now complete:



This release also includes the **first version of devon4node**. We consider that node.js should be a first-class citizen inside the devonfw platform and for that reason we have included the latest development technologies for this ecosystem. The devon4node CLI, schematics and other tools will allow our users to create powerful node.js applications with the same philosophy you may find in the other languages and frameworks included. More information at its section below.

The new **devon4net 3.2.0 version** is also included in this release. Based on the .NET Core 3.0 and containing lots of new features gathered from important and recent projects, it represents a great improvement and an intermediate step to provide support for the incoming .NET Core 3.1 LTS. More information at its section below.

This release includes the final version of the **new CobiGen CLI** and completely integrated with the new devonfw IDE. Now using commands, you will be able to generate code the same way as you do with Eclipse. This means that you can use CobiGen on other IDEs like Visual Studio Code or IntelliJ.

Besides the Update command has been implemented. Now you will be able to update easily all your CobiGen plug-ins and templates inside the CLI.

On the other hand, the refactoring process has been completely developed, improving the mergers and including input readers for any other languages and frameworks, allowing the creation of models to generate code from them. Last, but not least, this new version includes the new templates for devon4net, devon4ng and devon4j generation.

And as always, **My Thai Star** has been updated to the latest versions of devon4j, devon4node and devon4net including completely State Management with NgRx in its devon4ng implementation upgrade.

This is the last release with the current semantic versioning number and without a fixed release calendar. From now on the new devonfw releases will happen in April, August and December and will be named **YYYY.MM.NN**, being the first release of the next year the 2020.04.00.

21.2. Changes and new features

21.2.1. devonfw-ide

We have entirely rewritten our automated solution for your local IDE (integrated desktop environment). The former oasp4j-ide and devonfw distributions with their extra-large gigabyte zip files are not entirely replaced with devonfw-ide. This new solution is provided as a small *.tar.gz file that is publicly available. It works on all platforms and has been tested on Windows, MacOS, and Linux. After extraction you only need to run a `setup` script. Here you provide a settings git URL for your customer project or simply hit return for testing or small projects. After reading and confirming the terms of use it will download all required tools in the proper versions for your operating system and configure them. Instead of various confusing scripts there is now only one CLI command `devon` for all use-cases what gives a much better user experience.

To get started go to the home page. There is even a migration-guide if you are currently used to the old approach and want to quickly jump into the new solution.

21.2.2. My Thai Star Sample Application

The new release of My Thai Star has focused on the following improvements:

- Release 3.2.0.
- devon4j:
  - devon4j 3.2.0 integrated.
  - Spring Boot 2.1.9 integrated.
  - SAP 4/HANA prediction use case.
  - Bug fixes.
- devon4ng:
  - SAP 4/HANA prediction use case.

    - 2FA toggleable (two factor authentication).
    - NgRx full integrated (PR #285).
- devon4net
    - devon4net for dotnet core 3.0 updated
    - Updated the API contract compatible with the other stacks
    - JWT implementation reviewed to increase security
    - ASP.NET user database dependencies removed
    - HTTP2 support
    - Clearer CRUD pattern implementation
- devon4node
    - TypeScript 3.6.3.
    - Based on Nest framework.
    - Configuration Module
    - Added cors and security headers
    - Added mailer module and email templates.
    - Built in winston logger
    - Custom ClassSerializerInterceptor
- MrChecker
    - Example cases for end-to-end test.
    - Production line configuration.
- CICD
    - Improved integration with Production Line
    - New Traefik load balancer and reverse proxy
    - New deployment from artifact
    - New CICD pipelines
    - New deployment pipelines
    - Automated creation of pipelines in Jenkins

21.2.3. Documentation updates

This release addresses the new documentation workflow, being now possible to keep the documentation synced with any change. The new documentation includes the following contents:

- Getting started
- devonfw ide
- devon4j documentation
- devon4ng documentation

- devon4net documentation
- devon4node documentation
- CobiGen documentation
- devonfw-shop-floor documentation
- cicdgen documentation
- devonfw testing with MrChecker
- My Thai Star documentation
- Contribution guide
- Release notes

21.2.4. devon4j

The following changes have been incorporated in devon4j:

- Completed full support from Java8 to Java11
- Several security fixes
- Upgrade to Spring Boot 2.1.9
- Upgrade to Spring 5.1.8
- Upgrade to JUnit 5 (requires migration via devonfw-ide)
- Improved JPA support for IdRef
- Improved auditing metadata support
- Many improvements to documentation (added JDK guide, architecture-mapping, JMS, etc.)
- For all details see milestone.

21.2.5. devon4ng

The following changes have been incorporated in devon4ng:

- Angular CLI 8.3.1,
- Angular 8.2.11,
- Angular Material 8.2.3,
- Ionic 4.11.1,
- Capacitor 1.2.1 as Cordova replacement,
- NgRx 8.3 support for State Management,
- devon4ng Angular application template updated to Angular 8.2.11 with visual improvements and bugfixes https://github.com/devonfw/devon4ng-application-template
- devon4ng Ionic application template updated to 4.11.1 and improved https://github.com/devonfw/devon4ng-ionic-application-template
- Improved devon4ng Angular application template with state management using Angular 8 and

NgRx 8 https://github.com/devonfw/devon4ng-ngrx-template

- Documentation and samples updated to latest versions:
  - Web Components with Angular Elements
  - Initial configuration with App Initializer pattern
  - Error Handling
  - PWA with Angular and Ionic
  - Lazy Loading
  - Library construction
  - Layout with Angular Material
  - Theming with Angular Material

21.2.6. devon4net

The following changes have been incorporated in devon4net:

- Updated to latest .net core 3.0 version
- Template
  - Global configuration automated. devon4net can be instantiated on any .net core application template with no effort
  - Added support for HTTP2
  - Number of libraries minimized
  - Architecture layer review. More clear and scalable
  - Added red button functionality (aka killswitch) to stop attending API request with custom error
  - Improved API error management
  - Added support to only accept request from clients with a specific client certificate on Kestrel server. Special thanks to Bart Roozendaal (Capgemini NL)
  - All components use IOptions pattern to be set up properly
  - Swagger generation compatible with OpenAPI v3
- Modules
  - The devon4net netstandard libraries have been updated to netstandard 2.1
  - JWT:
    - Added token encryption (token cannot be decrypted anymore by external parties). Now You can choose the encryption algorithm depending on your needs
    - Added support for secret key or certificate encryption
    - Added authorization for swagger portal
  - Circuit breaker
    - Added support to bypass certificate validation

        - Added support to use a certificate for https communications using Microsoft's httpclient factory
    - Unit of Work
        - Repository classes unified and reviewed for increasing performance and reduce the consumed memory
        - Added support for different database servers: In memory, Cosmos, MySQL + MariaDB, Firebird, PostgreSQL, Oracle, SQLite, Access, MS Local.

21.2.7. devon4node

The following changes have been incorporated in devon4node:

- TypeScript 3.6.3.
- Based on Nest framework.
- Complete backend implementation.
- New devon4node CLI. It will provide you some commands
    - new: create a new devon4node interactively
    - generate: generate code based on schematics
    - db: manage the database
- New devon4node schematics
    - application: create a new devon4node application
    - config-module: add a configuration module to the project
    - mailer: install and configure the devon4node mailer module
    - typeorm: install TypeORM in the project
    - auth-jwt: add users and auth-jwt modules to the project
    - swagger: expose an endpoint with the auto-generated swagger
    - security: add cors and other security headers to the project.
    - crud: create all CRUD for an entity
    - entity: create an entity
- New mailer module
- New common library
- Build in winston logger
- Custom ClassSerializerInterceptor
- Extendable base entity
- New application samples

21.2.8. CobiGen

- CobiGen core new features:
  - CobiGen CLI: Update command implemented. Now you will be able to update easily all your CobiGen plug-ins and templates inside the CLI. Please take a look into the documentation for more info.
    - CobiGen CLI is now JDK11 compatible.
    - CobiGen CLI commandlet for devonfw-ide has been added. You can use it to setup easily your CLI and to run CobiGen related commands.
    - Added a version provider so that you will be able to know all the CobiGen plug-ins versions.
    - Added a process bar when the CLI is downloading the CobiGen plug-ins.
  - CobiGen refactoring finished: With this refactoring we have been able to decouple CobiGen completely from the target and input language. This facilitates the creation of parsers and mergers for any language. For more information please take a look here.
    - New TypeScript input reader: We are now able to parse any TypeScript class and generate code using the parsed information. We currently use TypeORM entities as a base for generation.
  - Improving CobiGen templates:
    - Updated devon4ng-NgRx templates to NgRx 8.
    - Generation of an Angular client using as input a TypeORM entity. This is possible thanks to the new TypeScript input reader.
    - .Net templates have been upgraded to .Net Core 3.0
  - CobiGen for Eclipse is now JDK11 compatible.
  - Fixed bugs when adapting templates and other bugs on the CobiGen core.

21.2.9. devonfw shop floor

- Added devon4ng OpenShift templates
- Added devon4j OpenShift templates
- Added devon4node OpenShift templates
- Added more methods to link https://github.com/devonfw-forge/devon-production-line-shared-lib [devonfw Production Line shared library]
- Updated link: devonfw Production Line templates

cicdgen

- Patched minor bugs

21.2.10. sonar-devon4j-plugin

sonar-devon4j-plugin is a SonarQube plugin for architecture governance of devon4j applications. It

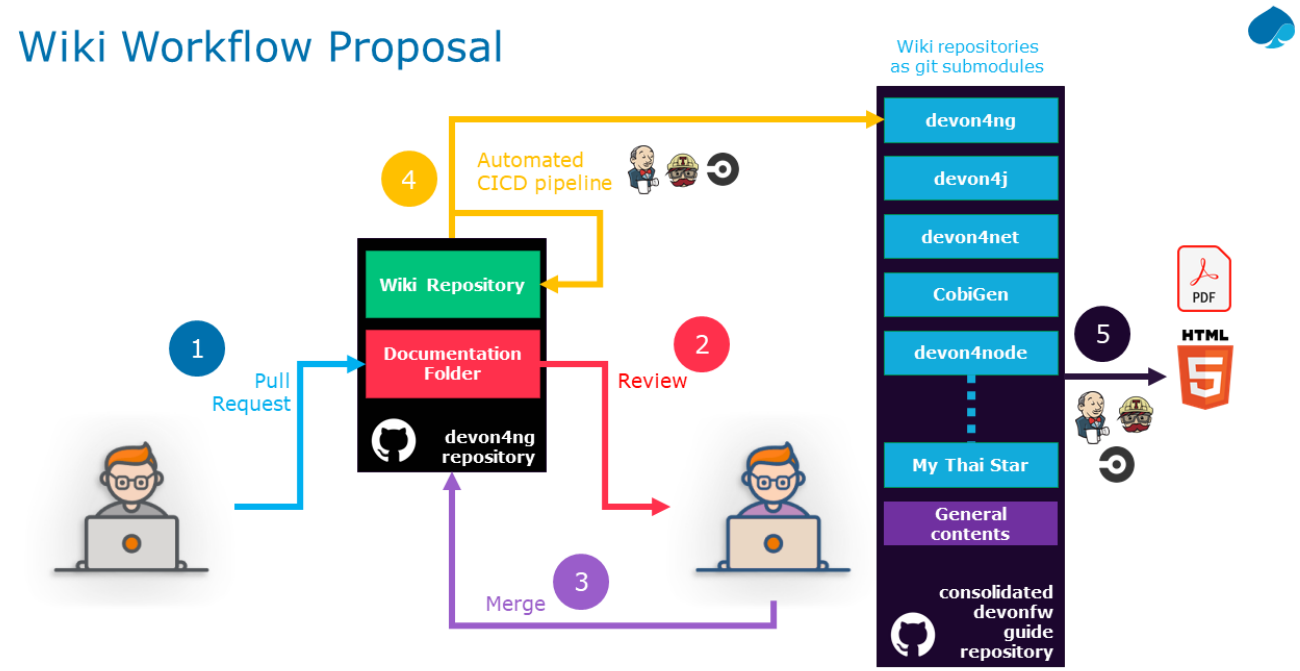
verifies the architecture and conventions of devon4j, the Java stack of devonfw. The following changes have been incorporated: * Plugin was renamed from sonar-devon-plugin to sonar-devon4j-plugin * Rules/checks have been added to verify naming conventions * New rule for proper JPA datatype mapping * Proper tagging of rules as architecture-violation and not as bug, etc. * Several improvements have been made to prepare the plugin to enter the SonarQube marketplace, what will happen with the very next release. * Details can be found here: https://github.com/devonfw/sonar-devon4j-plugin/milestone/2?closed=1

22. devonfw Release notes 3.1 “Goku”

22.1. Introduction

We are proud to announce the immediate release of devonfw version 3.1 (code named “Goku” during development). This version is the first one that implements our new documentation workflow, that will allow users to get the updated documentation at any moment and not to wait for the next devonfw release.

This is now possible as we have established a new workflow and rules during development of our assets. The idea behind this is that all the repositories contain a `documentation` folder and, in any pull request, the developer must include the related documentation change. A new Travis CI configuration added to all these repositories will automatically take the changes and publish them in the wiki section of every repository and in the new devonfw-guide repository that consolidates all the changes from all the repositories. Another pipeline will take changes from this consolidated repository and generate dynamically the devonfw guide in PDF and in the next weeks in HTML for the new planned devonfw website. The following schema explains this process:



This release includes the very first version of the new CobiGen CLI. Now using commands, you will be able to generate code the same way as you do with Eclipse. This means that you can use CobiGen on other IDEs like Visual Studio Code or IntelliJ. Please take a look at https://github.com/devonfw/cobigen/wiki/howto_Cobigen-CLI-generation for more info.

The devonfw-shop-floor project has got a lot of updates in order to make even easier the creation of devonfw projects with CICD pipelines that run on the Production Line, deploy on Red Hat OpenShift Clusters and in general Docker environments. See the details below.

This release includes the very first version of our devonfw-ide tool that will allow users to automate devonfw setup and update the development environment. This tool will become the default devonfw setup tool in future releases. For more information please visit the repository

https://github.com/devonfw/devon-ide.

Following the same collaboration model we used in order to improve the integration of devonfw with Red Hat OpenShift and which allowed us to get the Red Hat Open Shift Primed certification, we have been working alongside with SAP HANA developers in order to support this database in the devon4j. This model was based on the contribution and review of pull requests in our reference application My Thai Star. In this case, SAP developers collaborated with us in the following two new use cases:

- Prediction of future demand
- Geospatial analysis and clustering of customers

More info at https://blogs.sap.com/2019/06/17/introducing-devonfw-support-for-sap-hana/.

Last but not least the devonfw extension pack for VS Code has been improved with the latest extensions and helpers for this IDE. Among many others you can now use:

- Remote development on Docker containers and VMs https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.vscode-remote-extensionpack
- Dependency Analysis for maven and npm https://marketplace.visualstudio.com/items?itemName=redhat.fabric8-analytics
- React Native Tools https://marketplace.visualstudio.com/items?itemName=msjsdiag.vscode-react-native
- NgRx Snippets https://marketplace.visualstudio.com/itemdetails?itemName=hardikpthv.NgRxSnippets

Also it is worth the try of the updated support for Java and Spring Boot development in VS Code. Check it out for yourself!

More information at https://marketplace.visualstudio.com/items?itemName=devonfw.devonfw-extension-pack. Also, you can contribute to this extension in this GitHub repository https://github.com/devonfw/devonfw-extension-pack-vscode.

22.2. Changes and new features

22.2.1. Devonfw dist

- Eclipse 2018.12 integrated
  - CheckStyle Plugin updated.
  - SonarLint Plugin updated.
  - Git Plugin updated.
  - FindBugs Plugin updated.
  - CobiGen plugin updated.
- Other Software
  - Visual Studio Code latest version included and pre-configured with the devonfw Platform

Extension Pack.
    - Ant updated to latest.
    - Maven updated to latest.
    - Java updated to latest.
    - Nodejs LTS updated to latest.
    - @angular/cli included.
    - @devonfw/cicdgen included.
    - Yarn package manager updated.
    - Python3 updated.
    - Spyder3 IDE integrated in python3 installation updated.
    - devon4ng-application-template for Angular 8 at workspaces/examples
    - devon4ng-ionic-application-template for Ionic 4 at workspace/samples

22.2.2. My Thai Star Sample Application

The new release of My Thai Star has focused on the following improvements:

- Release 3.1.0.
- devon4j:
    - devon4j 3.1.0 integrated.
    - Spring Boot 2.1.6 integrated.
    - SAP 4/HANA prediction use case.
    - Bug fixes.
- devon4ng:
    - SAP 4/HANA prediction use case.
    - 2FA toggleable (two factor authentication).
    - NgRx integration in process (PR #234).
- devon4node
    - TypeScript 3.1.3.
    - Based on Nest framework.
    - Aligned with devon4j.
    - Complete backend implementation.
    - TypeORM integrated with SQLite database configuration.
    - Webpack bundler.
    - Nodemon runner.
    - Jest unit tests.
- Mr.Checker

- Example cases for end-to-end test.
- Production line configuration.
- CICD
- Improved integration with Production Line
- New Traefik load balancer and reverse proxy
- New deployment from artifact
- New CICD pipelines
- New deployment pipelines
- Automated creation of pipelines in Jenkins

22.2.3. Documentation updates

This release addresses the new documentation workflow, being now possible to keep the documentation synced with any change. The new documentation includes the following contents:

- Getting started
- Contribution guide
- Devcon
- Release notes
- devon4j documentation
- devon4ng documentation
- devon4net documentation
- devonfw-shop-floor documentation
- cicdgen documentation
- devonfw testing with MrChecker
- My Thai Star documentation

22.2.4. devon4j

The following changes have been incorporated in devon4j:

- Added Support for Java8 up to Java11
- Upgrade to Spring Boot 2.1.6.
- Upgrade to Spring 5.1.8
- Upgrade to JPA 2.2
- Upgrade to Hibernate 5.3
- Upgrade to Dozer 6.4.1 (ATTENTION: Requires Migration, use devon-ide for automatic upgrade)
- Many improvements to documentation (added JDK guide, architecture-mapping, JMS, etc.)
- Completed support (JSON, Beanmapping) for pagination, IdRef, and java.time

- Added MasterCto
- For all details see milestone.

22.2.5. devon4ng

The following changes have been incorporated in devon4ng:

- Angular CLI 8,
- Angular 8,
- Angular Material 8,
- Ionic 4,
- Capacitor 1.0 as Cordova replacement,
- NgRx 8 support for State Management,
- devon4ng Angular application template updated to Angular 8 with visual improvements and bugfixes https://github.com/devonfw/devon4ng-application-template
- devon4ng Ionic application template updated and improved https://github.com/devonfw/devon4ng-ionic-application-template
- New devon4ng Angular application template with state management using Angular 8 and NgRx 8 https://github.com/devonfw/devon4ng-ngrx-template
- New devon4ng library https://github.com/devonfw/devon4ng-library that includes the following libraries:
  - Cache Module for Angular 7+ projects.
  - Authorization Module for Angular 7+ projects.
- New use cases with documentation and samples:
  - Web Components with Angular Elements
  - Initial configuration with App Initializer pattern
  - Error Handling
  - PWA with Angular and Ionic
  - Lazy Loading
  - Library construction
  - Layout with Angular Material
  - Theming with Angular Material

22.2.6. devon4net

The following changes have been incorporated in devon4net:

- New circuit breaker component to communicate microservices via HTTP
- Resolved the update packages issue

22.2.7. AppSec Quick Solution Guide

This release incorporates a new Solution Guide for Application Security based on the state of the art in OWASP based application security. The purpose of this guide is to offer quick solutions for common application security issues for all applications based on devonfw. It's often the case that we need our systems to comply to certain sets of security requirements and standards. Each of these requirements needs to be understood, addressed and converted to code or project activity. We want this guide to prevent the wheel from being reinvented over and over again and to give clear hints and solutions to common security problems.

- The wiki can be accessed here: https://github.com/devonfw/devonfw-security/wiki
- The PDF can be accessed here: https://github.com/devonfw/devonfw-security

22.2.8. CobiGen

- CobiGen core new features:
  - CobiGen CLI: New command line interface for CobiGen. Using commands, you will be able to generate code the same way as you do with Eclipse. This means that you can use CobiGen on other IDEs like Visual Studio Code or IntelliJ. Please take a look into the documentation for more info.
    - Performance improves greatly in the CLI thanks to the lack of GUI.
    - You will be able to use path globs for selecting multiple input files.
    - We have implemented a search functionality so that you can easily search for increments or templates.
  - First steps taken on CobiGen refactoring: With the new refactoring we will be able to decouple CobiGen completely from the target and input language. This will facilitate the creation of parsers and mergers for any language.
    - NashornJS has been deprecated: It was used for executing JavaScript code inside JVM. With the refactoring, performance has improved on the TypeScript merger.
  - Improving CobiGen templates:
    - Removed Covalent from Angular templates as it is not compatible with Angular 8.
    - Added devon4ng-NgRx templates that implement reactive state management. Note: The TypeScript merger is currently being improved in order to accept NgRx. The current templates are set as overridable by default.
    - Test data builder templates now make use of Lambdas and Consumers.
    - CTOs and ETOs increments have been correctly separated.
  - TypeScript merger has been improved: Now it is possible to merge comments (like tsdoc) and enums.
  - OpenAPI parsing extended to read enums. Also fixed some bugs when no properties were set or when URLs were too short.
  - Java static and object initializers now get merged.
  - Fixed bugs when downloading and adapting templates.

22.2.9. Devcon

A new version of Devcon has been released. Fixes and new features include:

- Updated to match current devon4j
- Update to download Linux distribution.
- Custom modules creation improvements.
- Code Migration feature added.
- Bugfixes.

22.2.10. Devonfw OSS Modules

Modules upgraded to be used in new devon4j projects:

- Reporting module
- WinAuth AD Module
- WinAuth SSO Module
- I18n Module
- Async Module
- Integration Module
- Microservice Module
- Compose for Redis Module See: https://github.com/devonfw/devon/wiki#devonfw-modules

22.2.11. devonfw shop floor

- Industrialization oriented to configure the provisioning environment provided by Production Line and deploy applications on an OpenShift cluster.
- Added Jenkinsfiles to configure automatically OpenShift environments to deploy devonfw applications.
- Industrialization to start new projects and configure them with CICD.
- Upgrade the documentation with getting started guide to configure CICD in any devonfw project and deploy it.
- Added new tool cicdgen to generate CICD code/files.

cicdgen

cicdgen is a devonfw tool to generate all code/files related to CICD in your project. It's based on angular schematics and it has its own CLI. More information here.

- CICD configuration for devon4j, devon4ng and devon4node projects
- Option to deploy devonfw projects with Docker
- Option to deploy devonfw projects with OpenShift

22.2.12. Devonfw Testing

Mr.Checker

The Mr.Checker Test Framework is an automated testing framework for functional testing of web applications, API web services, Service Virtualization, Security and in coming future native mobile apps, and databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions. Mr.Checker updates and improvements:

- Examples available under embedded project “MrChecker-App-Under-Test” and in project wiki: https://github.com/devonfw/devonfw-testing/wiki
- How to install:
  - Wiki : https://github.com/devonfw/devonfw-testing/wiki/How-to-install
- Release Note:
  - module selenium - 3.8.2.1:
    - possibility to define version of driver in properties.file
    - automatic driver download if the version is not specified
    - possibility to run with different browser options
    - module webAPI – 1.2.1:
    - possibility to connect to the remote WireMock server

23. devonfw Release notes 3.0 “Fry”

23.1. Introduction

We are proud to announce the immediate release of devonfw version 3.0 (code named “Fry” during development). This version is the consolidation of Open Source, focused on the major namespace change ever in the platform, removing the OASP references and adopting the new devonfw names for each technical stack or framework.

The new stack names are the following:

- devon4j, former OASP4J, is the new name for Java.
- devon4ng, former OASP4JS, is the new one for Angular.
- devon4net, is the new .NET stack.
- devon4X, is the new stack for Xamarin development.
- devon4node, is the new devonfw incubator for node.js.

The new devon4j version was created directly from the latest oasp4j version (3.0.0). Hence it brings all the features and values that oasp4j offered. However, the namespace migration was used to do some housekeeping and remove deprecated code as well as reduce dependencies. Therefore your data-access layer will no longer have to depend on any third party except for devon4j as well as of course the JPA. We also have improved the application template that now comes with a modern JSON logging ready for docker and logstash based environments.

To help you upgrading we introduced a migration feature in devcon. This can automatically migrate your code from oasp4j (even older versions starting from 2.4.0) to the latest version of devon4j. There might be some small manual changes left to do but 90% of the migration will be done automatically for you.

Besides, the first version of the devonfw plugin for SonarQube has been released. It extends SonarQube with the ability to validate your code according to the devon4j architecture. More details at https://github.com/devonfw/sonar-devon-plugin.

This is the first release that integrates the new devonfw .NET framework, called devon4net, and Xamarin for mobile native development, devon4X. devon4NET and devon4X are the Capgemini standard frameworks for .NET and Xamarin software development. With the two new family members devonfw provides guidance and acceleration for the major software development platforms in our industry. Their interoperability provides you the assurance your multichannel solution will be consistent across web and mobile channels.

“Fry” release contains lots of improvements in our Mr.Checker E2E Testing Framework, including a complete E2E sample inside our reference application My Thai Star. Besides Mr.Checker, we include as an incubator Testar, a test tool (and framework) to test applications at the GUI level whose objective is to solve part of the maintenance problem affecting tests by automatically generating test cases based on a structure that is automatically derived from the GUI. Testar is not included to replace Mr.Checker but rather to provide development teams with a series of interesting options which go beyond what Mr.Checker already provides.

Apart from Mr.Checker, engagements can now use Testar as an extra option for testing. This is a tool that enables the automated system testing of desktop, web and mobile applications at the GUI level. Testar has been added as an incubator to the platform awaiting further development during 2019.

The new incubator for node.js, called devon4node, has been included and implemented in several internal projects. This incubator is based on the Nest framework https://www.nestjs.com/. Nest is a framework for building efficient, scalable Node.js server-side applications. It uses progressive JavaScript, is built with TypeScript (preserves compatibility with pure JavaScript) and combines elements of OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming). Under the hood, Nest makes use of Express, but also provides compatibility with a wide range of other libraries (e.g. Fastify). This allows for easy use of the myriad third-party plugins which are available.

In order to facilitate the utilization of Microsoft Visual Studio Code in devonfw, we have developed and included the new devonfw Platform Extension Pack with lots of features to develop and test applications with this IDE in languages and frameworks such as TypeScript, JavaScript, .NET, Java, Rust, C++ and many more. More information at https://marketplace.visualstudio.com/items?itemName=devonfw.devonfw-extension-pack. Also, you can contribute to this extension in this GitHub repository https://github.com/devonfw/devonfw-extension-pack-vscode.

There is a whole range of new features and improvements which can be seen in that light. The My Thai Star sample app has now been upgraded to devon4j and devon4ng, a new devon4node backend implementation has been included that is seamless interchangeable, an E2E MrChecker sample project, CICD and deployment scripts and lots of bugs have been fixed.

Last but not least, the projects wikis and the devonfw Guide has once again been updated accordingly before the big refactor that will be addressed in the following release in 2019.

23.2. Changes and new features

23.2.1. Devonfw dist

- Eclipse 2018.9 integrated
  - CheckStyle Plugin updated.
  - SonarLint Plugin updated.
  - Git Plugin updated.
  - FindBugs Plugin updated.
  - CobiGen plugin updated.
- Other Software
  - Visual Studio Code latest version included and pre-configured with the devonfw Platform Extension Pack.
  - Ant updated to latest.
  - Maven updated to latest.

- Java updated to latest.
- Nodejs LTS updated to latest.
- @angular/cli included.
- Yarn package manager updated.
- Python3 updated.
- Spyder3 IDE integrated in python3 installation updated.
- devon4ng-application-template for Angular 7 at workspaces/examples
- devon4ng-ionic-application-template for Ionic 3.20 at workspace/samples

23.2.2. My Thai Star Sample Application

The new release of My Thai Star has focused on the following improvements:

- Release 1.12.2.
- devon4j:
  - devon4j 3.0.0 integrated.
  - Spring Boot 2.0.4 integrated.
  - Spring Data integration.
  - New pagination and search system.
  - Bug fixes.
- devon4ng:
  - Client devon4ng updated to Angular 7.
  - Angular Material and Covalent UI frameworks updated.
  - Electron framework integrated.
- devon4node
  - TypeScript 3.1.3.
  - Based on Nest framework.
  - Aligned with devon4j.
  - Complete backend implementation.
  - TypeORM integrated with SQLite database configuration.
  - Webpack bundler.
  - Nodemon runner.
  - Jest unit tests.
- Mr.Checker
  - Example cases for end-to-end test.
  - Production line configuration.
  - CICD

- Improved integration with Production Line
- New deployment from artifact
- New CICD pipelines
- New deployment pipelines
- Automated creation of pipelines in Jenkins

23.2.3. Documentation updates

The following contents in the devonfw guide have been updated:

- Upgrade of all the new devonfw named assets.
  - devon4j
  - devon4ng
  - Mr.Checker
- Electron integration cookbook.
- Updated cookbook about Swagger.
- Removed deprecated entries.

Apart from this the documentation has been reviewed and some typos and errors have been fixed.

The current development of the guide has been moved to https://github.com/devonfw-forge/devon-guide/wiki in order to be available as the rest of OSS assets.

23.2.4. devon4j

The following changes have been incorporated in devon4j:

- Spring Boot 2.0.4 Integrated.
- Spring Data layer Integrated.
- Decouple mmm.util.*
- Removed depreciated restaurant sample.
- Updated Pagination support for Spring Data
- Add support for hana as dbType.
- Bugfixes.

23.2.5. devon4ng

The following changes have been incorporated in devon4ng:

- New client application architecture guide https://github.com/devonfw/devon4ng/wiki
- Angular CLI 7,
- Angular 7,

- Angular Material 7 and Covalent 2.0.0-beta.7,
- Ionic 3.20.0,
- Cordova 8.0.0,
- devon4ng Angular application template updated to Angular 7 with visual improvements and bugfixes https://github.com/devonfw/devon4ng-application-template
- devon4ng Ionic application template updated and improved https://github.com/devonfw/devon4ng-ionic-application-template
- PWA enabled.
- Electron integrated to run My Thai Star as a desktop application in Windows, Linux or macOS.

23.2.6. devon4net

Some of the highlights of devon4net 1.0 are:

- External configuration file for each environment.
- .NET Core 2.1.X working solution (Latest 2.1.402).
- Packages and solution templates published on nuget.org.
- Full components customization by config file.
- Docker ready (My Thai Star sample fully working on docker).
- Port specification by configuration.
- Dependency injection by Microsoft .NET Core.
- Automapper support.
- Entity framework ORM (Unit of work, async methods).
- .NET Standard library 2.0 ready.
- Multi-platform support: Windows, Linux, Mac.
- Samples: My Thai Star back-end, Google API integration, Azure login, AOP with Castle.
- Documentation site.
- SPA page support.

And included the following features:

- Logging:
  - Text File.
  - Sqlite database support.
  - Serilog Seq Server support.
  - Graylog integration ready through TCP/UDP/HTTP protocols.
  - API Call params interception (simple and compose objects).
  - API error exception management.
- Swagger:

    - Swagger auto generating client from comments and annotations on controller classes.
    - Full swagger client customization (Version, Title, Description, Terms, License, Json endpoint definition).
- JWT:
    - Issuer, audience, token expiration customization by external file configuration.
    - Token generation via certificate.
    - MVC inherited classes to access JWT user properties.
    - API method security access based on JWT Claims.
- CORS:
    - Simple CORS definition ready.
    - Multiple CORS domain origin definition with specific headers and verbs.
- Headers:
    - Automatic header injection with middleware.
    - Supported header definitions: AccessControlExposeHeader, StrictTransportSecurityHeader, XFrameOptionsHeader, XssProtectionHeader, XContentTypeOptionsHeader, ContentSecurityPolicyHeader, PermittedCrossDomainPoliciesHeader, ReferrerPolicyHeader.
- Reporting server:
    - Partial implementation of reporting server based on My-FyiReporting (now runs on linux container).
- Testing:
    - Integration test template with sqlite support.
    - Unit test template.
    - Moq, xunit frameworks integrated.

23.2.7. devon4X

Some of the highlights of the new devonfw Xamarin framework are:

- Based on Excalibur framework by Hans Harts (https://github.com/Xciles/Excalibur).
- Updated to latest MVVMCross 6 version.
- My Thai Star Excalibur forms sample.
- Xamarin Forms template available on nuget.org.

23.2.8. AppSec Quick Solution Guide

This release incorporates a new Solution Guide for Application Security based on the state of the art in OWASP based application security. The purpose of this guide is to offer quick solutions for common application security issues for all applications based on devonfw. It's often the case that we need our systems to comply to certain sets of security requirements and standards. Each of these requirements needs to be understood, addressed and converted to code or project activity. We

want this guide to prevent the wheel from being reinvented over and over again and to give clear hints and solutions to common security problems.

- The wiki can be accessed here: https://github.com/devonfw/devonfw-security/wiki
- The PDF can be accessed here: https://github.com/devonfw/devonfw-security

23.2.9. CobiGen

- CobiGen core new features:
  - CobiGen_Templates will not need to be imported into the workspace anymore. However, If you want to adapt them, you can still click on a button that automatically imports them for you.
  - CobiGen_Templates can be updated by one-click whenever the user wants to have the latest version.
  - Added the possibility to reference external increments on configuration level. This is used for reducing the number of duplicated templates.
- CobiGen_Templates project and docs updated:
  - Spring standard has been followed better than ever.
  - Interface templates get automatically relocated to the api project. Needed for following the new devon4j standard.
- CobiGen Angular:
  - Angular 7 generation improved based on the updated application template.
  - Pagination changed to fit Spring standard.
- CobiGen Ionic: Pagination changed to fit Spring standard.
- CobiGen OpenAPI plugin released with multiple bug-fixes and other functionalities like:
  - Response and parameter types are parsed properly when they are a reference to an entity.
  - Parameters defined on the body of a request are being read correctly.

23.2.10. Devcon

A new version of Devcon has been released. Fixes and new features include:

- Updated to match current devon4j
- Update to download Linux distribution.
- Custom modules creation improvements.
- Code Migration feature added
- Bugfixes.

23.2.11. Devonfw OSS Modules

Modules upgraded to be used in new devon4j projects:

- Reporting module
- WinAuth AD Module
- WinAuth SSO Module
- I18n Module
- Async Module
- Integration Module
- Microservice Module
- Compose for Redis Module

See: https://github.com/devonfw/devon/wiki#devonfw-modules

23.2.12. Devonfw Testing

Mr.Checker

The Mr.Checker Test Framework is an automated testing framework for functional testing of web applications, API web services, Service Virtualization, Security and in coming future native mobile apps, and databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions. Mr.Checker updates and improvements:

- Examples available under embedded project “MrChecker-App-Under-Test” and in project wiki: https://github.com/devonfw/devonfw-testing/wiki
- How to install:
  - Wiki : https://github.com/devonfw/devonfw-testing/wiki/How-to-install
- Release Note:
  - module selenium - 3.8.1.13:
    - headless browser
    - enable browser options
  - module DevOps :
    - Jenkinsfile align with ProductionLine

Testar

We have added Test*, Testar, as an incubator to the available test tools within devonfw. This ground-breaking tool is being developed by the Technical University of Valencia (UPV). In 2019 Capgemini will co-develop Testar with the UPV.

Testar is a tool that enables the automated system testing of desktop, web and mobile applications at the GUI level.

With Testar, you can start testing immediately. It automatically generates and executes test sequences based on a structure that is automatically derived from the UI through the accessibility API. Testar can detect the violation of general-purpose system requirements and you can use

plugins to customize your tests.

You do not need test scripts and maintenance of it. The tests are random and are generated and executed automatically.

If you need to do directed tests you can create scripts to test specific requirements of your application.

Testar is included in the devonfw distro or can be downloaded from https://testar.org/download/.

The Github repository can be found at o: https://github.com/TESTARtool/TESTAR.

24. devonfw Release notes 2.4 “EVE”

24.1. Introduction

We are proud to announce the immediate release of devonfw version 2.4 (code named “EVE” during development). This version is the first one that fully embraces Open Source, including components like the documentation assets and CobiGen. Most of the IP (Intellectual Property or proprietary) part of devonfw are now published under the Apache License version 2.0 (with the documentation under the Creative Commons License (Attribution-NoDerivatives)). This includes the GitHub repositories where all the code and documentation is located. All of these repositories are now open for public viewing as well.

“EVE” contains a slew of new features but in essence it is already driven by what we expect to be the core focus of 2018: strengthening the platform and improving quality.

This release is also fully focused on deepening the platform rather than expanding it. That is to say: we have worked on improving existing features rather than adding new ones and strengthen the qualitative aspects of the software development life cycle, i.e. security, testing, infrastructure (CI, provisioning) etc.

“EVE” already is very much an example of this. This release contains the Allure Test Framework (included as an incubator in version 2.3) update called MrChecker Test Framework. MrChecker is an automated testing framework for functional testing of web applications, API web services, Service Virtualization, Security and in coming future native mobile apps, and databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions.

Another incubator being updated is the devonfw Shop Floor which intended to be a compilation of DevOps experiences from the devonfw perspective. A new part of the release is the new Solution Guide for Application Security based on the state of the art in OWASP based application security.

There is a whole range of new features and improvements which can be seen in that light. OASP4j 2.6 changes and improves the package structure of the core Java framework. The My Thai Star sample app has now been upgraded to Angular 6, lots of bugs have been fixed and the devonfw Guide has once again been improved.

Last but not least, this release contains the formal publication of the devonfw Methodology or The Accelerated Solution Design - an Industry Standards based solution design and specification (documentation) methodology for Agile (and less-than-agile) projects.

24.2. Changes and new features

24.2.1. devonfw 2.4 is Open Source

This version is the first release of devonfw that fully embraces Open Source, including components like the documentation assets and CobiGen. This is done in response to intensive market pressure and demands from the MU´s (Public Sector France, Netherlands)

Most of the IP (Intellectual Property or proprietary) part of devonfw are now published under the Apache License version 2.0 (with the documentation under the Creative Commons License (Attribution-NoDerivatives)).

So you can now use the devonfw distribution (the "zip" file), CobiGen, the devonfw modules and all other components without any worry to expose the client unwittingly to Capgemini IP.

Note: there are still some components which are IP and are not published under an OSS license. The class room trainings, the Sencha components and some CobiGen templates. But these are not includes in the distribution nor documentation and are now completely maintained separately.

24.2.2. devonfw dist

- Eclipse Oxygen integrated
  - CheckStyle Plugin updated.
  - SonarLint Plugin updated.
  - Git Plugin updated.
  - FindBugs Plugin updated.
  - CobiGen plugin updated.
- Other Software
  - Visual Studio Code latest version included and pre-configured with https://github.com/oasp/oasp-vscode-ide
  - Ant updated to latest.
  - Maven updated to latest.
  - Java updated to latest.
  - Nodejs LTS updated to latest.
  - @angular/cli included.
  - Yarn package manager updated.
  - Python3 updated.
  - Spyder3 IDE integrated in python3 installation updated.
  - OASP4JS-application-template for Angular 6 at workspaces/examples

24.2.3. My Thai Star Sample Application

The new release of My Thai Star has focused on the following improvements:

- Release 1.6.0.
- Travis CI integration with Docker. Now we get a valuable feedback of the current status and when collaborators make pull requests.
- Docker compose deployment.
- OASP4J:

- Flyway upgrade from 3.2.1 to 4.2.0
- Bug fixes.

- OASP4JS:
  - Client OASP4JS updated to Angular 6.
  - Frontend translated into 9 languages.
  - Improved mobile and tablet views.
  - Routing fade animations.
  - Compodoc included to generate dynamically frontend documentation.

24.2.4. Documentation updates

The following contents in the devonfw guide have been updated:

- devonfw OSS modules documentation.
- Creating a new OASP4J application.
- How to update Angular CLI in devonfw.
- Include Angular i18n.

Apart from this the documentation has been reviewed and some typos and errors have been fixed.

The current development of the guide has been moved to https://github.com/oasp-forge/devon-guide/wiki in order to be available as the rest of OSS assets.

24.2.5. OASP4J

The following changes have been incorporated in OASP4J:

- Integrate batch with archetype.
- Application module structure and dependencies improved.
- Issues with Application Template fixed.
- Solved issue where Eclipse maven template oasp4j-template-server version 2.4.0 produced pom with missing dependency spring-boot-starter-jdbc.
- Solved datasource issue with project archetype 2.4.0.
- Decouple archetype from sample (restaurant).
- Upgrade to Flyway 4.
- Fix for issue with Java 1.8 and QueryDSL #599.

24.2.6. OASP4JS

The following changes have been incorporated in OASP4JS:

- First version of the new client application architecture guide https://github.com/oasp-forge/oasp4js-wiki/wiki

- Angular CLI 6,
- Angular 6,
- Angular Material 6 and Covalent 2.0.0-beta.1,
- Ionic 3.20.0,
- Cordova 8.0.0,
- OASP4JS Angular application template updated to Angular 6 with visual improvements and bugfixes https://github.com/oasp/oasp4js-application-template
- OASP4JS Ionic application template updated and improved https://github.com/oasp/oasp4js-ionic-application-template
- PWA enabled.

24.2.7. AppSec Quick Solution Guide

This release incorporates a new Solution Guide for Application Security based on the state of the art in OWASP based application security. The purpose of this guide is to offer quick solutions for common application security issues for all applications based on devonfw. It's often the case that we need our systems to comply to certain sets of security requirements and standards. Each of these requirements needs to be understood, addressed and converted to code or project activity. We want this guide to prevent the wheel from being reinvented over and over again and to give clear hints and solutions to common security problems.

- The wiki can be accessed here: https://github.com/devonfw/devonfw-security/wiki
- The PDF can be accessed here: https://github.com/devonfw/devonfw-security

24.2.8. CobiGen

- CobiGen_Templates project and docs updated.
- CobiGen Angular 6 generation improved based on the updated application template
- CobiGen Ionic CRUD App generation based on Ionic application template. Although a first version was already implemented, it has been deeply improved:
  - Changed the code structure to comply with Ionic standards.
  - Added pagination.
  - Pull-to-refresh, swipe and attributes header implemented.
  - Code documented and JSDoc enabled (similar to Javadoc)
- CobiGen TSPlugin Interface Merge support.
- CobiGen XML plugin comes out with new cool features:
  - Enabled the use of XPath within variable assignment. You can now retrieve almost any data from an XML file and store it on a variable for further processing on the templates. Documented here.
  - Able to generate multiple output files per XML input file.
  - Generating code from UML diagrams. XMI files (standard XML for UML) can be now read

and processed. This means that you can develop templates and generate code from an XMI like class diagrams.

- CobiGen OpenAPI plugin released with multiple bug-fixes and other functionalities like:
  - Assigning global and local variables is now possible. Therefore you can set any string for further processing on the templates. For instance, changing the root package name of the generated files. Documented here.
  - Enabled having a class with more than one relationship to another class (more than one property of the same type).
- CobiGen Text merger plugin has been extended and now it is able to merge text blocks. This means, for example, that the generation and merging of AsciiDoc documentation is possible. Documented here.

24.2.9. Devcon

A new version of Devcon has been released. Fixes and new features include:

- Now Devcon is OSS, with public repository at https://github.com/devonfw/devcon
- Updated to match current OASP4J
- Update to download Linux distribution.
- Custom modules creation improvements.
- Bugfixes.

24.2.10. devonfw OSS Modules

- Existing devonfw IP modules have been moved to OSS.
  - They can now be accessed in any OASP4J project as optional dependencies from Maven Central.
  - The repository now has public access https://github.com/devonfw/devon
- Starters available for modules:
  - Reporting module
  - WinAuth AD Module
  - WinAuth SSO Module
  - I18n Module
  - Async Module
  - Integration Module
  - Microservice Module
  - Compose for Redis Module

See: https://github.com/devonfw/devon/wiki#devonfw-modules

24.2.11. devonfw Shop Floor

- devonfw Shop Floor 4 Docker
  - Docker-based CICD environment
    - docker-compose.yml (installation file)
    - dsf4docker.sh (installation script)
    - Service Integration (documentation in Wiki)
  - devonfw projects build and deployment with Docker
    - Dockerfiles (multi-stage building)
      - Build artifact (NodeJS for Angular and Maven for Java)
      - Deploy built artifact (NGINX for Angular and Tomcat for Java)
      - NGINX Reverse-Proxy to redirect traffic between both Angular client and Java server containers.
- devonfw Shop Floor 4 OpenShift
  - devonfw projects deployment in OpenShift cluster
    - s2i images
    - OpenShift templates
    - Video showcase (OpenShift Origin 3.6)

This incubator is intended to be a compilation of DevOps experiences from the devonfw perspective. “How we use our devonfw projects in DevOps environments”. Integration with the Production Line, creation and service integration of a Docker-based CI environment and deploying devonfw applications in an OpenShift Origin cluster using devonfw templates. See: https://github.com/devonfw/devonfw-shop-floor

24.2.12. devonfw Testing

The MrChecker Test Framework is an automated testing framework for functional testing of web applications, API web services, Service Virtualization, Security and in coming future native mobile apps, and databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions.

- Examples available under embedded project “MrChecker-App-Under-Test” and in project wiki: https://github.com/devonfw/devonfw-testing/wiki
- How to install:
  - Wiki : https://github.com/devonfw/devonfw-testing/wiki/How-to-install
- Release Note:
  - module core - 4.12.0.8:
    - fixes on getting Environment values
    - top notch example how to keep vulnerable data in repo , like passwords

- module selenium - 3.8.1.8:
  - browser driver auto downloader
  - list of out off the box examples to use in any web page
- module webAPI - ver. 1.0.2 :
  - api service virtualization with REST and SOAP examples
  - api service virtualization with dynamic arguments
  - REST working test examples with page object model
- module security - 1.0.1 (security tests against My Thai Start)
- module DevOps :
  - dockerfile for Test environment execution
  - CI + CD as Jenkinsfile code

24.2.13. devonfw methodology: Accelerated Solution Design

One of the prime challenges in Distributed Agile Delivery is the maintenance of a common understanding and unity of intent among all participants in the process of creating a product. That is: how can you guarantee that different parties in the client, different providers, all in different locations and time zones during a particular period of time actually understand the requirements of the client, the proposed solution space and the state of implementation.

We offer the Accelerated Solution Design as a possible answer to these challenges. The ASD is carefully designed to be a practical guideline that fosters and ensures the collaboration and communication among all team members.

The Accelerated Solution Design is:

- A practical guideline rather than a “methodology”
- Based on industry standards rather than proprietary methods
- Consisting of an evolving, “living”, document set rather than a static, fixed document
- Encapsulating the business requirements, functional definitions as well as Architecture design
- Based on the intersection of Lean, Agile, DDD and User Story Mapping

And further it is based on the essential belief or paradigm that ASD should be:

- Focused on the design (definition) of the “externally observable behavior of a system”
- Promoting communication and collaboration between team members
- Guided by prototypes

For more on the devonfw Methodology / ASD, see: https://github.com/devonfw/devon-methodology/blob/master/design-guidelines/Accelerated_Solution_Design.adoc

25. devonfw Release notes 2.3 "Dash"

25.1. Release: improving & strengthening the Platform

We are proud to announce the immediate release of **devonfw version 2.3** (code named "*Dash*" during development). This release comes with a bit of a delay as we decided to wait for the publication of OASP4j 2.5. "Dash" contains a slew of new features but in essence it is already driven by what we expect to be the core focus of 2018: strengthening the platform and improving quality.

After one year and a half of rapid expansion, we expect the next release(s) of the devonfw 2.x series to be fully focused on deepening the platform rather than expanding it. That is to say: we should work on improving existing features rather than adding new ones and strengthen the qualitative aspects of the software development life cycle, i.e. testing, infrastructure (CI, provisioning) etc.

"Dash" already is very much an example of this. This release contains the Allure Test Framework as an incubator. This is an automated testing framework for functional testing of web applications. Another incubator is the devonfw Shop Floor which intended to be a compilation of DevOps experiences from the devonfw perspective. And based on this devonfw has been *OpenShift Primed* ("certified") by Red Hat.

There is a whole range of new features and improvements which can be seen in that light. OASP4j 2.5 changes and improves the package structure of the core Java framework. The My Thai Star sample app has now been fully integrated in the different frameworks and the devonfw Guide has once again been significantly expanded and improved.

25.2. An industrialized platform for the ADcenter

Although less visible to the overall devonfw community, an important driving force was (meaning that lots of work has been done in the context of) the creation of the ADcenter concept towards the end of 2017. Based on a radical transformation of on/near/offshore software delivery, the focus of the ADcenters is to deliver agile & accelerated "Rightshore" services with an emphasis on:

- Delivering Business Value and optimized User Experience
- Innovative software development with state of the art technology
- Highly automated devops; resulting in lower costs & shorter time-to-market

The first two ADcenters, in Valencia (Spain) and Bangalore (India), are already servicing clients all over Europe - Germany, France, Switzerland and the Netherlands - while ADcenter aligned production teams are currently working for Capgemini UK as well (through Spain).Through the ADcenter, Capgemini establishes industrialized innovation; designed for & with the user. The availability of platforms for industrialized software delivery like devonfw and the Production Line has allowed us to train and make available over a 150 people in very short time.

The creation of the ADcenter is such a short time is visible proof that we´re getting closer to a situation where devonfw and Production Line are turning into the default development platform for APPS2, thereby standardizing all aspects of the software development life cycle: from training and design, architecture, devops and development, all the way up to QA and deployment.

25.3. Changes and new features

25.3.1. devonfw dist

The **devonfw dist**, or distribution, i.e. the central zip file which contains the main working environment for the devonfw developer, has been significantly enhanced. New features include:

- Eclipse Oxygen integrated
  - CheckStyle Plugin installed and configured
  - SonarLint Plugin installed and configured
  - Git Plugin installed
  - FindBugs replaced by SpotBugs and configured
  - Tomcat8 specific Oxygen configuration
  - CobiGen Plugin installed
- Other Software
  - Cmder integrated (when console.bat launched)
  - Visual Studio Code latest version included and pre-configured with https://github.com/devonfw/extension-pack-vscode
  - Ant updated to latest.
  - Maven updated to latest.
  - Java updated to latest.
  - Nodejs LTS updated to latest.
  - @angular/cli included.
  - Yarn package manager included.
  - Python3 integrated
  - Spyder3 IDE integrated in python3 installation
  - OASP4JS-application-template for Angular5 at workspaces/examples
  - Devon4sencha starter templates updated

25.3.2. OASP4j 2.5

Support for JAX-RS & JAX-WS clients

With the aim to enhance the ease in consuming RESTful and SOAP web services, JAX-RS and JAX-WS clients have been introduced. They enable developers to concisely and efficiently implement portable client-side solutions that leverage existing and well-established client-side HTTP connector implementations. Furthermore, the getting started time for consuming web services has been considerably reduced with the default configuration out-of-the-box which can be tweaked as per individual project requirements.

See: https://github.com/oasp/oasp4j/issues/358

Separate security logs for OASP4J log component

Based on OWASP(Open Web Application Security Project), OASP4J aims to give developers more control and flexibility with the logging of security events and tracking of forensic information. Furthermore, it helps classifying the information in log messages and applying masking when necessary. It provides powerful security features while based on set of logging APIs developers are already familiar with over a decade of their experience with Log4J and its successors.

See: https://github.com/oasp/oasp4j/issues/569

Support for Microservices

Integration of an OASP4J application to a Microservices environment can now be leveraged with this release of OASP4J. Introduction of service clients for RESTful and SOAP web services based on Java EE give developers agility and ease to access microservices in the Devon framework. It significantly cuts down the efforts on part of developers around boilerplate code and stresses more focus on the business code improving overall efficiency and quality of deliverables.

See: https://github.com/oasp/oasp4j/pull/589/commits

25.3.3. Cobigen

A new version of Cobigen has been included. New features include:

- Swagger/Yaml Plugin for CobiGen. CobiGen is able to read a swagger definition file that follows the OpenAPI 3.0 spec and generate code. A preliminary release was already included in 2.2.1 but the current version is much more mature and stable. See: https://github.com/devonfw/cobigen/wiki/howto_openapi_generation
- Integration of CobiGen into Maven build process. This already existed but has been improved. It consists mainly of documentation + better log output and bug fixes. See: https://github.com/devonfw/cobigen/wiki/cobigen-maven_configuration
- CobiGen Ionic CRUD App generation based on https://github.com/oasp/oasp4js-ionic-application-template
- Cobigen_Templates project and docs updated
- Bugfixes and Hardening

25.3.4. My Thai Star Sample Application

From this release on the My Thai Star application has been fully integrated in the different frameworks in the platform. Further more, a more modularized approach has been followed in the current release of My Thai star application to decouple client from implementation details. Which provides better encapsulation of code and dependency management for API and implementation classes. This has been achieved with creation of a new “API” module that contain interfaces for REST services and corresponding Request/Response objects. With existing “Core” module being dependent on “API” module. To read further you can follow the link https://github.com/oasp/my-thai-star/wiki/java-design#basic-architecture-details

Furthermore: an email and Twitter micro service were integrated in my-thai-star. This is just for

demonstration purposes. A full micro service framework is already part of oasp4j 2.5.0

25.3.5. Documentation refactoring

The complete devonfw guide is restructured and refactored. Getting started guides are added for easy start with devonfw.Integration of the new Tutorial with the existing devonfw Guide whereby existing chapters of the previous tutorial were converted to Cookbook chapters. Asciidoctor is used for devonfw guide PDF generation. See: https://github.com/devonfw/devon-guide/wiki

25.3.6. OASP4JS

The following changes have been incorporated in OASP4JS:

- Angular CLI 1.6.0,
- Angular 5.1,
- Angular Material 5 and Covalent 1.0.0 RC1,
- PWA enabled,
- Core and Shared Modules included to follow the recommended Angular projects structure,
- Yarn and NPM compliant since both lock files are included in order to get a stable installation.

25.3.7. Admin interface for oasp4j apps

The new version includes an Integration of an admin interface for oasp4j apps (Spring Boot). This module is based on CodeCentric´s Spring Boot Admin (https://github.com/codecentric/spring-boot-admin). See: https://github.com/devonfw/devon-guide/wiki/Spring-boot-admin-Integration-with-devon4j

25.3.8. Devcon

A new version of Devcon has been released. Fixes and new features include:

- Renaming of system Commands.
- New menu has been added - “other modules”, if menus are more than 10, other modules will display some menus.
- A progress bar has been added for installing the distribution

25.3.9. devonfw Modules

Existing devonfw modules can now be accessed with the help of starters following namespace devonfw-<module_name>-starter. Starters available for modules:

- Reporting module
- WinAuth AD Module
- WinAuth SSO Module
- I18n Module

- Async Module
- Integration Module
- Microservice Module
- Compose for Redis Module

See: https://github.com/devonfw/devon/wiki#ip-modules

25.3.10. devonfw Shop Floor

This incubator is intended to be a compilation of DevOps experiences from the devonfw perspective. “How we use our devonfw projects in DevOps environments”. Integration with the Production Line, creation and service integration of a Docker-based CI environment and deploying devonfw applications in an OpenShift Origin cluster using devonfw templates.

See: https://github.com/devonfw/devonfw-shop-floor

25.3.11. devonfw-testing

The Allure Test Framework is an automated testing framework for functional testing of web applications and in coming future native mobile apps, web services and databases. All modules have tangible examples of how to build resilient integration test cases based on delivered functions.

- Examples available under embedded project “Allure-App-Under-Test” and in project wiki: https://github.com/devonfw/devonfw-testing/wiki
- How to install: https://github.com/devonfw/devonfw-testing/wiki/How-to-install
- Release Notes:
  - Core Module – ver.4.12.0.3:
    - Test report with logs and/or screenshots
    - Test groups/tags
    - Data Driven (inside test case, external file)
    - Test case parallel execution
    - Run on independent Operating System (Java)
    - Externalize test environment (DEV, QA, PROD)
  - UI Selenium module – ver. 3.4.0.3:
    - Malleable resolution (Remote Web Design, Mobile browsers)
    - Support for many browsers(Internet Explorer, Edge, Chrome, Firefox, Safari)
    - User friendly actions (elementCheckBox, elementDropdown, etc.)
    - Ubiquese test execution (locally, against Selenium Grid through Jenkins)
    - Page Object Model architecture
    - Selenium WebDriver library ver. 3.4.0

See: https://github.com/devonfw/devonfw-testing/wiki

25.3.12. DOT.NET Framework incubators

The .NET Core and Xamarin frameworks are still under development by a workgroup from The Netherlands, Spain, Poland, Italy, Norway and Germany. The 1.0 release is expected to be coming soon but the current incubator frameworks are already being used in several engagements. Some features to highlight are:

- Full .NET implementation with multi-platform support
- Detailed documentation for developers
- Docker ready
- Web API server side template :
  - Swagger auto-generation
  - JWT security
  - Entity Framework Support
  - Advanced log features
- Xamarin Templates based on Excalibur framework
- My Thai Star implementation:
  - Backend (.NET Core)
  - FrontEnd (Xamarin)

25.3.13. devonfw has been Primed by Red Hat for OpenShift

OpenShift is a supported distribution of Kubernetes from Red Hat for container-based software deployment and management. It is using Docker containers and DevOps tools for accelerated application development. Using OpenShift allows Capgemini to avoid Cloud Vendor lock-in. OpenShift provides devonfw with a state of the art CI/CD environment (devonfw Shop Floor), providing devonfw with a platform for the whole development life cycle: from development to staging / deploy.

See https://hub.openshift.com/primed/120-capgemini and https://github.com/oasp/s2i

25.3.14. Harvested components and modules

The devonfw Harvesting process continues to add valuable components and modules to the devonfw platform. The last months the following elements were contributed:

Service Client support (for Micro service Projects).

This client is for consuming microservices from other application.This solution is already very flexible and customizable.As of now,this is suitable for small and simple project where two or three microservices are invoked. Donated by Jörg Hohwiller. See: https://github.com/devonfw/devon-microservices

JHipster devonfw code generation

This component was donated by the ADcenter in Valencia. It was made in order to comply with strong requirements (especially from the French BU) to use jHipster for code generation.

JHipster is a code generator based on Yeoman generators. Its default generator generator-jhipster generates a specific JHipster structure. The purpose of generator-jhipster-DevonModule is to generate the structure and files of a typical OASP4j project. It is therefore equivalent to the standard OASP4j application template based CobiGen code generation.

See: https://github.com/devonfw/devon-guide/wiki/cookbook-devon-jhipster-module

Simple Jenkins task status dashboard

This component has been donated by, has been harvested from system in use by, Capgemini Valencia. This dashboard, apart from an optional gamification element, allows the display of multiple Jenkins instances. See: https://github.com/oasp/jenkins_view

25.3.15. And lots more, among others:

- OASP4J/devonfw docker based build IN a docker process. See: https://github.com/devonfw/devon-guide/wiki/Dockerfile-for-the-maven-based-spring.io-projects
- CI test boot archetype. This is for unit testing.This will create a sample project and add sample web service to it. A Jenkins job will start oasp4j server and will call web service. See: https://github.com/devonfw/devonfw-shop-floor/tree/master/testing/Oasp4jTestingScripts
- CI test Angular starterTemplate. Testing automation for Angular applications (My Thai Star) in Continuous Integration environments by using Headless browsers and creating Node.js scripts. See: https://github.com/oasp/my-thai-star/blob/develop/angular/package.json#L8-L12 and https://github.com/oasp/my-thai-star/blob/develop/angular/karma.conf.js

26. devonfw Release notes 2.2 "Courage"

26.1. Production Line Integration

devonfw is now fully supported on the Production Line v1.3 and the coming v2.0. Besides that, we now "eat our own dogfood" as the whole devonfw project, all "buildable assets", now run on the Production Line.

26.2. OASP4js 2.0

The main focus of the Courage release is the renewed introduction of "OASP for JavaScript", or OASP4js. This new version is a completely new implementation based on Angular (version 4). This new "stack" comes with:

- New application templates for Angular 4 application (as well as Ionic 3)
- A new reference application
- A new tutorial (and Architecture Guide following soon)
- Component Gallery
- New CobiGen templates for generation of both Angular 4 and Ionic 3 UI components ("screens")
- Integration of Covalent and Bootstrap offering a large number of components
- my-thai-star, a showcase and reference implementation in Angular of a real, responsive usable app using recommended architecture and patterns
- A new Tutorial using my-thai-star as a starting point

See: https://github.com/oasp/oasp4js-application-template https://github.com/oasp/oasp4js-angular-catalog https://github.com/oasp/my-thai-star/tree/develop/angular

26.3. A new OASP Portal

As part of the new framework(s) we have also done a complete redesign of the OASP Portal website at http://oasp.io/ which should make all things related with OASP more accessible and easier to find.

26.4. New Cobigen

Major changes in this release:

- Support for multi-module projects
- Client UI Generation:
  - New Angular 4 templates based on the latest - angular project seed
  - Basic Typescript Merger
  - Basic Angular Template Merger
  - JSON Merger

- Refactored oasp4j templates to make use of Java template logic feature
- Bugfixes:
  - Fixed merging of nested Java annotations including array values
  - more minor issues
- Under the hood:
  - Large refactoring steps towards language agnostic templates formatting sensitive placeholder descriptions automatically formatting camelCase to TrainCase to snake-case, etc.
- Easy setup of CobiGen IDE to enable fluent contribution
- CI integration improved to integrate with GitHub for more valuable feedback

See: https://github.com/devonfw/cobigen/releases

26.5. MyThaiStar: New Restaurant Example, reference implementation & Methodology showcase

A major part of the new devonfw release is the incorporation of a new application, "my-thai-star" which among others:

- serve as an example of how to make a "real" devonfw application (i.e. the application could be used for real)
- Serves as an attractive showcase
- Serves as a reference application of devonfw patterns and practices as well as the standard example in the new devonfw tutorial
- highlights modern security option like JWT Integration

The application is accompanied by a substantial new documentation asset, the devonfw methodology, which described in detail the whole lifecycle of the development of a devonfw application, from requirements gathering to technical design. Officially my-that-star is still considered to be an incubator as especially this last part is still not as mature as it could be. But the example application and tutorial are 100% complete and functional and form a marked improvement over the "old" restaurant example app. My-Thai-star will become the standard example app from devonfw 3.0 onwards.

See: https://github.com/oasp/my-thai-star https://github.com/oasp/my-thai-star/wiki

26.6. The new OASP Tutorial

The OASP Tutorial is a new part of the combined OASP / devonfw documentation which changes the focus of how people can get started with the platform

There are tutorials for OASP4j, OASP4js (Angular), OASP4fn and more to come. My-Thai-Star is used throughout the tutorial series to demonstrate the basic principles, architecture, and good practices of the different OASP "stacks". There is an elaborated exercise where the readers get to write their

own application "JumpTheQueue".

We hope that the new tutorial offers a better, more efficient way for people to get started with devonfw. Answering especially the question: how to make a devonfw application.

Oasp4j tutorial: https://github.com/oasp/oasp-tutorial-sources/wiki/OASP4jGettingStartedHome
Oasp4js tutorial: https://github.com/oasp/oasp-tutorial-sources/wiki/OASP4jsGettingStartedHome
Oasp4fn tutorial: https://github.com/oasp/oasp-tutorial-sources/wiki/OASP4FnGettingStartedHome

26.7. OASP4j 2.4.0

"OASP for Java" or OASP4j now includes updated versions of the latest stable versions of Spring Boot and the Spring Framework and all related dependencies. This allows guaranteed, stable, execution of any devonfw 2.X application on the latest versions of the Industry Standard Spring stack. Another important new feature is a new testing architecture/infrastructure. All database options are updated to the latest versions as well as guaranteed to function on all Application Servers which should cause less friction and configuration time when starting a new OASP4j project.

Details:

- Spring Boot Upgrade to 1.5.3
- Updated all underlying dependencies
- Spring version is 4.3.8
- Exclude Third Party Libraries that are not needed from sample restaurant application
- Bugfix:Fixed the 'WhiteLabel' error received when tried to login to the sample restaurant application that is deployed onto external Tomcat
- Bugfix:Removed the API api.org.apache.catalina.filters.SetCharacterEncodingFilter and used spring framework's API org.springframework.web.filter.CharacterEncodingFilter instead
- Bugfix:Fixed the error "class file for javax.interceptor.InterceptorBinding not found" received when executing the command 'mvn site' when trying to generate javadoc using Maven javadoc plugin
- Removed the deprecated API io.oasp.module.web.common.base.PropertiesWebApplicationContextInitializer
- Documentation of the usage of UserDetailsService of Spring Security

See: https://github.com/oasp/oasp4j

Wiki: https://github.com/oasp/oasp4j/wiki

26.8. Microservices Netflix

devonfw now includes a microservices implementation based on Spring Cloud Netflix. It provides a Netflix OSS integrations for Spring Boot apps through auto-configuration and binding to the Spring Environment. It offers microservices archetypes and a complete user guide with all the details to

start creating microservices with devonfw.

See: https://github.com/devonfw-forge/devon-guide/wiki/devon-microservices

26.9. devonfw distribution based on Eclipse OOMPH

The new Eclipse devonfw distribution is now based on Eclipse OOMPH, which allows us, an any engagement, to create and manage the distribution more effectively by formalizing the setup instructions so they can be performed automatically (due to a blocking issue postponed to devonfw 2.2.1 which will be released a few weeks after 2.2.0)

26.10. Visual Studio Code / Atom

The devonfw distro now contains Visual Studio Code alongside Eclipse in order to provide a default, state of the art, environment for web based development.

See: https://github.com/oasp/oasp-vscode-ide

26.11. More I18N options

The platform now contains more documentation and a conversion utility which makes it easier to share i18n resource files between the different frameworks.

See: https://github.com/devonfw/devon/wiki/cookbook-i18n-resource-converter

26.12. Spring Integration as devonfw Module

This release includes a new module based on the Java Message Service (JMS) and Spring Integration which provides a communication system (sender/subscriber) out-of-the-box with simple channels (only to send and read messages), request and reply channels (to send messages and responses) and request & reply asynchronously channels.

See: https://github.com/devonfw/devon/wiki/cookbook-integration-module

26.13. devonfw Harvest contributions

devonfw contains a whole series of new components obtained through the Harvesting process. Examples are :

- New backend IP module Compose for Redis: management component for cloud environments. Redis is an open-source, blazingly fast, key/value low maintenance store. Compose's platform gives you a configuration pre-tuned for high availability and locked down with additional security features. The component will manage the service connection and the main methods to manage the key/values on the storage. The library used is "lettuce".
- Sencha component for extending GMapPanel with the following functionality :
    - Markers management

- Google Maps options management
- Geoposition management
- Search address and coordinates management
- Map events management
- Map life cycle and behavior management

- Sencha responsive Footer that moves from horizontal to vertical layout depending on the screen resolution or the device type. It is a simple functionality but we consider it very useful and reusable.

See: https://github.com/devonfw/devon/wiki/cookbook-compose-for-redis-module

26.14. More Deployment options to JEE Application Servers and Docker/CloudFoundry

The platform now fully supports deployment on the latest version of Weblogic, WebSphere, Wildfly (JBoss) as well as Docker and Cloud Foundry.

See: https://github.com/devonfw/devon/wiki/Deployment-on-WebLogic https://github.com/devonfw/devon/wiki/cookbook-Deployment-on-WebSphere https://github.com/devonfw/devon/wiki/cookbook-Deployment-on-Wildfly

26.15. Devcon on Linux

Devcon is now fully supported on Linux which, together with the devonfw distro running on Linux, makes devonfw fully multi-platform and Cloud compatible (as Linux is the default OS in the Cloud!)

See: https://github.com/devonfw/devcon/releases

26.16. New OASP Incubators

From different Business Units (countries) have contributed "incubator" frameworks:

- OASP4NET (Stack based on .NET Core / .NET "Classic" (4.6))
- OASP4X (Stack based on Xamarin)
- OASP4Fn (Stack based on Node-js/Serverless): https://github.com/oasp/oasp4fn

An "incubator" status means that the frameworks are production ready, all are actually already used in production, but are still not fully compliant with the OASP definition of a "Minimally Viable Product".

During this summer the OASP4NET and OASP4X repos will be properly installed. In the mean time, if you want to have access to the source code, please contact the *devonfw Core Team*.

27. Release notes devonfw 2.1.1 "Balu"

27.1. Version 2.1.2: OASP4J updates & some new features

We've released the latest update release of devonfw in the *Balu* series: version 2.1.2. The next major release, code named *Courage*, will be released approximately the end of June. This current release contains the following items:

27.1.1. OASP4j 2.3.0 Release

Friday the 12th of May 2017 OASP4J version 2.3.0 was released. Major features added are :

- Database Integration with PostGres, MSSQL Server, MariaDB
- Added docs folder for gh pages and added oomph setups
- Refactored Code
- Refactored Test Infrastructure
- Added Documentation on debugging tests
- Added Two Batch Job tests in the restaurant sample
- Bugfix: Fixed the error received when the Spring Boot Application from sample application that is created from maven archetype is launched
- Bugfix: Fix for 404 error received when clicked on the link '1. Table' in index.html of the sample application created from maven archetype

More details on features added can be found at https://github.com/oasp/oasp4j/milestone/23?closed=1 . The OASP4j wiki and other documents are updated for release 2.3.0.

27.1.2. CobiGen Enhancements

Previous versions of CobiGen are able to generate code for REST services only. Now it is possible to generate the code for SOAP services as well. There are two use cases available in CobiGen:

- SOAP without nested data
- SOAP nested data

The "nested data" use case is when there are 3 or more entities which are interrelated with each other. CobiGen will generate code which will return the nested data. Currently CobiGen services return ETO classes, CobiGen has been enhanced as to return CTO classes (ETO + relationship).

Apart from the SOAP code generation, the capability to express nested relationships have been added to the existing ReST code generator as well.

See: https://github.com/devonfw/devon-guide/wiki/cookbook-cobigen-advanced-use-cases-soap-and-nested-data

27.1.3. Micro services module (Spring Cloud/Netflix OSS)

To make it easier for devonfw users to design and develop applications based on microservices, this release provides a series of archetypes and resources based on *Spring Cloud Netflix* to automate the creation and configuration of microservices.

New documentation ind de devonfw Guide contains all the details to start creating microservices with devonfw

27.1.4. Spring Integration Module

Based on the *Java Message Service* (JMS) and *Spring Integration*, the devonfw *Integration module* provides a communication system (sender/subscriber) out-of-the-box with simple channels (only to send and read messages), request and reply channels (to send messages and responses) and request & reply asynchronously channels. You can find more details about the implementation in the devonfw guide.

27.1.5. WebSphere & Wildfly deployment documentation

The new version of devonfw contains more elaborate and updated documentation about deployment on WebSpere and Wildfly.

27.2. Version 2.1.1 Updates, fixes & some new features

27.2.1. CobiGen code-generator fixes

The CobiGen incremental code generator released in the previous version contained a regression which has now been fixed. Generating services in Batch mode whereby a package can be given as an input, using all Entities contained in that package, works again as expected.

For more information see: The CobiGen documentation and the corresponding change in the devonfw Guide

27.2.2. Devcon enhancements

In this new release we have added devcon to the devonfw distribution itself so one can directly use devcon from the console.bat or ps-console.bat windows. It is therefore no longer necessary to independently install devcon. However, as devcon is useful outside of the devonfw distribution, this remains a viable option.

27.2.3. Devon4Sencha

in Devon4Sencha there are changes in the sample application. It now complies fully with the architecture which is known as "universal app", so now it has screens custom tailored for desktop and mobile devices. All the basic logic remains the same for both versions. (The StarterTemplate is still only for creating a desktop app. This will be tackled in the next release.)

27.2.4. New Winauth modules

The original *winauth* module that, in previous Devon versions, implemented the *Active Directory* authentication and the *Single Sign-on* authentication now has been divided in two independent modules. The *Active Directory* authentication now is included in the new *Winauth-ad* module whereas the *Single Sign-on* implementation is included in a separate module called *Winauth-sso*. Also some improvements have been added to *Winauth-sso* module to ease the way in which the module can be injected.

For more information about the update see: The Sencha docs within the devonfw Guide

27.2.5. General updates

There are a series of updates to the devonfw documentation, principally the devonfw Guide. Further more, from this release on, you can find the devonfw guide in the *doc* folder of the distribution.

Furthermore, the OASP4J and devonfw source-code in the "examples" workspace, have been updated to the latest version.

27.3. Version 2.1 New features, improvements and updates

27.3.1. Introduction

We are proud to present the new release of devonfw, version "2.1" which we've baptized "Balu". A major focus for this release is developer productivity. So that explains the name, as Balu is not just big, friendly and cuddly but also was very happy to let Mowgli do the work for him.

27.3.2. Cobigen code-generator UI code generation and more

The Cobigen incremental code generator which is part of devonfw has been significantly improved. Based on a single data schema it can generate the JPA/Hibernate code for the whole service layer (from data-access code to web services) for all CRUD operations. When generating code, Cobigen is able to detect and leave untouched any code which developers have added manually.

In the new release it supports Spring Data for data access and it is now capable of generating the whole User Interface as well: data-grids and individual rows/records with support for filters, pagination etc. That is to say: Cobigen can now generate automatically all the code from the server-side database access layer all the way up to the UI "screens" in the web browser.

Currently we support Sencha Ext JS with support for Angular 2 coming soon. The code generated by Cobigen can be opened and used by Sencha Architect, the visual design tool, which enables the programmer to extend and enhance the generated UI non-programmatically. When Cobigen regenerates the code, even those additions are left intact. All these features combined allow for an iterative, incremental way of development which can be up to an order of an magnitude more productive than "programming manual"

Cobigen can now also be used for code-generation within the context of an engagement. It is easily extensible and the process of how to extend it for your own project is well documented. This becomes already worthwhile ("delivers ROI") when having 5+ identical elements within the project.

For more information see: The Cobigen documentation and the corresponding changer in the devonfw Guide and

27.3.3. Angular 2

With the official release of Angular 2 and TypeScript 2, we're slowly but steadily moving to embrace these important new players in the web development scene. We keep supporting the Angular 1 based OASP4js framework and are planning a migration of this framework to Angular 2 in the near future. For "Balu" we've have decided to integrate "vanilla" Angular 2.

We have migrated the Restaurant Sample application to serve as a, documented and supported, blueprint for Angular 2 applications. Furthermore, we support three "kickstarter" projects which help engagement getting started with Angular2 - either using Bootstrap or Google´s Material Design - or, alternatively, Ionic 2 (the mobile framework on top of Angular 2). For more information see: Angular 2 Kickstarter and Ionic 2 Kickstarter

27.3.4. OASP4J 2.2.0 Release

A new release of OASP4J, version 2.2.0, is included in this release of devonfw. This release mainly focuses on server side of oasp. i.e oasp4j.

Major features added are :

- Upgrade to Spring Boot 1.3.8.RELEASE
- Upgrade to Apache CXF 3.1.8
- Database Integration with Oracle 11g
- Added Servlet for HTTP-Debugging
- Refactored code and improved JavaDoc
- Bugfix: mvn spring-boot:run executes successfully for oasp4j application created using oasp4j template
- Added subsystem tests of SalesmanagementRestService and several other tests
- Added Tests to test java packages conformance to OASP conventions

More details on features added can be found at https://github.com/oasp/oasp4j/milestone/19?closed=1(here). The OASP4j wiki and other documents are updated for release 2.2.0.

27.3.5. Devon4Sencha

Devon4Sencha is an alternative view layer for web applications developed with devonfw. It is based on Sencha Ext JS. As it requires a license for commercial applications it is not provided as Open Source and is considered to be part of the IP of Capgemini.

These libraries provide support for creating SPA (Single Page Applications) with a very rich set of

components for both desktop and mobile. In the new version we extend this functionality to support for "Universal Apps", the Sencha specific term for true multi-device applications which make it possible to develop a single application for desktop, tablet as well as mobile devices. In the latest version Devon4Sencha has been upgraded to support Ext JS 6.2 and we now support the usage of Cobigen as well as Sencha Architect as extra option to improve developer productivity. For more information about the update see: The Sencha docs within the devonfw Guide

27.3.6. Devcon enhancements

The Devon Console, Devcon, is a cross-platform command line tool running on the JVM that provides many automated tasks around the full life-cycle of Devon applications, from installing the basic working environment and generating a new project, to running a test server and deploying an application to production. It can be used by the engagements to integrate with their proprietary tool chain.

In this new release we have added an optional graphical user interface (with integrated help) which makes using Devcon even easier to use. Another new feature is that it is now possible to easily extend it with commands just by adding your own or project specific Javascript files. This makes it an attractive option for project task automation. You can find more information in the Devcon Command Developers Guide

27.3.7. Ready for the Cloud

devonfw is in active use in the Cloud, with projects running on IBM Bluemix and on Amazon AWS. The focus is very much to keep Cloud-specific functionality decoupled from the devonfw core. The engagement can choose between - and easily configure the use of - either CloudFoundry or Spring Cloud (alternatively, you can run devonfw in Docker containers in the Cloud as well. See elsewhere in the release notes). For more information about how to configure devonfw for use in the cloud see: devonfw on Docker and devonfw in IBM Bluemix

27.3.8. Spring Data

The java server stack within devonfw, OASP4J, is build on a very solid DDD architecture which uses JPA for its data access layer. We now offer integration of Spring Data as an alternative or to be used in conjunction with JPA. Spring Data offers significant advantages over JPA through its query mechanism which allows the developer to specify complex queries in an easy way. Overall working with Spring Data should be quite more productive compared with JPA for the average or junior developer. And extra advantage is that Spring Data also allows - and comes with support for - the usage of NoSQL databases like MongoDB, Cassandra, DynamoDB etc. THis becomes especially critical in the Cloud where NoSQL databases typically offer better scalability than relational databases. For more information see: Integrating Spring Data in OASP4J

27.3.9. Videos content in the devonfw Guide

The devonfw Guide is the single, authoritative tutorial and reference ("cookbook") for all things devonfw, targeted at the general developer working with the platform (there is another document for Architects). It is clear and concise but because of the large scope and wide reach of devonfw, it comes with a hefty 370+ pages. For the impatient - and sometimes images do indeed say more than

words - we've added 17 videos to the Guide which significantly speed up getting started with the diverse aspects of devonfw.

For more information see: Video releases on TeamForge

27.3.10. Containerisation with Docker and the Production Line

Docker (see: https://www.docker.com/) containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. Docker containers resemble virtual machines but are far more resource efficient. Because of this, Docker and related technologies like Kubernetes are taking the Enterprise and Cloud by storm. We have certified and documented the usage of devonfw on Docker so we can now firmly state that "devonfw is Docker" ready. All the more so as the iCSD Production Line is now supporting devonfw as well. The Production Line is a Docker based set of methods and tools that make possible to develop custom software to our customers on time and with the expected quality. By having first-class support for devonfw on the Production Line, iCSD has got an unified, integral solution which covers all the phases involved on the application development cycle from requirements to testing and hand-off to the client.

See: devonfw on Docker and devonfw on the Production Line

27.3.11. Eclipse Neon

devonfw comes with its own pre configured and enhanced Eclipse based IDE: the Open Source "OASP IDE" and "devonfw Distr" which falls under Capgemini IP. We've updated both versions to the latest stable version of Eclipse, Neon. From Balu onwards we support the IDE on Linux as well and we offer downloadable versions for both Windows and Linux.

See: The Devon IDE

27.3.12. Default Java 8 with Java 7 compatibility

From version 2.1. "Balu" onwards, devonfw is using by default Java 8 for both the tool-chain as well as the integrated development environments. However, both the framework as well as the IDE and tool-set remain fully backward compatible with Java 7. We have added documentation to help configuring aspects of the framework to use Java 7 or to upgrade existing projects to Java 8. See: Compatibility guide for Java7, Java8 and Tomcat7, Tomcat8

27.3.13. Full Linux support

In order to fully support the move towards the Cloud, from version 2.1. "Balu" onwards, devonfw is fully supported on Linux. Linux is the de-facto standard for most Cloud providers. We currently only offer first-class support for Ubuntu 16.04 LTS onward but most aspects of devonfw should run without problems on other and older distributions as well.

27.3.14. Initial ATOM support

Atom is a text editor that's modern, approachable, yet hackable to the core - a tool you can customize to do anything but also use productively without ever touching a config file. It is turning

into a standard for modern web development. In devonfw 2.1 "Balu" we provide a script which installs automatically the most recent version of Atom in the devonfw distribution with a pre-configured set of essential plugins. See: OASP/devonfw Atom editor ("IDE") settings & packages

27.3.15. Database support

Through JPA (and now Spring Data as well) devonfw supports many databases. In Balu we've extended this support to prepared configuration, extensive documentations and supporting examples for all major "Enterprise" DB servers. So it becomes even easier for engagements to start using these standard database options. Currently we provide this extended support for Oracle, Microsoft SQL Server, MySQL and PostgreSQL. For more information see: OASP Database Migration Guide

27.3.16. File upload and download

File up and download was supported in previous version of the framework, but as these operations are common but complex, we've extended the base functionality and improved the available documentation so it becomes substantially easier to offer both File up- as well as download in devonfw based applications. See: devonfw Guide Cookbook: File Upload and Download

27.3.17. Internationalisation (I18N) improvements

Likewise, existing basic Internationalisation (I18N) support has been significantly enhanced through an new devonfw module and extended to support Ext JS and Angular 2 apps as well. This means that both server as well as client side applications can be made easily to support multiple languages ("locales"), using industry standard tools and without touching programming code (essential when working with teams of translators). For more information see: The I18N (Internationalization) module and Client GUI Sencha i18n

27.3.18. Asynchronous HTTP support

Asynchronous HTTP is an important feature allowing so-called "long polling" HTTP Requests (for streaming applications, for example) or with requests sending large amounts of data. By making HTTP Requests asynchronous, devonfw server instances can better support these types of use-cases while offering far better performance. Documentation about how to include the new devonfw module implementing this feature can be found at: The devonfw async module

27.3.19. Security and License guarantees

In devonfw security comes first. The components of the framework are designed and implemented according to the recommendations and guidelines as specified by OWASP in order to confront the top 10 security vulnerabilities.

From version 2.1 "Balu" onward we certify that devonfw has been scanned by software from "Black Duck". This verifies that devonfw is based on 100% Open Source Software (non Copyleft) and demonstrates that at moment of release there are no known, critical security flaws. Less critical issues are clearly documented.

27.3.20. Documentation improvements

Apart from the previously mentioned additions and improvements to diverse aspects of the devonfw documentation, principally the devonfw Guide, there are a number of other important changes. We've incorporated the Devon Modules Developer´s Guide which describes how to extend devonfw with its Spring-based module system. Furthermore we've significantly improved the Guide to the usage of web services. We've included a Compatibility Guide which details a series of considerations related with different version of the framework as well as Java 7 vs 8. And finally, we've extended the F.A.Q. to provide the users with direct answers to common, Frequently Asked Questions.

27.3.21. Contributors

Many thanks to adrianbielewicz, aferre777, amarinso, arenstedt, azzigeorge, cbeldacap, cmammado, crisjdiaz, csiwiak, Dalgar, drhoet, Drophoff, dumbNickname, EastWindShak, fawinter, fbougeno, fkreis, GawandeKunal, henning-cg, hennk, hohwille, ivanderk, jarek-jpa, jart, jensbartelheimer, jhcore, jkokoszk, julianmetzler, kalmuczakm, kiran-vadla, kowalj, lgoerlach, ManjiriBirajdar, MarcoRose, maybeec, mmatczak, nelooo, oelsabba, pablo-parra, patrhel, pawelkorzeniowski, PriyankaBelorkar, RobertoGM, sekaiser, sesslinger, SimonHuber, sjimenez77, sobkowiak, sroeger, ssarmokadam, subashbasnet, szendo, tbialecki, thoptr, tsowada, znazir and anyone who we may have forgotten to add!

