

インストールマニュアル

目次

I. Testabliish サーバ環境構築手順(管理者作業)

- [1. Testabliish サーバの構築【Windows10】](#)
 - [1.1. Testabliish サーバをインストールする](#)
 - [1.1.1. インストーラの起動](#)
 - [1.1.2. 仮想プラットフォームのチェックと設定](#)
 - [1.1.3. インストールの継続](#)
 - [1.1.4. WSL2インストール](#)
 - [1.1.5. Ubuntuインストール](#)
 - [1.1.6. プロキシ設定](#)
 - [1.1.7. Dockerインストール](#)
 - [1.1.8. Testabliishサーバコンテナインストール](#)
 - [1.1.9. インストール完了](#)
 - [1.2. Testabliishサーバの起動確認をする](#)
 - [1.3. Testabliish サーバの起動・停止](#)
 - [1.3.1. サーバ起動](#)
 - [1.3.2. サーバ停止](#)
 - [1.3.3. 外部接続設定1](#)
 - [1.3.4. 外部接続設定2](#)
 - [1.4. Testabliish サーバをバージョンアップする](#)
- [2. Testabliish サーバの構築【Linux】](#)
 - [2.1. Dockerをインストールする](#)
 - [2.2. Testabliishサーバを構築する](#)
 - [2.3. Testabliish サーバをバージョンアップする](#)

II. Extension(拡張機能)の導入

- [1. Windowsでの拡張機能導入](#)
 - [1.1. Testabliish タスクトレイアプリのインストール](#)
 - [1.1.1. インストーラの指示に従ってインストールする](#)
 - [1.1.2. Testabliish タスクトレイアプリを起動する](#)
 - [1.1.3. Testabliish タスクトレイアプリの設定をする](#)
 - [1.2. Google Chromeでの拡張機能設定](#)
 - [1.2.1. Chrome Extensionを入手・導入する](#)
 - [1.2.2. ブラウザポップアップを許可する](#)
 - [1.2.3. Chrome Extensionのオプション設定](#)
 - [1.3. Internet Explorerでの拡張機能設定](#)
 - [1.3.1. Internet Explorer でのアドオン\(TestabliishBho Class\)の有効化](#)
 - [1.3.2. Internet Explorer での「Windows Defender SmartScreen」設定](#)
 - [1.3.3. Internet Explorer での互換表示設定の解除](#)
 - [1.3.4. Internet Explorer でのセキュリティ設定](#)

III. 自動実行環境の構築手順

- 1. 自動実行環境を構築する
 - 1.1 Windowsでの実行環境導入
 - 1.1.1. インストーラー の実行
 - 1.1.2. 使用許諾画面
 - 1.1.3. ネットワーク接続確認
 - 1.1.4. WebDriverインストール方法選
 - 1.1.5. インストール先の指定
 - 1.1.6. インストール
 - 1.1.7. インストール完了
 - 1.2 手動で導入する
 - 1.2.1 Selenium をインストールする
 - 1.2.2 Internet Explorer の設定を確認する
 - 1.2.3 Windows10 の設定をする
 - 1.2.4 Java をインストールする
 - 1.2.5 Gradle を設定する
 - 1.2.6 テストコードを実行する

IV. 証明書のインストール

- 1. セキュリティ証明書の生成
- 2. セキュリティ証明書のインポート
 - 2.1. certlm.msc を起動する
 - 2.2. インポートウィザードの起動
- 3. セキュリティ証明書の確認

I. Testabliish サーバ環境構築手順(管理者作業)

- [1. Testabliish サーバの構築【Windows10】](#)
 - [1.1. Testabliish サーバをインストールする](#)
 - [1.1.1. インストーラの起動](#)
 - [1.1.2. 仮想プラットフォームのチェックと設定](#)
 - [1.1.3. インストールの継続](#)
 - [1.1.4. WSL2インストール](#)
 - [1.1.5. Ubuntuインストール](#)
 - [1.1.6. プロキシ設定](#)
 - [1.1.7. Dockerインストール](#)
 - [1.1.8. Testabliishサーバコンテナインストール](#)
 - [1.1.9. インストール完了](#)
 - [1.2. Testabliishサーバの起動確認をする](#)
 - [1.3. Testabliish サーバの起動・停止](#)
 - [1.3.1. サーバ起動](#)
 - [1.3.2. サーバ停止](#)
 - [1.3.3. 外部接続設定1](#)
 - [1.3.4. 外部接続設定2](#)
 - [1.4. Testabliish サーバをバージョンアップする](#)
- [2. Testabliish サーバの構築【Linux】](#)
 - [2.1. Dockerをインストールする](#)
 - [2.2. Testabliishサーバを構築する](#)
 - [2.3. Testabliish サーバをバージョンアップする](#)

1. Testabliish サーバの構築【Windows10】

Testabliish v1.4.4 からはWindows 10 上でのTestabliish サーバは、WSL2の環境下で実行されるUbuntu(Linuxデистриビューション)のDockerのコンテナとして作成されます。

本章の手順ではインストーラにより新規にWSL2、Ubuntu、Docker環境、Testabliish をインストールする流れを説明しています。

- すでにUbuntuがインストールされている場合は削除してからインストールを開始してください。
- Windows環境ですでにv1.4.3以前のバージョンで利用されている方は『Testabliish マイグレーションマニュアル.pdf』を参照して移行してください。
- Windows 10 環境で WSL2 を利用するためには、仮想化プラットフォーム「Hyper-V」を有効にする必要があります。

Hyper-V は、Windows 10 Pro、Enterprise、および Education の64ビットバージョンで使用できます。システム要件について、詳しくは 以下をご確認ください。

[Windows 10 Hyper-V のシステム要件](#)

Hyper-Vのインストール方法について詳しい情報は以下をご参照ください。

[Windows 10 上に Hyper-V をインストールする](#)

- Testabliishサーバのインストールは、**インターネットに接続できる環境**で行ってください。
- **Windows 10 バージョン2004以降(KB5004296 適用済み)**の環境で行ってください。

1.1. Testabliish サーバをインストールする

Testabliish サーバのインストールはインストーラを利用します。

インストーラ内では再起動を要求されるので使用する前に、**作業中のソフトウェアはすべて終了して** 実行してください。

また、他のWSLディストリビューションがインストールされている場合は**必ずStoppedになった状態で実行**してください。

コマンドプロンプトで以下のコマンドで確認ができます。

```
wsl -l -v
```

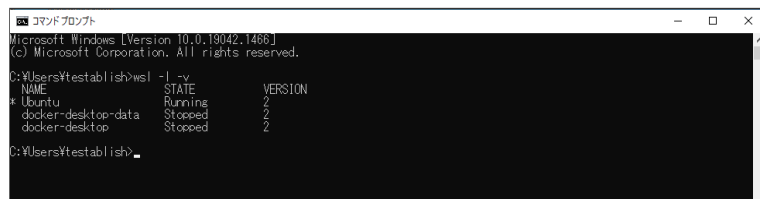


Fig. 1.1 WSLステータス

準備ができたなら任意の場所に、配布されているインストーラおよび関連ファイル(arcフォルダ)を配置しておいてください。

1.1.1. インストーラの起動

Testablisth インストーラのファイル名は**Testablisth_vx.x.x_Installer.exe**(x部はバージョン番号)です。ファイルをダブルクリックして起動し、ウィザードに従ってインストールを進めていきます。

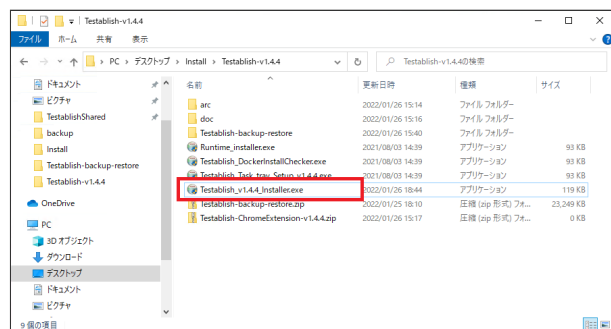


Fig. 1.1.1 解凍ディレクトリ

1.1.2. 仮想プラットフォームのチェックと設定

インストーラの言語設定、ライセンス画面で使用許諾に同意した後に、仮想プラットフォームのチェックと設定が始まります。

完了後、次の画面が表示されます。

すぐに再起動したくない場合は"いいえ"を選択するとインストーラが終了します。次回起動したときに自動的にインストールが再開されます。

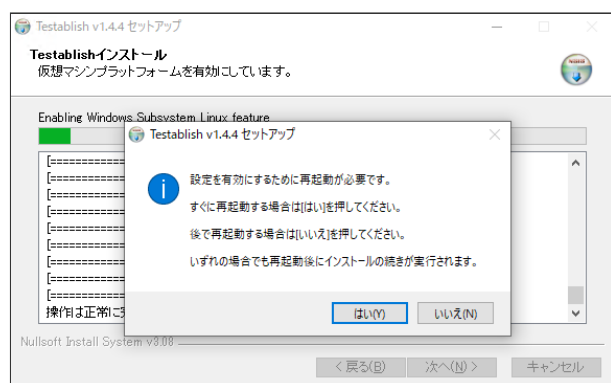


Fig. 1.1.2 仮想プラットフォームインストール

1.1.3. インストールの継続

PCの再起動後、自動的にインストーラーが起動します。

インストーラの言語設定、ライセンス画面で使用許諾に同意した後に、インストール先の選択画面に遷移します。

インストール先を指定インストールボタンを押下します。

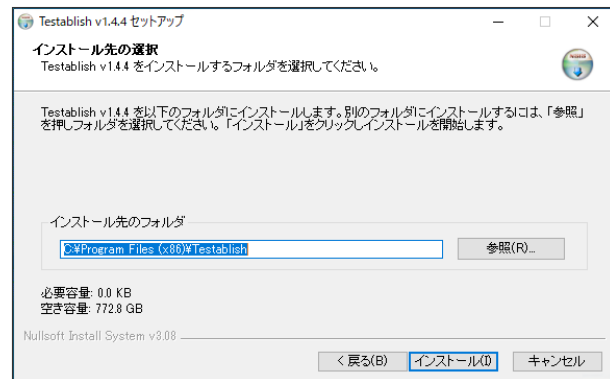


Fig. 1.1.3 インストール先選択

1.1.4. WSL2インストール

WSL2のインストールが始まります。

途中でWSLのインストールを促す画面が出てきますので進めていきます。

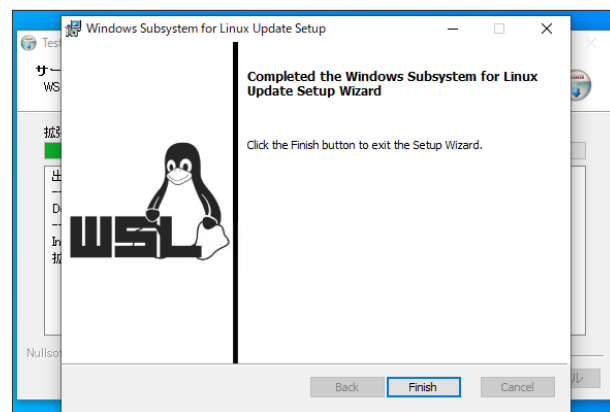


Fig. 1.1.4 WSL2インストール

1.1.5. Ubuntuインストール

しばらくするとUbuntuのインストールが始まります。

この画面は環境にもよりますが10分以上がかかる場合がありますので停止させずにそのままお待ちください。

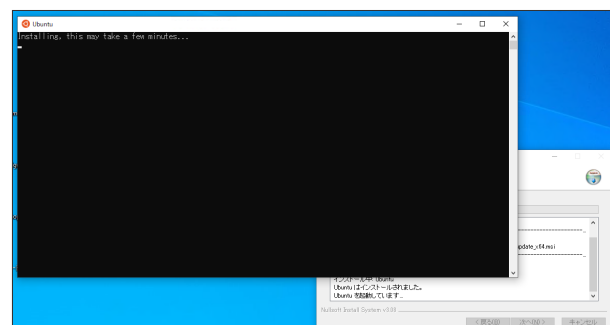


Fig. 1.1.5 Ubuntuインストール

1.1.6. プロキシ設定

Ubuntuのインストールが完了するとプロキシ設定の画面に遷移します。

ここで設定するのはUbuntu内で使用されるプロキシ設定となります。

プロキシが無い環境であればそのまま次の画面に進んでください。

プロキシがある場合はHTTP,HTTPSの両方の設定を行ってください。

プロキシが認証を要求する場合はそのユーザID/パスワードを入力してください。

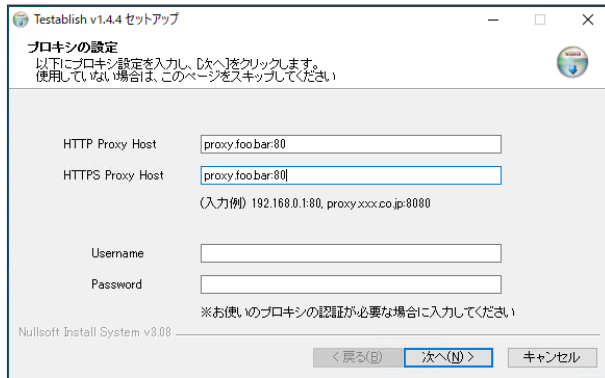


Fig. 1.1.6 プロキシ設定

次の画面に進むとプロキシの設定画面になります。

ここでチェックボタンを押すと前の画面で設定したプロキシ設定を使ってUbuntu内部からインターネットへアクセスできるかをチェックします。

チェックが成功するとインストールボタンが有効になりますので次の画面に進んでください。

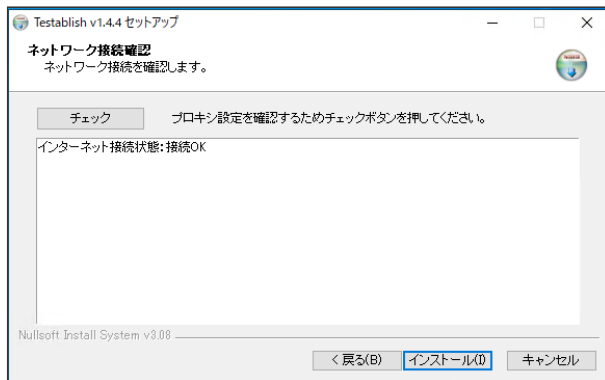


Fig. 1.1.6a プロキシチェック成功

チェックに失敗する場合、前の画面に戻ってプロキシ設定を確認してください。

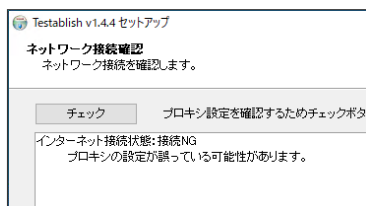


Fig. 1.1.6b プロキシチェック失敗

チェックには時間がかかる場合があり、ウィンドウが一時的に応答なしになる場合があります。

そのまましばらくお待ちください。

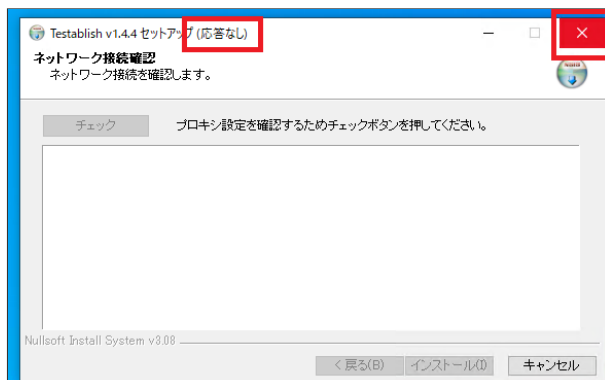


Fig. 1.1.6c プロキシチェックに時間がかかる場合

1.1.7. Dockerインストール

プロキシ設定が成功するとDockerインストール画面に遷移します。

Dockerインストールも時間がかかる場合がありますのでしばらくお待ちください。

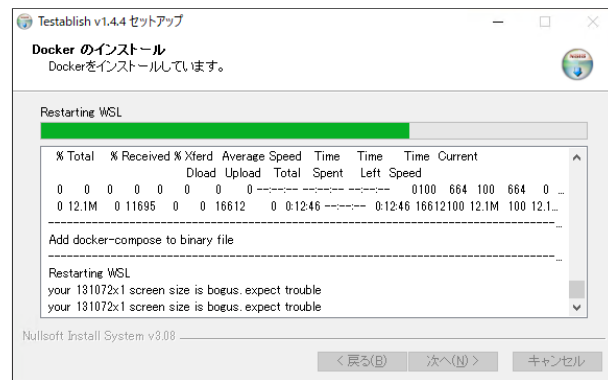


Fig. 1.1.7 Dockerインストール

1.1.8. Testabishサーバコンテナインストール

Dockerインストールが完了するとTestabishのサーバコンテナのインストールが始まります。

Testabishのtarファイルを指定してください。

tarファイルの配置場所はローカルマシン上に配置して下さい。ファイルサーバ上のファイルを指定するとインストールに失敗します。

この時選択したtarファイルの同階層にあるdocker-compose.ymlがインストール時に使用されます。

```
├─ Testabish_Installer.exe
└─ arc/
   ├─ testabish-pro-vx.x.x.tar // x.x.x は バージョン番号
   └─ docker-compose.yml     // コンテナ設定ファイル
```

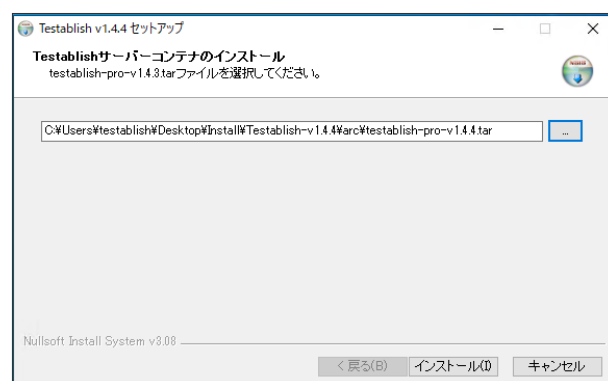


Fig. 1.1.8 Testabishインストール

ファイルを指定して次へ進むとサーバコンテナのインストールが始まります。

このインストールも時間がかかる場合がありますのでしばらくお待ちください。

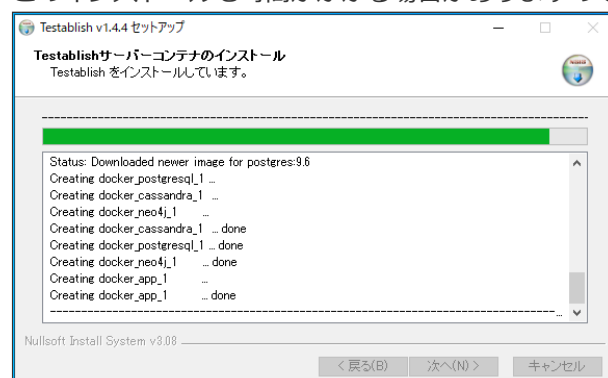


Fig. 1.1.8b Testabishコンテナインストール

1.1.9. インストール完了

コンテナのインストールが完了したら終了です。
完了ボタンを押して終了してください。

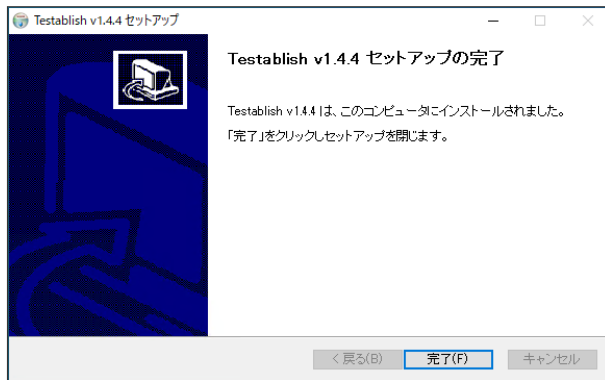


Fig. 1.1.9 インストール完了

**旧バージョンからのマイグレーションの方はここまで
マイグレーションマニュアルに戻ってデータのリストアを実施して
ください。**

**バージョンアップの方はここまで
バージョンアップの手順に戻ってデータのリストアを実施してくだ
さい。**

1.2. Testabliishサーバの起動確認をする

<http://localhost/> に接続して、Testabliish サーバの画面が表示されることを確認します。

インストール後の初回の表示には、数分間程度の時間がかかることがあります。

起動中は起動中画面が表示されます。

ログイン画面が表示されるまでしばらくお待ちください。

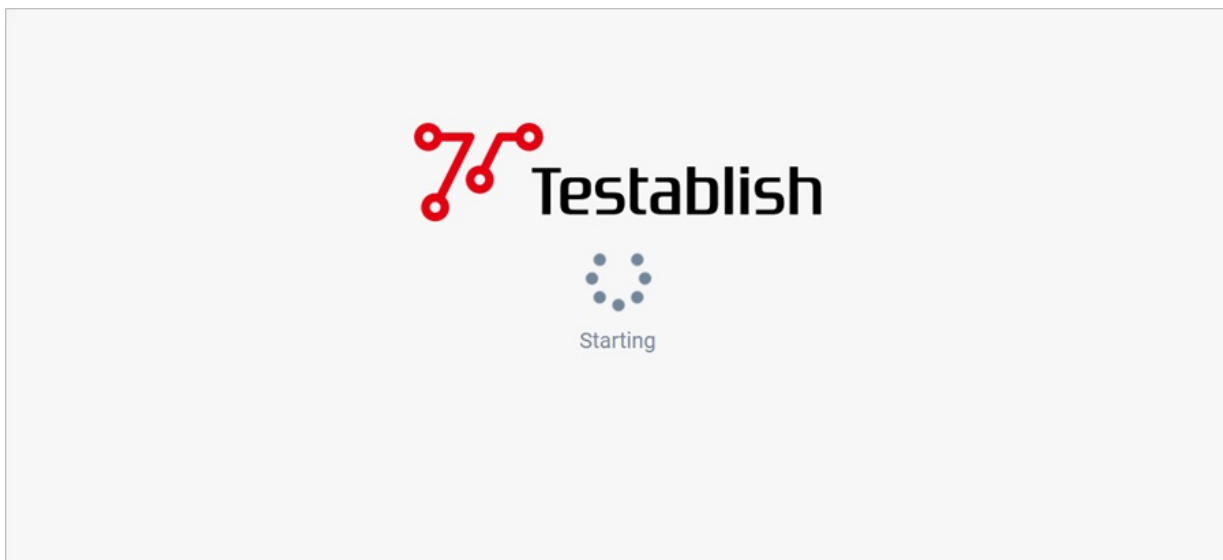


Fig. 1.2-1 Testabliish サーバ起動中画面

ログイン画面表示が表示されたら **デフォルトユーザ** でログインしてください。
※ デフォルトユーザ名およびパスワードは配布メディアのパッケージに記載されています。



Fig. 1.2-2 Testabliish 起動後ログイン画面

ログインすると **Setup Wizard** が表示されますので、**Start** ボタンか **Next** をクリックして画面を進めてください。

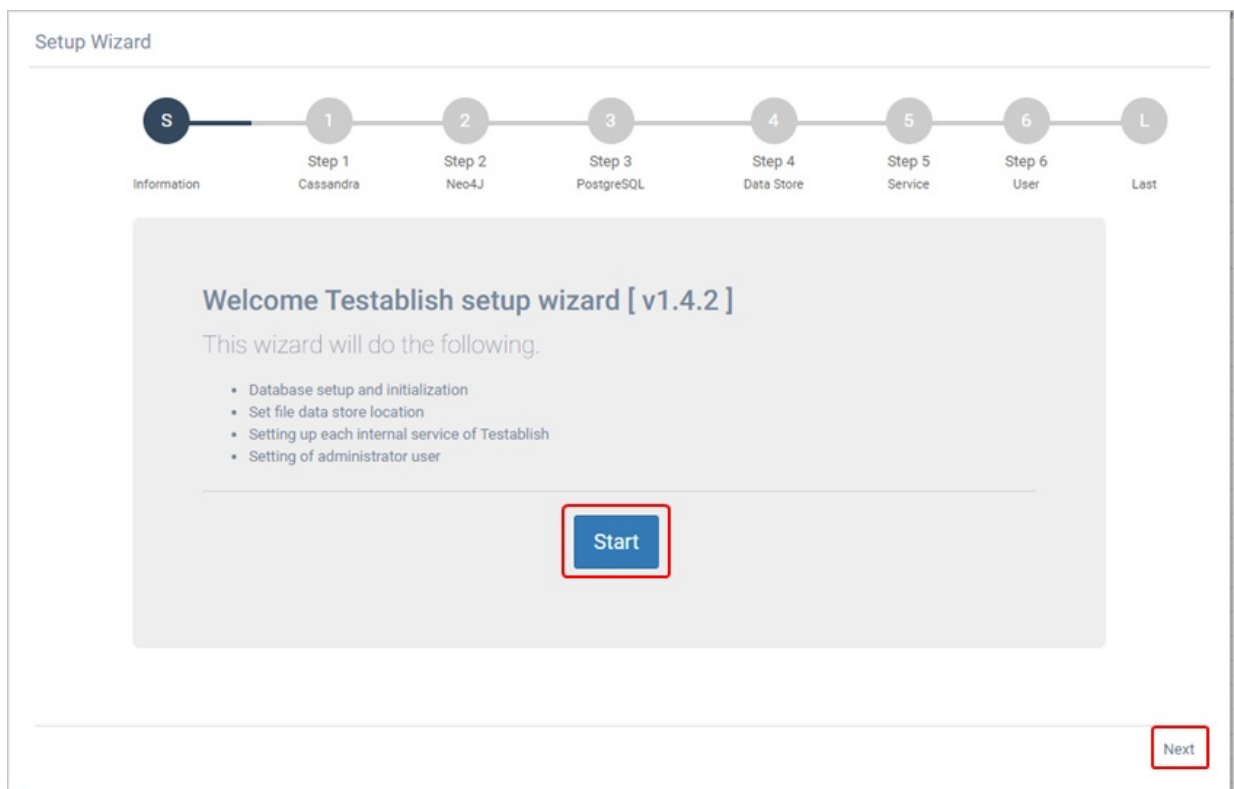


Fig. 1.2-3 Setup Wizard Information

画面を進めると入力できるテキストボックスが表示されますが、そのままの状態に進めてください。

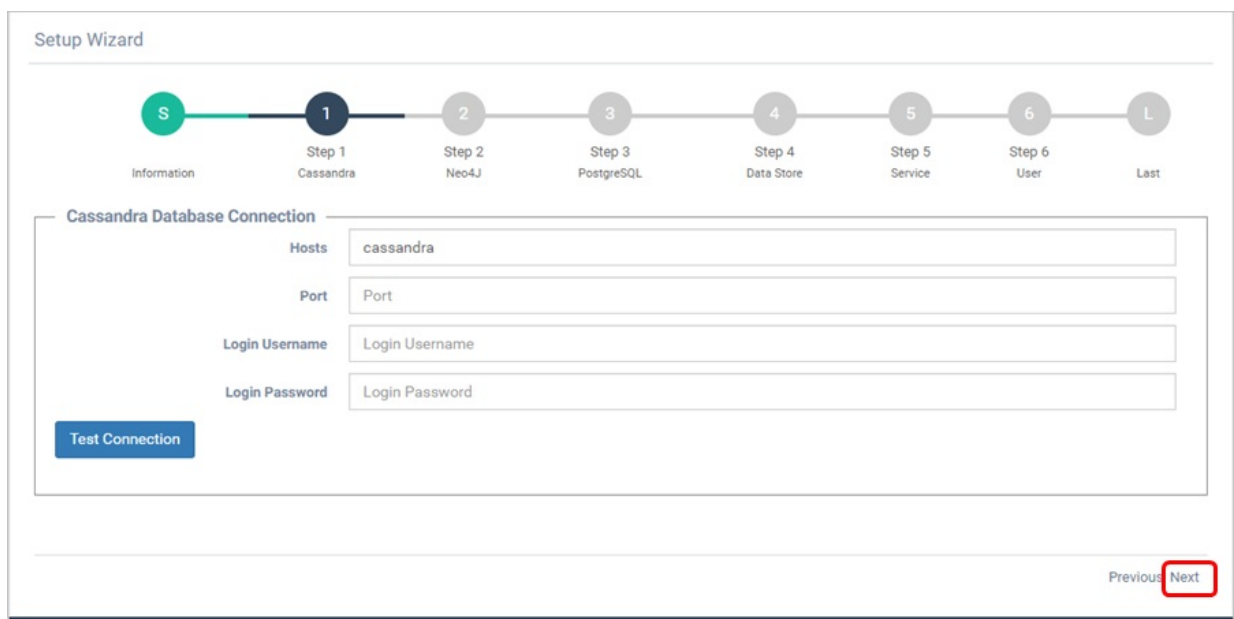


Fig. 1.2-4 Setup Wizard Cassandra

Setup Wizard

S — 1 — 2 — 3 — 4 — 5 — 6 — L

Information Step 1 Cassandra Step 2 Neo4J Step 3 PostgreSQL Step 4 Data Store Step 5 Service Step 6 User Last

Neo4J Database Connection

Connection URL

Login Username

Login Password

[Test Connection](#)

[Previous](#) [Next](#)

Fig. 1.2-5 Setup Wizard Neo4j

Setup Wizard

S — 1 — 2 — 3 — 4 — 5 — 6 — L

Information Step 1 Cassandra Step 2 Neo4J Step 3 PostgreSQL Step 4 Data Store Step 5 Service Step 6 User Last

PostgreSQL Database Connection

Host

Port

Database Name

Login Username

Login Password

Max Pool Connections

Idle Timeout Milliseconds

[Test Connection](#)

[Previous](#) [Next](#)

Fig. 1.2-6 Setup Wizard PostgreSQL

Setup Wizard

S

1

2

3

4

5

6

L

Information

Step 1
Cassandra

Step 2
Neo4J

Step 3
PostgreSQL

Step 4
Data Store

Step 5
Service

Step 6
User

Last

Data Store

Session Data Store Path

/testablish/session

Thumbnail Data Store Path

/testablish/thumb

Page Sample Data Store Path

/testablish/sample

Testdoc Template Data Store Path

/testablish/testdoc-template

Test Access

Previous

Next

Fig. 1.2-7 Setup Wizard DataStore

Setup Wizard

S

1

2

3

4

5

6

L

Information

Step 1
Cassandra

Step 2
Neo4J

Step 3
PostgreSQL

Step 4
Data Store

Step 5
Service

Step 6
User

Last

Testablish PM (Dashboard)

Port

1879

UI

Port

1881

Importer

Host

127.0.0.1

Port

1880

Create Thumbnail Image Size

200

Converter

Host

127.0.0.1

Port

1885

Test Code Generator

Host

127.0.0.1

port

1884

Test Code Generator

Host

127.0.0.1

port

1884

Test Document Generator

Host

127.0.0.1

Port

1886

Scraping

Host

127.0.0.1

Port

1882

Previous

Next

Fig. 1.2-8 Setup Wizard Service

Administrator User の設定画面では **ID**、**Name**、**Password**、**Confirm Password**、**Language** の設定を行い、**Next** で進んでください。

設定した内容は **Setup Wizard** の完了後、ログインする際に必要になります。

The screenshot shows the 'Setup Wizard' interface. At the top, a progress bar contains eight steps: 'S' (Information), '1' (Step 1: Cassandra), '2' (Step 2: Neo4J), '3' (Step 3: PostgreSQL), '4' (Step 4: Data Store), '5' (Step 5: Service), '6' (Step 6: User), and 'L' (Last). Step 6 is currently selected. Below the progress bar, the 'Administrator User' section contains a form with the following fields: 'ID' (admin), 'Name' (admin), 'Password' (masked with dots), 'Confirm Password' (masked with dots), and 'Language' (a dropdown menu with 'English' and 'Japanese' options, where 'Japanese' is highlighted in blue). At the bottom right of the form, there are 'Previous' and 'Next' buttons, with 'Next' being highlighted with a red border.

Fig. 1.2-9 Setup Wizard User

ラストページまで進めたら、**Setup & Restart** ボタンをクリックしてください。

The screenshot shows the 'Setup Wizard' interface at the final step. The progress bar at the top is identical to the previous screenshot, with Step 6 ('User') selected. The main content area is a light gray box with the text 'This is last step.' followed by 'Click on the "Setup & Restart" button to initialize the database, Make Testablish available'. Below this text is a large blue button labeled 'Setup & Restart', which is highlighted with a red border. At the bottom right, there are 'Previous' and 'Finish' buttons.

Fig. 1.2-10 Setup Wizard Last

確認ダイアログが表示されますので、**OK** をクリックしてください。

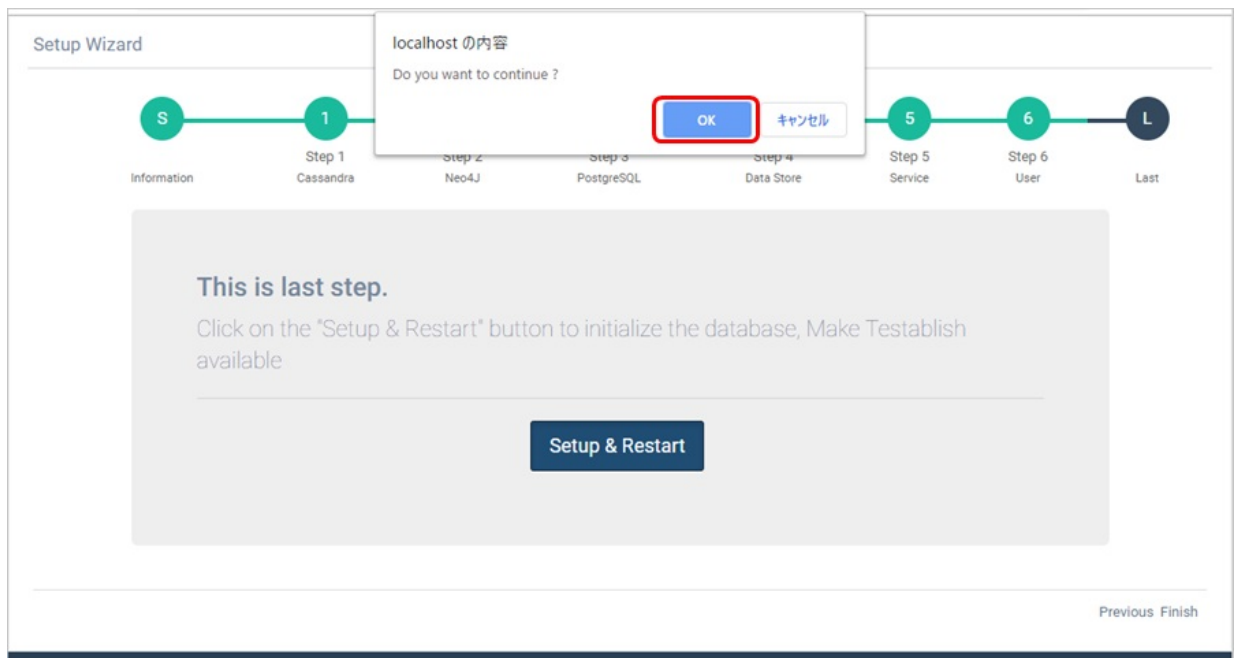


Fig. 1.2-11 Setup Wizard ダイアログ

Setup successful のポップアップが表示されれば成功です。

この後サーバのシステムが再起動しますので、しばらく接続が落ちますがそのままお待ちください。

サーバの再起動が完了するとログイン画面に変わりますので、**Administrator User** の画面で設定したユーザ情報でログインしてご使用ください。

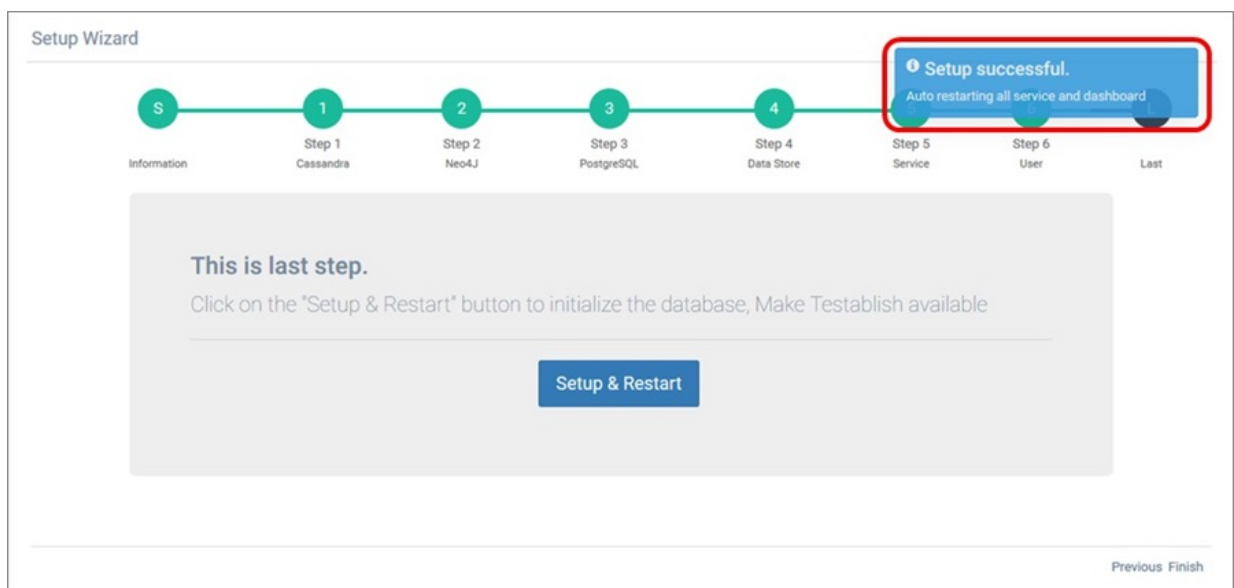


Fig. 1.2-12 Setup Wizard succesful

ログインすると、プロジェクト一覧画面が表示されます。右上のユーザ名をクリックして表示されるプルダウンメニューから、**システム設定** を選択します。



Fig. 1.2-13 Testablish Dashboard の起動

Testabliish のDashboard画面が表示されることを確認してください。

Services

サービス システム設定

Testabliishサーバー側のサービスダッシュボード

#	サービス名	PID	状態	アップタイム	不安定な再起動	CPU	メモリ	オペレーション
#	UI	32	online	4分	0	0 %	95.20 MiB	Start Restart Stop Log
#	Importer	25	online	4分	0	0 %	49.95 MiB	Start Restart Stop Log
#	Converter	31	online	4分	0	0 %	50.19 MiB	Start Restart Stop Log
#	Scraping	38	online	4分	0	0 %	84.82 MiB	Start Restart Stop Log
#	Generator-Java	44	online	4分	0	0 %	50.52 MiB	Start Restart Stop Log
#	Generator-TestDoc	50	online	4分	0	0 %	39.70 MiB	Start Restart Stop Log
(合計)							370.37 MiB	

ログ

ログ行: 100

Fig. 1.2-14 Testabliish Dashboard の起動確認

上記が表示されれば、Testabliishのサーバ構築は **完了** です。

1.3. Testabliish サーバの起動・停止とネットワーク設定

Testabliishサーバの起動はコマンドスクリプトから実行できるようになります。PCの再起動に伴ってTestabliishサーバが立ち上がらなくなった場合以下の手順で実行してください。

1.3.1. サーバ起動

スタートメニュー -> TestabliishServer -> StartTestabliish を選択するとスクリプトが起動してサーバが起動します。

実行には管理者権限が必要です。※一覧にエントリが存在しません のメッセージが表示されることがありますが無視してください。

正常に完了するとsuccessfully のメッセージが表示されます。

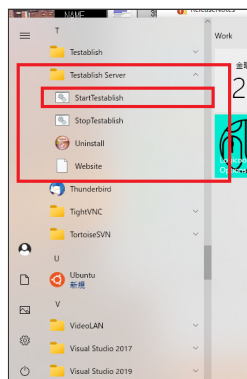
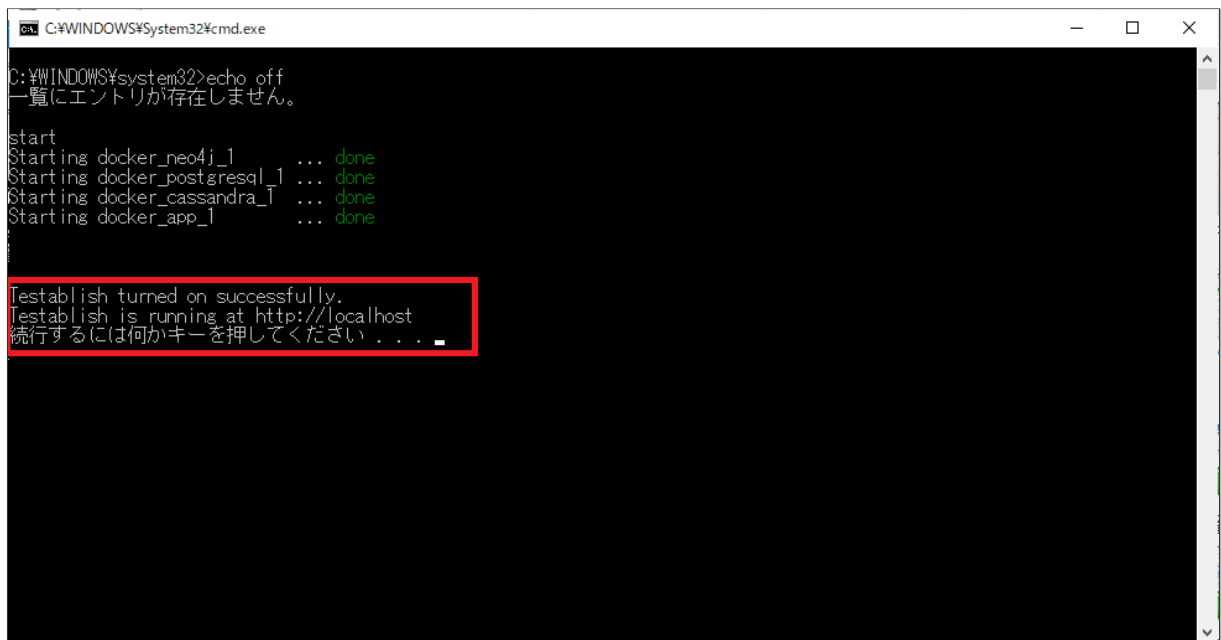


Fig. 1.3.1a サーバ起動



```
C:\WINDOWS\system32\cmd.exe

C:\WINDOWS\system32>echo off
一覧にエントリが存在しません。

start
Starting docker_neo4j_1 ... done
Starting docker_postgresql_1 ... done
Starting docker_cassandra_1 ... done
Starting docker_app_1 ... done

Testabliish turned on successfully.
Testabliish is running at http://localhost
続行するには何かキーを押してください . . .
```

Fig. 1.3.1b サーバ起動コマンドプロンプト

1.3.2. サーバ停止

スタートメニュー -> TestabliishServer -> StopTestabliish を選択するとスクリプトが起動してサーバが停止します。

実行には管理者権限が必要です。※一覧にエントリが存在しません のメッセージが表示されることがありますが無視してください。

正常に完了するとsuccessfully のメッセージが表示されます。

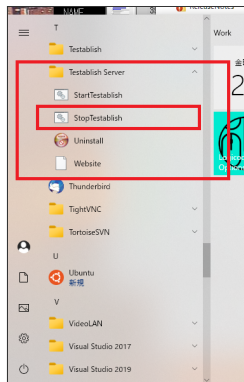


Fig. 1.3.1a サーバ停止

```
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>echo off
一覧にエントリが存在しません。

Stopping Testabli sh
docker_cassandra_1
docker_neo4j_1
docker_postgresql_1
docker_app_1

Testabli sh turned off successfully.
```

Fig. 1.3.1b サーバ停止コマンドプロンプト

1.3.3. 外部接続設定1

インストールしたPC以外から接続を許すためには Windowsの設定が必要になります。

コマンドプロンプトを管理者権限で起動して以下のコマンドで設定してください。

外部からport80で接続するための設定は以下になります。

Testabli shのサーバに外部からアクセスする場合は port80とport1880の開放が必要になります。

※ 『listenaddress=XXX.XXX.XXX.XXX』の部分はお使いのPCのIPアドレスに置き換えてください

```
$ netsh interface portproxy add v4tov4 listenport=80 listenaddress=XXX.XXX.XXX.XXX connectport=80
connectaddress=localhost
```

```
$ netsh interface portproxy add v4tov4 listenport=1880 listenaddress=XXX.XXX.XXX.XXX connectport=1880
connectaddress=localhost
```

この設定はPCを再起動しても有効です。

設定の確認

```
$ netsh interface portproxy show all
ipv4 をリッスンする:ipv4 に接続する:
```

Address	Port	Address	Port
-----	-----	-----	-----
XXX.XXX.XXX.XXX	80	localhost	80
XXX.XXX.XXX.XXX	1880	localhost	1880

設定を削除する場合は以下のコマンドで削除して下さい。

```
$ netsh interface portproxy delete v4tov4 listenport=80 listenaddress=XXX.XXX.XXX.XXX
$ netsh interface portproxy delete v4tov4 listenport=1880 listenaddress=XXX.XXX.XXX.XXX
```

1.3.4. 外部接続設定2

Windows Defender が有効になっていて、インストールしたPC以外から接続を許すためには Windows Defender ファイアーウォールの設定が必要になります。

1 ファイアウォールを開く

コントロールパネルを開き、システムとセキュリティ > Windows Defender ファイアウォールをクリックする。

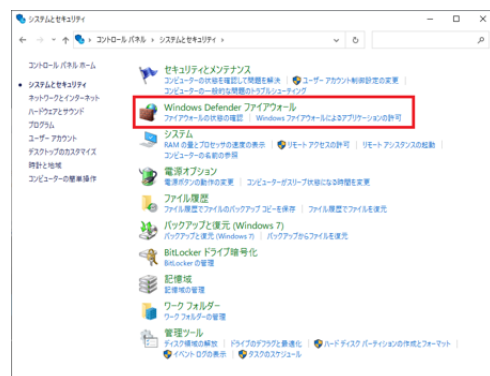


Fig. 1.3.4-1 Windows Defender ファイアウォール

2 詳細設定を開く

左メニューから詳細設定をクリックする。

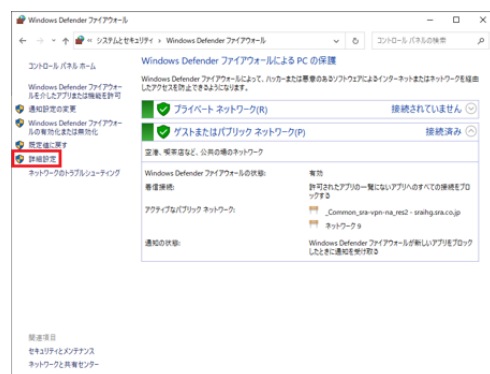


Fig. 1.3.4-2 詳細設定

3 規則の追加

左メニューから受信の規則を選択し、右メニューから新しい規則を選択する。

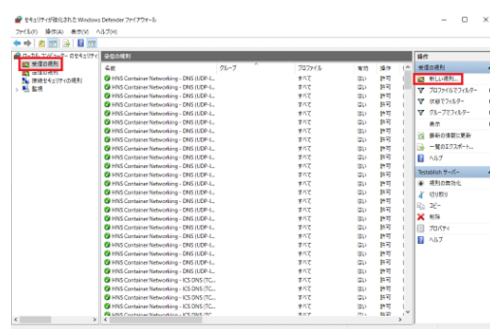


Fig. 1.3.4-3 規則の追加

4 登録内容

以下の内容で順に登録する

規則の種類	ポートを選択
プロトコル	TCP を選択
ポート	特定のローカルポートを選択して 80 を入力

操作	接続を許可する
プロファイル	ドメイン、プライベート、パブリックを全て選択する
名前	Testabliish 外部接続80

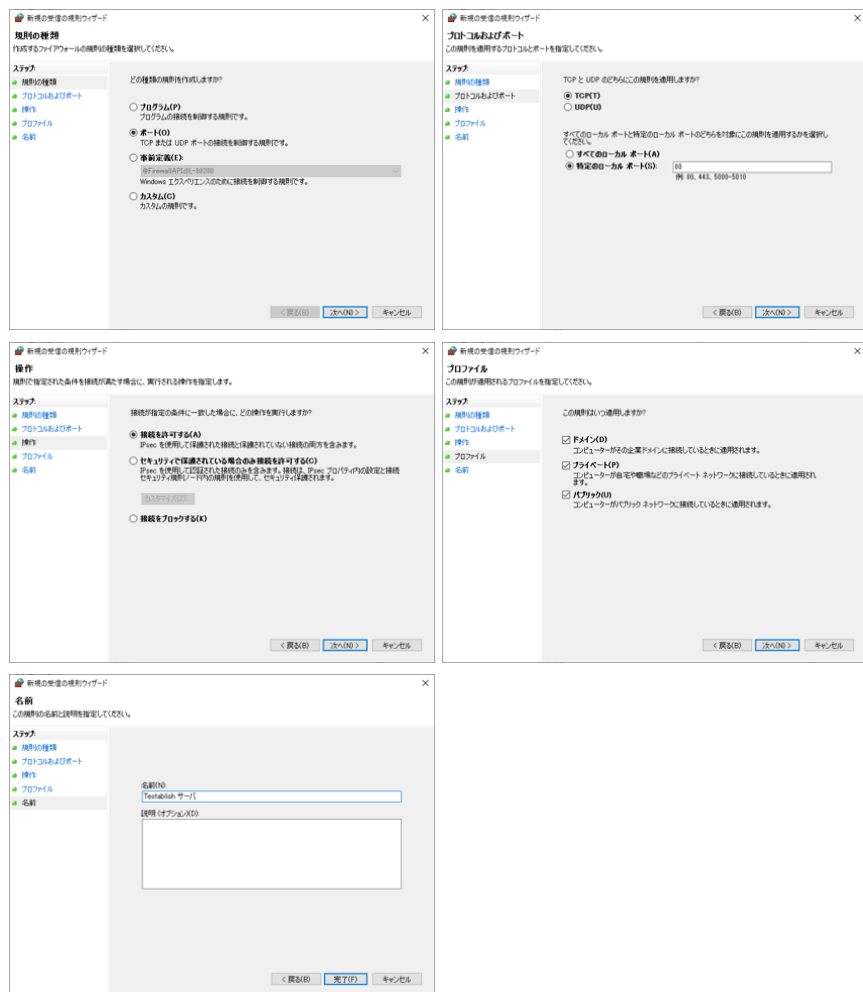


Fig. 1.3.4-4 登録内容

5 ローカルポート1880

上記、3,4の操作をローカルポート1880に対しても行う。

1.4. Testabliish サーバをバージョンアップする

Windows版サーバーのバージョンアップ手順は データベースバックアップ～アンインストール～インストール～リストア の手順で実施します。

1.4.1. データベースバックアップ

バックアップコマンドを使ってデータベースのバックアップを行います。
このツールを利用すると利用中のすべてのデータのバックアップを行います。
バックアップ中はTestabliishの利用ができませんので注意してください。

1 バックアップツールを解凍する

Testabliish-v1.X.X.zip ファイルに含まれる Testabliish-backup-restore.zipを解凍します。

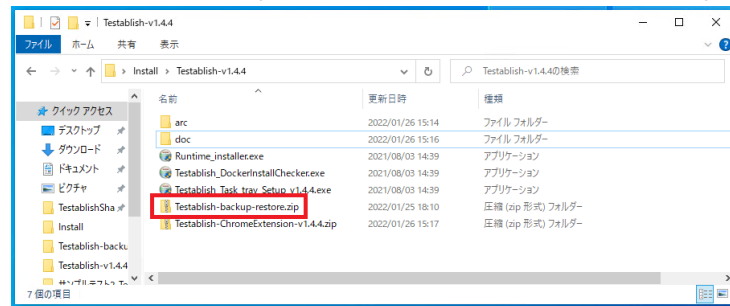


Fig. 1.4.1-1 Testabliish-v1.X.X 解凍フォルダ

2 バックアップの設定を変更する

解凍したフォルダの setting.yaml ファイルをテキストエディタで開きます。

backup_platformと**restore_platform**の設定をお使いのサーバーバージョンに合わせて変更します。

```
#####
####
#### バックアップ (backup.bat) の設定
####
#####

# バックアップのプラットフォーム(host or wsl)
# host: v1.4.3 以前 (dockerDesktop)の場合
# wsl : v1.4.4 以降 (Ubuntu)の場合

backup_platform: wsl

# バックアップするコンテナの名前
backup_container_name:

. . .

#####
####
#### リストア (restore.bat) の設定
####
#####

# リストア先のプラットフォーム(host or wsl)
# host: v1.4.3 以前 (dockerDesktop)の場合
# wsl : v1.4.4 以降 (Ubuntu)の場合

restore_platform: wsl

# バックアップファイルがあるディレクトリ
restore_directory: ./backup

. . .
```

3 バックアップツールを実行する

解凍したフォルダに移動して backup.bat ファイルをダブルクリックします。バックアップが始まります。

完了するまでしばらく時間がかかります。データの量に応じてバックアップの時間が変わります。

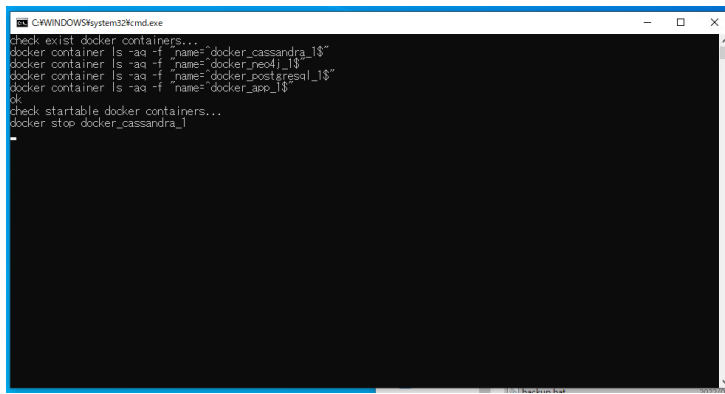


Fig. 1.4.1-2 バックアップ中

4 バックアップ完了後

バックアップが完了すると backup successfully completed! のメッセージが表示されます。

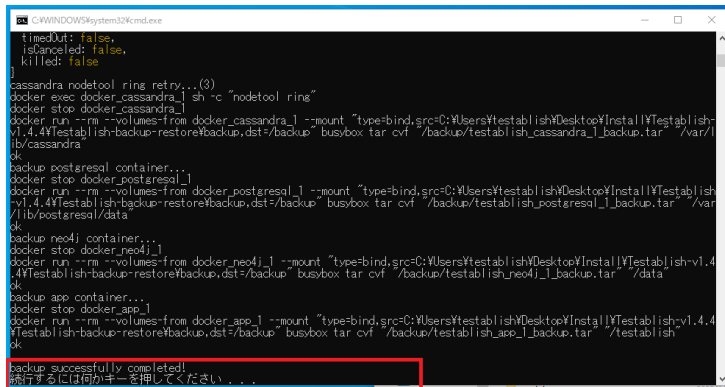


Fig. 1.4.1-3 バックアップ完了

バックアップファイルはこのフォルダに "backup" というフォルダに保存されます。
各データベースの tar ファイルが配置されます。

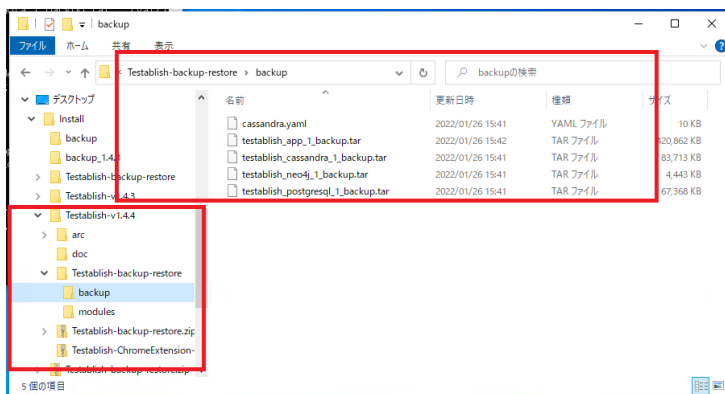


Fig. 1.4.1-4 バックアップフォルダ

1.4.2. アンインストール

Windowsのコントロールパネルのプログラムの追加と削除からTestab\shを削除します。

1.4.3. インストール

[『Testab\sh サーバの構築【Windows10】』](#)を参照してインストールを行ってください。

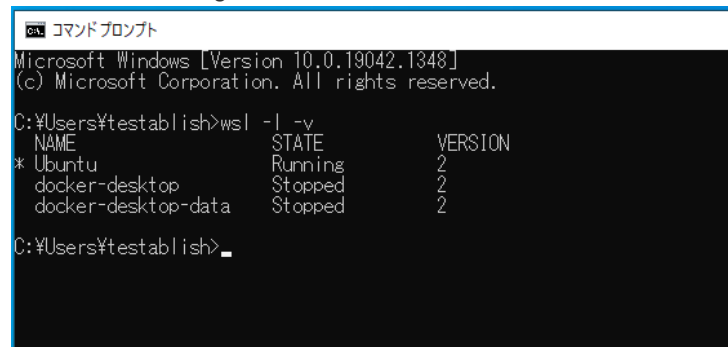
1.4.4. リストア

1 リストア前の確認

コマンドプロンプトからリストアするサーバーの状態を確認します。以下のコマンドを入力して確認します。

```
C:\Users\testabliash>wsl -l -v
NAME STATE VERSION
* Ubuntu Running 2
docker-desktop-data Stopped 2
docker-desktop Stopped 2
```

Ubuntu が Running になっていることを確認してください。



```
コマンドプロンプト
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\testabliash>wsl -l -v
NAME STATE VERSION
* Ubuntu Running 2
docker-desktop Stopped 2
docker-desktop-data Stopped 2

C:\Users\testabliash>
```

Fig. 1.4.4-1 Ubuntuの状態

2 リストアツールを実行する

バックアップツールのフォルダを開きます。

restore.bat とバックアップを行ったときに作成されたbackupファイルが配置されているのを確認します。

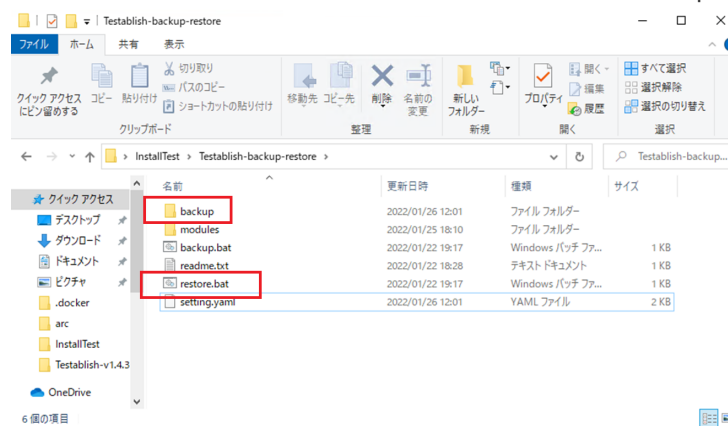
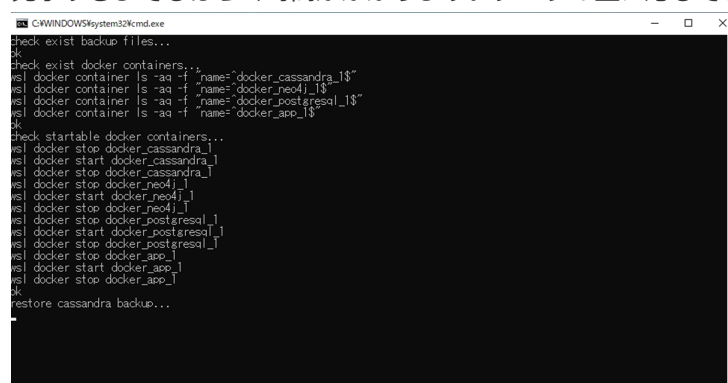


Fig. 1.4.4-2 バックアップフォルダ

restore.bat ファイルをダブルクリックします。リストアが始まります。

完了するまでしばらく時間がかかります。データの量に応じてリストアの時間が変わります。



```
C:\WINDOWS\system32\cmd.exe
check exist backup files...
pk
check exist docker containers...
wsl docker container ls -a -f "name=docker_cassandra_1$"
wsl docker container ls -a -f "name=docker_neo4j_1$"
wsl docker container ls -a -f "name=docker_postgresql_1$"
wsl docker container ls -a -f "name=docker_app_1$"
pk
check startable docker containers...
wsl docker stop docker_cassandra_1
wsl docker start docker_cassandra_1
wsl docker stop docker_cassandra_1
wsl docker start docker_neo4j_1
wsl docker stop docker_neo4j_1
wsl docker start docker_postgresql_1
wsl docker stop docker_postgresql_1
wsl docker start docker_app_1
wsl docker stop docker_app_1
wsl docker start docker_app_1
wsl docker stop docker_app_1
pk
restore cassandra backup...
```

Fig. 1.4.4-3 リストア中

3 リストア完了

リストアが完了すると backup successfully completed! のメッセージが表示されます。

```
C:\WINDOWS\system32\cmd.exe
wsl docker start docker_cassandra_1
wsl docker cp $(wslpath -u 'C:\Users\testabli\h\Desktop\Install\Test\testabli\h-backup-restore\backup\cassandra.yaml')
docker_cassandra_1:/etc/cassandra/cassandra.yaml
wsl docker run --rm --volumes-from docker_cassandra_1 --mount "type=bind,src=$(wslpath -u 'C:\Users\testabli\h\Desktop\Install\Test\testabli\h-backup-restore\backup'),dst=/backup" busybox tar xvf /backup/testabli\h_cassandra_1_backup.tar
wsl docker exec docker_cassandra_1 sh -c 'rm -rf /var/lib/cassandra/data/system/*'
wsl docker stop docker_cassandra_1
wsl docker start docker_cassandra_1
ok
restore neo4j backup...
wsl docker stop docker_neo4j_1
wsl docker run --rm --volumes-from docker_neo4j_1 --mount "type=bind,src=$(wslpath -u 'C:\Users\testabli\h\Desktop\Install\Test\testabli\h-backup-restore\backup'),dst=/backup" busybox tar xvf /backup/testabli\h_neo4j_1_backup.tar
wsl docker start docker_neo4j_1
ok
restore postgresql backup...
wsl docker stop docker_postgresql_1
wsl docker run --rm --volumes-from docker_postgresql_1 --mount "type=bind,src=$(wslpath -u 'C:\Users\testabli\h\Desktop\Install\Test\testabli\h-backup-restore\backup'),dst=/backup" busybox tar xvf /backup/testabli\h_postgresql_1_backup.tar
wsl docker start docker_postgresql_1
ok
restore app backup...
wsl docker stop docker_app_1
wsl docker run --rm --volumes-from docker_app_1 --mount "type=bind,src=$(wslpath -u 'C:\Users\testabli\h\Desktop\Install\Test\testabli\h-backup-restore\backup'),dst=/backup" busybox tar xvf /backup/testabli\h_app_1_backup.tar
wsl docker start docker_app_1
ok
restore successfully completed!
```

Fig. 1.4.4-4 リストア完了

4 リストアの確認

ブラウザでTestabliサーバーを開きます

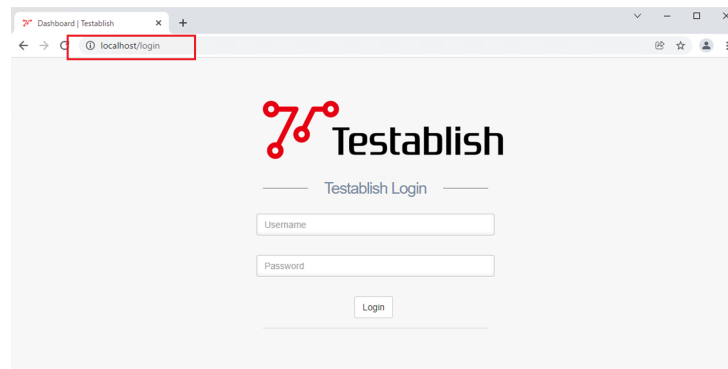


Fig. 1.4.4-5 サーバー画面

ログインするとウィザードの画面が表示されます。

右下の Next をクリックして最後まで進めます。

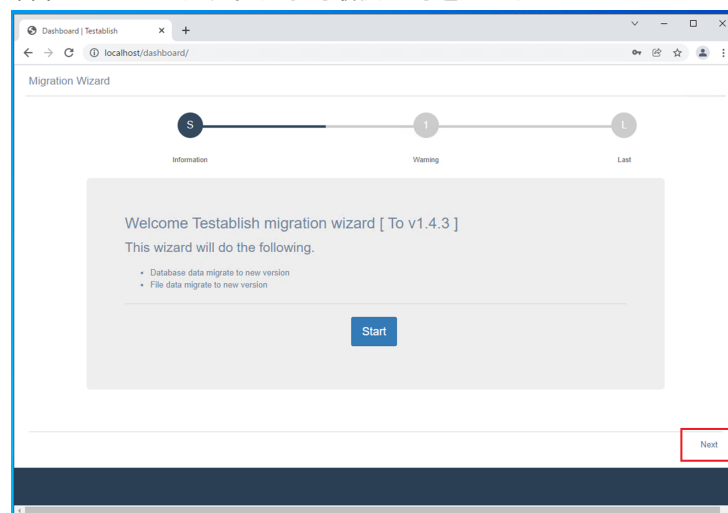


Fig. 1.4.4-6 wizard画面1

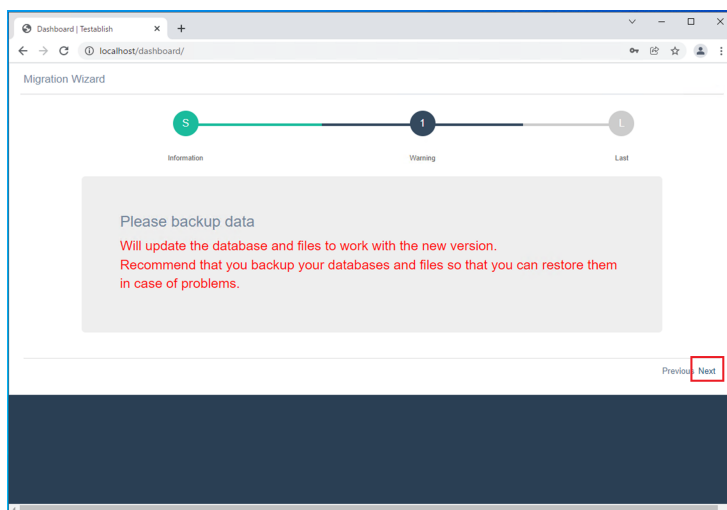


Fig. 1.4.4-6 wizard画面2

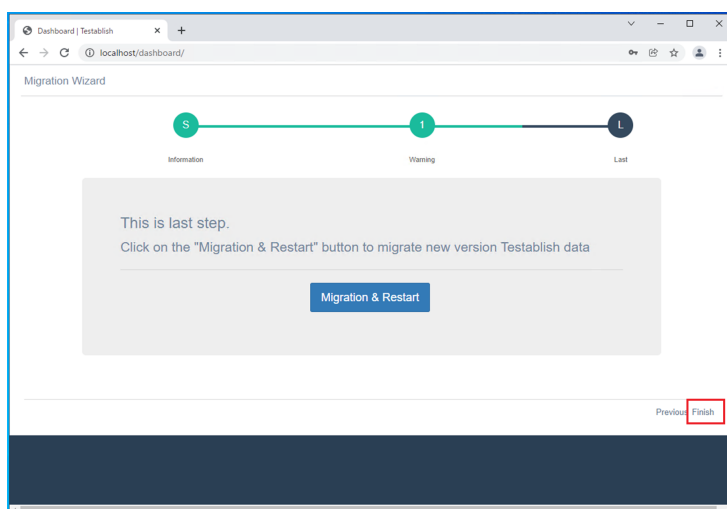


Fig. 1.4.4-7 wizard画面3

Finish まで完了するとログイン画面が表示されますのでログインしてプロジェクトがリストアップされていることを確認します。

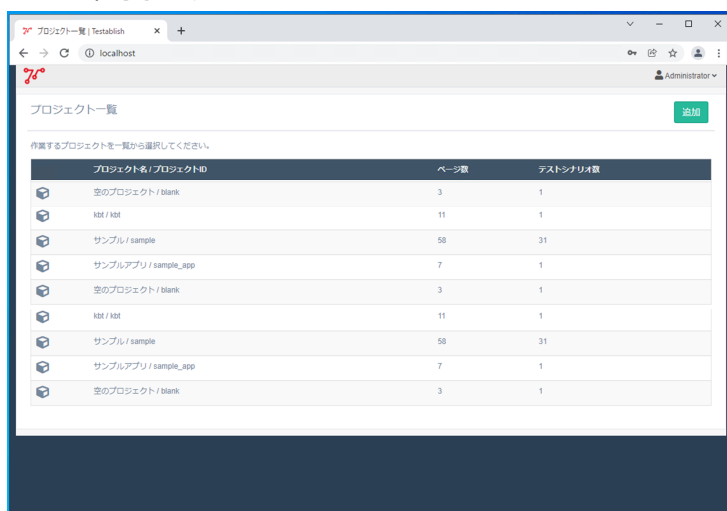


Fig. 1.4.4.-8 プロジェクト画面

2. Testabish サーバの構築【Linux】

LinuxでのTestabliishサーバを構築します。

2.1. Dockerをインストールする

Docker のインストールについては、以下のサイトを参考にしてください。

⇒ Docker Documentation <https://docs.docker.com/>

2.2. Testabliishサーバを構築する

Testabliishサーバを構築していきます。

1. 以下のように Testabliishサーバ用の docker ディレクトリを作成してください。
(任意の場所) /testabliish/docker
2. 以下の tarファイルと ymlファイルを上記の docker ディレクトリに配置してください。
 - **testabliish-pro-vx.x.x.tar** (x.x.x はバージョン番号)
 - **docker-compose.yml**

ディレクトリ構造は以下のようになります。

```
(任意の場所)
├─ testabliish
│   └─ docker
│       ├── testabliish-pro-vx.x.x.tar
│       └─ docker-compose.yml
```

イメージの導入からコンテナを作成し、起動します。
ターミナルを開き、以下のコマンドを使って作成を行ってください。

- docker ディレクトリに移動します。

```
$ cd (任意のディレクトリ) /testabliish/docker
```
- DockerイメージファイルをDockerに取り込みます。

```
$ docker load -i ./testabliish-pro-vx.x.x.tar
```
- イメージが取り込めたことを確認してください。
REPOSITORY に **(Testabliish アドレス) /testabliish-pro** が表示されていれば取り込み完了です。

```
$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
121.1.152.66:80/testabliish-pro	latest	7ff98e508e66	2 weeks ago	1.26GB
121.1.152.66:80/testabliish-pro	v1.4.2	7ff98e508e66	2 weeks ago	1.26GB
docker4w/nsenter-dockerd	latest	2f1c802f322f	8 months ago	187kB

- コンテナ設定ファイルをもとにコンテナを作成します。

```
$ docker-compose up -d
```

```
Creating network "docker_default" with the default driver
Creating volume "docker_app" with local driver
Creating volume "docker_cassandra" with local driver
```

```

Creating volume "docker_postgresql" with local driver
Creating volume "docker_neo4j" with local driver
Pulling neo4j (neo4j:3.1)...
3.1: Pulling from library/neo4j
4fe2ade4980c: Pull complete
6fc58a8d4ae4: Pull complete
819f4a45746c: Pull complete
6b424cc95557: Pull complete
5226b1091189: Pull complete
e3530dee375a: Pull complete
26bc5bf1aa3e: Pull complete
Pulling cassandra (cassandra:3)...
3: Pulling from library/cassandra
fc7181108d40: Pull complete
5d30a29d1ca7: Pull complete
71b1da59a33e: Pull complete
eef322d59447: Pull complete
1b972645cba8: Pull complete
adcafb3ea5c7: Pull complete
a1f579d537e1: Pull complete
d6d4053ef622: Pull complete
ebe94652798a: Pull complete
0db0a9a149fd: Pull complete
Pulling postgresql (postgres:9.6)...
9.6: Pulling from library/postgres
fc7181108d40: Already exists
81cfa12d39e9: Pull complete
793d305ca761: Pull complete
d1465f4e119c: Pull complete
32ecb3afd85d: Pull complete
e79a18e1d8ce: Pull complete
c9527f80b877: Pull complete
ab10bee9b69f: Pull complete
f0ed14592ade: Pull complete
78a22e18ffb9: Pull complete
9d42fc2722c1: Pull complete
82fc6cf195b8: Pull complete
5daba5b03264: Pull complete
0e1e7117e844: Pull complete
Creating docker_neo4j_1 ... done
Creating docker_postgresql_1 ... done
Creating docker_cassandra_1 ... done
Creating docker_app_1 ... done

```

- コンテナの作成が正常に終了したら、コンテナを確認してください。

```
$ docker ps -a
```

CONTAINER ID	IMAGE	~ STATUS	~ NAMES
08b8ab7556e0	121.1.152.66:80/testabliish-pro:v1.4.2	~ Up 6 minutes	~ docker_app_1
4d268f93df2e	cassandra:3	~ Up 6 minutes	~ docker_cassandra_1
b91ecfaa65d2	postgres:9.6	~ Up 6 minutes	~ docker_postgresql_1
e58448f5190f	neo4j:3.1	~ Up 6 minutes	~ docker_neo4j_1

- 以下の4つのコンテナが表示されていれば作成完了です。

- **docker_app_1**
- **docker_postgresql_1**
- **docker_neo4j_1**
- **docker_cassandra_1**

- 各コンテナの STATUS が **Up** になっていればコンテナが起動している状態です。
- 停止したdockerコンテナを起動するときも、同じコマンドを使用してください。

```
$ docker-compose up -d
```

```
Starting docker_neo4j_1      ... done
Starting docker_cassandra_1  ... done
Starting docker_postgresql_1 ... done
Starting docker_app_1        ... done
```

2.3. Testabliishサーバをバージョンアップする

新しいバージョンのTestabliishがリリースされたときなど、**既に稼働しているTestabliishサーバをバージョンアップする** 手順です。

現在 Docker 上で稼働している Testabliish サーバを停止・削除し、新しいバージョンの Testabliishサーバを構築します。

1. 起動しているコンテナを停止します。

docker ps コマンドでコンテナのIDを確認し、docker stop コマンドでコンテナを停止します。

```
$ docker ps
```

```
$ docker stop [コンテナID]
```

2. コンテナの停止を確認します。

各コンテナの STATUS が Exited になっていればコンテナが停止している状態です。

```
$ docker ps -a
```

CONTAINER ID	IMAGE	STATUS	NAMES
08b8ab7556e0	121.1.152.66:80/testabliish-pro:v1.4.2	Exited (137) 13 seconds ago	
docker_app_1			
4d268f93df2e	cassandra:3	Exited (137) 11 seconds ago	
docker_cassandr			
b91ecfaa65d2	postgres:9.6	Exited (0) 12 seconds ago	
docker_postgres			
e58448f5190f	neo4j:3.1	Exited (143) 12 seconds ago	
docker_neo4j_1			

3. コンテナを削除します。

```
$ docker rm [コンテナID]
```

4. イメージを削除します。

- ・イメージIDを確認します。

```
$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	9.6	edb112fd2766	4 days ago	230MB
cassandra	3	a34c036183d1	4 days ago	323MB
121.1.152.66:80/testabliish-pro	latest	7ff98e508e66	2 weeks ago	1.26GB
121.1.152.66:80/testabliish-pro	v1.4.2	7ff98e508e66	2 weeks ago	1.26GB
neo4j	3.1	89de6a594a03	8 months ago	177MB
docker4w/nsenter-dockerd	latest	2f1c802f322f	8 months ago	187kB

- ・イメージIDを指定して、イメージを削除します。

```
$ docker rmi -f [イメージID]
```

5. バージョンアップファイルを配置します。

[2.2. Testabliishサーバを構築する](#)で設定したTestabliishサーバ用dockerフォルダに、バージョンアップするDockerイメージファイルとコンテナ設定ファイルを置いてください。

ディレクトリ構造は以下のようになります。(x.x.x は バージョン番号)

```
(任意の場所)
└─ testabliish
   └─ docker
      ├── testabliish-pro-vx.x.x.tar
      └─ docker-compose.yml
```

6. 新しいバージョンをインストールします。

[2.2. Testabliishサーバを構築する](#)を参照し、新しいバージョンをインストールしてください。

7. 起動確認

インストールができれば、<http://localhost/> に接続してください。

8. マイグレーション

ログイン画面が表示されますので **管理者権限のアカウント** でログインしてください。



Fig. 2.4-2 バージョンアップ ログイン

Migration Wizard が表示されますので、**Start** ボタンか **Next** をクリックして画面を進めてください。

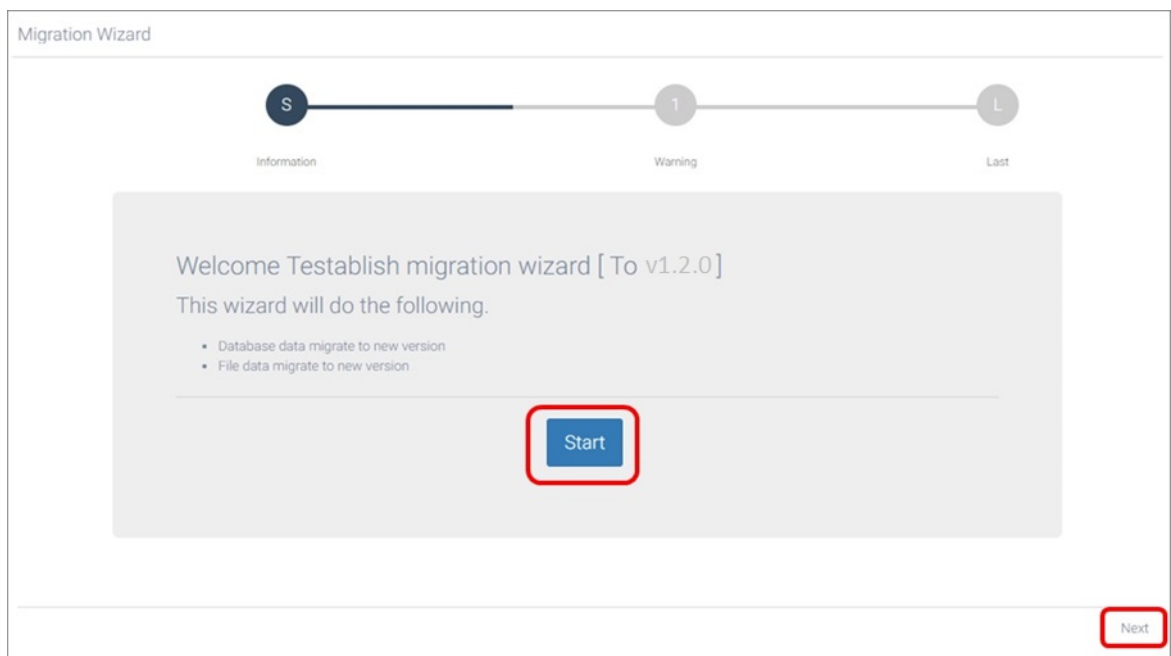


Fig. 2.4-3 Migration Wizard information

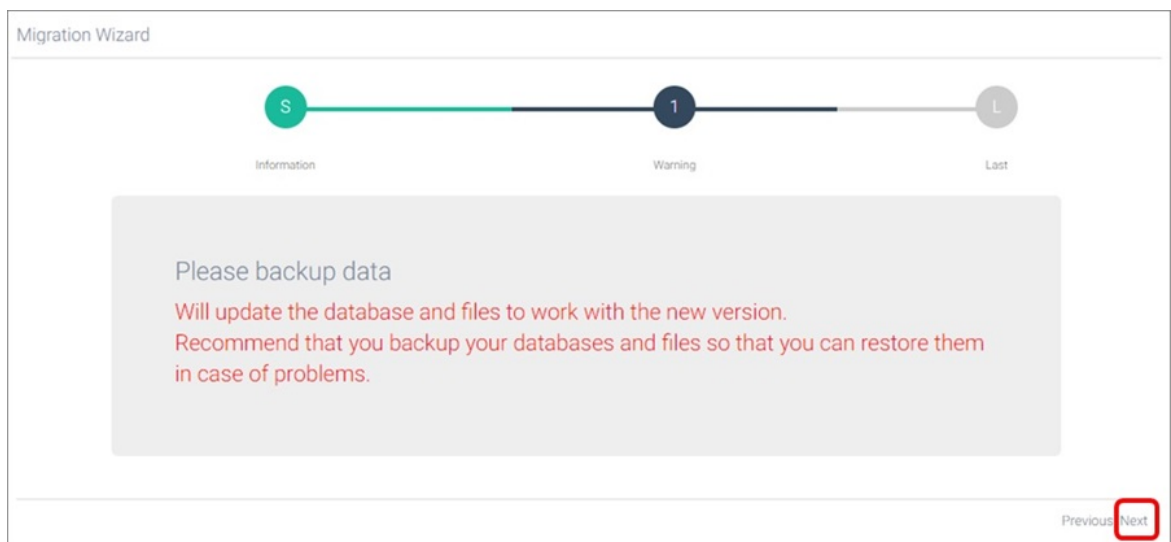


Fig. 2.4-4 Migration Wizard Warning

ラストページまで進めたら、**Migration & Restart** ボタンをクリックしてください。

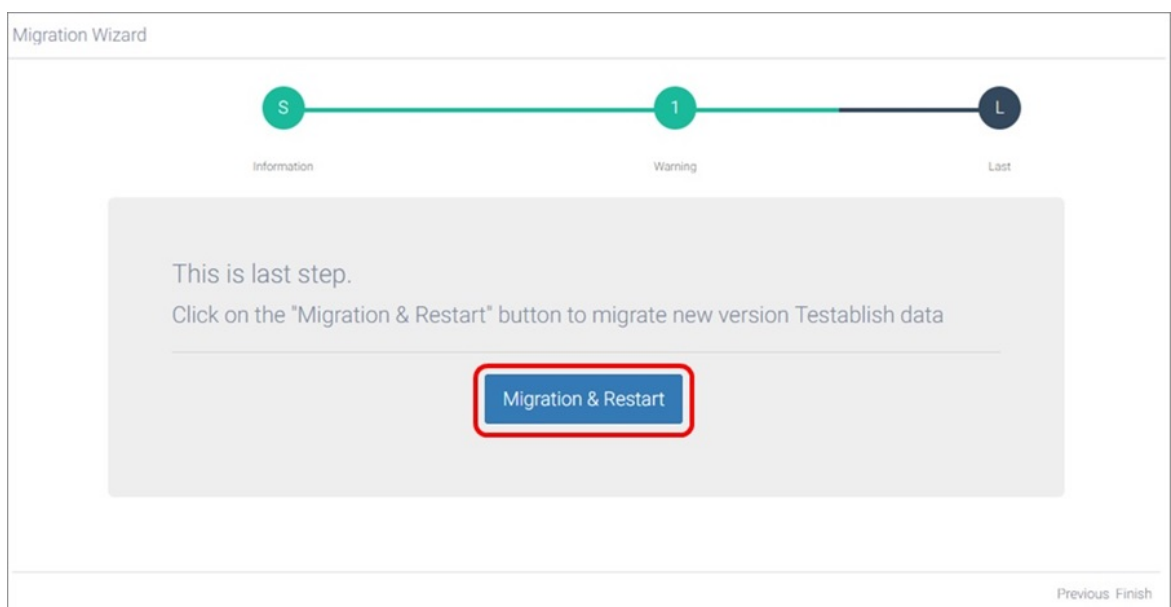


Fig. 2.4-5 Migration Wizard Last

確認ダイアログが表示されますので、**OK** をクリックしてください。

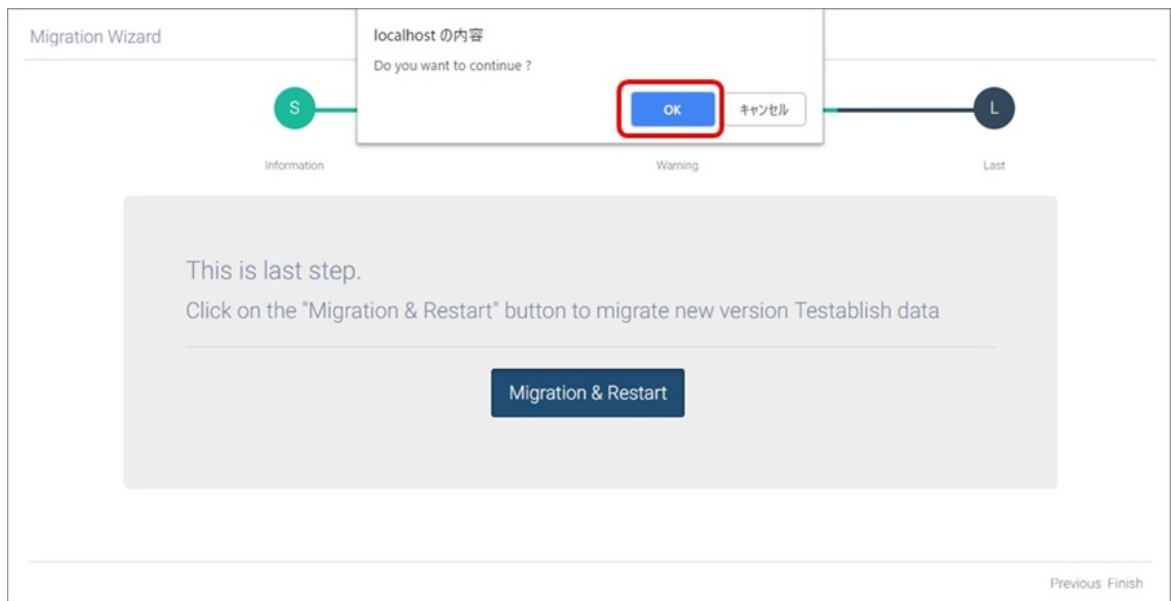


Fig. 2.4-6 Migration Wizard ダイアログ

Migration successful のポップアップが表示されたら成功です。

この後サーバのシステムが再起動しますので、しばらく接続が落ちますがそのままお待ちください。
サーバの再起動が完了するとログイン画面に変わりますので、ログインしてご使用ください。

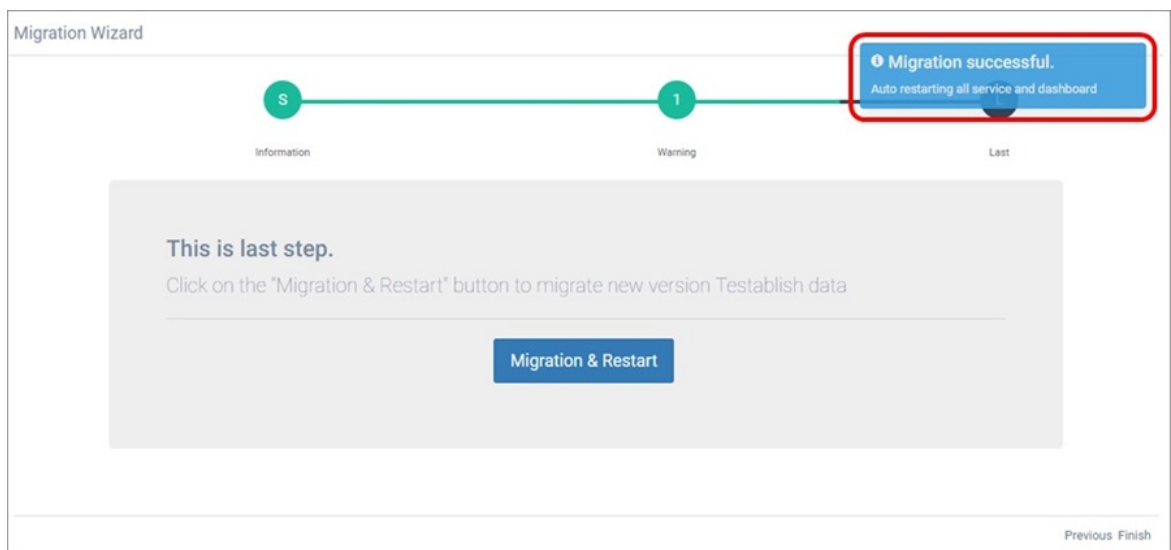


Fig. 2.4-7 Migration Wizard succesful

アップデートの内容によっては **Setup Wizard** が表示されることがありますが、[1.4. Testabliishサーバの起動確認をする](#) を参照して進めてください。

[目次](#) に戻る

II. Extension(拡張機能)の導入

- [1. Windowsでの拡張機能導入](#)
 - [1.1. Testabliish タスクトレイアプリのインストール](#)
 - [1.1.1. インストーラの指示に従ってインストールする](#)
 - [1.1.2. Testabliish タスクトレイアプリを起動する](#)
 - [1.1.3. Testabliish タスクトレイアプリの設定をする](#)
 - [1.2. Google Chromeでの拡張機能設定](#)
 - [1.2.1. Chrome Extensionを入手・導入する](#)
 - [1.2.2. ブラウザポップアップを許可する](#)
 - [1.2.3. Chrome Extensionのオプション設定](#)
 - [1.3. Internet Explorerでの拡張機能設定](#)
 - [1.3.1. Internet Explorer でのアドオン\(TestabliishBho Class\)の有効化](#)
 - [1.3.2. Internet Explorer での「Windows Defender SmartScreen」設定](#)
 - [1.3.3. Internet Explorer での互換表示設定の解除](#)
 - [1.3.4. Internet Explorer でのセキュリティ設定](#)

1. Windowsでの拡張機能導入

1.1. Testabliish タスクトレイアプリのインストール

Testabliish タスクトレイアプリは、ブラウザ上の操作を Testabliishサーバに取り込むための拡張機能です。タスクトレイアプリは情報収集(キャプチャ)を行うクライアントのPC (Windows10)に導入して利用します。

タスクトレイアプリでは、情報収集(キャプチャ)を行うための**各種サービスを起動・停止**を行います。

また、情報収集(キャプチャ)を行う対象Webアプリケーションの設定もタスクトレイアプリで行います。

この他、**Internet Explorer** での情報収集(キャプチャ)の起動・停止を行うことができます。

Chromeでの情報収集(キャプチャ)の起動・停止は、別途 Chrome用ブラウザ拡張機能をインストールして行います。⇒ [1.2. Google Chromeでの拡張機能設定](#)

1.1.1. インストーラの指示に従ってインストールする

インストーラのファイル名は **Testabliish_Task_tray_Setup_<VersionNo>.exe** です。

インストーラを起動し、インストール ウィザードに従って拡張機能をインストールしていきます。

このインストーラでは、Testabliish タスクトレイアプリ本体のインストールの他、Internet Explorer での情報収集(キャプチャ)を行うための設定や、SSL通信を利用するための証明書のインストールも同時に行います。

インストールは以下の順序で進みます。

1. Welcome画面
2. ライセンス画面
3. カスタム設定
4. インストール先選択
5. インストール
6. 証明書インストール
7. Internet Explorer の設定
8. 完了画面

7. Internet Explorer の設定 では、以下の項目を設定しています。

※ Internet Explorer 11がインストールされていない場合はスキップされます。

- Internet Explorer でのアドオン(TestabliSh Bho Class)の有効化
- Internet Explorer での「Windows Defender SmartScreen」設定有効化
- Internet Explorer での互換表示設定の解除
 - ・「イントラネット サイトを互換表示で表示する」の無効化
 - ・「Microsoft互換性リストの使用」の無効化
- Internet Explorer でのインターネットオプション設定
 - セキュリティタブ：各ゾーンの「保護モード」を有効化：
「イントラネット」「信頼済みサイト」「インターネット」「制限付きサイト」
 - 詳細設定タブ：
 - ・設定 > セキュリティ
 - ・「拡張保護モードを有効にする」の無効化
 - ・「拡張保護モードで64ビットプロセッサを有効にする」の無効化
 - ・設定 > ブラウズ
 - ・「サードパーティ製のブラウザー拡張を有効にする」の有効化

何らかの問題でエラーが発生した場合、メッセージエリアにエラー内容が表示されます。

発生したエラーに対応する方法については、II. Extension(拡張機能)の導入 > [1.3. Internet Explorerでの拡張機能設定](#)を参照してください。

1.1.2. TestabliSh タスクトレイアプリを起動する

デスクトップに作成された TestabliSh のショートカットアイコンをダブルクリックします。



Fig. 1.1.2-1 TestabliSh ショートカットアイコン

起動すると**タスクトレイ**に**常駐**します。

1.1.3. TestabliSh タスクトレイアプリの設定をする

TestabliSh タスクトレイアプリが起動したら、各種項目設定を行います。

タスクトレイの上向きの三角をクリックして表示された TestabliSh アイコンをクリックし、表示された「**TestabliSh タスクトレイアプリ**」画面で各項目を設定します。

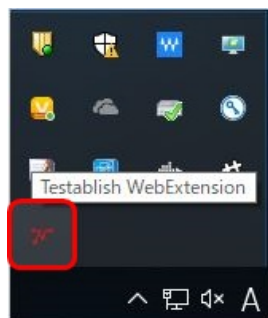


Fig. 1.1.3-1 TestabliSh タスクトレイアイコン

TestabliSh のタスクトレイアプリは、**ローカルプロキシ** と **バックサービス** の二つを起動します。
ブラウザ拡張機能によりブラウザのインターネットプロキシの接続先がローカルプロキシに置き換えら

れ、ローカルプロキシはやり取りされたHTMLや通信(http request/response)の内容を Testablish サーバの **Importer** へ送り、これらはサーバに蓄積されます。

また、ブラウザ拡張機能により「ボタンをクリックした」「文字を入力した」等の操作ログがサーバの **Importer** へ送られますが、Internet Explorerは一旦 **バックサービス** を経由して **Importer** と通信します。

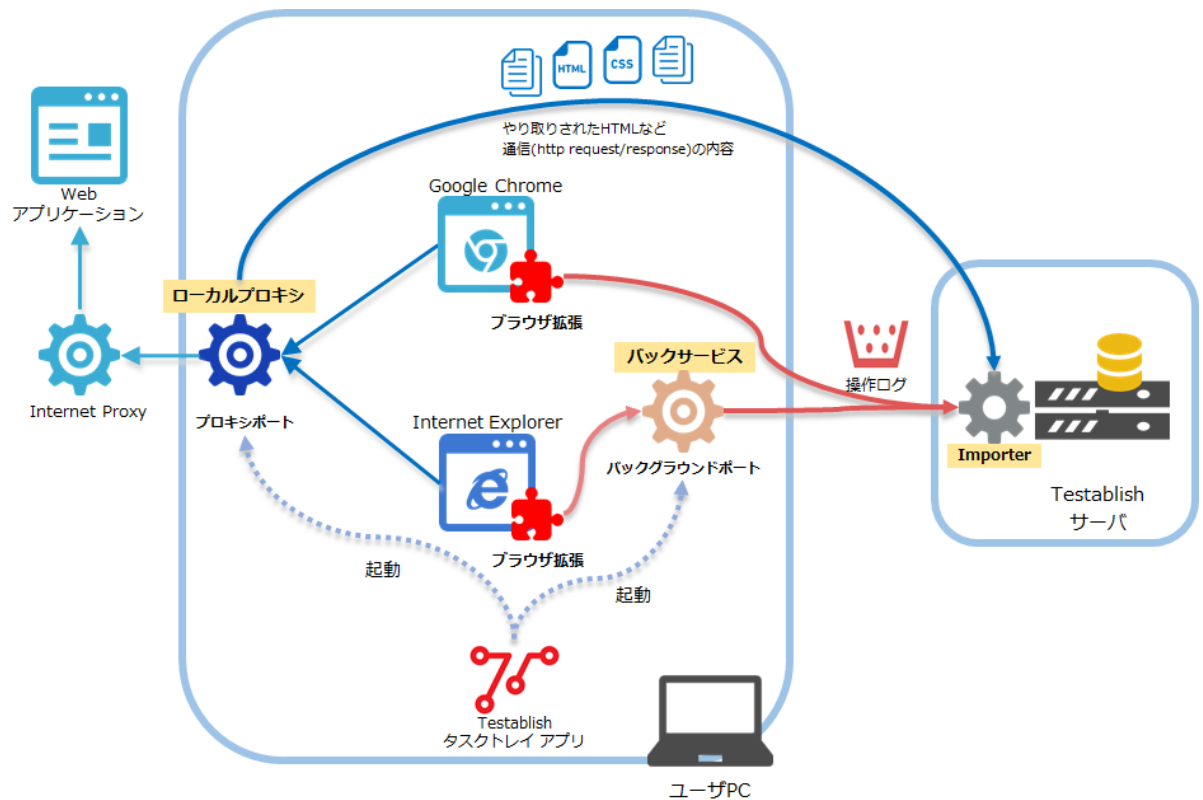


Fig. 1.1.3-2 Testablish ブラウザ拡張機能とデータの流れ

タスクトレイアプリの各タブのそれぞれの項目を設定していきます。

◦ WebExtension

WebExtension (タスクトレイアプリ) の設定を行います。

設定できたら、必ず保存してください。

Fig. 1.1.3-3 WebExtentionタブ

バックグラウンド	
バックグラウンドポート	バックサービスが利用するポート。 PCの空いているポートを指定してください。 デフォルト値は "12444" です。
ローカルプロキシ	
ローカルプロキシポート	ローカルプロキシが利用するポート。 PCの空いているポートを指定してください。 デフォルト値は "3128" です。
インターネットプロキシ	
プロキシなし	プロキシを使用しない場合に選択します。
システムプロキシ	システムのプロキシ設定を利用する場合に選択します。
手動プロキシ	手動でプロキシを設定する場合に選択します。
インターネットプロキシ	ブラウザのプロキシを指定します。
例外アドレス	インターネットプロキシを利用しないアドレスを指定します。 複数指定する場合は コンマ(,)で区切ってください。

。サーバ

Testabishサーバの設定を行います。

設定できたら、必ず保存してください。

The screenshot shows the 'Testabishタスクトレイアプリ' window with the 'サーバ' (Server) tab selected. The configuration fields are as follows:

- Host: localhost
- Port: 80
- User ID: admin
- Password: [Masked with dots]
- Project ID: installer_test (selected from a dropdown)
- Auth Token: 9ff0f48d92b94356a65e8c276d96b2a01e8a47074ae7d42a8ef1

Buttons at the bottom: 保存 (Save), キャンセル (Cancel).

Fig. 1.1.3-4 Extension サーバタブ

ホスト	Testabishサーバのアドレスを指定します。
ポート	Testabishサーバで動いている UI のポートを指定します。※1 デフォルト値は "80" です。
ユーザID	Testabish を利用するユーザIDを指定します。
パスワード	上記ユーザのパスワードを指定します。
プロジェクトID	指定しているホストの Testabishサーバ上に登録されているプロジェクトのID をプルダウンから選択します。

	(先にTestabliishサーバの構築とプロジェクト作成を行っておく必要があります。)
認証トークン	選択したプロジェクトIDの認証トークンが表示されます。

※1 Testabliish v1.4.1 まではこのポートはImporterのポート番号を設定していましたが、v1.4.2 以降では UI のポート番号の指定に変更されました。バージョンアップされた場合にはこのポート番号の設定を確認することをお勧めします。

。サービス

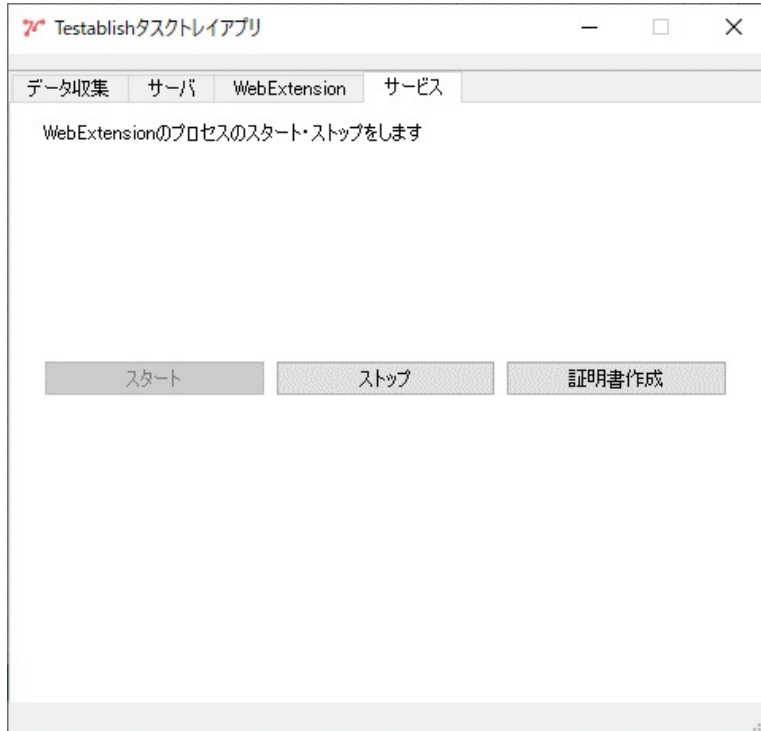


Fig. 1.1.3-5 Extension サービスタブ

スタートボタン	Testabliish タスクトレイアプリのサービス(ローカルプロキシ・バックサービス)を起動します。
ストップボタン	Testabliish タスクトレイアプリのサービス(ローカルプロキシ・バックサービス)を停止します。
証明書作成ボタン	ローカルプロキシの証明書を生成します。 生成した証明書のインストールについては、IV. 証明書のインストールを参照してください。 なお、タスクトレイアプリをインストールする際に証明書の生成・インポートが行われています。(v1.4.2～)

。データ収集

Internet Explorerでのデータ収集開始・終了を行います。

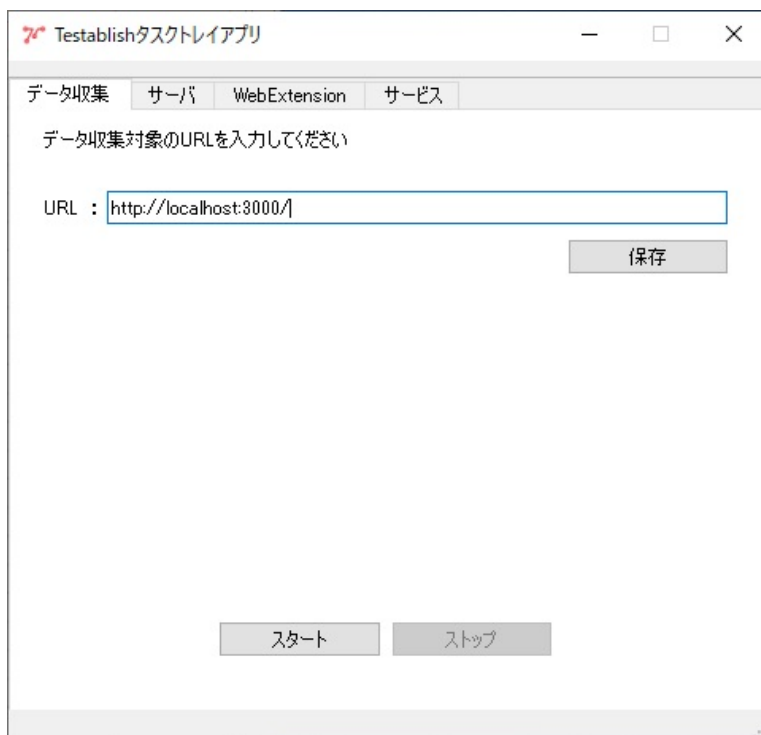


Fig. 1.1.3-6 Extension データ収集タブ

URL	データ収集対象のURLを入力します。
保存ボタン	入力されている URL を保存します。
スタートボタン	Internet Explorer が起動して上記URLで指定したサイトを表示し、操作記録のデータ収集を開始します。
ストップボタン	Internet Explorer での操作記録データ収集を終了します。

1.2. Google Chromeでの拡張機能設定

Google Chrome で操作情報の収集(キャプチャ)を行う場合は、別途 ブラウザへの拡張機能の導入と設定が必要になります。

1.2.1. Chrome Extensionを入手・導入する

chrome ウェブストアの [Testabish WebExtension のページ](#) にアクセスします。

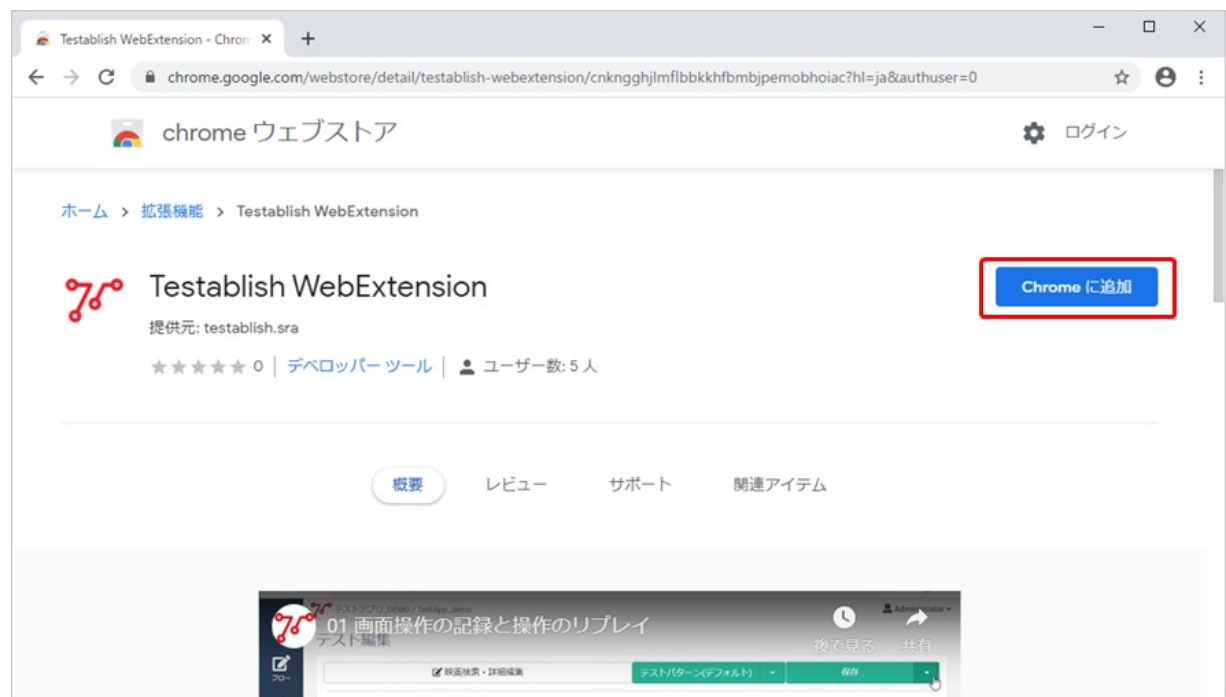


Fig. 1.2.1-1 Chrome ウェブストア

「Chromeに追加」ボタンをクリックしてブラウザにインストールします。
アドレスバーの右にTestabliishのロゴアイコンが表示されているのを確認します。



Fig. 1.2.1-2 Testabliishのロゴアイコン

何らかの理由で chrome ウェブストアを利用しない場合は、インストールパッケージに含まれている zip ファイルから手動で導入することができます。

1. zip ファイルを入手・展開する

Google Chrome への機能拡張の導入には、**Testabliish-ChromeExtension-<VersionNo>.zip** を使用します。

入手した zip ファイルを展開します。(展開する場所はどこでも構いません。)

2. Chromeで拡張機能を設定する

Google Chrome を起動し、アドレスバーの右にあるメニューボタンから、「**その他のツール(L) > 拡張機能(E)**」を選択します。

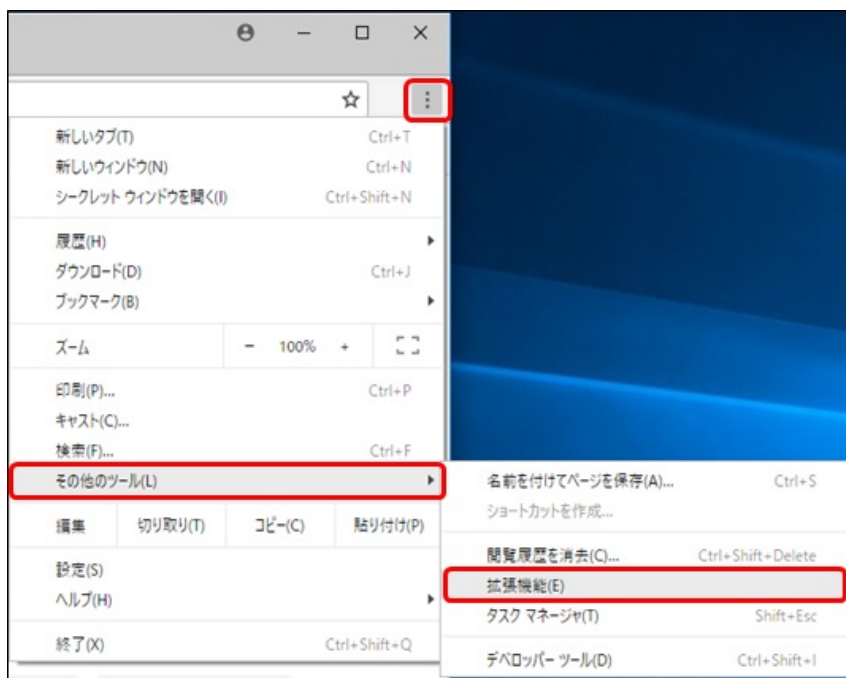


Fig. 1.2.1-3 その他のツール(L) > 拡張機能(E)

「デベロッパーモード」をONにします。

「パッケージ化されていない拡張機能を読み込む」をクリックし、フォルダの参照ウィンドウから、先ほど展開した拡張機能の **フォルダ** を指定します。

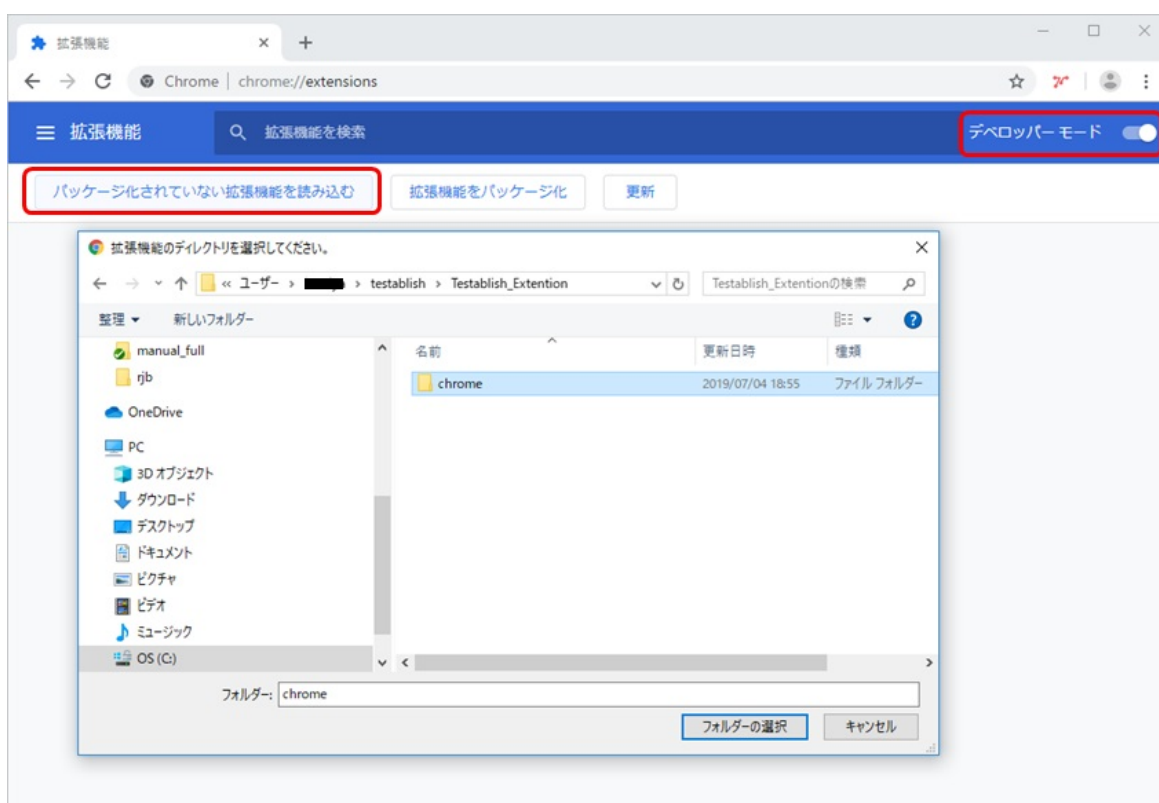


Fig. 1.2.1-4 パッケージ化されていない拡張機能を読み込む

アドレスバーの右にTestablishのロゴアイコンが表示されているのを確認します。

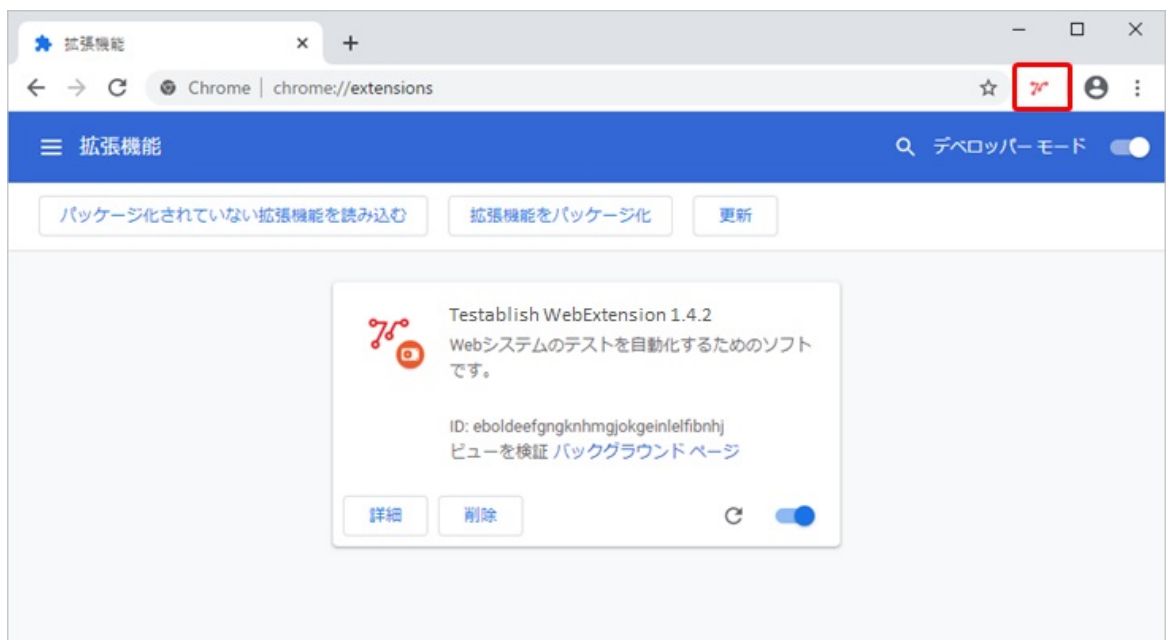


Fig. 1.2.1-5 Testablishのロゴアイコン

1.2.2. ブラウザポップアップを許可する

アドレスバーの右にあるメニューボタンから「設定(S)」を選択し、表示された設定画面下部の「詳細設定 ▼ > プライバシーとセキュリティ > サイトの設定 > ポップアップとリダイレクト」より、ポップアップを「許可」してください。



Fig. 1.2.2-1 メニュー > 設定(S)

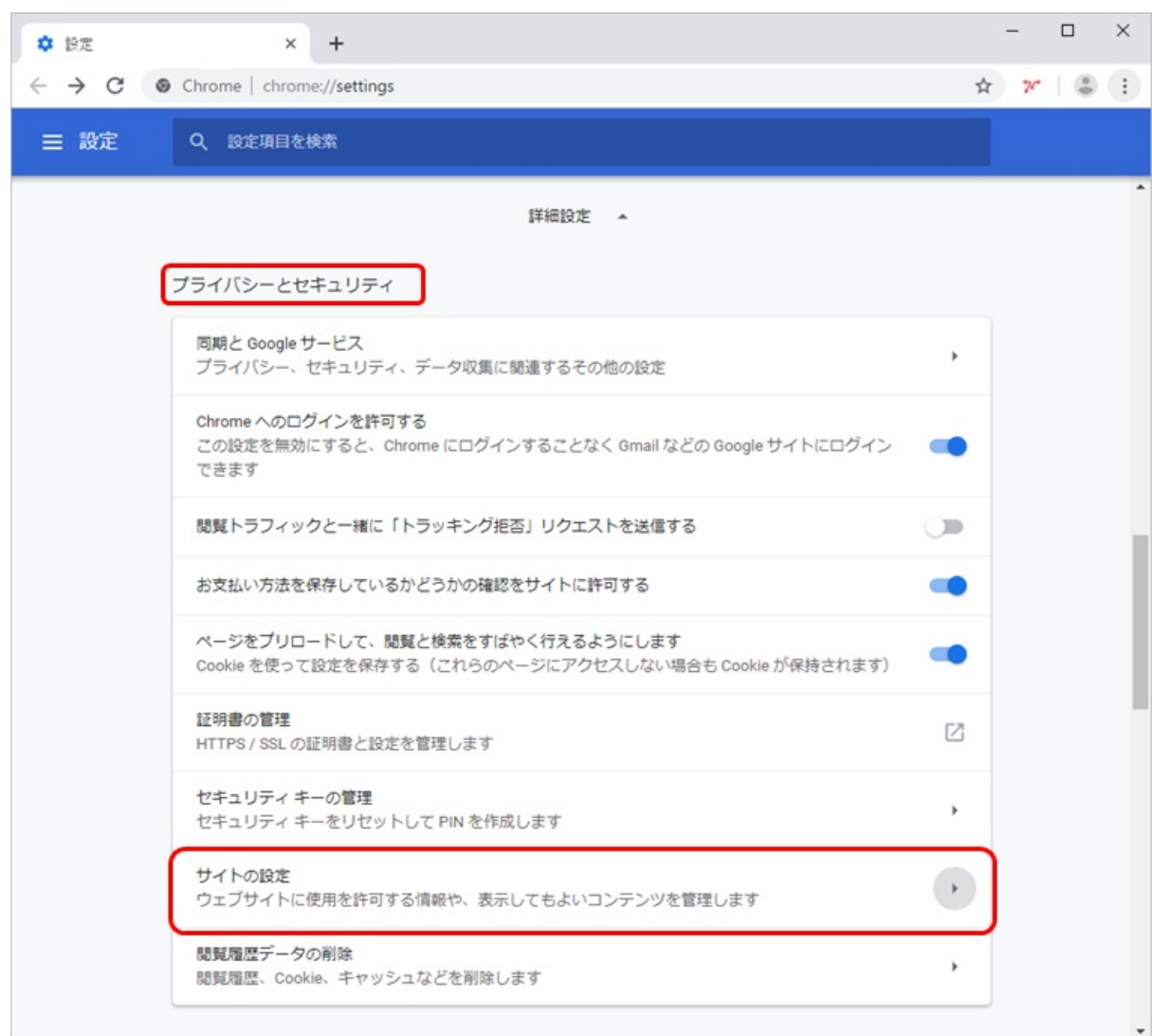


Fig. 1.2.2-2 プライバシーとセキュリティ - サイトの設定

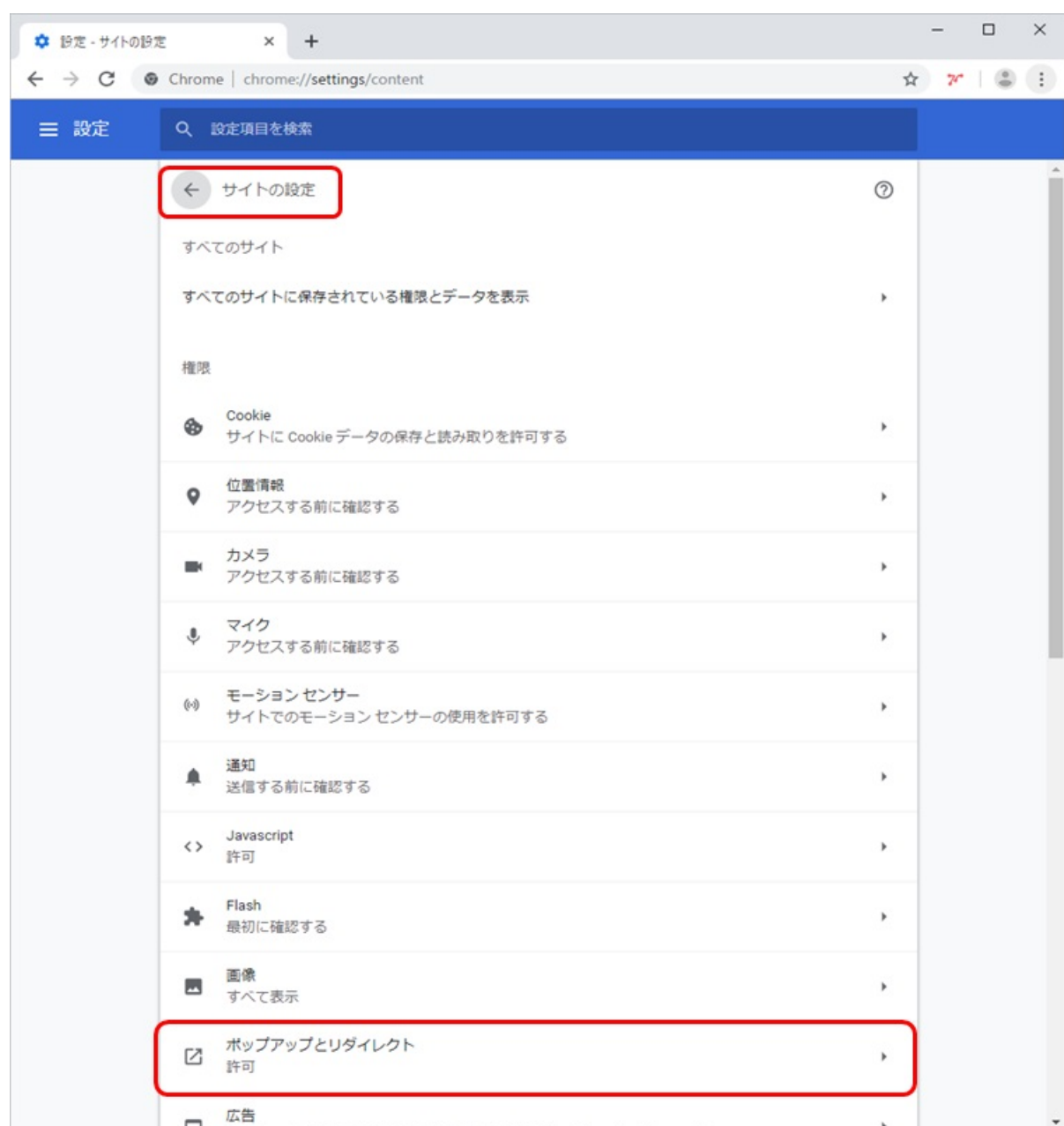


Fig. 1.2.2-3 サイトの設定 - ポップアップ

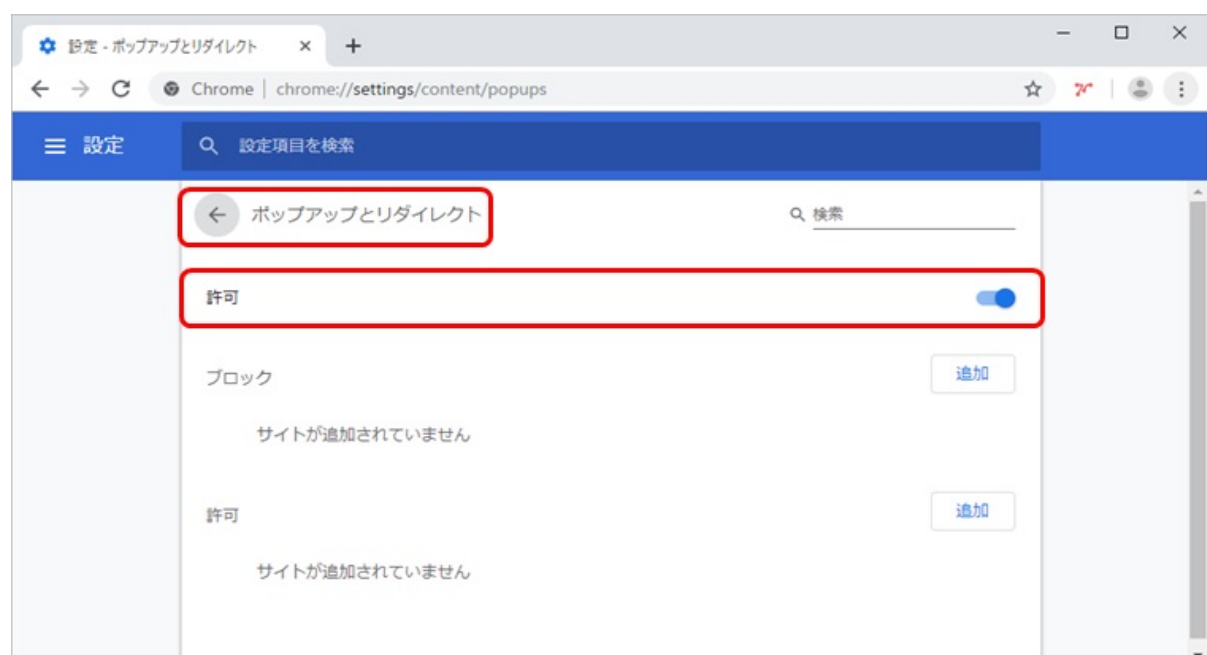


Fig. 1.2.2-4 ポップアップ - 許可

1.2.3. Chrome Extensionのオプション設定

1. Testabishのロゴアイコンを右クリックして表示されるメニューから、**オプション** を選択します。

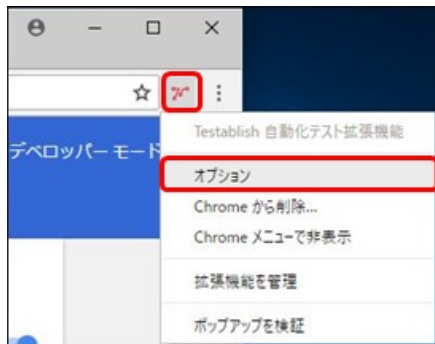


Fig. 1.2.3-1 オプション

2. ポップアップした「Testabish WebExtension」ダイアログでローカルプロキシポートを設定します。



Fig. 1.2.3-2 「Testabish WebExtension」ダイアログ

各設定項目の内容は、以下の通りです。

プロキシポート	ローカルプロキシが利用するポート番号を指定します。 タスクトレイアプリのローカルプロキシポートの設定と一致させてください。 未入力の場合は 3128番ポートを利用します。
----------------	---

設定後、データ収集をおこなう方法については、

別マニュアル：Testabish 使い方マニュアル » III. 操作の流れ » 1. ページ情報および操作の収集を参照してください。

1.3. Internet Explorerでの拡張機能設定

Internet Explorer で拡張機能を利用する場合、以下の設定が必要です。

これらの設定は タスクトレイアプリのインストール時に設定されますが、何らかの理由で設定できていない場合は、後述の設定方法を確認してください。

1.3.1. Internet Explorer でのアドオン(TestabishBho Class)の有効化

TestabishWebExtensionは Internet Explorer のアドオンであるため、これが有効であることを確認します。

1. Internet Explorer を起動し、画面右上の「ツール」（歯車マーク）をクリックします。

2. 表示された一覧から「インターネットオプション」を選択します。

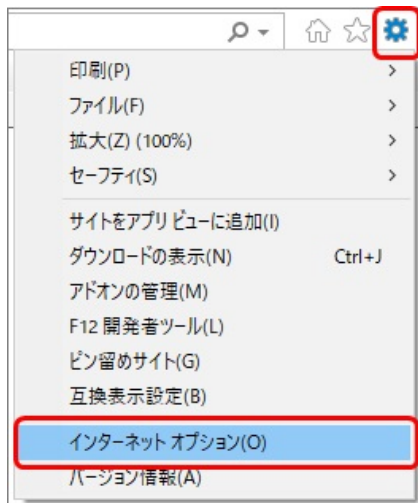


Fig. 1.3.1-1 「ツール(T) > インターネット オプション(O)」

3. 表示された「インターネットオプション」ダイアログで「プログラム」タブを開き、「アドオンの管理」ボタンをクリックします。

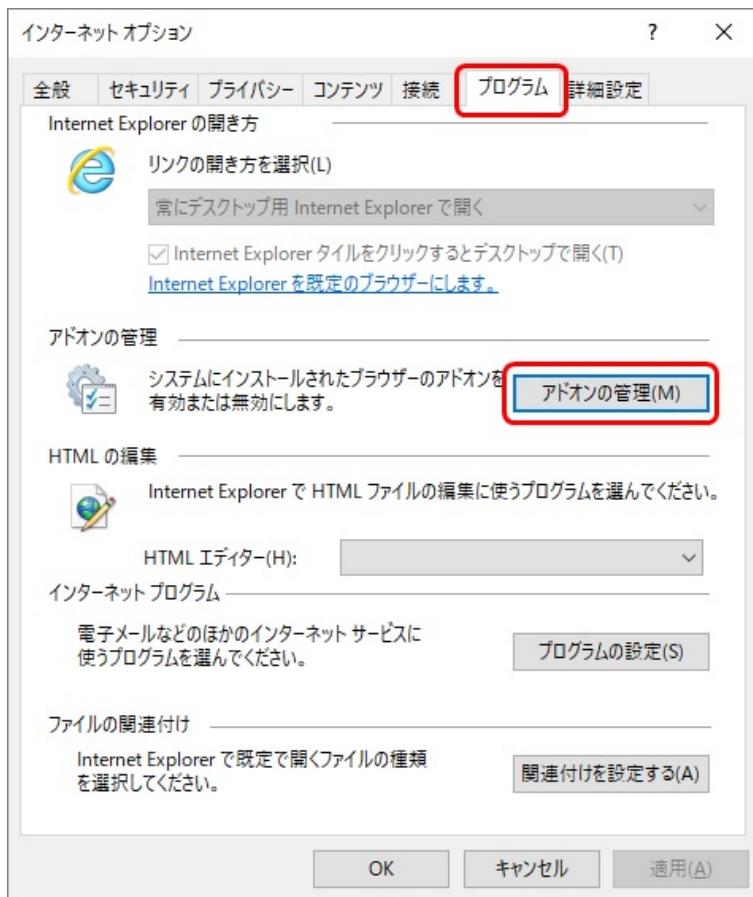


Fig. 1.3.1-2 「プログラムタブ > アドオンの管理(M)」

4. 「TestablishBho Class」が、有効になっていることを確認します。無効になっている場合は、有効に設定してください。



Fig. 1.3.1-3 「アドオンの管理ダイアログ」

1.3.2. Internet Explorer での「Windows Defender SmartScreen」設定

Internet Explorer で、**Windows Defender SmartScreen 機能** を **有効** にします。

Windows Defender SmartScreenとは、フィッシング詐欺サイトや悪意のあるWebページへのアクセス、ウイルスやスパイウェアなどのダウンロードをブロックする機能です。

設定を有効にすると、該当するWebページを表示した際に警告画面が表示されます。

TestablishWebExtensionでは、このSmartScreenをハンドルしているため、有効にしておく必要があります。

1. Internet Explorer を起動し、画面右上の「ツール」（歯車マーク）をクリックします。
2. 表示されたメニューから「セーフティ」にマウスポインタを合わせて、「Windows Defender SmartScreenを有効にする」をクリックします。

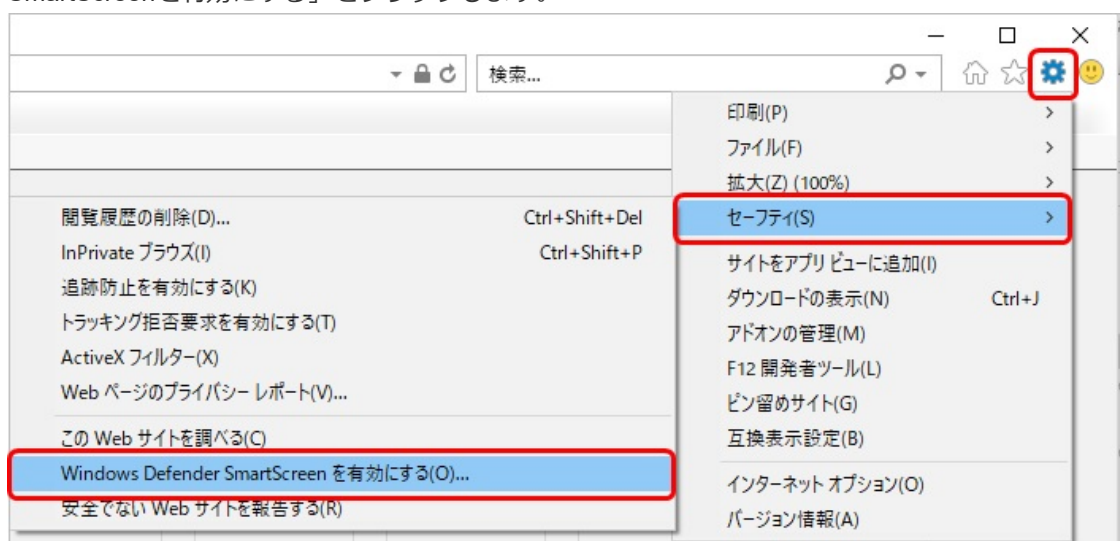


Fig. 1.3.2-1 「ツール > セーフティ(S) > Windows Defender SmartScreenを有効にする」

※ 「Windows Defender SmartScreenを無効にする」と表示された場合、Windows Defender SmartScreenは有効に設定されています。

- 表示されたダイアログで「Windows Defender SmartScreenを有効にする」をクリックし、「OK」をクリックします。

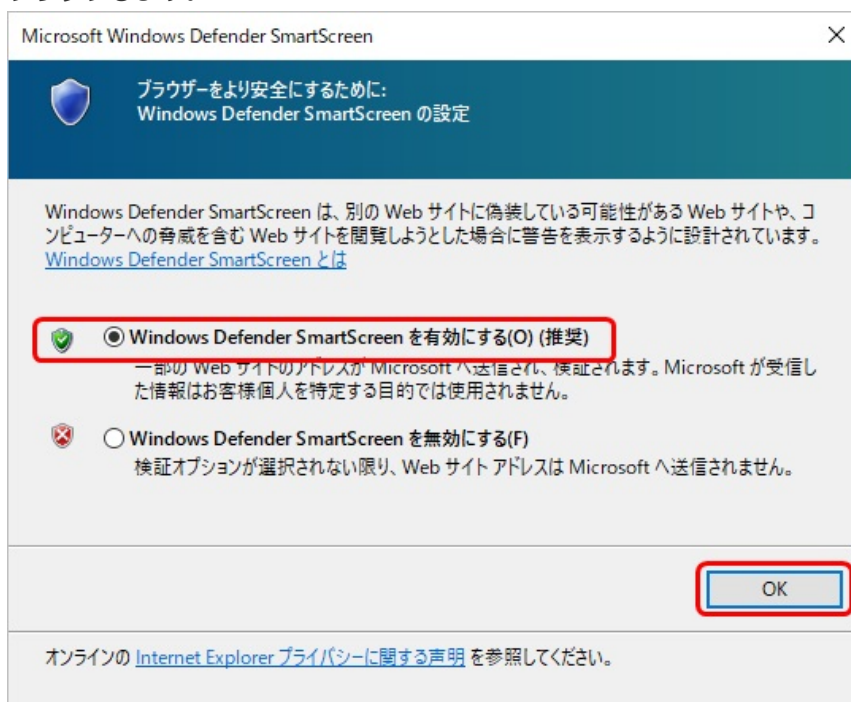


Fig. 1.3.2-2 Microsoft Windows Defender SmartScreen ダイアログ

1.3.3. Internet Explorer での互換表示設定の解除

互換表示設定が有効になっている場合、Internet Explorer が古いバージョンで動作し、Testabish タスクトレイアプリのユーザ操作を取得するのスクリプトが正常に動作しなくなるため、この設定を解除しておきます。

- Internet Explorer を起動し、画面右上の「ツール」（歯車マーク）をクリックします。
- 表示された一覧から「互換表示設定(B)」を選択します。

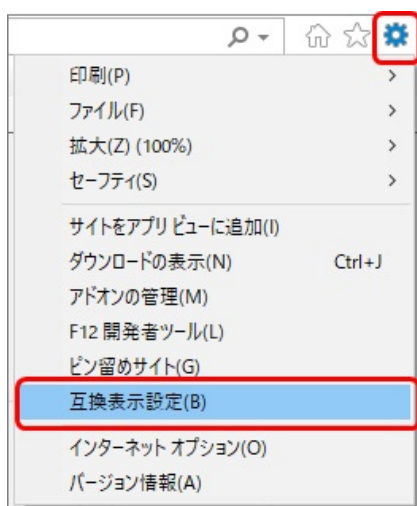


Fig. 1.3.3-1 ツール > 互換表示設定(B)

- 表示されたダイアログで「イントラネット サイトを互換表示で表示する(I)」、「Microsoft 互換性リストの使用(U)」のチェックが外れていることを確認します。チェックがついている場合は外してください。

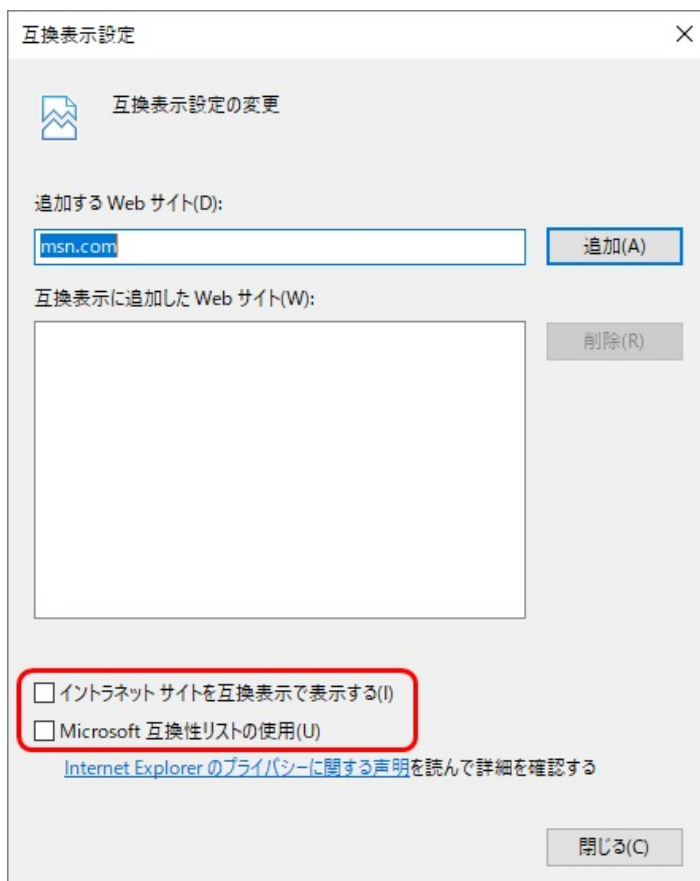


Fig. 1.3.3-2 互換表示設定ダイアログ

1.3.4. Internet Explorer でのセキュリティ設定

インターネットオプションのセキュリティタブで、各ゾーンの「保護モード」を **有効** にしておきます。また、詳細設定タブで**拡張保護モード**等の設定もしておきます。

1. Internet Explorer を起動し、画面右上の「ツール」（歯車マーク）をクリックします。
2. 表示された一覧から「インターネットオプション」を選択します。

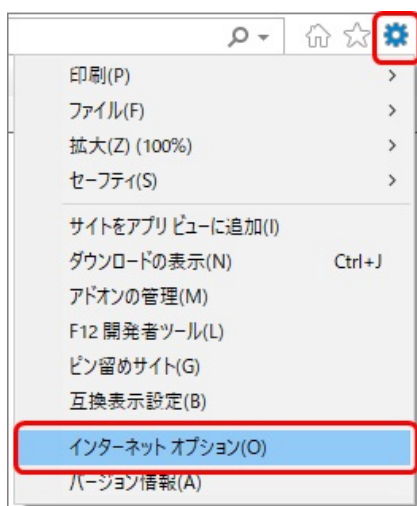


Fig. 1.3.4-1 「ツール(T) > インターネット オプション(O)」

3. 立ち上がった「インターネットオプション」ダイアログで「セキュリティ」タブを開きます。
4. 表示されている、「インターネット」「ローカル イントラネット」「信頼済みサイト」「制限付きサイト」のそれぞれのゾーンについて、「保護モードを有効にする」をチェックします。

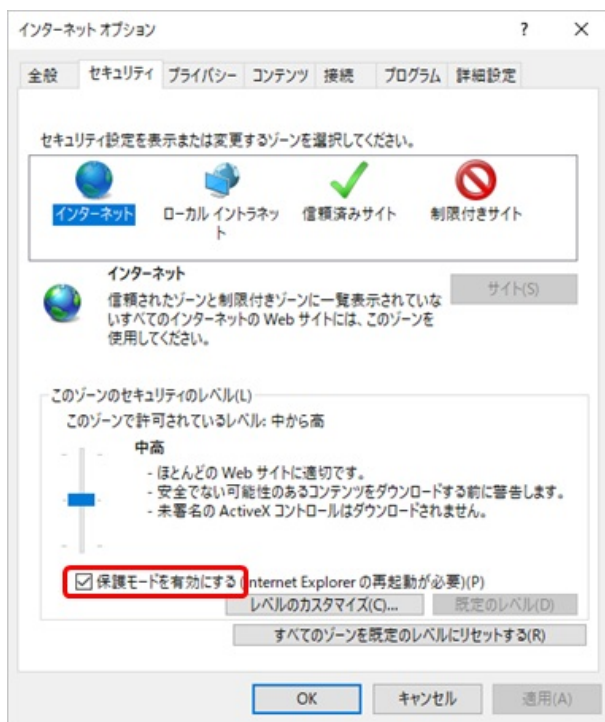


Fig. 1.3.4-2 「ゾーン：インターネット」

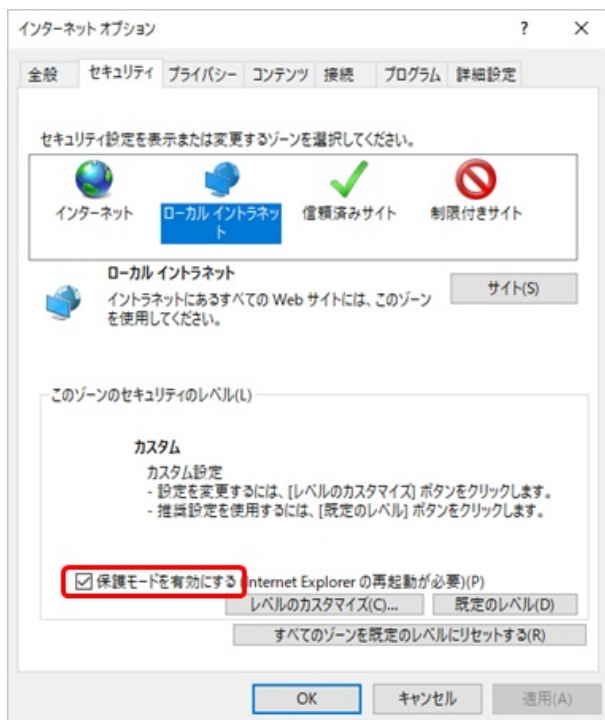


Fig. 1.3.4-3 「ゾーン：ローカル イントラネット」

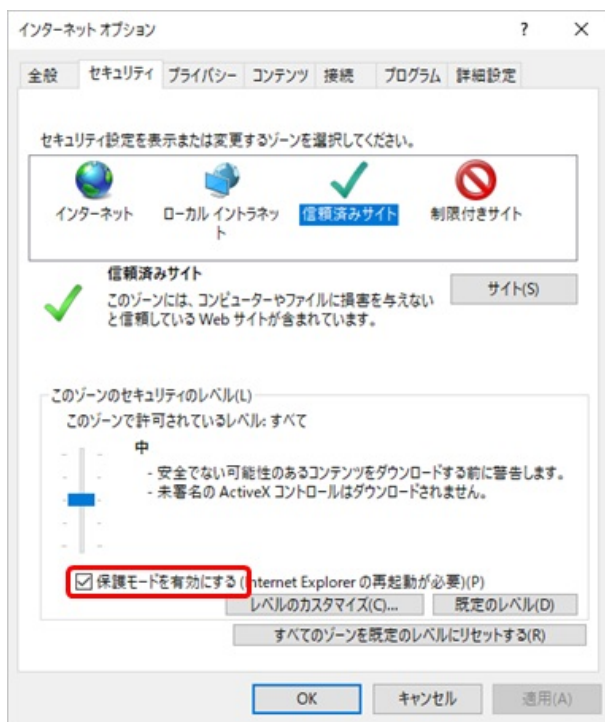


Fig. 1.3.4-4 「ゾーン：信頼済みサイト」

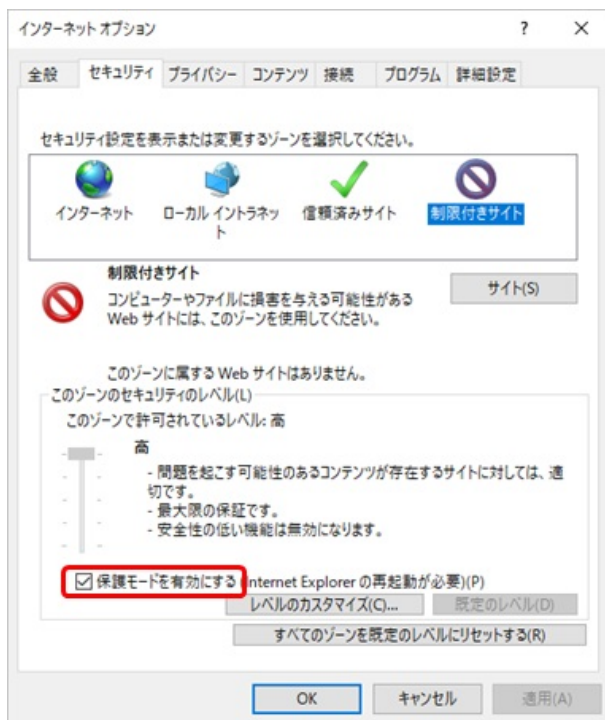


Fig. 1.3.4-5 「ゾーン：制限付きサイト」

5. 「インターネットオプション」ダイアログで「詳細設定」タブを開きます。
6. 「設定」のセキュリティセクションで
 - ・「拡張保護モードを有効にする」のチェックを外します。
 - ・「拡張保護モードで64ビットプロセッサを有効にする」のチェックを外します。

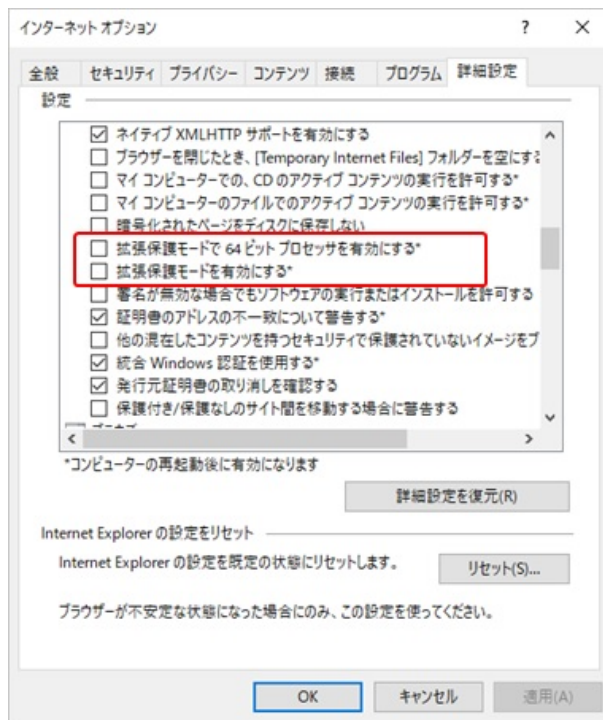


Fig. 1.3.4-6 「詳細設定タブ セキュリティセクション」

「設定」のブラウズセクションで

- ・「サードパーティ製のブラウザー拡張を有効にする」のチェックをつけます。

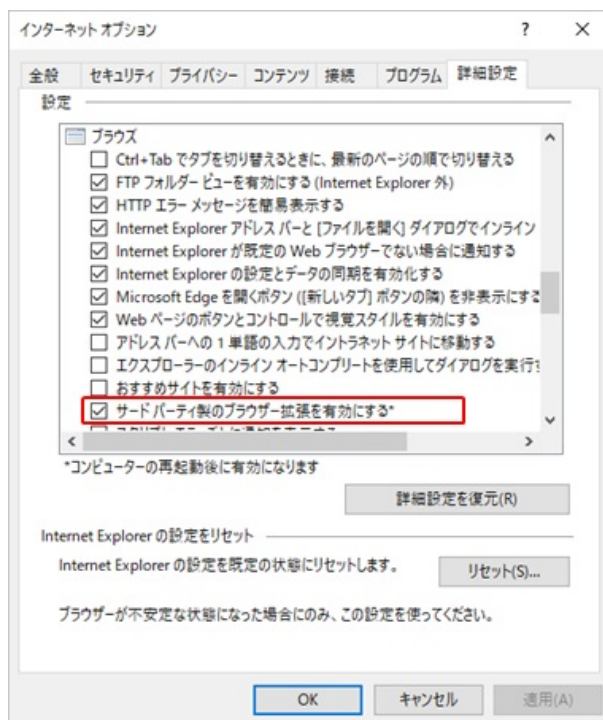


Fig. 1.3.4-7 「詳細設定タブ ブラウズセクション」

[目次](#)に戻る

III. 自動実行環境の構築手順

- 1. 自動実行環境を構築する
 - 1.1 Windowsでの実行環境導入
 - 1.1.1. インストーラー の実行
 - 1.1.2. 使用許諾画面
 - 1.1.3. ネットワーク接続確認
 - 1.1.4. WebDriverインストール方法選
 - 1.1.5. インストール先の指定
 - 1.1.6. インストール
 - 1.1.7. インストール完了
 - 1.2 手動で導入する
 - 1.2.1 Selenium をインストールする
 - 1.2.2 Internet Explorer の設定を確認する
 - 1.2.3 Windows10 の設定をする
 - 1.2.4 Java をインストールする
 - 1.2.5 Gradle を設定する
 - 1.2.6 テストコードを実行する

1. 自動実行環境を構築する

Testabliishから出力したテストコードを実行する環境を構築します。

実行する環境には Javaと使用するブラウザに合わせた Selenium Web Driver等のインストールが必要です。
この環境構築はインストーラを使う方法(Windowsの場合のみ)と手動で導入するやり方があります。

1.1 Windowsでの実行環境導入

Windows10で自動実行を行う場合はインストーラーを使って環境構築が行えます。

Javaは Amazon Corretto をインストールします。すでにお使いのPCにJavaがインストールされている場合はスキップして既存のJavaを使うことも可能です。

Selenium Web Driver はInternet Explorer、Google Chrome、Safari、Microsoft Edge Legacyの最新のものを選んでインストールできますが、すでにお使いのものがある場合はスキップすることができます。

1.1.1. インストーラー の実行

インストーラのファイル名はRuntime_installer.exe です。

1. インストールディレクトリのはRuntime_installer.exeを起動し、インストール ウィザードに従ってインストールしていきます。

名前	更新日時	種類	サイズ
arc	2021/07/21 19:01	ファイル フォルダー	
Runtime_installer.exe	2021/08/03 15:43	アプリケーション	10,163 KB
Testabliish_DockerInstallChecker.exe	2021/08/03 14:39	アプリケーション	93 KB
Testabliish_Installer.exe	2021/08/03 14:39	アプリケーション	97 KB
Testabliish_Task_tray_Setup_v1.4.3.exe	2021/08/03 14:39	アプリケーション	22,977 KB
Testabliish-ChromeExtension-v1.4.3.zip	2021/07/21 17:31	ZIP ファイル	143 KB

Fig. 1.1.1-1 ランタイムインストーラー

2. 起動後、『このアプリがデバイスに変更を加えることを許可しますか？』のダイアログが表示される場合は『はい』を選択して進んでください。

1.1.2. 使用許諾画面

1. 使用許諾画面が表示されます。一通り目を通して『同意する(A)』を押下して先に進みます。

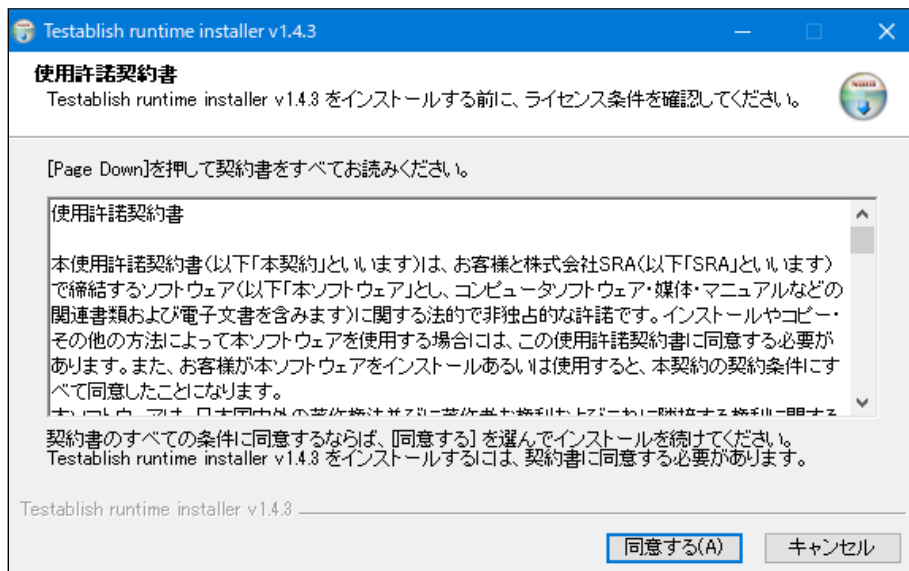


Fig. 1.1.2-1 使用許諾画面

1.1.3. ネットワーク接続確認

インストールはネットワークから必要なファイルをダウンロードします。この画面ではネットワークに接続されていて、ファイルをダウンロードできるかをチェックします。

1. ネットワーク接続確認ダイアログが表示されますので『確認』ボタンを押下します。

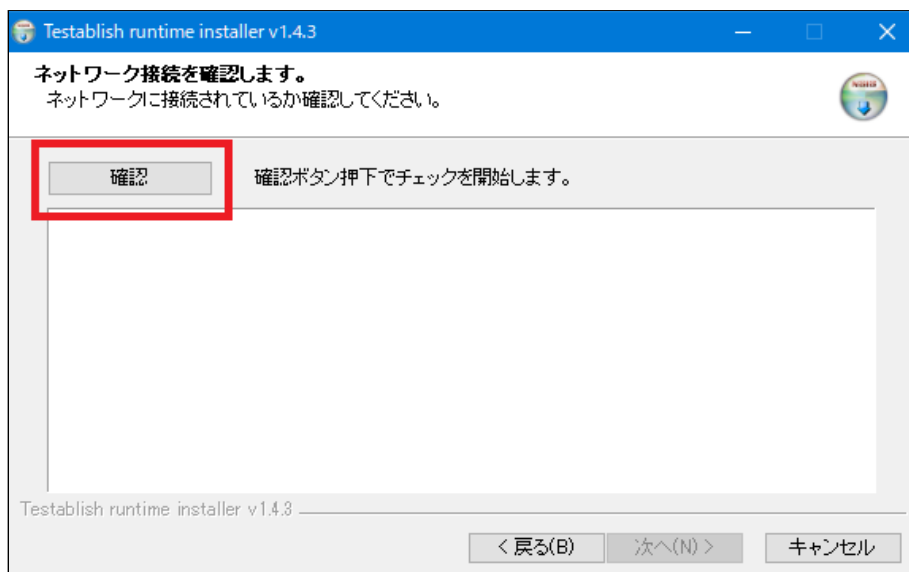


Fig. 1.1.3-1 ネットワーク接続確認画面

2. ネットワークの接続確認を行い、『インターネット接続:OK』のメッセージが現れるとチェックOKです。『次へ(N)>』ボタンを押下して先に進みます。

エラーになる場合

プロキシ環境下であり、以下のようなエラーが出る場合はプロキシの設定を確認してください。

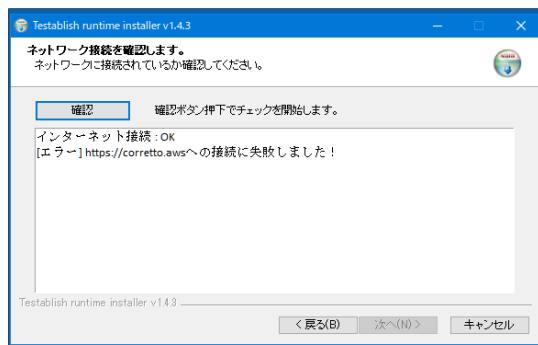


Fig. 1.1.3-2 ネットワーク接続確認画面_失敗メッセージ

プロキシの設定は環境変数の `HTTP_PROXY` と `HTTPS_PROXY` が設定されていることを確認してください。スタートメニュー右クリック -> システム -> 関連設定のシステムの詳細設定 の画面から『環境変数』ボタンを押下すると下の画面が開きます。

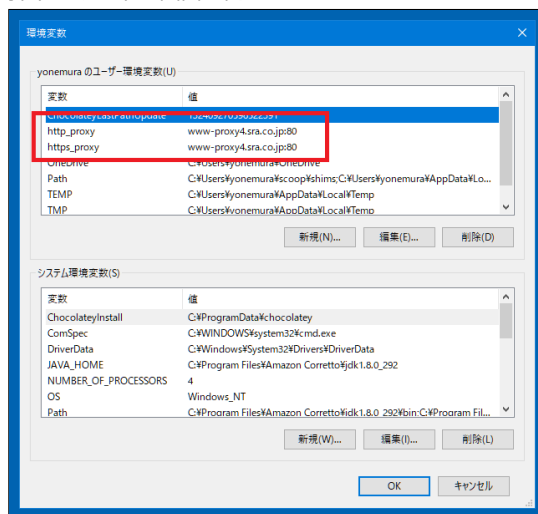


Fig. 1.1.3-3 プロキシ環境変数設定

1.1.4. WebDriverインストール方法選択

WebDriverのインストール方法を選択する画面が現れます。新規に最新のものをダウンロードする場合は『インターネットからWebDriverをインストールする』を選択してください『すでにインストールされているWebDriverを使う』を選択した場合は次の画面で既存のWebDriverを選択してインストールすることができます。

1. 『インターネットからWebDriverをインストールする』を選択した場合

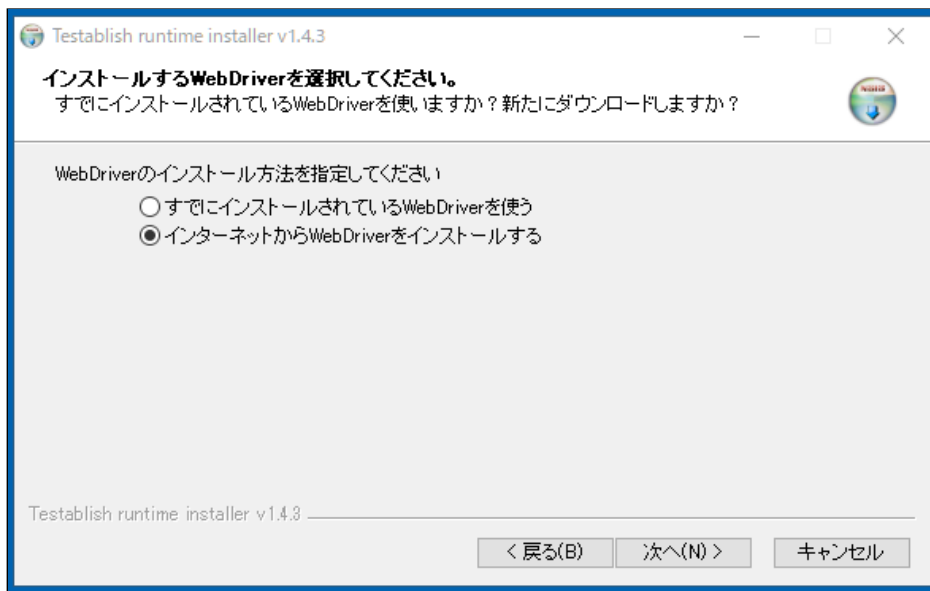


Fig. 1.1.4-1 WebDriverインストール方法確認画面

2. インストールするWebDriverのチェックを入れて『次へ(N)>』ボタンを押下して先に進みます。

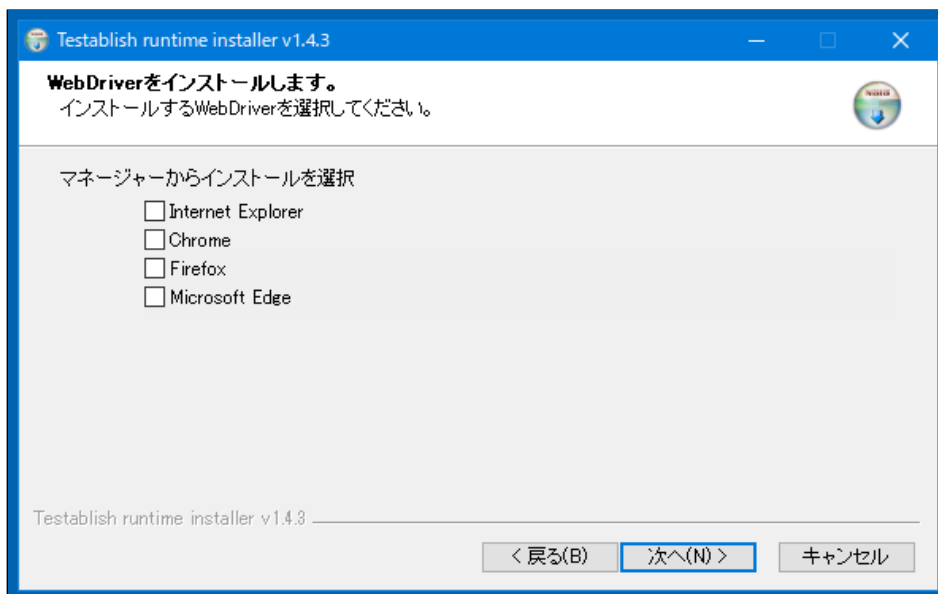


Fig. 1.1.4-2 WebDriver選択画面

既存のWebDriverを使う場合

既存のWebDriverを指定してそのパスが設定情報に反映されます。
WebDriverの指定は一つしかできません。

使用するWebDriverの場所を入力してください。exeファイルの場所を正しく入力するか、『...』ボタンからエクスプローラで指定してください。入力後『次へ(N)>』ボタンを押下して先に進みます。

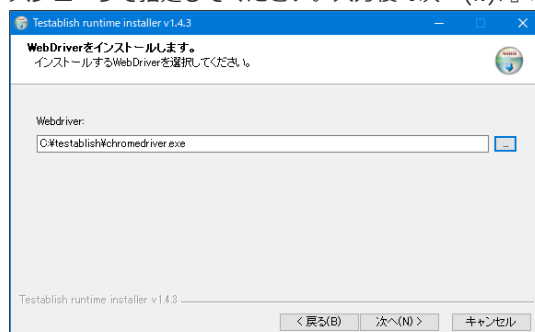


Fig. 1.1.4-3 既存のWebDriver選択画面

WebDriverの選択画面が出てくるので上の画面で選択したWebDriverを選択します。

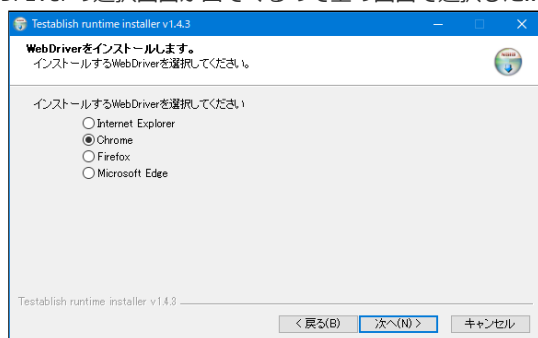


Fig. 1.1.4-4 選択したWebDriverの指定画面

1.1.5. インストール先の指定

1. インストールするフォルダを指定します。フォルダが存在しない場合は作成されます。

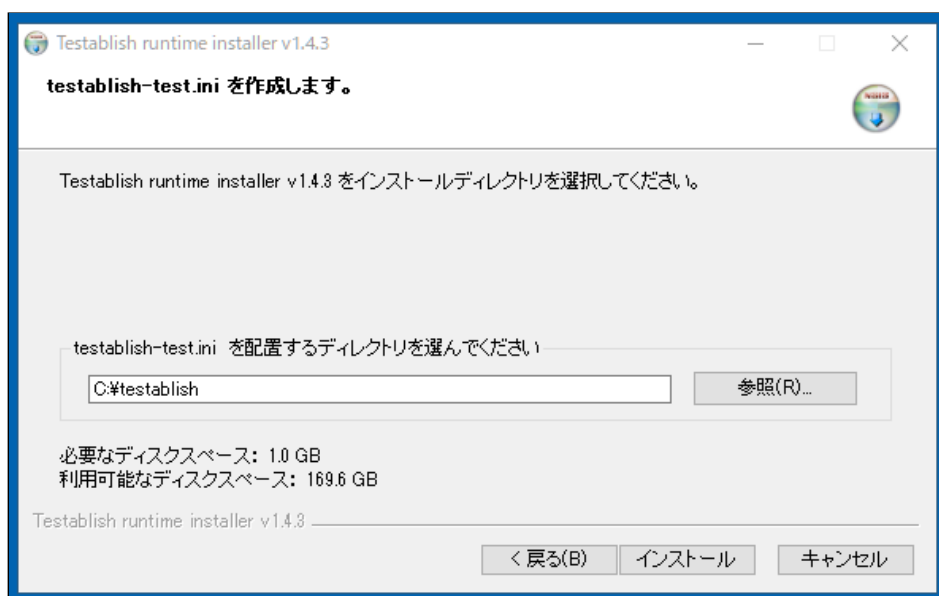


Fig. 1.1.5-1 インストール先指定画面

1.1.6. インストール

インストール先の指定画面で『インストール』を選択するとインストールが始まります。この時Javaが存在するかチェックを行って、すでにインストールされていればそのJavaを使用した設定が行われます。Javaがインストールされていない場合は Amazon Corretto をインストールします。

1. Javaがなかった場合、Javaのインストールを行う旨のメッセージが表示されるので『はい』を押下して先に進みます。『いいえ』を選択するとインストールを中断して終了します。

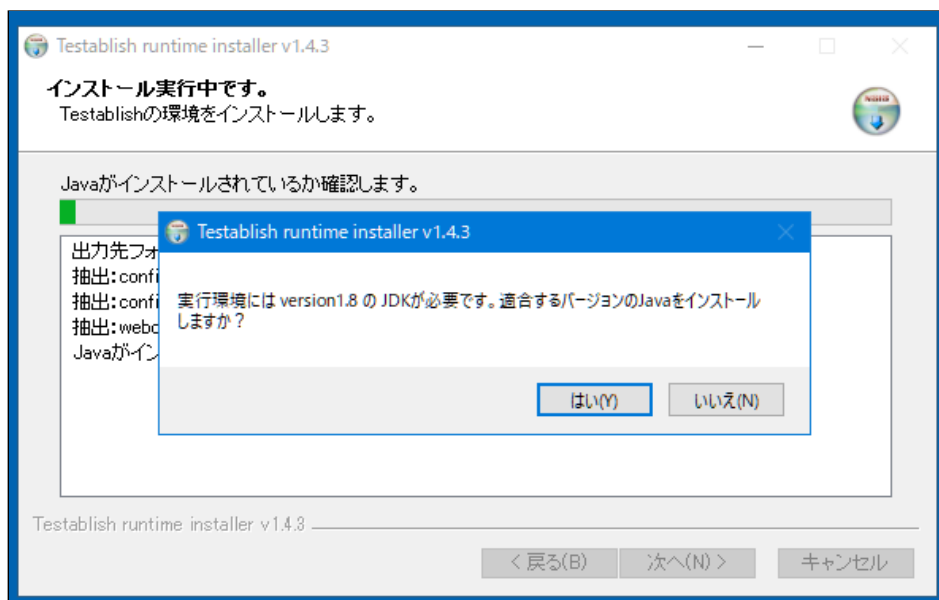


Fig. 1.1.6-1 Javaがなかった場合のインストール中画面

2. Amazon Correttoのダウンロード、インストールが始まり、完了すると以下のメッセージが表示されます。『OK』を押下して先に進みます。

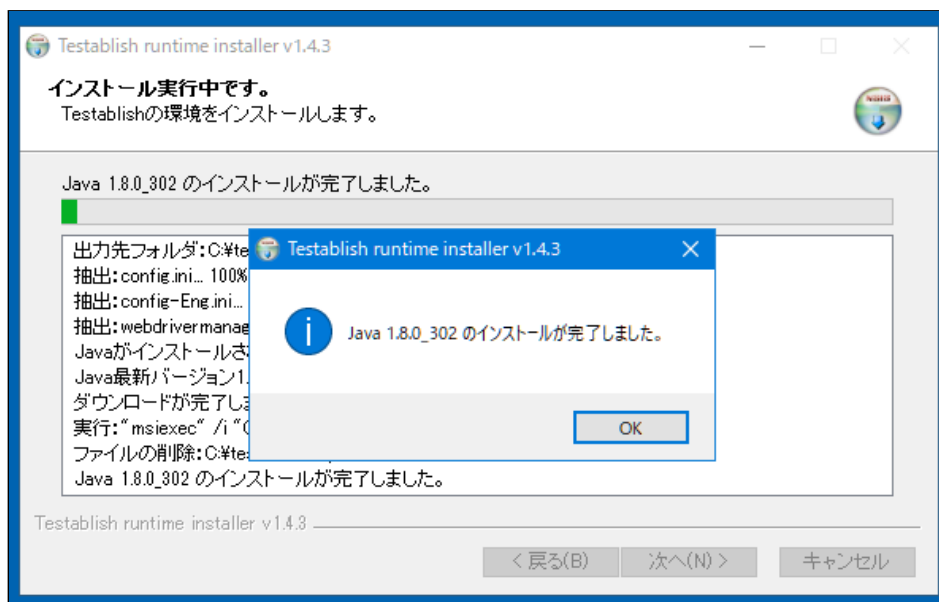


Fig. 1.1.6-2 Javaインストール完了画面

Javaがインストール済みの場合

Java1.8.0 がすでにインストールされている場合、以下の画面が表示されます。

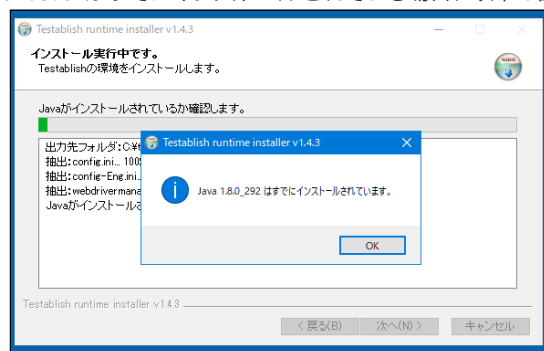


Fig. 1.1.6-3 既存のJavaがある場合の画面

3. Javaのインストールが終わると、レジストリのバックアップ、設定変更、WebDriverのダウンロード、Testabish.ini ファイルの作成を行います。これらの処理が完了すると以下の画面が表示されます。
『次へ(N)>』を押下して先に進みます。

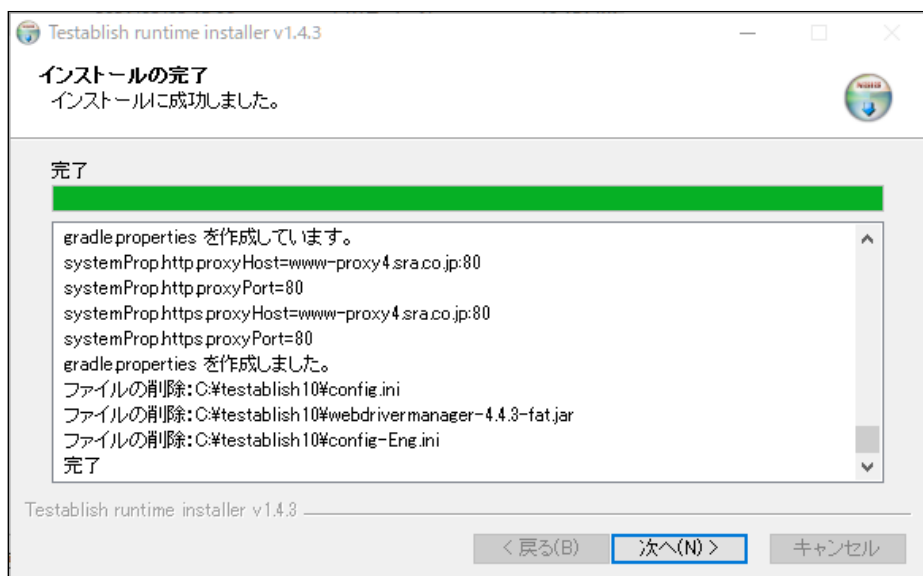


Fig. 1.1.6-4 インストール完了画面

1.1.7. インストール完了

1. 完了画面が表示されます。『完了(F)』を押下するとインストーラが終了します。

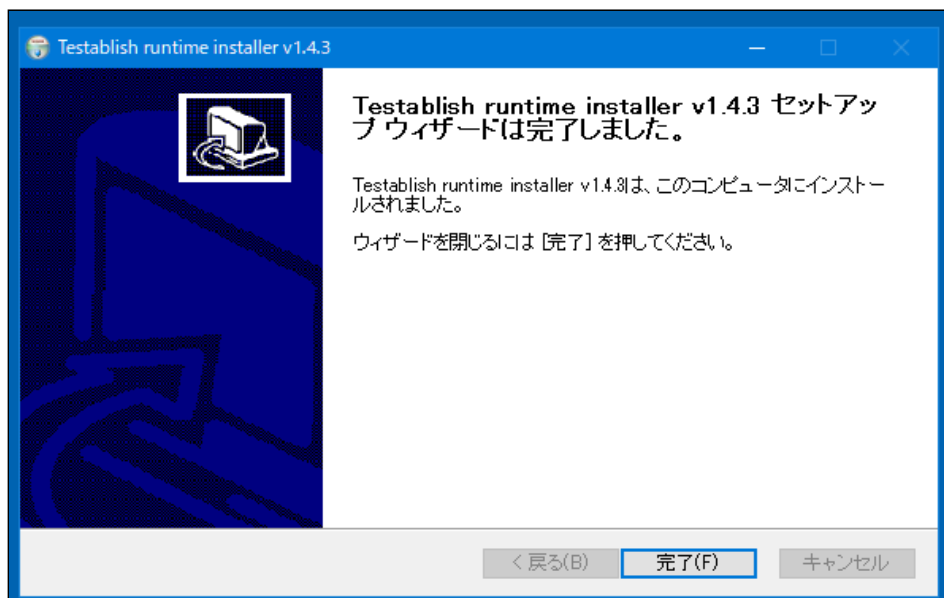


Fig. 1.1.7-1 完了画面

1.2 手動で導入する

手動で実行環境をセットアップする際は本章の説明に従ってください。LinuxやMacの環境で実行する際も同様の設定が必要となりますので参考にしてください。

1.2.1 Selenium をインストールする

Testabish で出力したテストコードの自動実行には、Seleniumを使います。

以下のURLから Selenium（3.8系）をダウンロードしてください。

http://selenium-release.storage.googleapis.com/3.8/IEDriverServer_x64_3.8.0.zip

展開して「C:/selenium/IEDriverServer_x64_3.8.0/IEDriverServer.exe」となるように配置してください。

1.2.2 Internet Explorer の設定を確認する

- 。ブラウザの ツール メニュー > インターネットオプション を開き、以下の項目を確認してください。
 - プライバシータブ > 「ポップアップブロックを有効にする」が **無効** になっていること。
 - セキュリティタブ > 全ゾーンの「保護モードを有効にする」が **有効** になっていること。
 - 詳細設定タブ > 設定一覧の「拡張保護モードで64ビットプロセッサを有効にする」のチェックがなく **無効** になっていること。
 - 詳細設定タブ > 設定一覧の「拡張保護モードを有効にする」のチェックがなく **無効** になっていること。
- 。ブラウザの右上の「ツール」アイコン（歯車の形のアイコン）をクリックし以下の項目を確認してください。
 - 拡大 が「**100%**」に設定されていること。

1.2.3 Windows10 の設定をする

- 。ディスプレイの設定をします。
 1. デスクトップ左下の Windows アイコンをクリックしてください。
 2. 設定（歯車の形のアイコン）をクリックしてください。
 3. システム > ディスプレイ の「テキスト、アプリ、その他の項目のサイズを変更する」を「**100%**」に設定してください。
- 。レジストリの設定のため、レジストリエディターを開きます。
 1. Windows システムツール > ファイル名を指定して実行 を選択してください。
 2. [regedit]と入力して、レジストリエディターを起動してください。
 3. HKEY_LOCAL_MACHINE を右クリック > エクスポート でレジストリのバックアップを取ってください。
レジストリの設定は、誤ると Windows 自体の動作が異常になる恐れがあります。**必ずバックアップを取ってください。**
- 。レジストリエディターでレジストリの設定をします。
 1. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl に、キー(ディレクトリ) [FEATURE_BFCACHE] を追加してください。
 2. 1 で追加したキーの中に DWORD値 で 名前 : [iexplore.exe]、値 : [0] を追加してください。
 3. HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet Explorer\Main\FeatureControl に、キー(ディレクトリ) [FEATURE_BFCACHE] を追加してください。
 4. 3 で追加したキーの中に DWORD値 で 名前 : [iexplore.exe]、値 : [0] を追加してください。

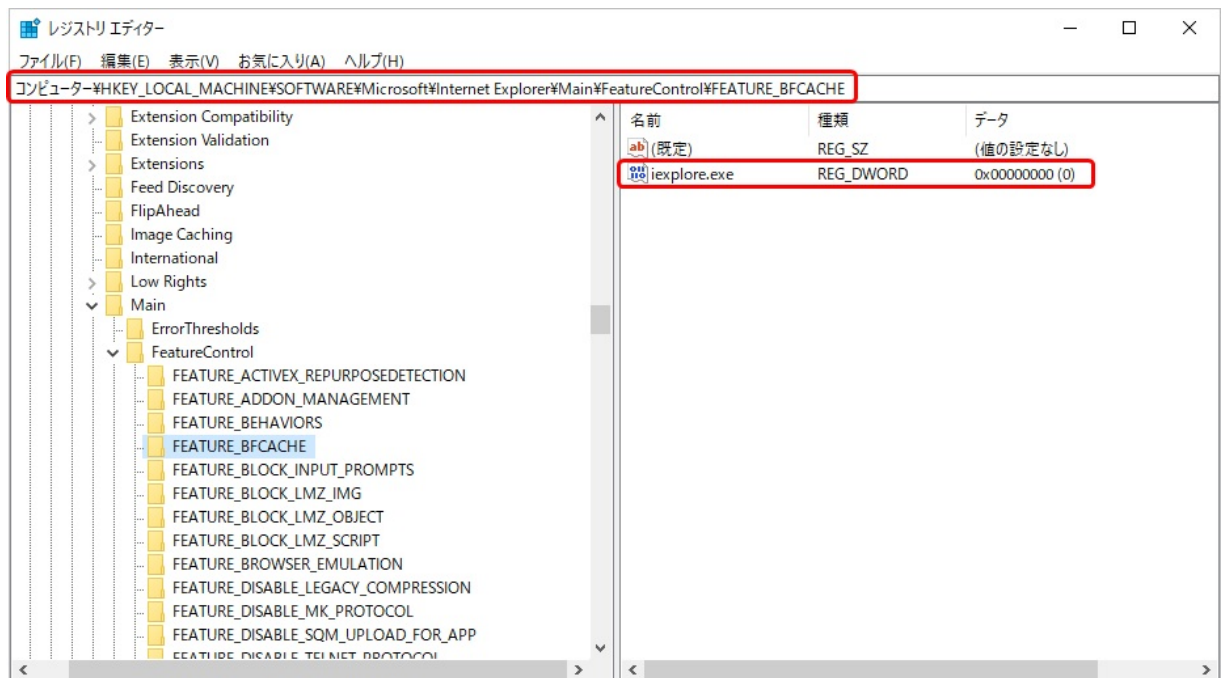


Fig. 1.3-1 レジストリエディタの設定

1.2.4 Java をインストールする

- **JDK (1.8系)** をインストールしてください。

お使いのPCにJDKがインストールされていない場合、JDK は Oracle やそれ以外にもさまざまな有償/無償のものが存在しますので、お使いの環境に応じて選択し、ダウンロードしてください。

既に java がインストールされているかどうかは、以下の方法で確認できます。

また、インストール後に Java の PATH がうまく設定されているかどうかと同じ方法で確認できます。

- Windows 10の場合: [スタートメニュー]→[Windows システムツール]→[コマンドプロンプト]
- コマンドプロンプトが起動されたら 以下のコマンドを入力してください。

```
% java -version
```

- JDKがインストールされて正常に動作している場合は、次のようにインストールされているJDKのバージョンが表示されます。

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\>java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)

C:\>
```

Fig. 1.4-1 java のバージョン確認

- WindowsにJavaのパスを設定してください。
 - (変数名) JAVA_HOME
(変数値) C:\Program Files\Java\jdk1.8.x_xxx

- 「jdk1.8.x_xxx」は、インストールしたバージョンによって異なります。
「C:\Program Files\Java」フォルダを確認して設定してください。
- (変数名) Path
(変数値) %JAVA_HOME%\bin;(既存のものはそのまま)

1.2.5 Gradle を設定する

- テスト自動実行を行うネットワーク環境で**プロキシを利用する場合に必要な**設定です。
- ホームディレクトリ [C:\Users\<ログインユーザ>\] に [.gradle] というディレクトリを作成します。
 - コマンドプロンプトで [mkdir .gradle] と実行してください。
- 作成した [.gradle] ディレクトリの中に、gradle.properties というファイルを作成し、中身を以下のよう
に設定してください。

```
systemProp.http.proxyHost=proxy.sample.com
systemProp.http.proxyPort=80
systemProp.https.proxyHost=proxy.sample.com
systemProp.https.proxyPort=80
```

- **proxyHost** および **proxyPort** は、お使いの環境で使用するインターネットプロキシを記載してください。

1.2.6 テストコードを実行する

テストの自動実行は、

別マニュアル：Testabliish 使い方マニュアル » III. 操作の流れ » 2. テストの自動実行
を参照してください。

[目次](#) に戻る

IV. 証明書のインストール

- [1. セキュリティ証明書の生成](#)
- [2. セキュリティ証明書のインポート](#)
 - [2.1. certlm.msc を起動する](#)
 - [2.2. インポートウィザードの起動](#)
- [3. セキュリティ証明書の確認](#)

SSLを利用するにはSSLサーバ証明書をインストールする必要があります。

Testabishのタスクトレイアプリを起動後、ブラウザ拡張機能によりブラウザのインターネットプロキシの接続先がローカルプロキシに置き換えられます。

このローカルプロキシの証明書をインストールします。

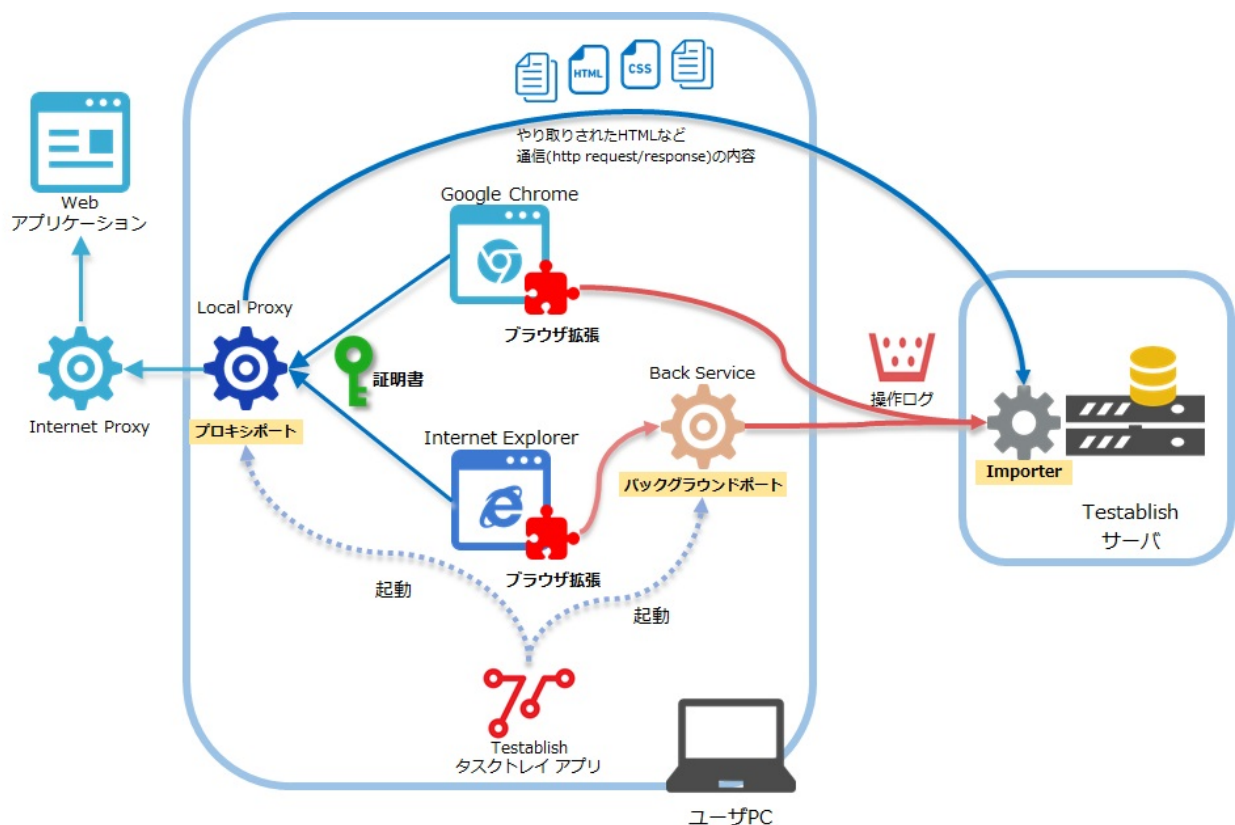


Fig. 証明書のインストール

1. セキュリティ証明書の生成

Testabish拡張機能(タスクトレイアプリ)の サービスタブにある「証明書作成」ボタンをクリックすると、以下のフォルダにセキュリティ証明書ファイル (**proxy.crt**) が生成されます。

C:\Users\<ユーザ名>\AppData\Local\Testabish\IE

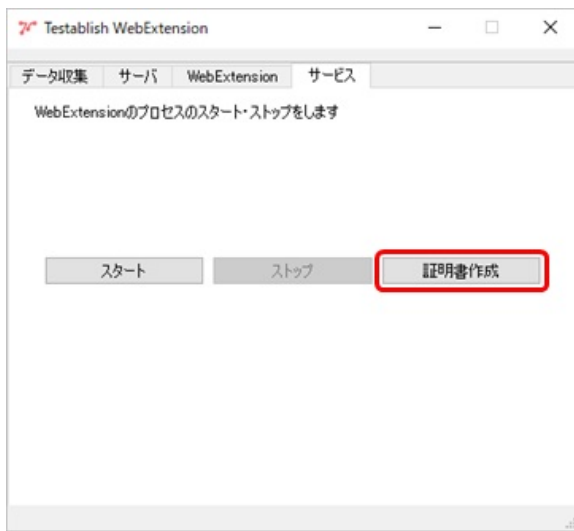


Fig. 1-1 証明書作成

これをローカルコンピュータの「信頼されたルート証明機関」にインストール(インポート)します。

2. セキュリティ証明書のインポート

Windows標準のGUIツールを用いてインストールを行う手順を示します。

2.1. certlm.msc を起動する

「ファイル名を指定して実行」ダイアログに "certlm.msc" と入力し、OKボタンをクリックします。

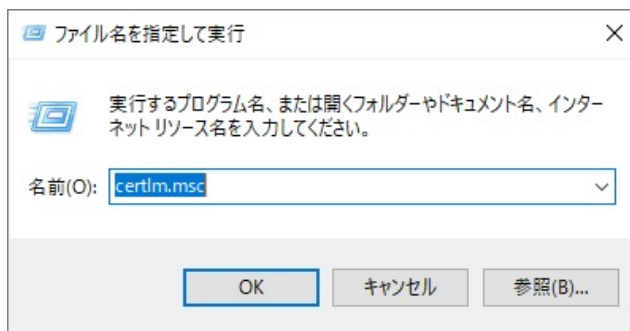


Fig. 2.1 certlm.msc を起動

2.2. インポートウィザードの起動

起動した証明書管理ツール画面で、

信頼されたルート証明機関 を右クリック > **すべてのタスク** > **インポート** を選択します。

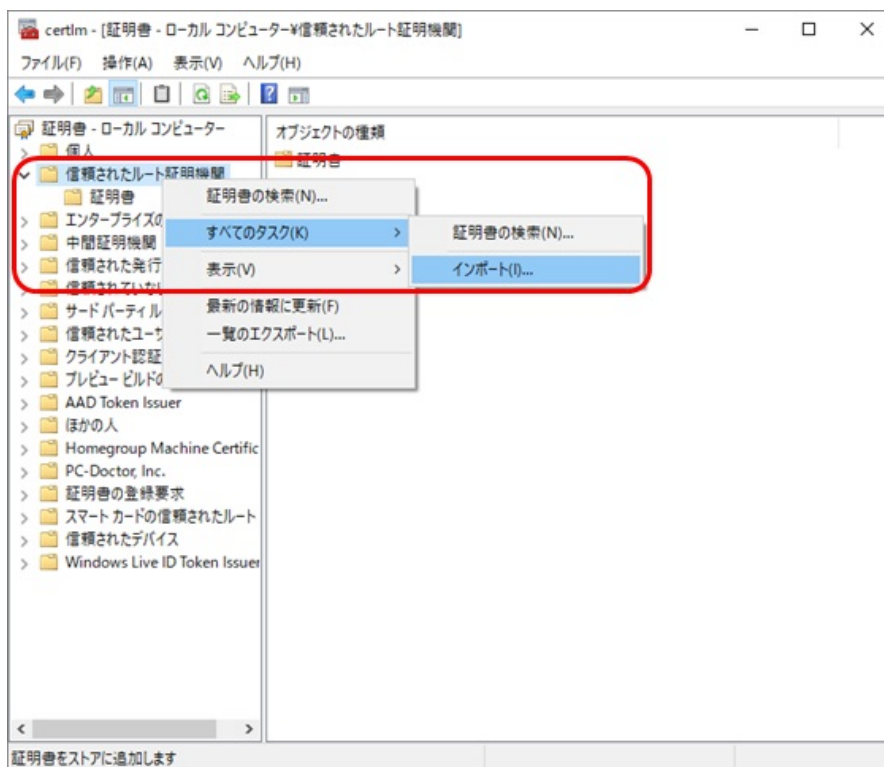


Fig. 2.2-1 インポートウィザードを起動

起動した証明書のインポートウィザードに従ってインポートしていきます。
「次へ」をクリックします。



Fig. 2.2-2 ウィザードの開始

インポートしたい証明書を「参照」ボタンで選択し、「次へ」をクリックします。
Testablish タスクトレイアプリの証明書ボタンで作成された証明書は、
C:\Users\<ユーザ名>\AppData\Local\TestablishIE 以下に **proxy.crt** という名称で作成されています。

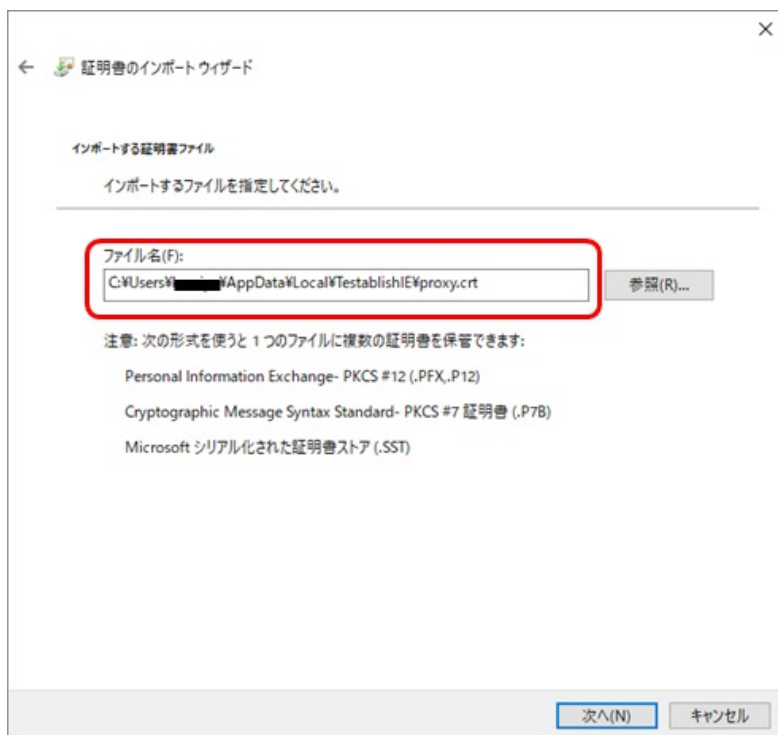


Fig. 2.2-3 インポートファイル名の指定

「参照」ボタンで証明書ストアを選択します。

「証明書をすべて次のストアに配置する」にチェックを入れ、「信頼されたルート証明機関」を選択し、「次へ」をクリックします。

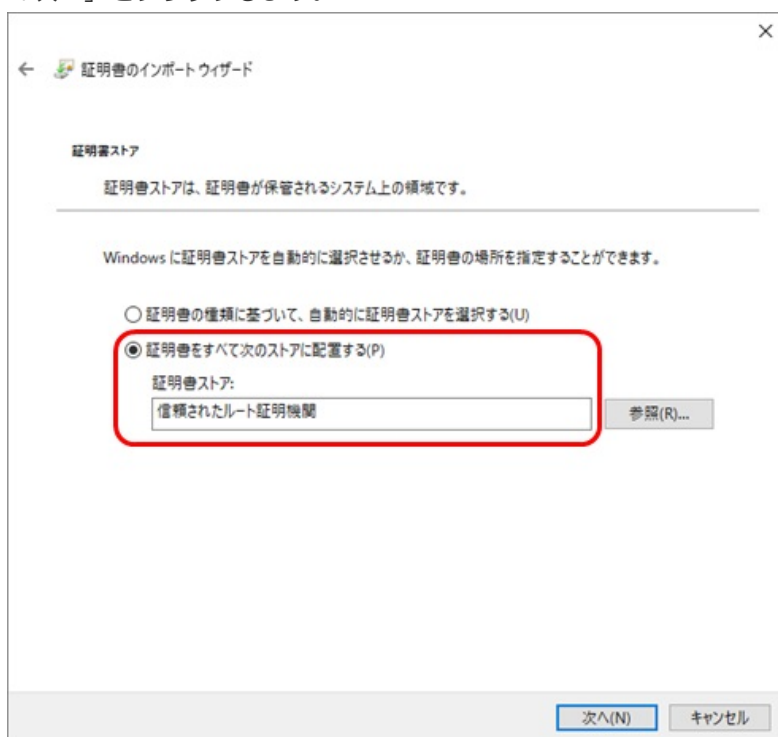


Fig. 2.2-4 証明書ストアの指定

「完了」ボタンをクリックし、証明書をインポートします。

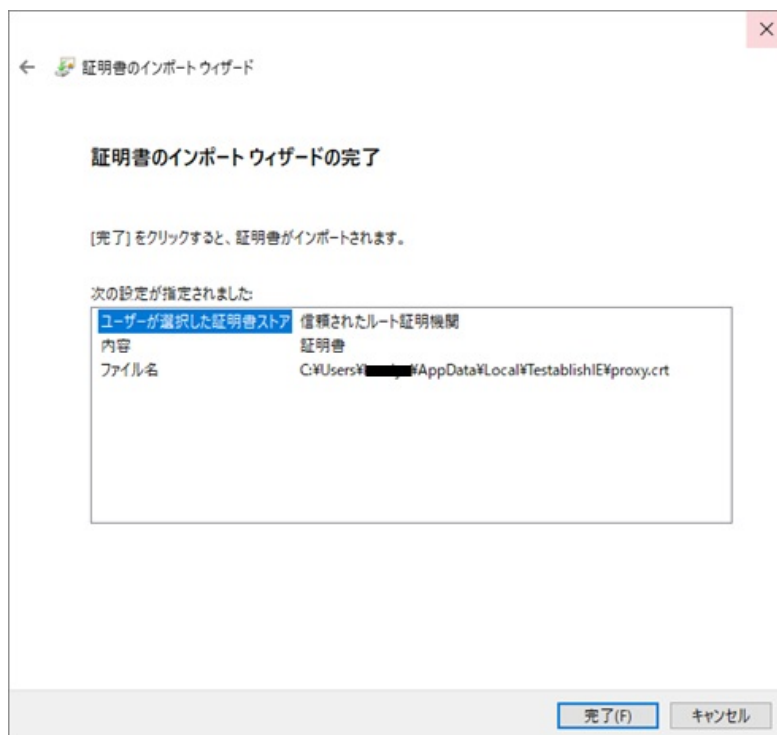


Fig. 2.2-5 インポートウィザードの完了

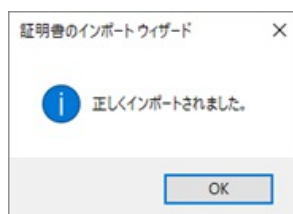


Fig. 2.2-6 インポートウィザードの終了

3. セキュリティ証明書の確認

証明書管理ツール画面で証明書がインポートされていることを確認します。

信頼されたルート証明機関 を右クリック > **最新の情報に更新** を選択します。

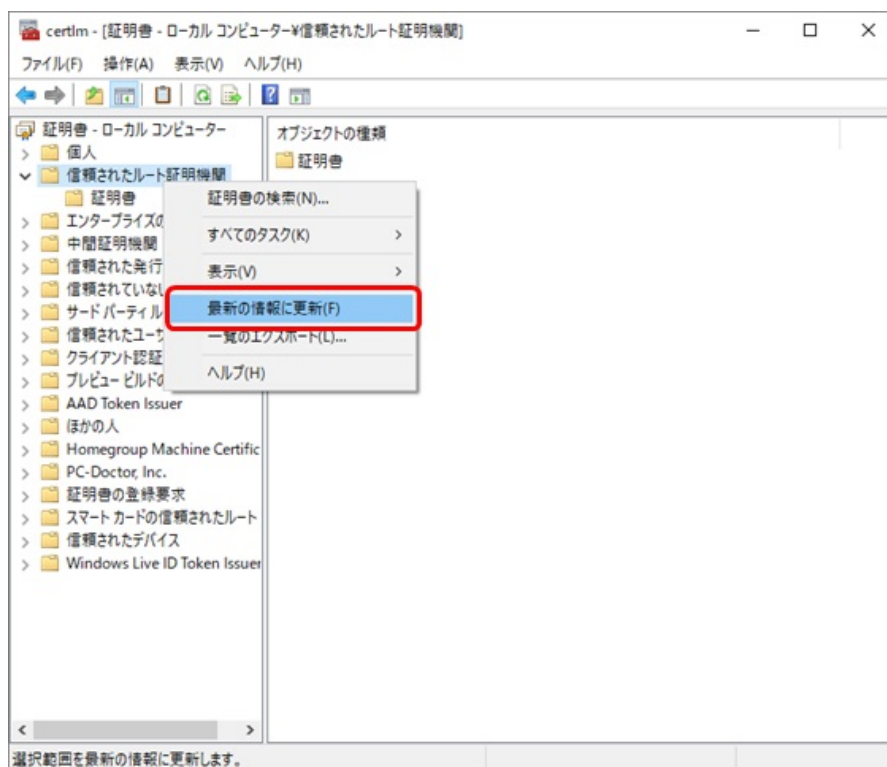


Fig. 2.3-1 証明書情報の更新

Testblishの証明書がインポートされていることを確認します。

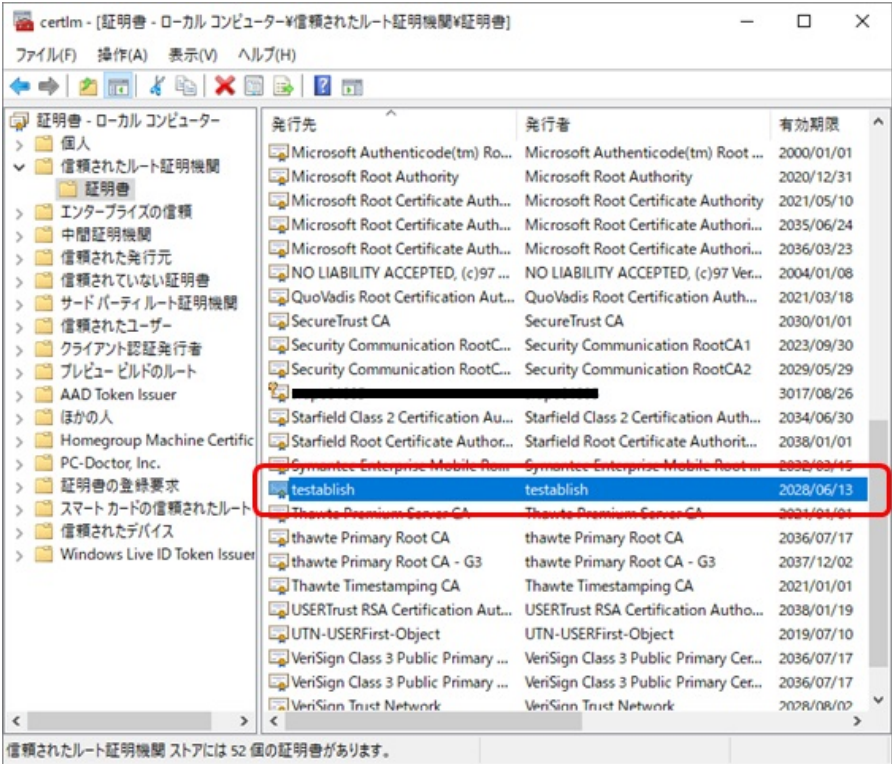


Fig. 1.4-2 証明書情報の確認

[目次](#)に戻る

Base version Testablish_v1.4.7
\$Revision: 844 \$ \$Date:: 2023-06-01#\$