

BI - Report (Eclipse BIRT, Spring boot)

目次

はじめに	1
1. Report ツール.....	1
2. BIRT のインストール.....	1
2.1. Spring Tools Suite (STS).....	2
2.2. BIRT プラグイン.....	2
3. レポートフォーマットの作成 (BIRT デザイナ)	4
4. 開発環境でのデータベース接続	5
4.1. JDBC データソース	5
4.2. データセットの登録.....	6
4.3. レポートの編集	7
4.4. 開発中の確認 (PDF 作成の例)	9
5. レポートの作成 (BIRT ビューア)	11
5.1. バッチ起動によるレポート作成.....	11
5.2. Web アプリ (WebViewerExample)	11
5.2.1. アプリサーバの準備.....	11
5.2.2. BIRT ビューアの配備	12
5.2.3. 新規作成したレポートの追加	14
5.2.4. アプリサーバ (Tomcat) の再起動	15
5.2.5. 新規レポートの表示.....	15
6. セキュリティの追加.....	16
6.1. Web アプリの構成.....	16
6.2. STS を使ったプロジェクトの作成.....	17
6.2.1. Web アプリ開発用の Java 設定	17
6.2.2. 新規プロジェクト	18
6.2.3. 作成する資材	20
6.2.4. STS からの起動.....	24
6.3. 実行形式 (jar) ファイルの作成.....	25
6.4. Jar の実行	27
6.5. セキュリティに関する注意点	28

BI - Report (Eclipse BIRT, Spring boot)

はじめに

BIという言葉が最初に使われたのは1959年の刊行物¹ (IBM Journal vol.2) で「産業、科学、または政府機関のさまざまな部門に情報を広めるための自動システムが…」という説明と共にでてきます。現代でも IBM 社からは BI 関連製品が出荷され、AI (による分析) を活用した BI を謳っています。Microsoft 社も BI を名前の一部に冠した製品を出していますが、こちらは会話型のデータ視覚化ツールという位置づけです。日本では BI という言葉自体に馴染みが薄く、「分析...何を?」という感じですがコードレスの視覚化ツールとして利用できる場面は多いと思います。特に定型的な帳票出力に使う場合はシステムと会話する必要もありません。

帳票のファイル形式には作成ツールの独自形式のものもありますが、多くの場合 PDF (ISO 32000) が選択できます。PDF はスクリプトで編集・出力するためのライブラリも作られています (Python ReportLab 他) が、BI ツールを使うと複雑な様式の帳票も定義体を作って管理することができます。

1. Report ツール

紙媒体から電子媒体に変化しても一定の期間保存を義務付けられている会計帳簿、社外取引文書のようなものがあるので帳票イメージの作成ツールが製品としていくつも市場にでています。帳票ツールを扱うとき、デザイン (レイアウトや固定出力部分の設定) とビューア (データの埋め込みと表示) の2つの要素があるのでこの組み合わせを考える必要があります。例えば、少量・簡易なレイアウトであれば Web ブラウザにデータを表示して印刷や PDF 化を利用者に任せる。あるいは、SVG を Illustrator 等のソフトで作成してブラウザのスクリプトでデータを埋め込む方法等があります。

BI に分類される帳票ツールを使う利点は帳票のデザインとビューアがセットで構成されていることと、データソースを関係づけられることでデータに纏わる制御が要らなくなり、定型の帳票を人手を介さずに作成することができるようになります。ここではオープンソースで企業色が薄く商用利用も可能な Eclipse BIRT (Business Intelligence and Reporting Tools、Eclipse Public License 2.0…以下 BIRT) を紹介します。

2. BIRT のインストール

BIRT の帳票デザインは Java の開発等に使う Eclipse IDE (以下、Eclipse) のプラグインです。

ビューアは構成の形態が幾つかありますが、Web アプリが必要で少量のコード実装が必要になる場合があります。使った (個別にインストールした) ツールは以下のものです。

- ① JDK 11
- ② Spring Tools Suite (STS) 4.16.1.RELEASE (Spring を含まない Eclipse でもレポート作成は可能)
- ③ Eclipse BIRT Designer 4.10.0

¹ ビジネスインテリジェンスシステム <https://ieeexplore.ieee.org/document/5392644>

BI - Report (Eclipse BIRT, Spring boot)

2.1. Spring Tools Suite (STS)

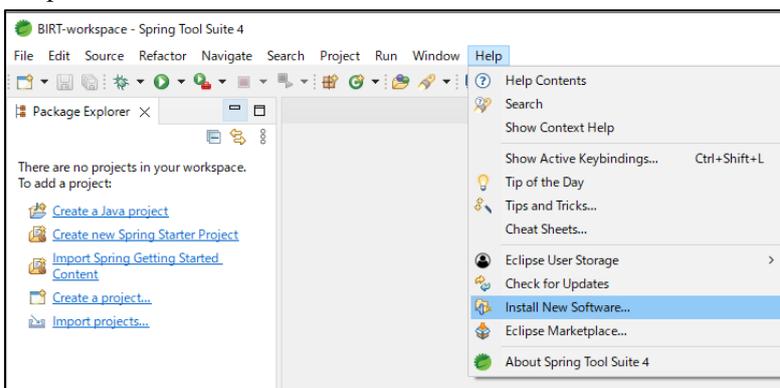
STS は Eclipse に Spring のプラグインを格納したものです。Spring Framework を使うと Web の認証機能等、ライブラリを使った機能追加が比較的簡単にできます。

Spring の公式サイト <https://spring.io/tools> からダウンロードする以外にも以下のいずれかの方法で STS を構成することができます。

- Pleiades のサイト <https://mergedoc.osdn.jp/> から（日本語化もされている）ダウンロードする
- Help メニューの「Eclipse マーケットプレイス」から「Spring Tool Suite」を追加する

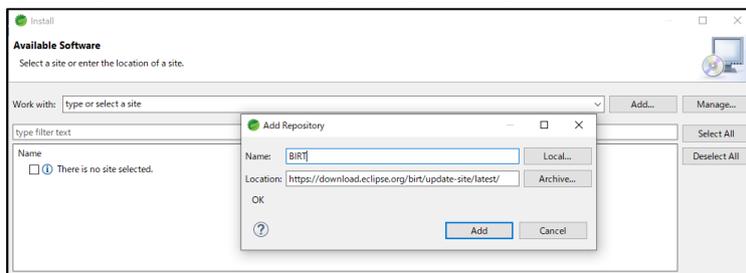
2.2. BIRT プラグイン

Help → Install New Software...

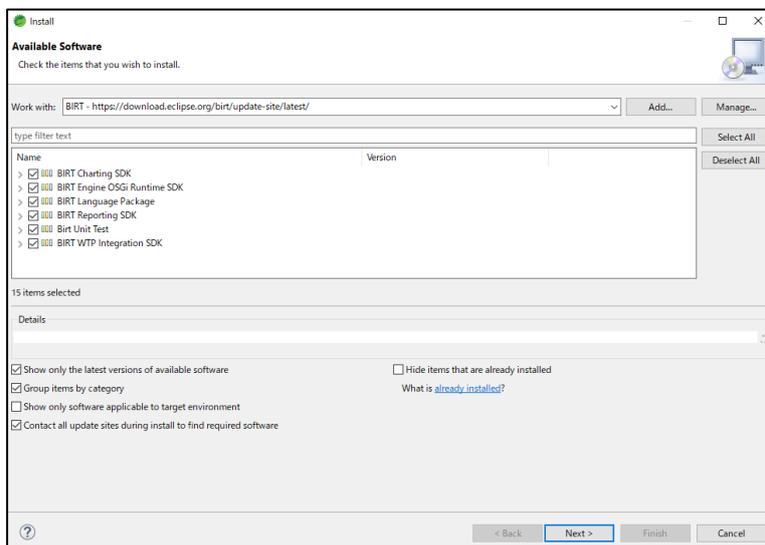


リポジトリ（インストール元サイト）追加…Location に次の URL（Name は適宜）を登録します

<https://download.eclipse.org/birt/update-site/latest/>

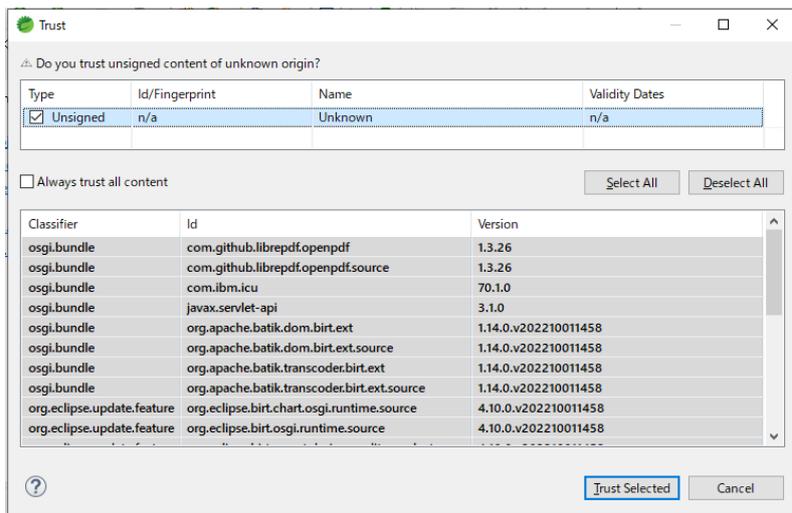
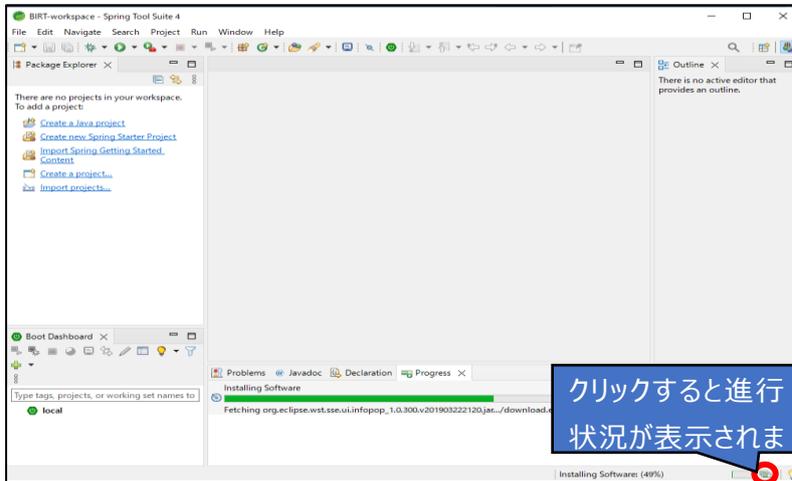
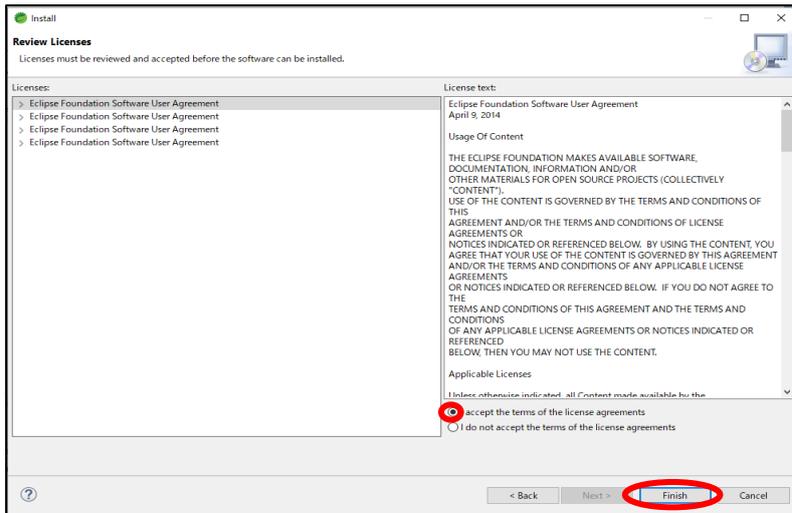


追加ソフトの選択…全てにチェックを入れて、Next ボタンを押下します



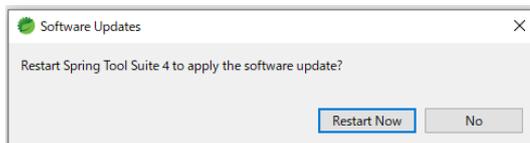
BI - Report (Eclipse BIRT, Spring boot)

「ライセンス条項に同意する」をチェックし、Finish ボタンを押下します



終盤で証明書が付属していない資材を信頼するか確認が出来ます。

⇒ 画面上段の Type 欄をチェック Trust Selected ボタンを押下 ⇒ Restart Now で再起動すると完了

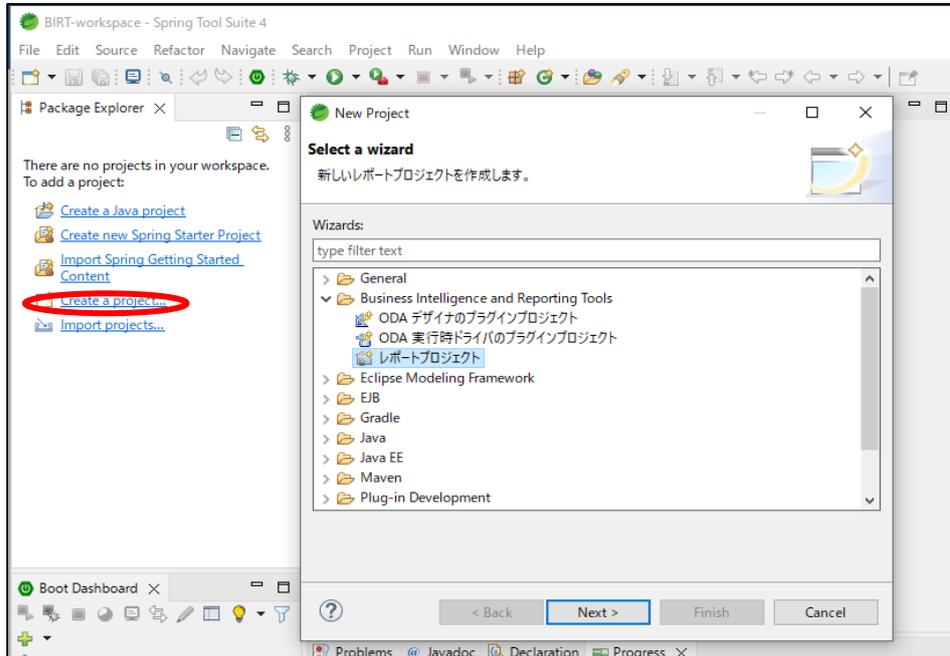


BI - Report (Eclipse BIRT, Spring boot)

3. レポートフォーマットの作成 (BIRT デザイナ)

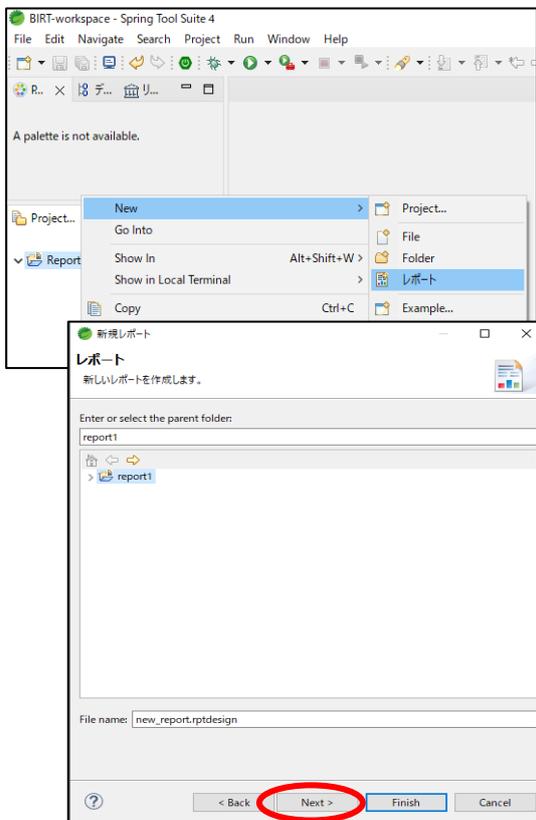
(1) レポートプロジェクトの作成

初期状態の Project Explorer で Create Project か、メニュー File → New → Project...から進みます



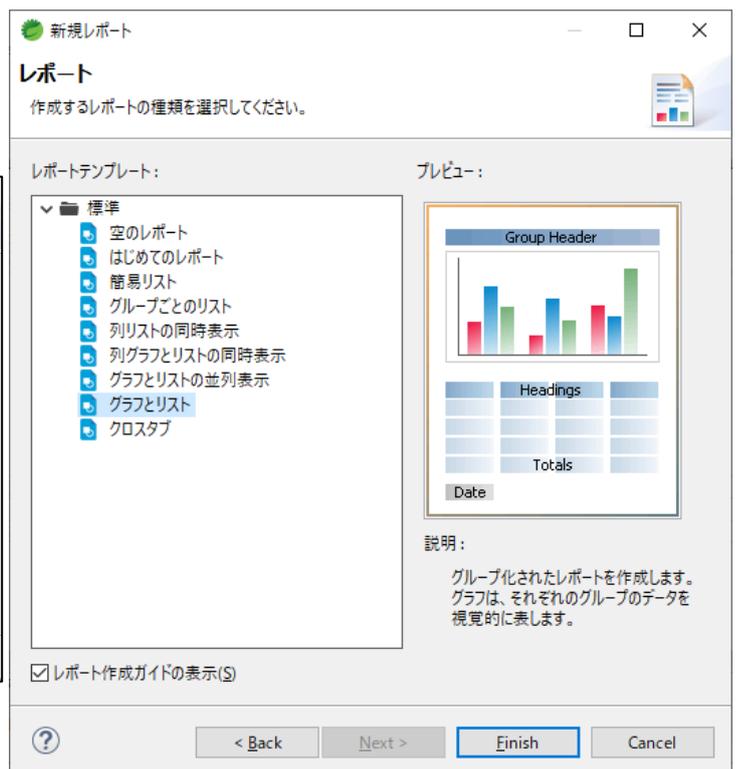
(2) 新規レポートの作成

レポートプロジェクトを右クリックし、New → レポート を選びます。



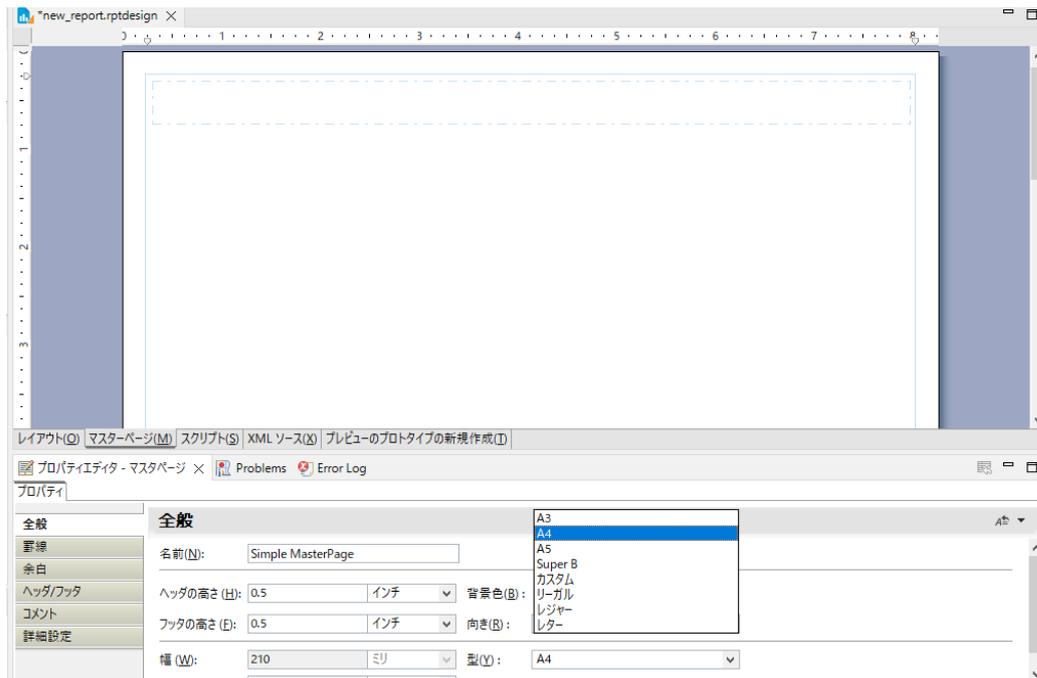
Next ボタン押下で

テンプレートが選択できます ⇒



BI - Report (Eclipse BIRT, Spring boot)

作成したレポートの「マスターページ」タブで用紙設定



4. 開発環境でのデータベース接続

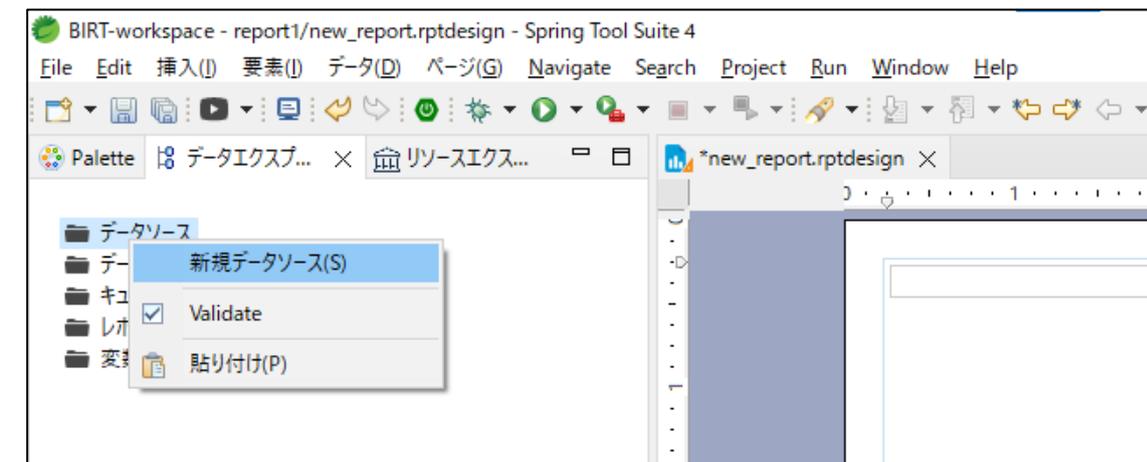
データベースと接続するための環境は帳票開発時（プレビュー）と稼働時で異なり、以下の説明はプレビューで帳票にデータベースのデータを載せるための設定です。

4.1. JDBC データソース

JDBC ドライバを用意²して STS(Eclipse)から参照できる場所に保存します。

データエクスプローラからレポート毎に使うデータを設定します。新規レポートを作成すると「データソース」が表示されるので右クリックし、新規データソースを選びます。

※ データソース等はレポートを削除すると同時に消えます（レポート間のコピー/貼り付けは可能）



² サンプルで使った RDBMS は SQLite で、ドライバは以下サイトの `sqlite-jdbc-<バージョン>.jar`
<https://github.com/xerial/sqlite-jdbc/releases>

BI - Report (Eclipse BIRT, Spring boot)

新規データソースの登録

① 初めて使うドライバは「ドライバの管理」から登録する

② 追加ボタンでドライバの場所を指定する

追加したドライバが選べるようになる

データベースの URL を指定後、接続テストで「接続完了」がでたら OK

JDBC Data Source を選び、Next ボタン押下⇒

4.2. データセットの登録

データセットを右クリックし、データセットの新規作成を行います。

BIRT-workspace - Report1/new_report.rptdesign - Spring Tool Suite 4

File Edit 挿入(I) 要素(I) データ(D) ページ(G) Navigate Search Proj

▼ データソース
レポートデータ
データセット
キュー
レポ
変数

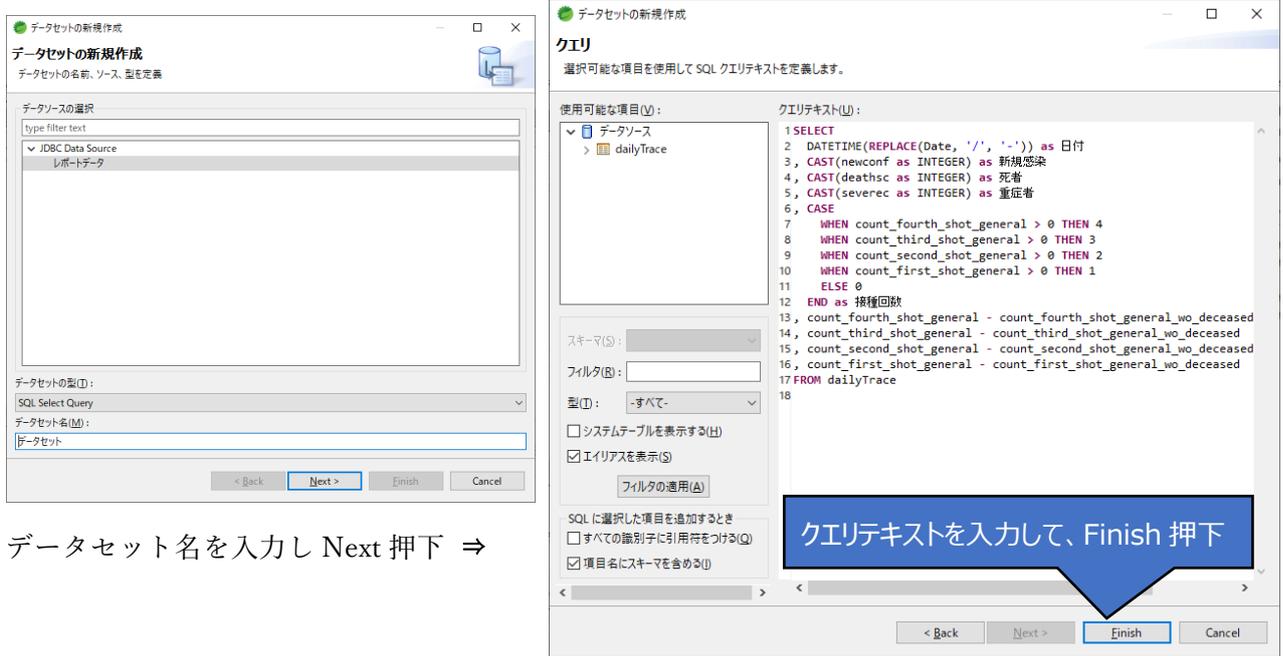
- データセットの新規作成(S)
- ジョインデータセットの新規作成(J)
- Validate
- 貼り付け(P)

BI - Report (Eclipse BIRT, Spring boot)

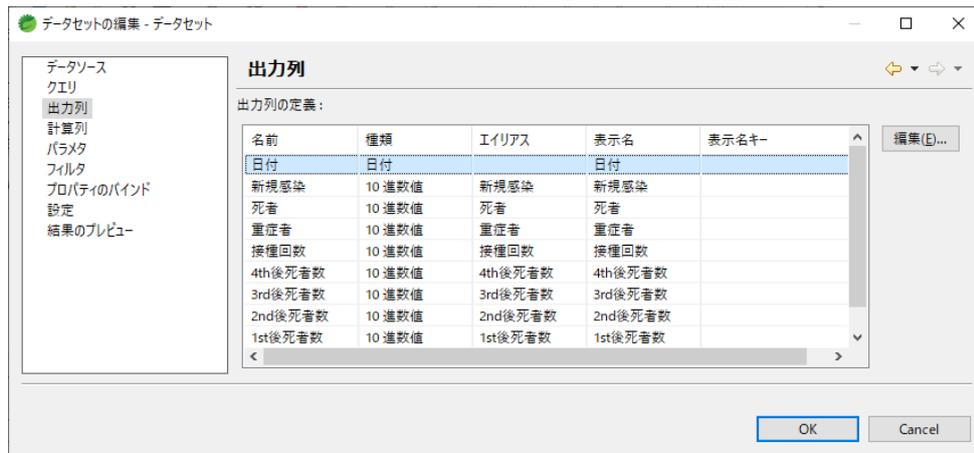
データセットの新規作成

※ クエリもレポートを削除すると消えます。必要であればテキストで保存してください

※ クエリは全角の空白が含まれないよう注意してください



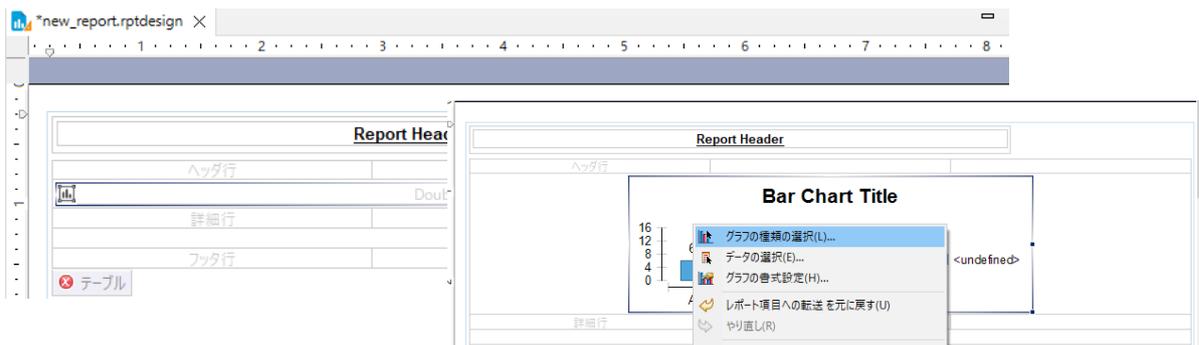
データセット名を入力し Next 押下 ⇒



4.3. レポートの編集

(1) グラフ

グラフを含めレポートに配置する要素は Palette のアイコンをレポートにドラッグして追加できます。テンプレートからのグラフは、グラフのアイコンをダブルクリックして種類を決めます



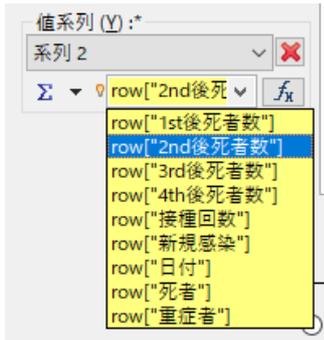
BI - Report (Eclipse BIRT, Spring boot)

【折れ線グラフ】

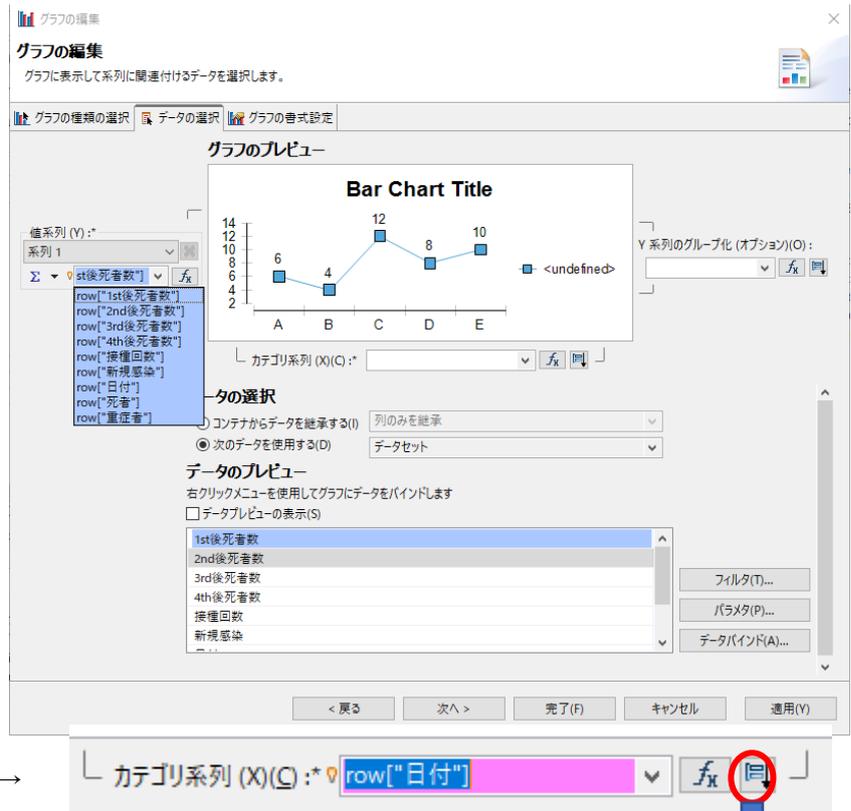
線を描く系列を作成し、



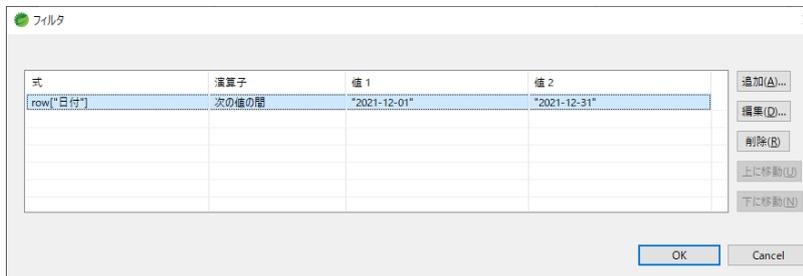
項目を追加



x 軸の項目を指定 →

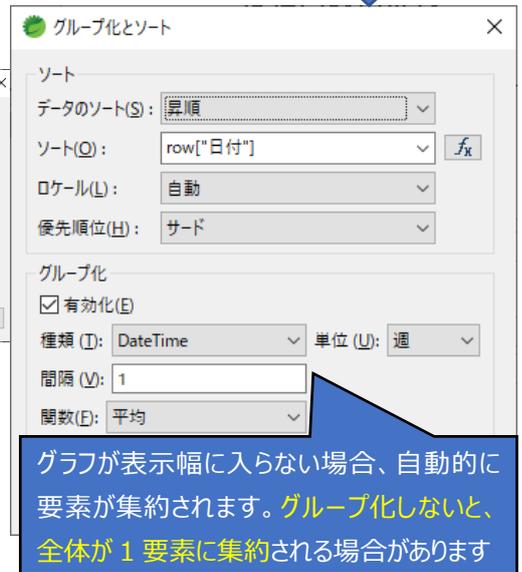


「フィルタ」からデータの選別条件を設定



<グラフの書式設定タブ>

v 系列 - タイトルでグラフ線の凡例を設定

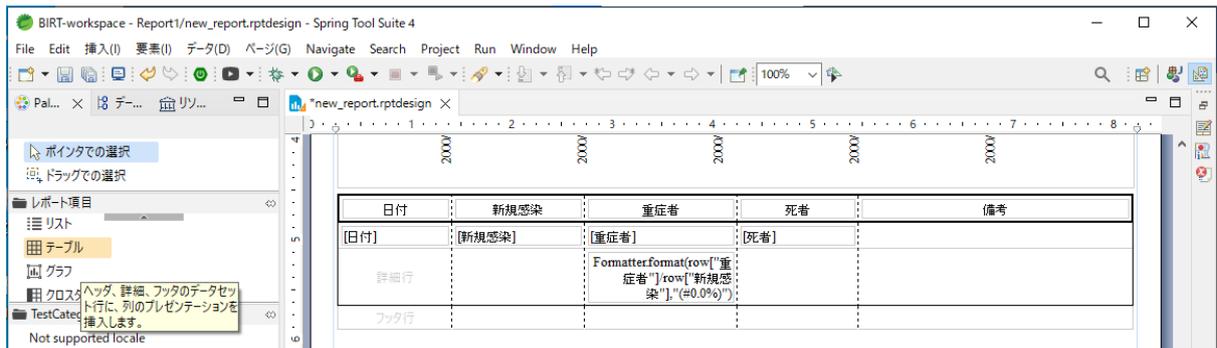


BI - Report (Eclipse BIRT, Spring boot)

(2) テーブル

データをリスト形式で表示したい場合は、テーブルを Palett からドラッグします。複数行/明細にしたり 1セルに複数項目設定することが可能です。表示するデータはデータセットで定義した項目、固定文字列、計算や文字列操作した結果等を表中に埋め込むことができます。

テーブルは HTML に似て、配置（右揃え、左揃え...）や罫線、背景色、幅等の各種の属性は階層的に持っています。例えば表上にテキストボックスを配置して文字列を右寄せで出力する場合、テキストボックスと表のセルの両方の書式が右揃えになっていないと希望通りに表示されません。



<出力>デフォルトのままですフッタ出力や改ページも行われます。

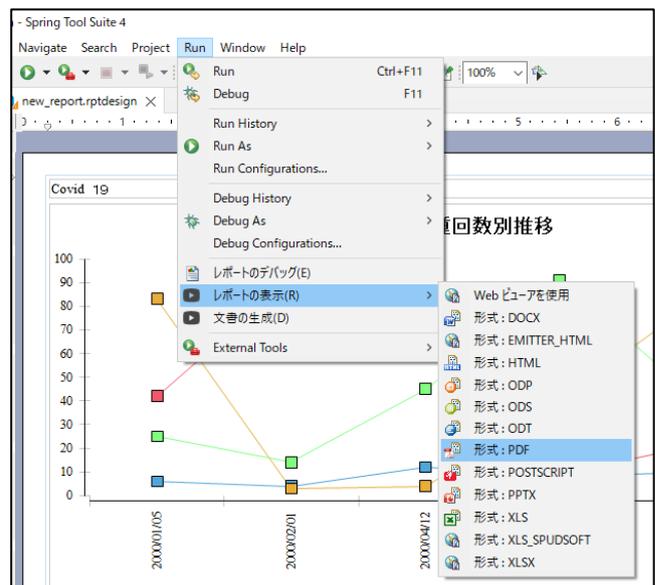
日付	新規感染	重症者	死者	備考
2021/04/12	2100	596 (28.4%)	9422	
2021/04/13	3445	608 (17.6%)	9466	

2022/11/11 13:17

日付	新規感染	重症者	死者	備考
2021/04/14	4302	63	9497	

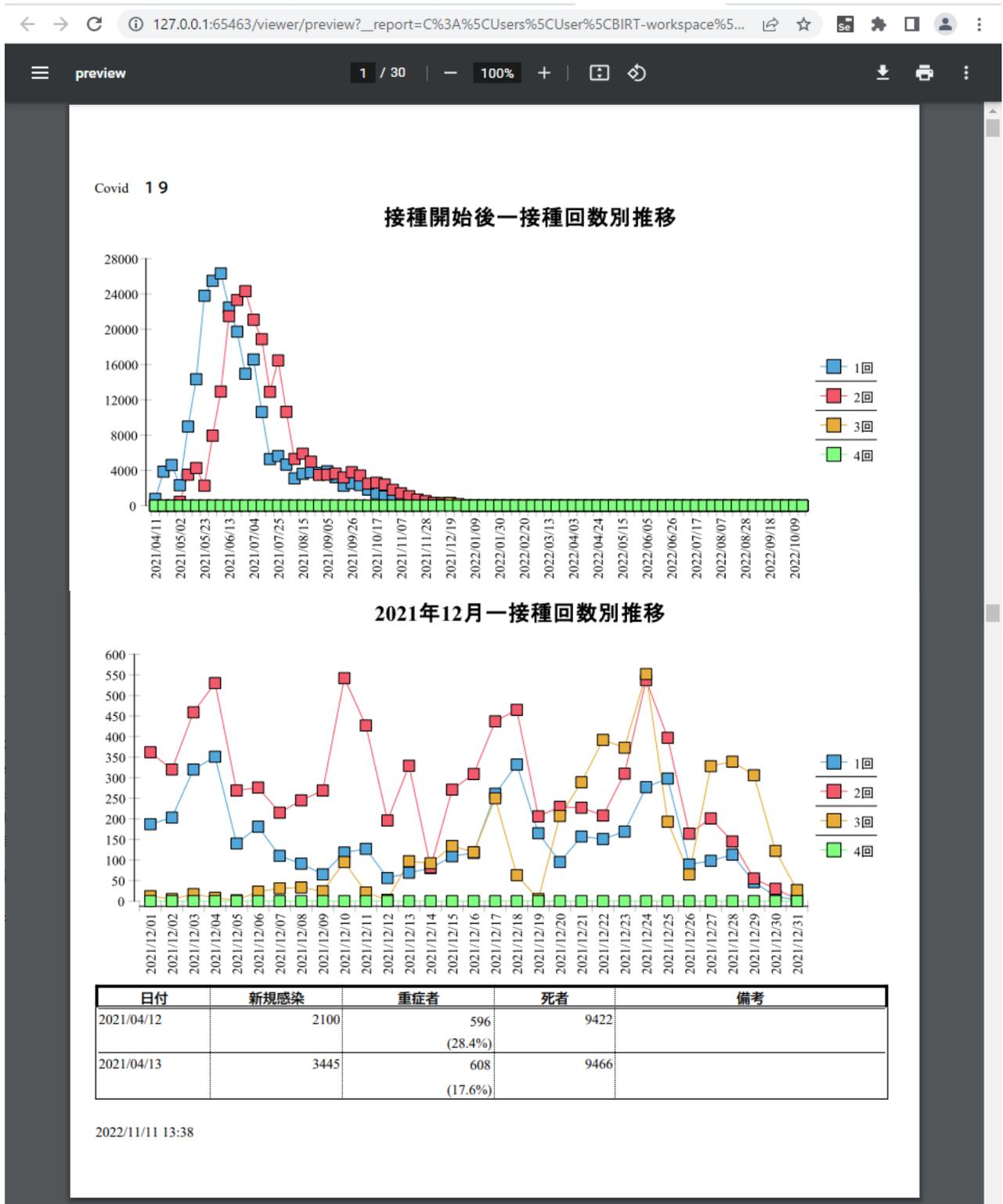
4.4. 開発中の確認 (PDF 作成の例)

メニュー Run → レポートの表示(R)



BI - Report (Eclipse BIRT, Spring boot)

<全体の出力例>



(2 頁以降は省略)

BI - Report (Eclipse BIRT, Spring boot)

5. レポートの作成 (BIRT ビューア)

デザインしたレポート (拡張子 .rptdesign) からは html や PDF が作成できます。運用時の作成の方法は BIRT ビューアを Web アプリとして起動するか、BIRT ランタイムをコマンドで起動します。

5.1. バッチ起動によるレポート作成

Eclipse のダウンロードサイト³から birt-runtime-<バージョン>.zip を取得し、zip ファイルを解凍して ReportEngine ディレクトリに入っている genReport.bat(Linux は genReport.sh)を実行します。

(1) 実行環境

コマンド実行時に JDBC ドライバが JAVA クラスパスから参照できる必要があります。簡単な方法としては ReportEngine¥lib に JDBC ドライバを含む jar ファイルを格納してください。

その他、Java 11 と BIRT 4.10 の組合せで正常に動作しています。

(2) コマンド実行

```
C:¥Users¥User>< birt-runtime のパス¥ReportEngine¥genReport.bat <レポート定義の格納パス>report.rptdesign
```

※ デフォルトは HTML で出力されます。PDF はオプションに -f pdf を指定してください

※ デフォルトの出力先はレポート定義の格納パスです。-o オプションで出力先を変更できます

5.2. Web アプリ (WebViewerExample)

Eclipse のダウンロードサイトにある birt-runtime-<バージョン>.zip には、バッチ起動ツールの ReportEngine とは別に Web アプリの WebViewerExample フォルダが入っています。このフォルダをアプリサーバに配備します。以下、オープンソースの Apache Tomcat 上で動作させる例です。

5.2.1. アプリサーバの準備

Tomcat はサポートする仕様や利用する Java バージョンの組合せでサポート中のバージョン⁴が複数ありますが、現役でサポートされているバージョンであればどれでも構いません。以下は apache-tomcat-9.0.69 を C:¥Tools に設置し、コマンドプロンプトで起動させる例です。

① ダウンロードサイトより 64-bit Windows zip を取得します

② 取得した zip ファイルを C:¥Tools¥apache-tomcat-9.0.69 ディレクトリに解凍します

③ 以下のコードを書いたサーバ起動用バッチファイル (拡張子 .bat) を作ります

```
chcp 65001
set CATALINA_HOME=C:¥Tools¥apache-tomcat-9.0.69
set JAVA_HOME=C:¥Tools¥AdoptOpenJDK¥jdk-11.0.10.9-hotspot
call C:¥Tools¥apache-tomcat-9.0.69¥bin¥startup.bat
cmd /k
```

a.コードページを UTF-8 に変更
b. 環境変数 CATALINA_HOME 設定
c. 環境変数 JAVA_HOME 設定
d. 起動スクリプト実行
e. スクリプト実行後ウィンドを残す

※ Windows Terminal 等、コマンドプロンプト以外のツールから起動する場合は文字コードに関わ

³ Eclipse Downloads <https://download.eclipse.org/birt/downloads/drops/latest/>

⁴ Apache Tomcat Versions <https://tomcat.apache.org/whichversion.html>

BI - Report (Eclipse BIRT, Spring boot)

る変更は不要になる場合があります

【サーバ起動用バッチファイルの内容】

- a. Tomcat は UTF-8 の文字コードで出力するので、コマンドプロンプトを合わせます
- b. c. Tomcat は起動時に以下の環境変数の設定が必要です
 - ・ CTALINA_HOME : Tomcat のインストール先。bin,conf,lib,webapps 等の親ディレクトリ
 - ・ JAVA_HOME : JDK のインストール先。bin,conf,lib 等の親ディレクトリ

④ アプリサーバのログ出力を UTF-8 から SJIS に変更します。

【変更資材／箇所】 **変更前の資材はコピー&リネームで必ずバックアップを取ってください**

資材 : C:\Tools\apache-tomcat-9.0.69\conf\logging.properties

```

42
43 4host-manager.org.apache.juli.AsyncFileHandler.level = FINE
44 4host-manager.org.apache.juli.AsyncFileHandler.directory = ${catalina.base}/logs
45 4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.
46 4host-manager.org.apache.juli.AsyncFileHandler.maxDays = 90
47 4host-manager.org.apache.juli.AsyncFileHandler.encoding = UTF-8
48
49 java.util.logging.ConsoleHandler.level = FINE
50 java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
51 java.util.logging.ConsoleHandler.encoding = SJIS
52

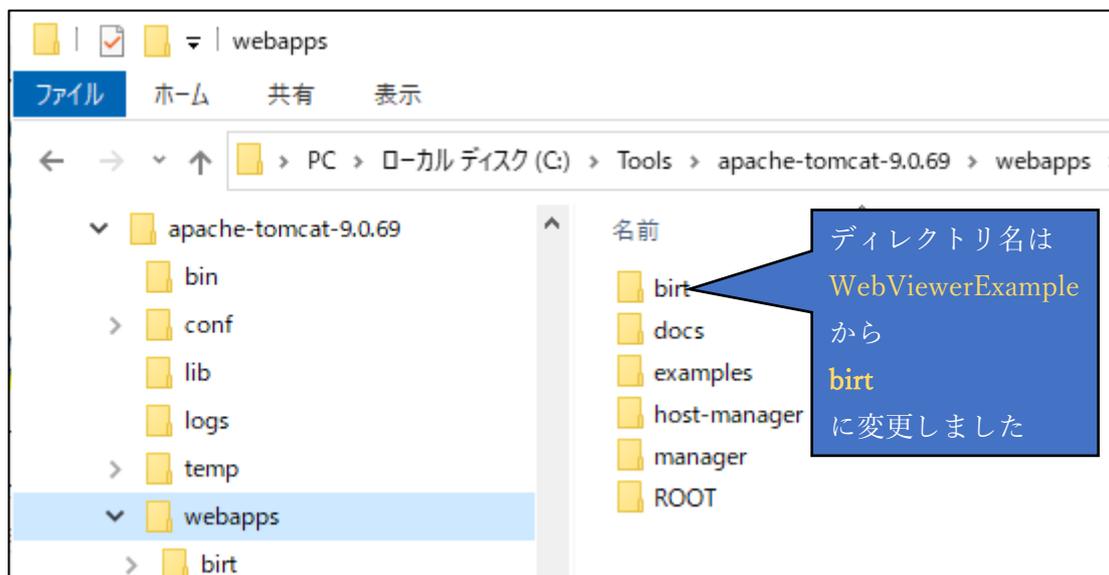
```

※ Tomcat の起動スクリプトは startup.bat ⇒ catalina.bat ⇒ java /Tomcat の起動と進み、サーバ起動用バッチファイルとは別のプロセスになるためコードページの切り替えが効きません。

5.2.2. BIRT ビューアの配備

birt-runtime-<バージョン>.zip に含まれている WebViewerExample を

C:\Tools\apache-tomcat-9.0.69\webapps に格納します。このとき webapps 内のディレクトリ名が Web アプリ名になるので、分かり易くかつ使い易い名前に変更します。



BI - Report (Eclipse BIRT, Spring boot)

● BIRT ビューアの初期画面表示

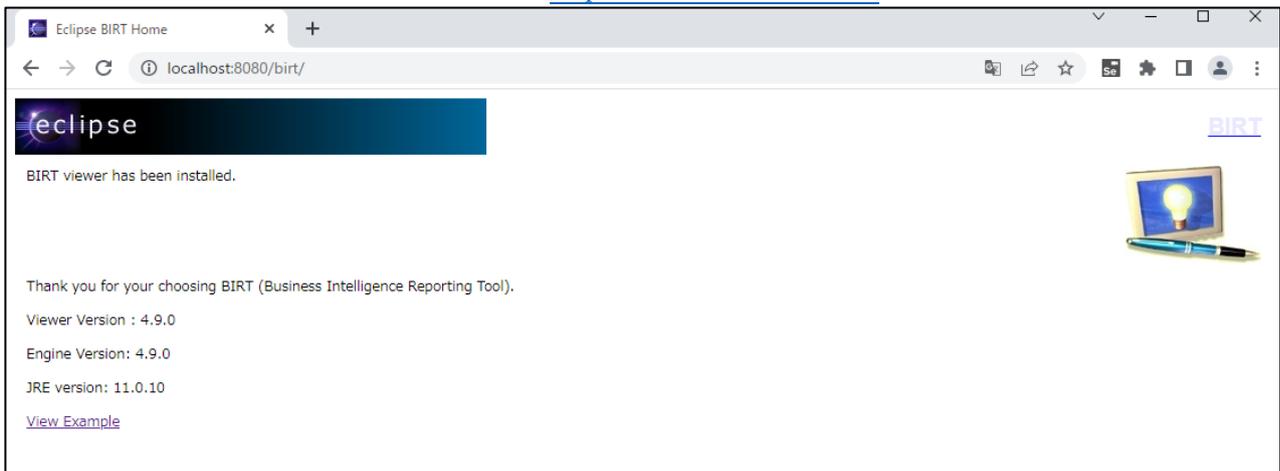
サーバ起動用バッチファイルを起動すると別画面が開いて birt の Web アプリケーション配備が完了したと、サーバが起動した旨のログが出力されます。

```

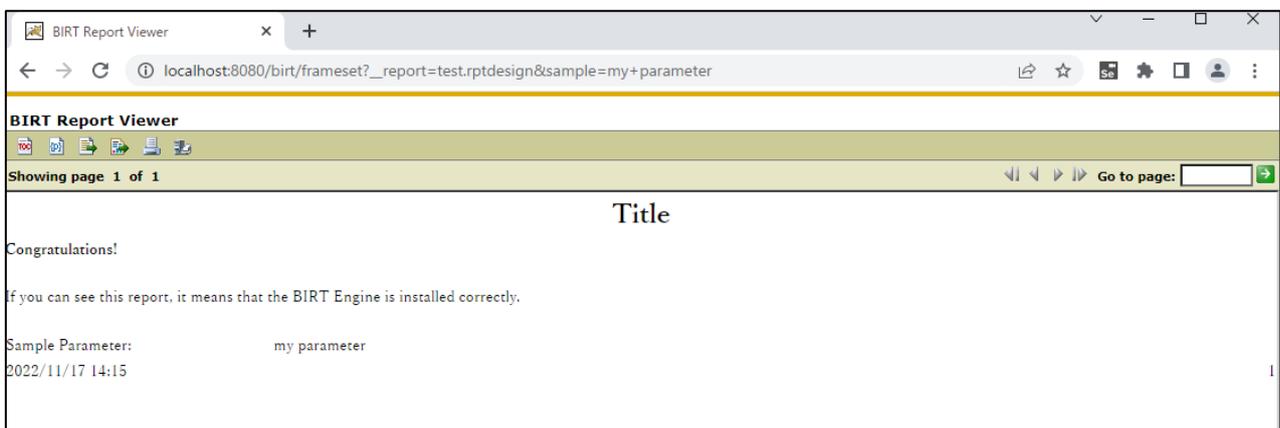
選択Tomcat
17-Nov-2022 13:44:38.218 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\birt] を配備します
17-Nov-2022 13:44:52.704 情報 [main] org.apache.jasper.servlet.TldScanner.scanJars 少なくとも1つのJARをTLDをスキャンしましたがTLDが含まれていませんでした。スキャンしたもののTLDが見つからなかったJARの完全なリストについては、このログのデバッグログを有効にしてください。スキャン中に不要なJARをスキップすると、起動時間とJSPのコンパイル時間が短縮されます。
17-Nov-2022 13:44:55.682 警告 [main] org.apache.catalina.util.SessionIdGeneratorBase.createSecureRandom セッション ID を生成するための SecureRandom インスタンスの作成に [264] ミリ秒かかりました。アルゴリズムは [SHA1PRNG] です。
17-Nov-2022 13:44:55.722 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\birt] の Web アプリケーションの配備は [17,505] ミリ秒で完了しました。
17-Nov-2022 13:44:55.723 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\docs] を配備します
17-Nov-2022 13:44:55.763 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\docs] の Web アプリケーションの配備は [41] ミリ秒で完了しました。
17-Nov-2022 13:44:55.763 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\examples] を配備します
17-Nov-2022 13:44:55.237 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\examples] の Web アプリケーションの配備は [474] ミリ秒で完了しました。
17-Nov-2022 13:44:56.238 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\host-manager] を配備します
17-Nov-2022 13:44:56.295 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\host-manager] の Web アプリケーションの配備は [56] ミリ秒で完了しました。
17-Nov-2022 13:44:56.296 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\manager] を配備します
17-Nov-2022 13:44:56.355 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\manager] の Web アプリケーションの配備は [59] ミリ秒で完了しました。
17-Nov-2022 13:44:56.356 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory Webアプリケーションディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\ROOT] を配備します
17-Nov-2022 13:44:56.400 情報 [main] org.apache.catalina.startup.HostConfig.deployDirectory ディレクトリ [C:\%Tools%\apache-tomcat-9.0.69\webapps\ROOT] の Web アプリケーションの配備は [44] ミリ秒で完了しました。
17-Nov-2022 13:44:56.411 情報 [main] org.apache.coyote.AbstractProtocol.start プロトコルハンドラー ["http-nio-8080"] を開始しました。
17-Nov-2022 13:44:56.447 情報 [main] org.apache.catalina.startup.Catalina.start サーバーの起動 [18370] ミリ秒

```

サーバの起動が表示されたらブラウザで <http://localhost:8080/birt> を開きます。



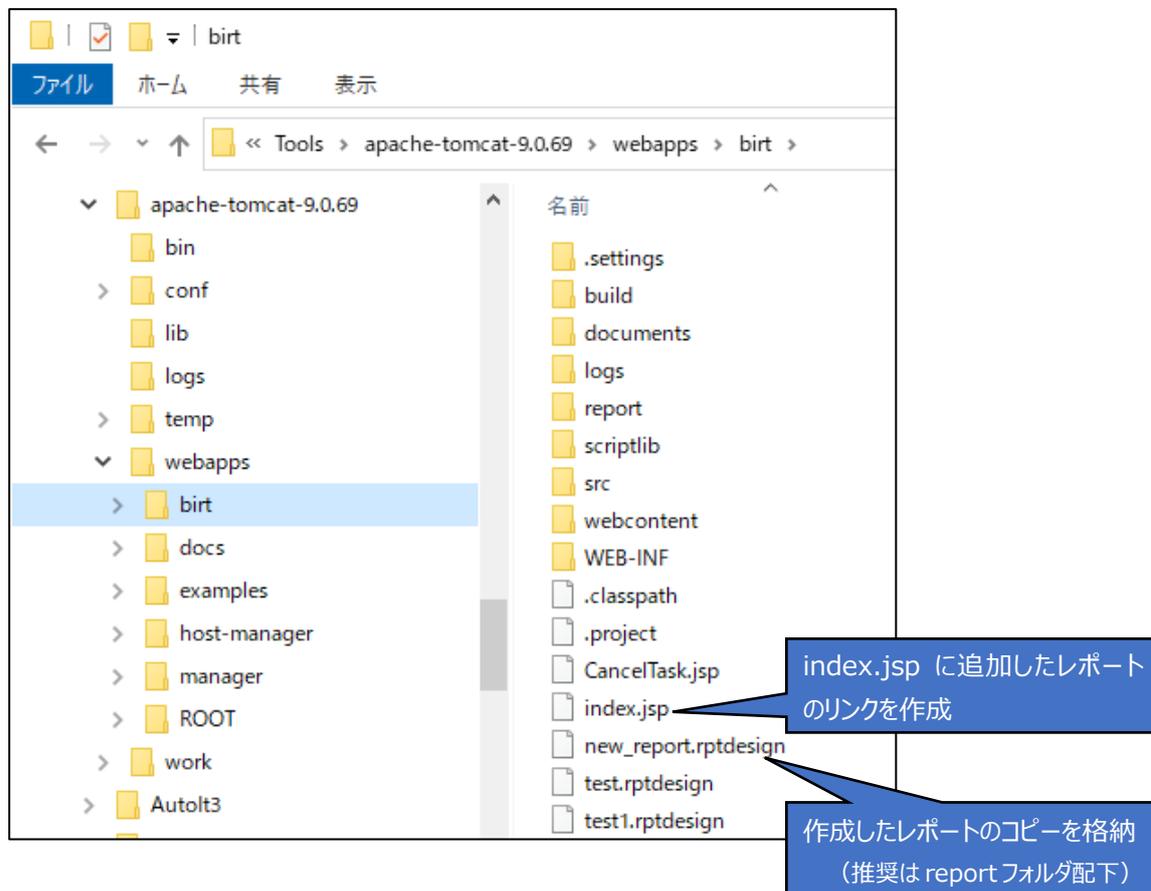
一番下のリンク View Example をクリックして以下のサンプル画面が表示されたら完成です。



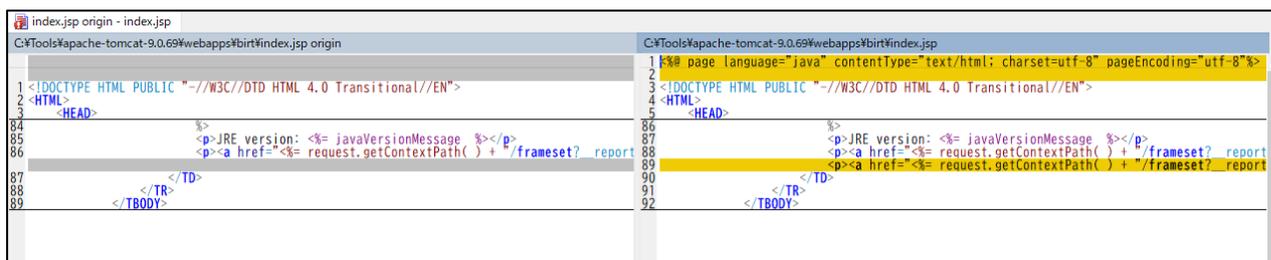
BI - Report (Eclipse BIRT, Spring boot)

5.2.3. 新規作成したレポートの追加

BIRT デザイナーで作成したレポート (拡張子.rptdesign) のファイルを webapps/birt ディレクトリに格納し、index.jsp にリンクを作ります。



(1) index.jsp の改修点



① 原状だと日本語が表示できないので page ディレクティブで文字コード utf-8 を指定します。

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
```

② 新しいレポートを表示するためのリンクを追加します。

【追加例】

```
<p><a href="<%= request.getContextPath() + "/frameset?_report=new_report.rptdesign&_format=pdf" %>">Covid19 近況</a>
```

【パラメータの内容】

__report: BIRT デザイナーで作成したレポートのファイル名

__format: html (デフォルト) または pdf を指定します

BI - Report (Eclipse BIRT, Spring boot)

(2) JDBC ドライバの追加

新規レポートの JDBC データソースで使ったドライバをクラスパスに配置します。汎用的に使うのであれば tomcat の lib (webapps と並列のディレクトリ)、レポート固有で使うのであれば以下の場所に jar ファイルを格納して Tomcat を再起動します。

(Windows のパス表記) webapps¥birt¥WEB-INF¥lib

5.2.4. アプリサーバ (Tomcat) の再起動

以上の操作を行った後に Tomcat の停止スクリプトを実行し、もう一度サーバ起動バッチファイルを実行します。

(停止スクリプト) C:¥Tools¥apache-tomcat-9.0.69¥bin¥shutdown.bat

5.2.5. 新規レポートの表示

サーバを再起動後にブラウザで <http://localhost:8080/birt> を開くと以下のように表示されます。

The screenshot shows the Eclipse BIRT viewer interface. At the top, it displays the Eclipse logo and the message "BIRT viewer has been installed." Below this, it says "Thank you for your choosing BIRT (Business Intelligence Reporting Tool)." and provides version information: "Viewer Version : 4.9.0", "Engine Version: 4.9.0", and "JRE version: 11.0.10". There are links for "View Example" and "Covid19 近況". A blue callout box points to the "Covid19 近況" link, stating: "index.jsp に追加したリンクをクリックすると、パラメータで指定したレポートが表示されます".

The main content area displays a report titled "Covid 19" with a subtitle "接種開始後—接種回数別推移". The chart shows the number of vaccinations (Y-axis, 0 to 28000) over time (X-axis, from 2021.04.11 to 2022.10.09). The data is categorized by the number of doses: 1回 (blue), 2回 (red), 3回 (orange), and 4回 (green). The chart shows a significant peak in 2021, followed by a decline and then a period of low activity in 2022. A secondary title "2021年12月—接種回数別推移" is visible at the bottom of the chart area.

BI - Report (Eclipse BIRT, Spring boot)

6. セキュリティの追加

前項のように birt-runtime に同梱されている WebViewerExample のサーブレットで自作のレポートを作成することができますが、このままでは不特定多数のネットワーク接続に対して情報を開示してしまいます。営業秘密や部内限り等の扱いは利用者の識別・認証やアクセス制御が必要になります。

6.1. Web アプリの構成

BIRT は継続的にバージョンアップされてきて今後の予定も出ています。BIRT ビューアも簡単に置き換えができ、レポートの追加や更新がしやすいように構成する例をあげます。

- ・ビューアのオリジナル部分を改変しない（できればフォルダごと入替える）
- ・ビューアに直接リクエストを出さず、認証とアクセス制御（簡易なもの）を挟むようにする

【画面構成】

The diagram illustrates the web application structure for BI-Report. It consists of several components:

- Login Page:** A page titled "BIRTレポート出力 利用者確認" with input fields for "利用者ID" and "Password", and a "Sign In" button. An arrow labeled "一般ユーザ" (General User) points to the "出力可能なレポート by user" page.
- Report Listing Pages:** Two pages titled "出力可能なレポート by user" and "出力可能なレポート by user2". The first page shows reports for "[ROLE_USER]" (General User向けレポート, 一般ユーザ向けレポート (続報)). The second page shows reports for "[ROLE_ADMIN, ROLE_USER]" (Covid19 近況, 一般ユーザ向けレポート, 一般ユーザ向けレポート (続報)). A red dashed circle highlights "Covid19 近況" with an arrow labeled "特権ユーザ" (Special User).
- BIRT Viewer:** A page titled "BIRT ビューア (レポート定義体以外は既存のまま)" displaying a report titled "Covid 19" with a chart showing "接種開始後—接種回数別推移".
- Annotations:** A large blue bracket on the right side of the report listing pages is labeled "新たに作る部分" (Newly added parts).

BI - Report (Eclipse BIRT, Spring boot)

6.2. STS を使ったプロジェクトの作成

Spring Boot を使い、Tomcat に認証を行うアプリと BIRT ビューアの二つの Web アプリを配備した実行形式の jar にします (BIRT ビューアは jar の起動中に動的に配備します)。以下の操作は全て STS で行います。

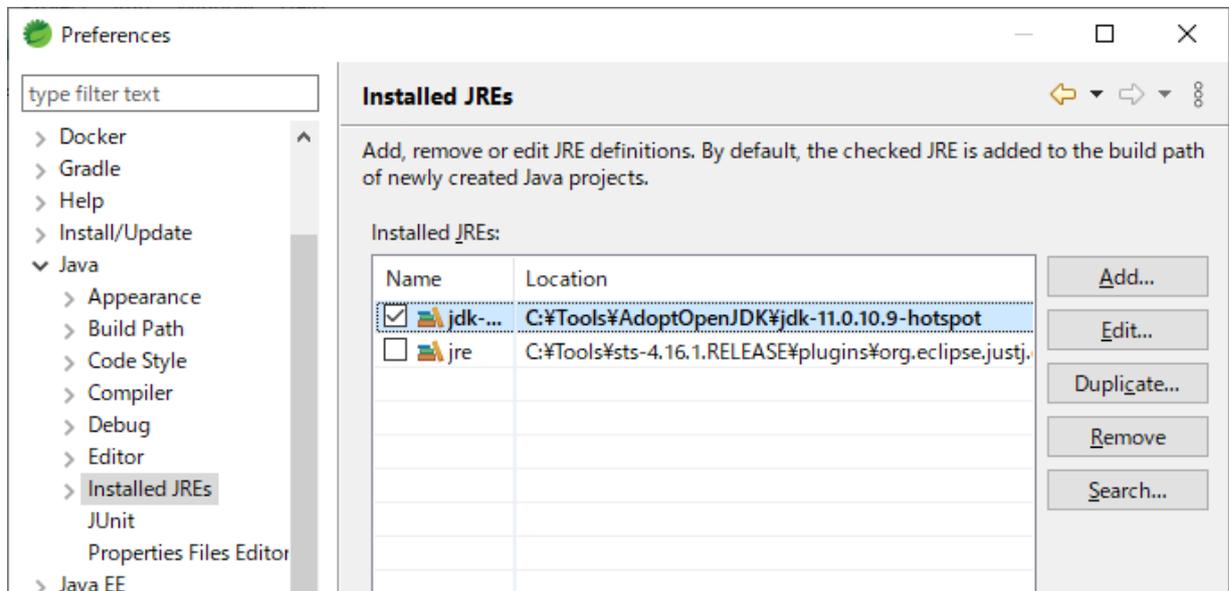
6.2.1. Web アプリ開発用の Java 設定

STS 4.16 は JDK を同梱しておりデフォルトでは Java17 のコンパイラが使われます。これ以外、特に下位バージョンの実行環境では動作しない場合があるので目的の実行環境に合わせておきます。

(以下、日本語化していない STS を使い Java11 環境用の設定をする例)

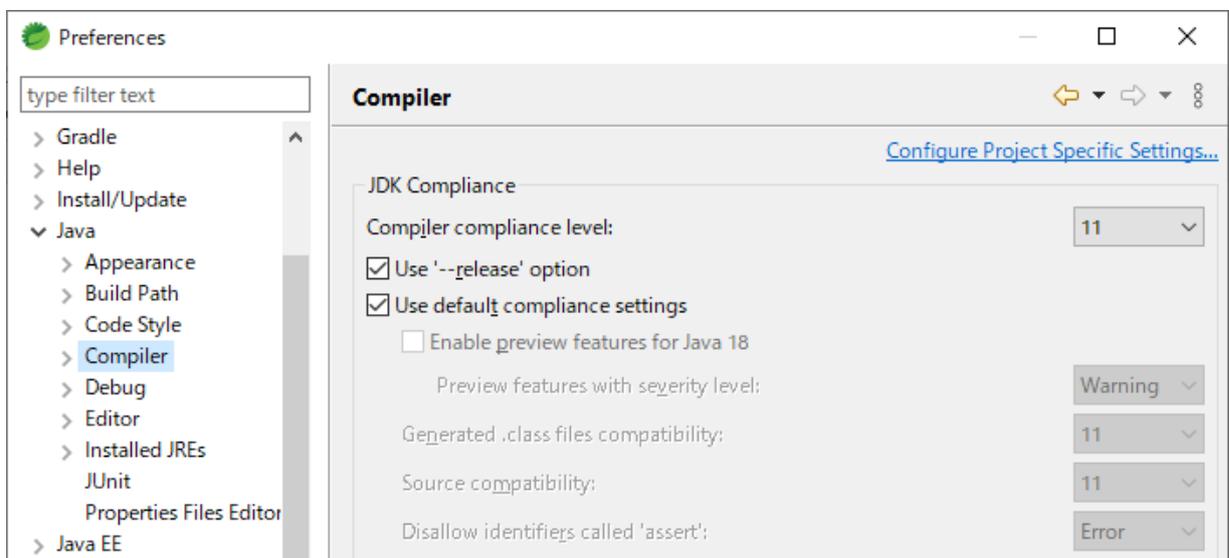
(1) JRE

メニューの Window → Preferences から Java > Installed JREs Add... ボタンで別途インストール済の JDK を追加し、デフォルトに設定します。



(2) コンパイラ

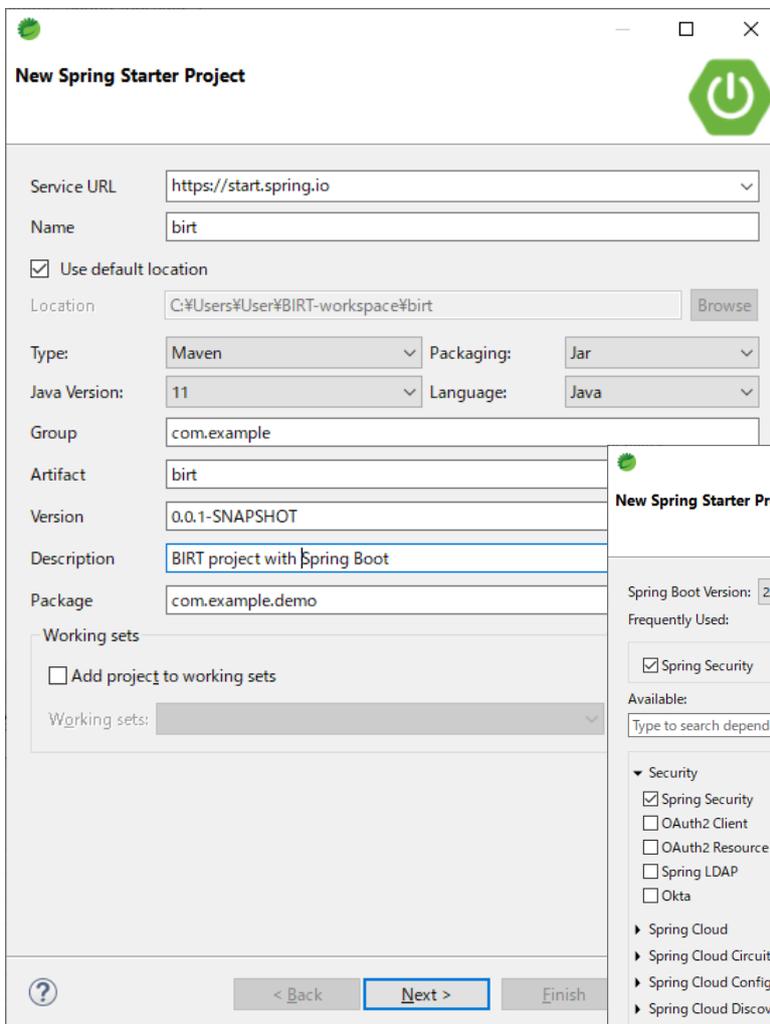
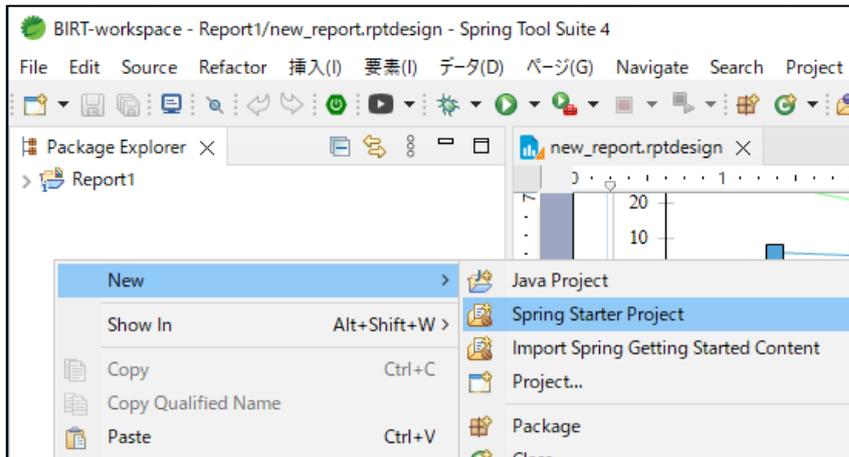
前項と同じく Java > Compiler の Compiler compliance level: を 11 にします。



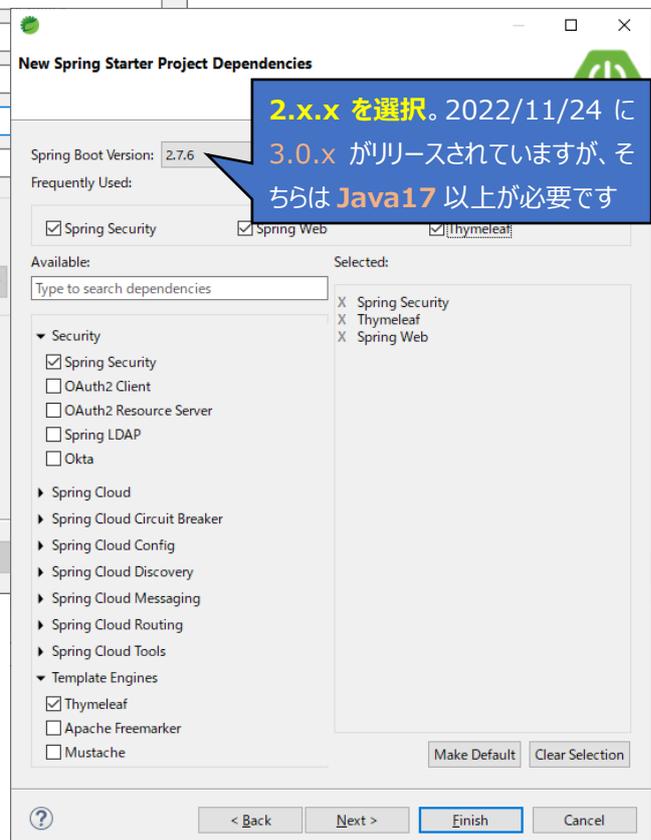
BI - Report (Eclipse BIRT, Spring boot)

6.2.2. 新規プロジェクト

Package Explorer を右クリックし、New → Spring Starter Project を選択します。



- Name: birt
(プロジェクト名: 適宜変更可)
- Type: Maven
- Java Version: 11
- その他の項目適宜
(意味が解らない場合はそのまま)



- Next> で以下のライブラリを選択
- ▶ Security > Spring Security
 - ▶ Template Engines > Thymeleaf
 - ▶ Web > Spring Web
- Finish ボタン押下

BI - Report (Eclipse BIRT, Spring boot)

【pom.xml】

プロジェクトの作成で作られるファイルです。

但し 44 行目~60 行目は Spring Starter Project で取り込めないため手作業で追加した行です。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd>
5 <modelVersion>4.0.0</modelVersion>
6 <parent>
7 <groupId>org.springframework.boot</groupId>
8 <artifactId>spring-boot-starter-parent</artifactId>
9 <version>2.7.6</version>
10 <relativePath/> <!-- lookup parent from repository -->
11 </parent>
12 <groupId>com.example</groupId>
13 <artifactId>birt</artifactId>
14 <version>0.0.1-SNAPSHOT</version>
15 <name>birt</name>
16 <description>BIRT project with Spring Boot</description>
17 <properties>
18 <java.version>11</java.version>
19 </properties>
20 <dependencies>
21 <dependency>
22 <groupId>org.springframework.boot</groupId>
23 <artifactId>spring-boot-starter-thymeleaf</artifactId>
24 </dependency>
25 <dependency>
26 <groupId>org.springframework.boot</groupId>
27 <artifactId>spring-boot-starter-web</artifactId>
28 </dependency>
29 <dependency>
30 <groupId>org.springframework.boot</groupId>
31 <artifactId>spring-boot-starter-security</artifactId>
32 </dependency>
33 <dependency>
34 <groupId>org.springframework.security</groupId>
35 <artifactId>spring-security-test</artifactId>
36 <scope>test</scope>
37 </dependency>
38 <dependency>
39 <groupId>org.springframework.boot</groupId>
40 <artifactId>spring-boot-starter-test</artifactId>
41 <scope>test</scope>
42 </dependency>
43
44 <!-- 以下、Spring Securityとthymeleafを組み合わせて使うための設定 Start -->
45 <dependency>
46 <groupId>org.thymeleaf.extras</groupId>
47 <artifactId>thymeleaf-extras-springsecurity5</artifactId>
48 </dependency>
49 <!-- Spring Securityとthymeleafを組み合わせて使うための設定 End -->
50
```

BI - Report (Eclipse BIRT, Spring boot)

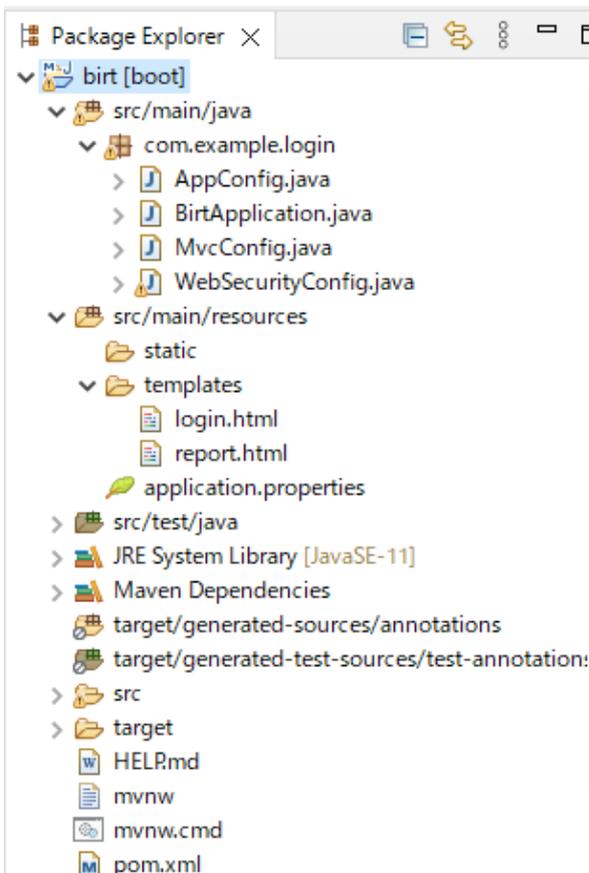
```

51     <!-- 以下、Spring Bootの組み込みTomcatでJSPを使うための設定 Start -->
52     <dependency>
53         <groupId>org.apache.tomcat.embed</groupId>
54         <artifactId>tomcat-embed-jasper</artifactId>
55     </dependency>
56     <dependency>
57         <groupId>javax.servlet</groupId>
58         <artifactId>jstl</artifactId>
59     </dependency>
60     <!-- Spring Bootの組み込みTomcatでJSPを使うための設定 End -->
61 </dependencies>
62
63 <build>
64     <plugins>
65         <plugin>
66             <groupId>org.springframework.boot</groupId>
67             <artifactId>spring-boot-maven-plugin</artifactId>
68         </plugin>
69     </plugins>
70 </build>
71
72 </project>

```

※pom.xml は Maven が参照／更新しますが、手作業で更新すると保存時に STS が必要なライブラリのダウンロードを行います

6.2.3. 作成する資材



src/main/java/

com.example.login パッケージ：以下の構成用 Java

- ①AppConfig…BIET ビューアの配備とログ出力先
- ②BirtApplication…main（自動生成のまま）
- ③MvcConfig…リクエストと表示 html の紐付け
- ④WebSecurityConfig…Spring Security 認証関連

src/main/resources

template フォルダ：thymeleaf タグを使う html

- ①login…認証画面
- ②report…利用者の役割に応じたレポートを表示

Application.properties：birtWeb アプリ配備元 path

src/test/java

BirtApplicationTest…自動生成されたテストクラス

※src/main/java/の BirtApplication とパッケージ名を一致させておかないと **ビルドの工程でエラー**が発生します

BI - Report (Eclipse BIRT, Spring boot)

src/main/java/com.example.login

① 【AppConfig.java】 クラス名に関係無く@ConfigurationによりSpringがこのクラスを処理します

```

1 package com.example.login;
2 /*
3  * BIRTビューア (webapps)をSpring Bootの環境に取り込むための環境設定
4  */
5 import java.io.File;
6
7 import org.apache.catalina.Context;
8 import org.apache.catalina.startup.Tomcat;
9 import org.springframework.beans.factory.annotation.Value;
10 import org.springframework.boot.autoconfigure.condition.ConditionalOnProperty;
11 import org.springframework.boot.web.embedded.tomcat.TomcatServletWebServerFactory;
12 import org.springframework.boot.web.embedded.tomcat.TomcatWebServer;
13 import org.springframework.context.annotation.Bean;
14 import org.springframework.context.annotation.Configuration;
15
16 @Configuration
17 public class AppConfig {
18
19     @Bean
20     @ConditionalOnProperty(name = "external.webapp.dir") //application.propertiesにexternal.webappがあ
21     ったら以下のメソッドを使う
22     public TomcatServletWebServerFactory servletContainerFactory(
23         @Value("${external.webapp.dir}") String path,
24         @Value("${external.webapp.context}") String contextPath
25     ) {
26         return new TomcatServletWebServerFactory() {
27
28             @Override
29             protected TomcatWebServer getTomcatWebServer(Tomcat tomcat) {
30                 //Springbootからだとorg.apache.commons.discovery.DiscoveryExceptionが発生する
31                 //対策としてシステムプロパティでLogFactoryを割り当てます。
32                 System.setProperty("org.apache.commons.logging.LogFactory",
33                 "org.apache.commons.logging.impl.LogFactoryImpl");
34
35                 logger.info("-----");
36                 logger.info("configuring app with basedir: " + new File(path).getAbsolutePath());
37
38                 Context ctx = tomcat.addWebapp(contextPath, new File(path).getAbsolutePath());
39                 ctx.setParentClassLoader(getClass().getClassLoader());
40
41                 logger.info("Context : " + ctx.toString());
42                 logger.info("-----");
43
44                 return super.getTomcatWebServer(tomcat);
45             }
46         };
47     }
48 }

```

BI - Report (Eclipse BIRT, Spring boot)

② 【BirtApplication.java】 自動生成のまま

```
1 package com.example.login;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class BirtApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(BirtApplication.class, args);
11     }
12
13 }
```

③ 【MvcConfig.java】

```
1 package com.example.login;
2
3 import org.springframework.context.annotation.Configuration;
4 import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
5 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
6
7 @Configuration
8 public class MvcConfig implements WebMvcConfigurer {
9
10     public void addViewControllers(ViewControllerRegistry registry) {
11         registry.addViewController("/").setViewName("login");
12         registry.addViewController("/login").setViewName("login");
13         registry.addViewController("/report").setViewName("report");
14     }
15 }
```

④ 【WebSecurityConfig.java】

```
1 package com.example.login;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
7 import org.springframework.security.core.userdetails.User;
8 import org.springframework.security.core.userdetails.UserDetails;
9 import org.springframework.security.core.userdetails.UserDetailsService;
10 import org.springframework.security.provisioning.InMemoryUserDetailsManager;
11 import org.springframework.security.web.SecurityFilterChain;
12
13 @Configuration
14 @EnableWebSecurity
15 public class WebSecurityConfig {
16
17     @Bean
18     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
19         http
20             .authorizeHttpRequests((requests) -> requests
21                 .antMatchers("/").permitAll() //ルートのURL パスだけ認証無のリクエストを許す
22                 .anyRequest().authenticated()
```

BI - Report (Eclipse BIRT, Spring boot)

```

23     )
24     .formLogin((form) -> form
25         .loginPage("/login")
26         .defaultSuccessUrl("/report", true)
27         .permitAll()
28     )
29     .logout((logout) -> logout.permitAll());
30
31     return http.build();
32 }
33
34 @Bean
35 public UserDetailsService userDetailsService() {
36     UserDetails user =
37         User.withDefaultPasswordEncoder() //デモ専用の簡易実装です。(詳細は Spring 公式…
38             .username("user")
39             .password("password")
40             .roles("USER")
41             .build();
42     UserDetails user2 =
43         User.withDefaultPasswordEncoder() //デモ専用の簡易実装です。(詳細は Spring 公式…
44             .username("user2")
45             .password("password")
46             .roles("ADMIN", "USER")
47             .build();
48
49     return new InMemoryUserDetailsManager(user, user2);
50 }
51 }

```

src/main/resources/templates

① 【login.html】

```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
3     xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity5">
4 <head>
5     <title>BIRT レポート出力 利用者確認 </title>
6 </head>
7 <body>
8     <div th:if="${param.error}">
9         利用者 ID か password が無効です.
10    </div>
11    <div th:if="${param.logout}">
12        ログアウトしました.
13    </div>
14    <form th:action="@{/login}" method="post">
15        <div><label> 利用者 ID : <input type="text" name="username"/> </label></div>
16        <div><label> Password: <input type="password" name="password"/> </label></div>
17        <div><input type="submit" value="Sign In"/></div>
18    </form>
19 </body>
20 </html>
21

```

BI - Report (Eclipse BIRT, Spring boot)

② 【report.html】

```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="https://www.thymeleaf.org"
3     xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity5">
4     <head>
5         <title>レポート</title>
6     </head>
7     <body>
8         <h1 th:inline="text">出力可能なレポート by
9     [[${#httpServletRequest.remoteUser}]]</h1>
10        <span sec:authentication="principal.authorities"></span>
11
12        <div sec:authorize="hasRole('ADMIN')">
13            <p><a href=birtviewer/frameset?__report=report/new_report.rptdesign&__format=pdf>Covid19 近況</a></p>
14        </div>
15        <div sec:authorize="hasRole('USER')">
16            <p><a href=birtviewer/frameset?__report=test.rptdesign&__format=html>一般ユーザー向けレポート</a></p>
17            <p><a href=birtviewer/frameset?__report=test1.rptdesign&__format=html>一般ユーザー向けレポート (続報)
18        </a></p>
19        </div>
20
21        <form th:action="@{/logout}" method="post">
22            <input type="submit" value="Sign Out"/>
23        </form>
24    </body>
25 </html>

```

src/main/resources/

【application.properties】

```

1 #埋込 Tomcat のログ出力先 (デフォルト) ...C:\Users\利用者 I D\AppData\Local\Temp\tomcat.8080.n-数字 18 桁
2 -n
3 server.tomcat.accesslog.enabled=true
4
5 external.webapp.dir=/C:/Users/User/Desktop/birtviewer
6 external.webapp.context=/birtviewer

```

※application.properties の 4 行目は WEB-INF、webcontent ディレクトリの上の階層を指定 (起動後もこのディレクトリの資材が参照されます)。他の資材は上記コード例のままで動作します

6.2.4. STS からの起動

pom.xml と application.properties を修正し、その他の資材をコピーすると STS から起動できます。

STS 画面左下 (初期) Boot Dashboard からプロジェクト(birt)を選んで起動します

Console にポート番号 8080 で Tomcat 開始のメッセージがたら、ブラウザから localhost:8080 を開いて login します

```

c-11.0.10.9-hotspot#bin\javaw.exe (2022/11/29 10:10:21) [pid: 6244]
main] o.apache.catalina.startup.ContextC
main] o.a.c.c.C.[.localhost].[/birtview
main] org.apache.jasper.servlet.TldScanne
main] o.a.c.util.SessionIdGeneratorBase
main] o.s.security.core.userdetails.User
main] o.s.s.core.userdetails.User
main] o.s.s.web.DefaultSecurityFilterChain
main] o.s.b.w.embedded.tomcat.TomcatWebServer
main] com.example.login.BirtApplication

```

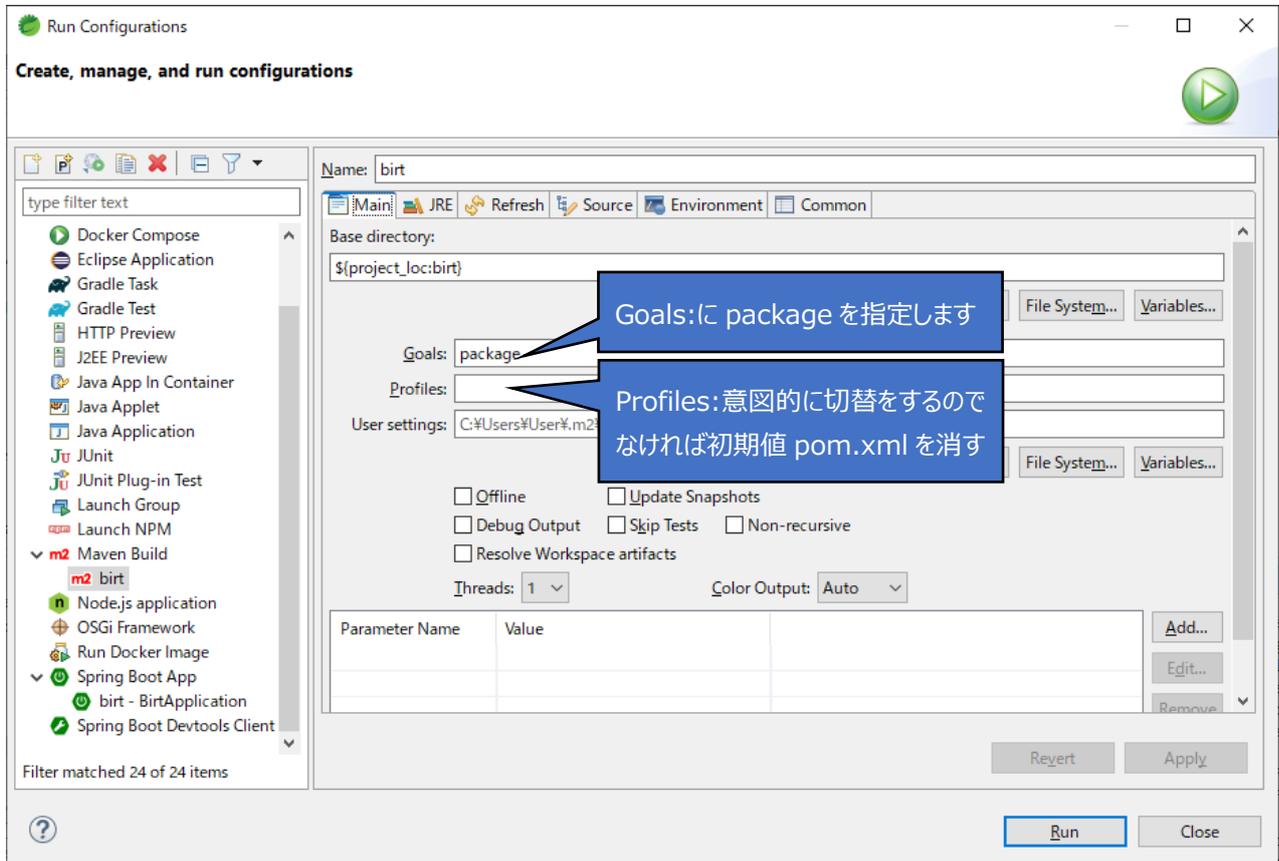
Copyright(C)2022 Future Office Coordinate Service Corporation All rights reserved. p. 21

BI - Report (Eclipse BIRT, Spring boot)

6.3. 実行形式 (jar) ファイルの作成

プロジェクトを選択し、メニュー Run ⇒ Run Configurations または
右クリックで Run as → Maven build...

Goals を設定して Run ボタンを押下します。

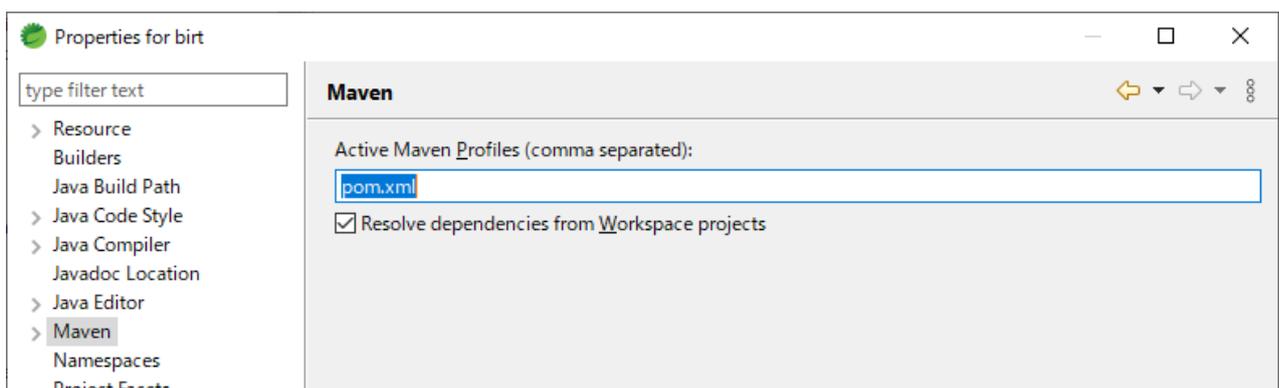


※Maven は初期化、依存ライブラリのコピー、パッケージ化の工程を順に実行します。この画面で指定する Goals は実行する最後の工程を指定するので、以下のコマンドを実行したのと同義になります

`mvn clean dependency:copy-dependencies package`

<エラー／ワーニング>

● [WARNING]The requested profile "pom.xml" could not be activated because it does not exist.
が出力されたら、プロジェクトを右クリックして Properties を開き、Maven に書かれている pom.xml を消します。(このワーニングが出ていてもビルドは正常に終了します)



BI - Report (Eclipse BIRT, Spring boot)

- TESTS 工程で[ERROR]が表示される場合

```
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ birt ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.example.demo.BirtApplicationTests
(中略)
14:50:52.066 [main] DEBUG org.springframework.test.context.support.ActiveProfilesUtils - Could not find any profiles to activate.
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.376 s <<< FAILURE!
[ERROR] com.example.demo.BirtApplicationTests Time elapsed: 0.368 s <<< ERROR!
java.lang.IllegalStateException: Unable to find a @SpringBootConfiguration, you need to use @SpringBootTest
[INFO]
[INFO] Results:
[INFO] |
[ERROR] Errors:
[ERROR] BirtApplicationTests » IllegalStateException Unable to find a @SpringBootConfiguration, ...
[INFO]
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 0.352 s
```

Mavenが実行するテストは、src/main/javaに@SpringBootApplicationを書いたソースとsrc/test/javaに@SpringBootTestと書いたソースが対になって存在する前提になっています。

※具体的な動作は@SpringBootTestのクラスから@SpringBootConfigurationを探してテスト対象を特定する処理で@SpringBootConfigurationを継承している@SpringBootApplicationを見つけます

したがって、Spring Boot のプロジェクトを作ったときに自動生成された以下の構成が維持されていればこのエラーは発生しません。

以下の対応関係を壊すパッケージ名やパスの変更した場合は手作業で整合性を合わせてください。

```
src/main/java
    com.example.login
        BirtApplication    ...@SpringBootApplication
src/test/java
    com.example.login
        BirtApplicationTests    ...@SpringBootTest
```

- 「BUILD SUCCESS」と表示されたら成功です。

```
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ birt ---
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.6:repackage (repackage) @ birt ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 15.729 s
[INFO] Finished at: 2022-11-29T14:33:31+09:00
[INFO] -----
```

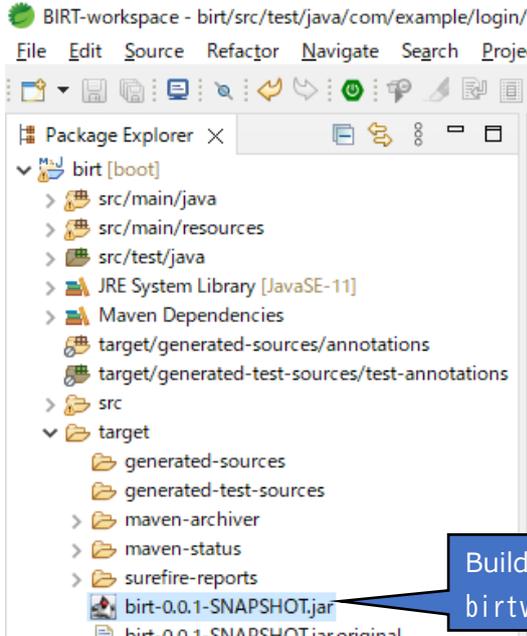
BI - Report (Eclipse BIRT, Spring boot)

6.4. Jar の実行

Maven でビルドした Jar ファイルはプロジェクトの target ディレクトリ内に作られます。
以下のコマンドで実行してください。

> java -jar <jar ファイル名>

《実行例》



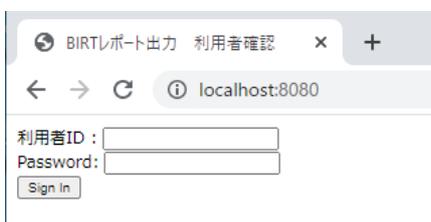
```

C:\Users\User>java -jar C:\birt-0.0.1-SNAPSHOT.jar

:: Spring Boot ::
          (v2.7.6)

2022-11-30 09:20:20.927 INFO 10960 --- [           main] com.example.login.BirtApplication : Starting BirtApplication v
0.0.1-SNAPSHOT using Java 11.0.10 on WIN-GBON7J2169G with PID 10960 (C:\birt-0.0.1-SNAPSHOT.jar started by User in C:\Users\User
r)
2022-11-30 09:20:20.936 INFO 10960 --- [           main] com.example.login.BirtApplication : No active profile set, fal
ling back to 1 default profile: "default"
2022-11-30 09:20:23.515 INFO 10960 --- [           main] com.example.login.AppConfig$1 : -----
(中略)
y.web.authentication.AnonymousAuthenticationFilter@797f1aea, org.springframework.security.web.session.SessionManagementFilter@c
6120a4, org.springframework.security.web.access.ExceptionTranslationFilter@5cfe703, org.springframework.security.web.access.int
ercept.AuthorizationFilter@46c635d]
2022-11-30 09:20:47.452 INFO 10960 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s):
8080 (http) with context path ''
2022-11-30 09:20:47.474 INFO 10960 --- [           main] com.example.login.BirtApplication : Started BirtApplication in
27.652 seconds (JVM running for 28.659)
  
```

終了は ctrl + c …停止時に「WARNING: An illegal reflective access operation has occurred」が表示 (Tomcat9 と Java11 の組合せで発生 issues#15101) されても、処理に問題はありません



Tomcat の起動メッセージ出力後、アクセスが可能になります

BI - Report (Eclipse BIRT, Spring boot)

6.5. セキュリティに関する注意点

(1) 利用者管理について

実装例の認証では WebSecurityConfig に利用者 ID とパスワード、役割等を直書きしていますが実際の運用では基幹システムの認証に統合するか、管理権限者が運用中に利用者 ID を停止できるシステムが必要です (OpenLDAP や Keycloak 等 Spring で使えるオープンソースもあります)。

(2) セキュリティレベル

BIRT のレポートビューアは http リクエストに認証情報を求めないので、レポート名 (定義名) さえ知っていれば curl 等のコマンドを使って誰でもレポートを取得できます。このため情報の重要度によって以下の対策が必要です。

- ① レポート定義名をランダムにして推測できなくする (アクセス用の html にだけ帳票名を表示)
- ② BIRT ビューアの前にリバースプロキシ (サーバ側のプロキシ) を用意して認証する
- ③ 重要な帳票はサーバ/LAN を分けてファイアウォールで要求元を選別する

以上