QualityForward

BTS 連携マニュアル

VERISERVE

最終更新日：2020/10/28

目次

第1章　BTS 連携を行う

ご使用の BTS を選択し、レポート機能と連携設定することができます。

社内ネットワークを使用されている場合、外部のシステムである QualityForward からのアクセスが弾かれてしまう場合がございます。その場合は以下の IP アドレスからのアクセスを許可していただく必要がございます。

QualityForward IP アドレス

13.112.115.12

13.113.53.12

52.197.246.217

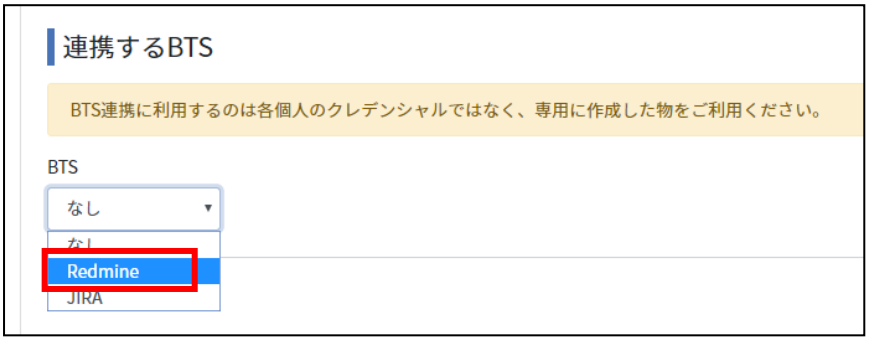
52.197.44.200

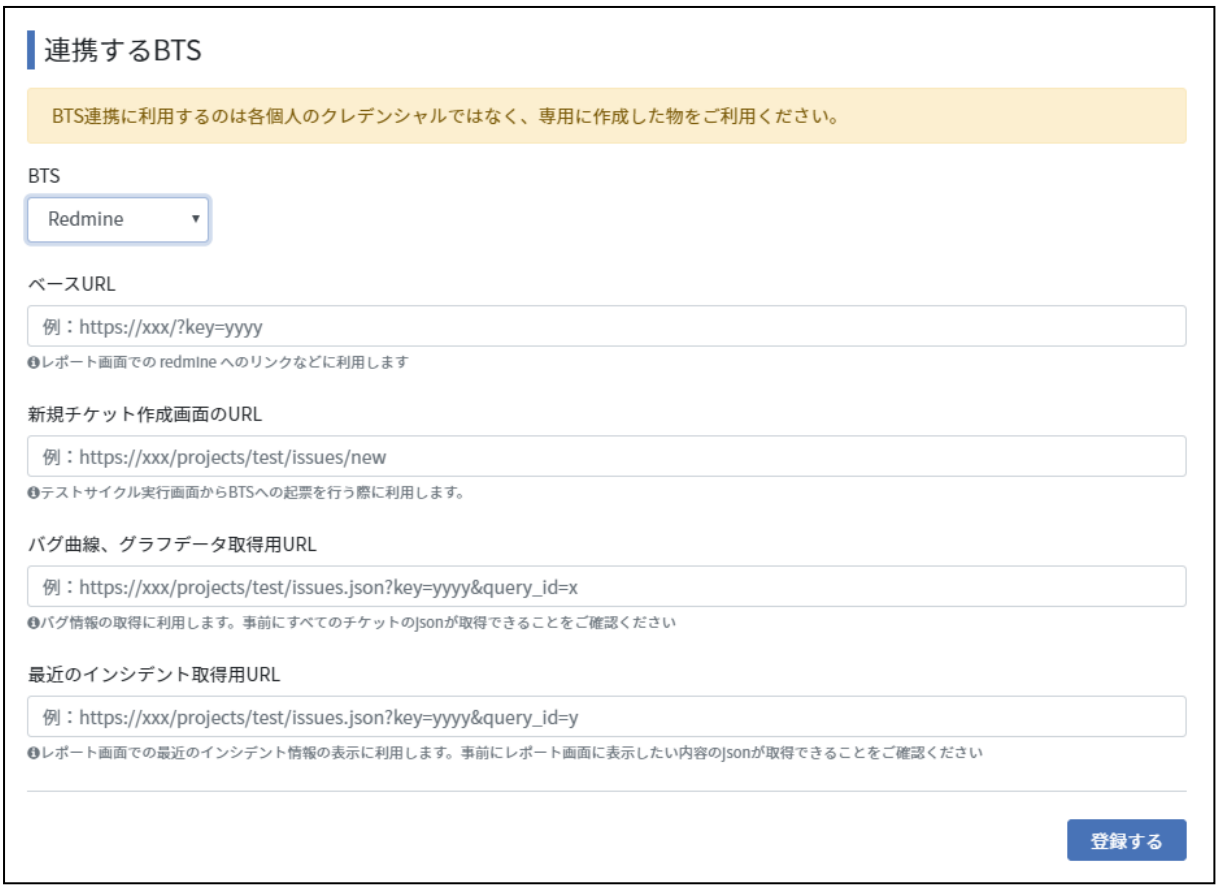
18.177.168.126

52.69.201.102

1.1. Redmine と連携する

連携する BTS で「Redmine」を選択すると Redmine に連携するための設定項目が表示されます。



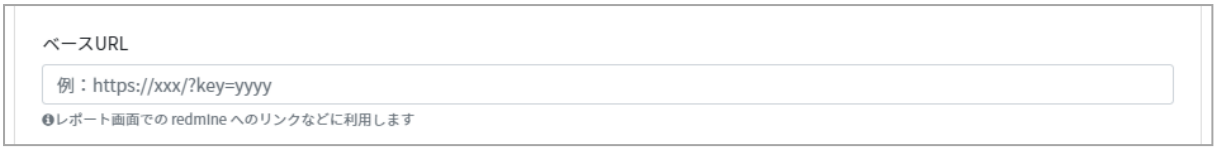


1.1.1. Redmine のベース URL を設定する

Redmine のベース URL の入力を行います。ベース URL はバグの優先度設定の取得、および「最近のインシデント一覧」のチケットのリンク生成のため利用します。

ベース URL は、Redmine のホームの URL です。

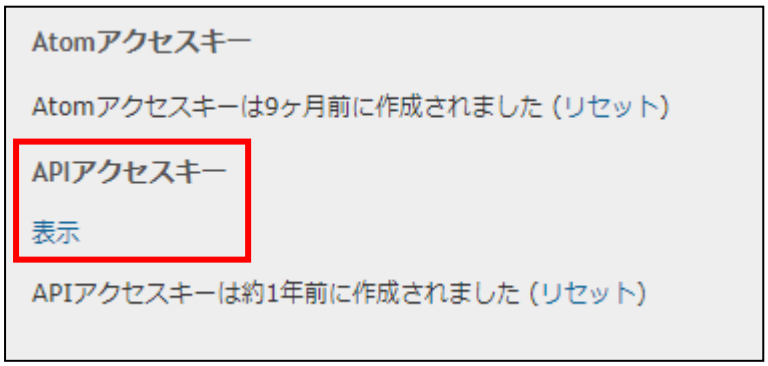
Redmine のルートにあたる URL を指し、 [https://xxxxxx.xxxx/] の場合と[https://xxxxxx.xxxx/redmine/] である場合の 2 パターンあります。URL の後には「?key=API キー」を指定します。



API キーは Redmine の個人設定から取得することができます。手順は以下の通りです。

(1)　Redmine にログインし、画面右上の個人設定を開きます。

(2)　API アクセスキーの表示をクリックすると API キーが表示されます。



1.1.2.　新規チケット作成画面の URL

新規チケット作成画面の URL を設定すると、テストサイクルでテストを実行中に右クリックから簡単にチケットへの起票を行うことができます。

チケットの起票画面の URL を記載してください。

例）https://xxx/projects/test/issue/new

1.1.3.　バグ一覧取得用の URL を設定する

バグ取得用の URL の入力を行います。バグ曲線を描画するため、指定された範囲におけるバグの OPEN/CLOSE 数の取得に利用します。

(1)　Redmine の個人設定から API キーを取得します。

(2)　Redmine の対象のプロジェクト開き、URL を取得します。

例）https://xxx.xxxx/projects/xxxx/

(3)　手順(2)で取得した URL に「issues.json?key=API キー」を追加します。

例）https://xxx.xxxx/projects/xxxx/issues.json?key=xxxx

(4)　Redmine で対象チケットの絞込みを行います。バグ一覧を取得するために「すべて」の「バグ」を対象にし、適用ボタンを押します。

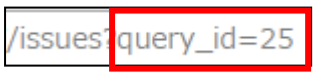
※ここでは「すべて」の「バグ」を対象にしていますが、フィルタ条件はプロジェクト方針に合わせて自由に設定していただけます。

(5)　すべてのバグの一覧が表示されたら保存ボタンを押します。

(6)　新しいクエリに名前を付けて保存します。

(7)　チケット一覧右側のカスタムクエリ一覧に手順(6)で作成したクエリが表示されます。作成したクエリ名をクリックします。

(8)　URL の最後にクエリ ID が表示されるので、「query_id=xx」をコピーします。



(9)　手順(3)までで作成した URL の最後に手順(8)の「&query_id=xx」を入力します。

例)https://xxx.xxxx/projects/xxxx/issues.json?key=xxxx&query_id=xx

(10)　手順(9)でできた URL をブラウザのアドレス欄に直接入力します。json の取得が確認できたら URL を登録欄に入力し、登録ボタンを押します。

1.1.4.　最近のインシデント取得用 URL を設定する

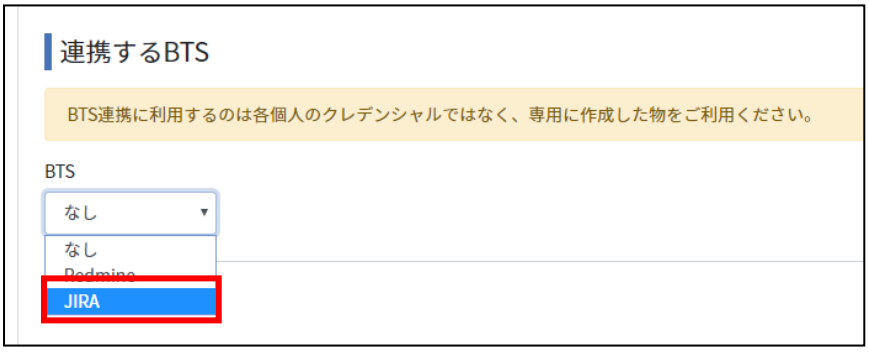
最近のインシデント取得用 URL「最近のインシデント一覧」「優先度別グラフ」「ステータス別グラフ」画面を生成するため、指定された範囲のバグのタイトル、優先度、ステータスを取得します。

※URL の取得は手順 1.1.3 と同様に行います。

※バグ一覧取得用 URL と最近のインシデント取得用 URL が個別に設定可能なのは、プロジェクトによって取得したい先のフィルタが異なる場合にも対応するためです。
集計したいバグの範囲が同一の場合には、同一の指定で問題ありません。

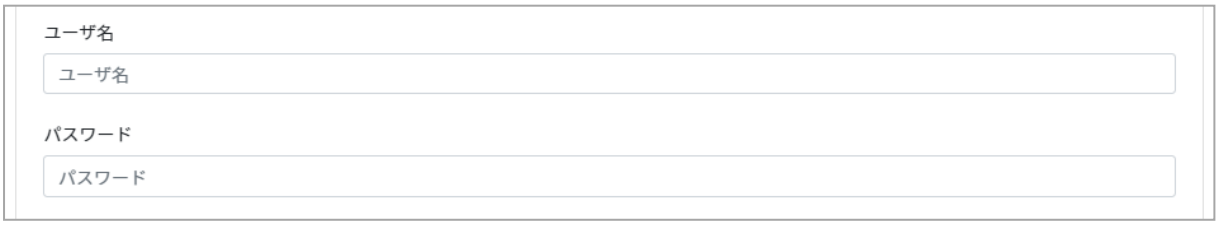
1.2. JIRA サーバー型と連携する

連携する BTS で「JIRA」を選択すると JIRA に連携するための設定項目が表示されます。



1.2.1.　ユーザ名とパスワードを入力する

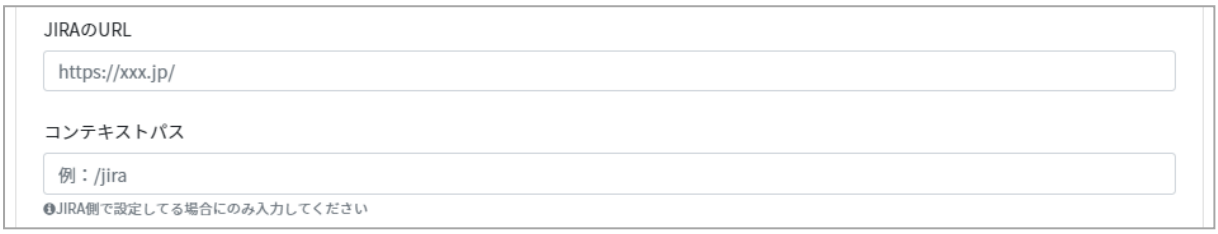
情報を取得するために、JIRA に登録済みのユーザ名とパスワードを入力します。



1.2.2.　URL・コンテキストパスを設定する

取得したい課題の登録されたプロジェクトを含む JIRA の URL を設定します。コンテキストパスは JIRA 側で設定を行っている場合にのみ入力してください。

※コンテキストパスは「/xxx」の形式で入力してください。



(1)　JIRA の管理メニューからシステムを選択します。

(2)　一般設定を開きます。

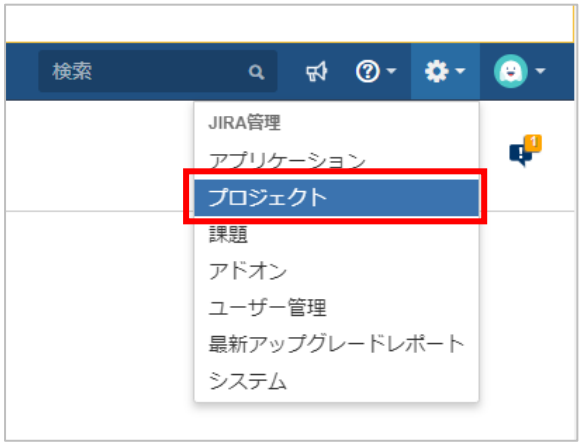
(3)　一般設定内にあるベース URL をコピーし、JIRA の URL に入力します。

1.2.3.　新規チケット作成画面の URL を設定する

新規チケット作成画面の URL を設定すると、テストサイクルでテストを実行中に右クリックから簡単にチケットへの起票を行うことができます。

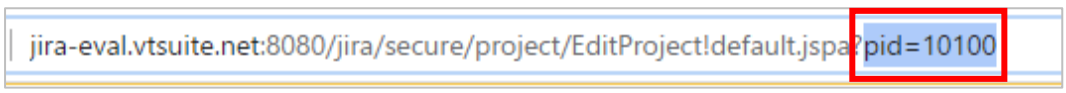
（1） JIRA の URL の後に「/secure/CreateIssueDetails!init.jspa?pid=**yyy**&issuetype=**zzz**」を追加します。PID と issuetype は以下手順で設定します。

（2） プロジェクト設定を開きます。



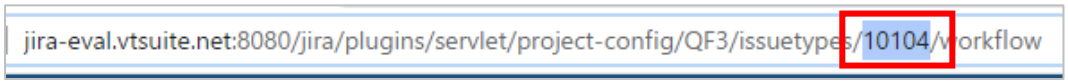
（3） 詳細情報を選択します。

（4） 詳細情報画面の URL の最後に記載されている PID を新規チケット作成画面の URL に設定します。



（5） プロジェクト設定画面で課題タイプを選択します。

（6） 課題タイプの画面 URL に記載されている issuetype を新規チケット作成画面の URL に設定します。



1.2.4. JQL を設定する

JQL は「JIRA Query Language」の略で、JIRA 専用のクエリ言語を指します。取得するプロジェクトや課題のタイプなどを絞り込むために JQL を設定する必要があります。未設定の場合は JIRA に登録されている全てのプロジェクト、課題が対象となります。

例）特定のプロジェクトを対象とする場合は project=”xxxxx”

例）特定の課題タイプを対象とする場合は issueType = “xxx”

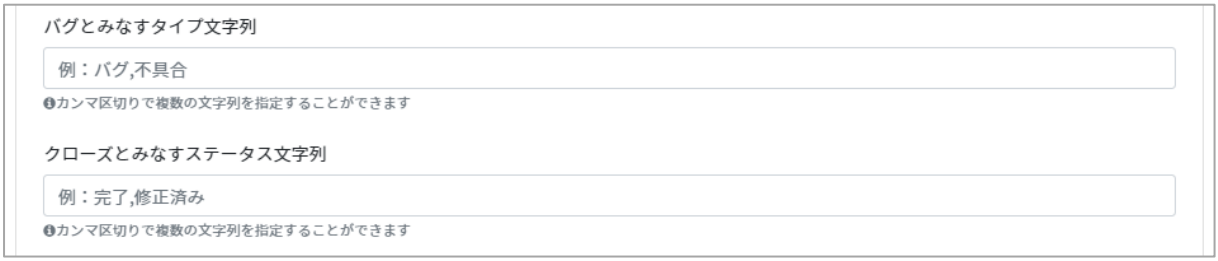
具体的な入力値について、プロジェクトを対象とする場合は JIRA に登録しているプロジェク

トキーを、課題タイプを対象とする場合は、先の手順で確認した issuetype の値を確認してください。

1.2.5. バグやクローズの文字列を設定する

バグの OPEN・CLOSE を集計する際に、どの課題タイプをバグとみなすか、どのステータスで完了とするかを設定することができます。

バグとみなすタイプ文字列に指定した課題タイプがレポート画面に表示される対象の課題となります。



(1) JIRA の管理メニューから課題を選択します。

(2) 課題タイプを開きます。バグとみなすタイプ文字列はこの課題タイプの名前を指定します。

(3)　課題タイプメニューからステータスを開きます。クローズとみなすステータス文字列はこのステータスの名前を指定します。

※課題が一部しか表示されない場合は jira.search.views.default.max の値を確認してください

※JIRA との接続に失敗する場合は以下の項目を確認してください

1. ログインに失敗する場合は JIRA で直接ログインをした後に連携設定を試みてください
   (JIRA のログインを複数回失敗すると CAPTCHA 認証が必要となります)
2. プロジェクト設定権限があることを確認してください
3. JIRA の認証設定が BASIC 認証となっていることを確認してください

1.3. JIRA クラウド型と連携する

1.3.1. ユーザ名とパスワードを入力する

クラウド型をご利用の方は API トークンの作成を行い、メールアドレスとパスワードを入力し

てください。

(1)　https://ja.confluence.atlassian.com/cloud/api-tokens-938839638.html を参考に API トークンの作成を行ってください。

(2)　ユーザ名／パスワードを以下に設定してください。

ユーザ名：メールアドレス

パスワード：API トークン

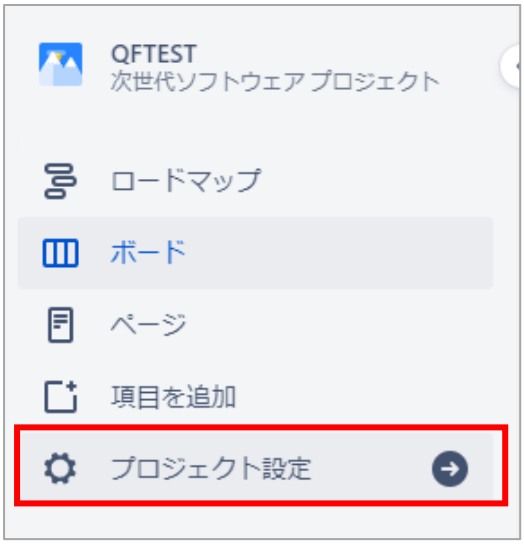
1.3.2.　新規チケット作成画面の URL を設定する

新規チケット作成画面の URL を設定すると、テストサイクルでテストを実行中に右クリックから簡単にチケットへの起票を行うことができます。

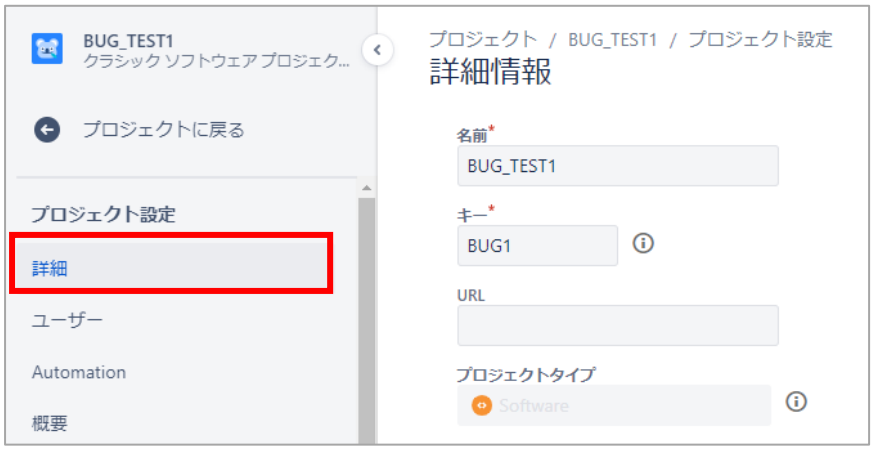
クラシックプロジェクトをご利用の場合

(1)　「クラッシックプロジェクト」をご利用の場合、JIRA の URL の後に「/secure/CreateIssueDetails!init.jspa?pid=yyy&issuetype=zzz」を追加します。PID と issuetype は以下手順で設定します。

(2)　プロジェクト設定を開きます。



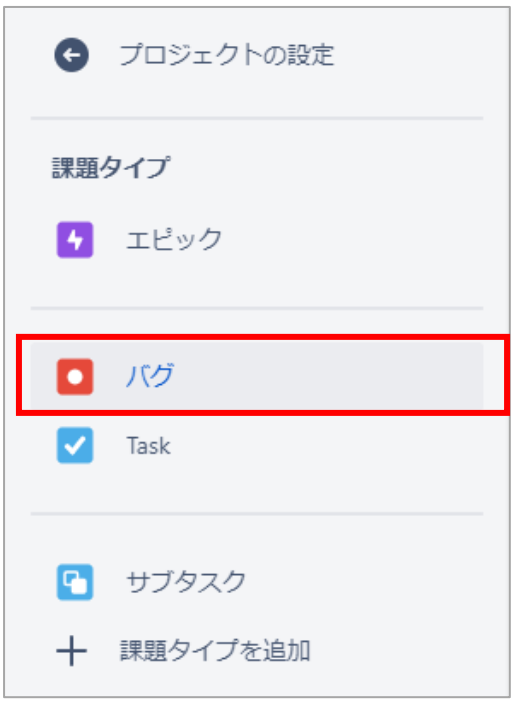
（３）　詳細情報を選択します。



（４）　詳細情報画面の URL の最後に記載されている PID を新規チケット作成画面の URL に設定します。

（５）　プロジェクト設定画面で課題タイプを選択します。

（６）　チケット新規作成時にデフォルトにしたい課題タイプを選択します。



（７）　課題タイプの画面 URL に記載されている issuetype を新規チケット作成画面の URL に設定します。

次世代プロジェクトをご利用の場合

次世代ソフトウェアプロジェクトでは、RestAPI から PID をご参照ください。手順は以下の通りです。

（１）　「次世代プロジェクト」をご利用の場合、JIRA の URL の後に「/secure/CreateIssueDetails!init.jspa?pid=yyy&issuetype=zzz」を追加します。PID と issuetype は以下手順で設定します。

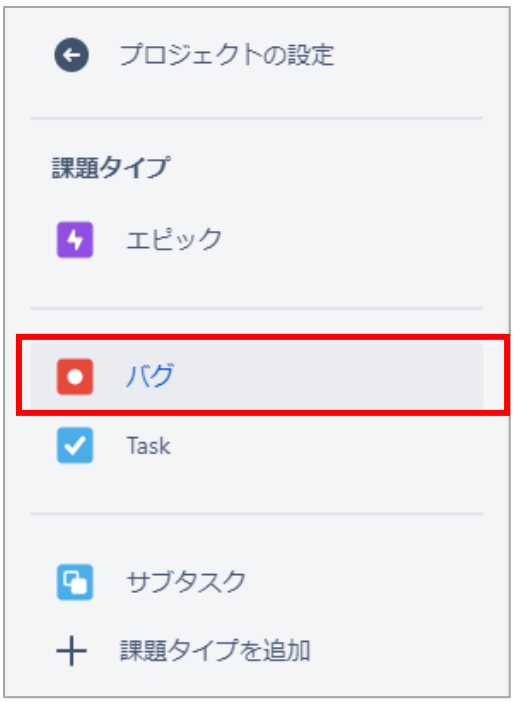
（２）　ブラウザに以下の URL を入力してください。JSON の「id」の項目が PID です。

https://[JIRA の URL]/rest/api/2/project/[プロジェクトキー]

（3） プロジェクト設定画面を開きます。

（4） プロジェクト設定画面で課題タイプを選択します。

（5） チケット新規作成時にデフォルトにしたい課題タイプを選択します。



（6） 課題タイプの画面 URL に記載されている issuetype を新規チケット作成画面の URL に設定します。

1.3.3. JQL を設定する

JQL は「JIRA Query Language」の略で、JIRA 専用のクエリ言語を指します。取得するプロジェクトや課題のタイプなどを絞り込むために JQL を設定する必要があります。未設定の場合は JIRA に登録されている全てのプロジェクト、課題が対象となります。

例）特定のプロジェクトを対象とする場合は project=”xxxxx”

例）特定の課題タイプを対象とする場合は issueType = ”xxx”

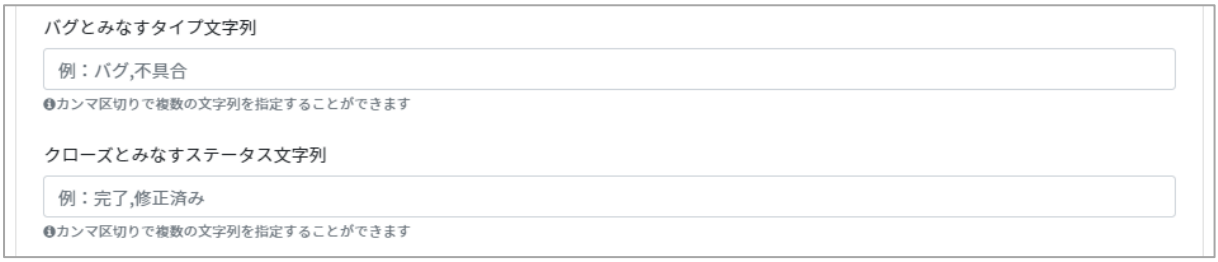
具体的な入力値について、プロジェクトを対象とする場合は JIRA に登録しているプロジェクトキーを、課題タイプを対象とする場合は、先の手順で確認した issuetype の値を確認してくだ

さい。

1.3.4.　バグやクローズの文字列を設定する

　バグの OPEN・CLOSE を集計する際に、どの課題タイプをバグとみなすか、どのステータスで完了とするかを設定することができます。

　バグとみなすタイプ文字列に指定した課題タイプがレポート画面に表示される対象の課題となります。



第2章　よくある質問と回答

2.1. Q: BTS 疎通ができているか確認したい

BTS の疎通確認は疎通確認画面より行っていただけます。確認手順は以下の通りです。

(1)　テストフェーズ一覧のテストフェーズ名下部にある「Redmine（または JIRA）との疎通確認」をクリックします。

(2)　疎通確認画面で「ベース URL の疎通確認」の項目が「〇」となっていれば疎通に成功しています。「×」となっている場合は設定を見直していただく必要がございます。

2.2. Q: BTS 疎通ができない

BTS に接続する際に外部のシステムである QualityForward からのアクセスが弾かれてしまう場合がございます。その場合は QualityForward の IP アドレス（第 1 章参照）からのアクセスを許可していただく必要がございます。

本設定は Redmine のシステム管理者権限を持っているユーザ様にご対応いただく必要がございます。

2.3. Q: BTS の件数がグラフに反映されない

フェーズ毎チャート画面の右上にある設定から「Redmine（または JIRA）と同期する」ボタ

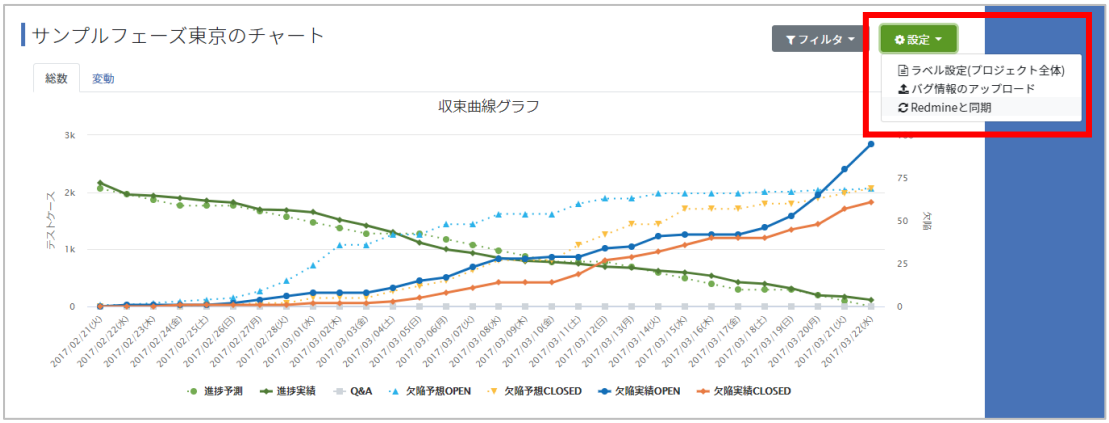
ンを押してください。※BTS の情報が自動で反映されるタイミングは 1 日 3 回（8 時・12 時・18 時）です。

2.4. Q: BTS の Open/Close の件数がレポートに反映されるのはいつですか？

1 日 3 回（8 時・12 時・18 時）です。任意のタイミングで更新したい場合は、設定から「Redmine（または JIRA）と同期する」ボタンを押すことでレポートに反映されます。

2.5. Q: 過去の BTS の Open/Close 数を修正したい

フェーズ毎チャート画面右上にある設定の「バグ情報のアップロード」から修正が可能です。



2.6. Q: JIRA のクラウド型を使用したい

以下の手順に従い設定を行ってください。

(1) こちらを参考に API トークンの作成を行ってください。

(2) ユーザ名／パスワードを以下のように設定してください。

ユーザ名：メールアドレス

パスワード：API トークン

2.7. Q: Redmine で BTS 疎通ができない

Redmine の設定が有効になっていない可能性があります。

(1) 「管理」→「設定」→「API」タブを開きます。

(2)　「REST による Web サービスを有効にする」にチェックを入れます。

(3)　「保存」ボタンを押します。

2.8. Q: BTS で集計されるチケット範囲を絞りたい

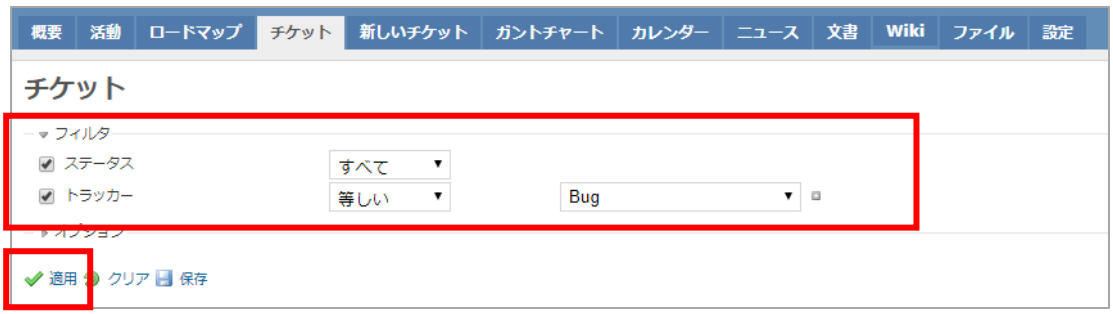
集計されるチケットの範囲を絞るには、Redmine 側で絞り込みを行ったものを指定していただく必要がございます。Redmine での絞り込み手順は以下の通りです。

(1)　Redmine の個人設定から API キーを取得します。

(2)　Redmine の対象のプロジェクト開き、URL を取得します。
例）https://xxx.xxxx/projects/xxxx/

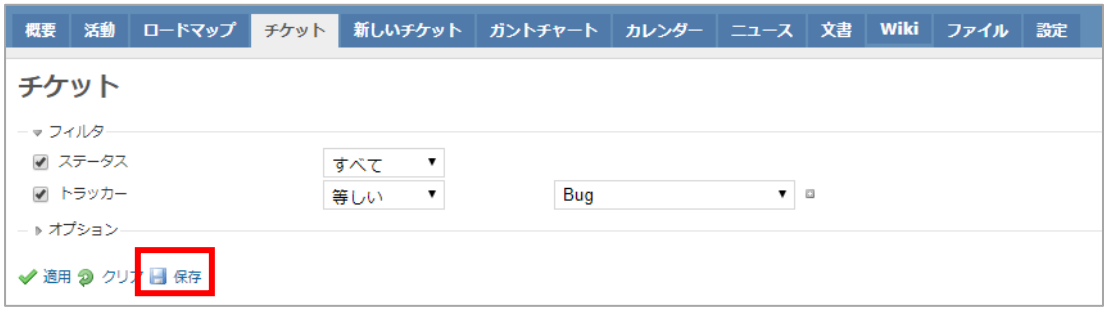
(3)　手順(2)で取得した URL に「issues.json?key=API キー」を追加します。例）
https://xxx.xxxx/projects/xxxx/issues.json?key=xxxx

(4)　Redmine で対象チケットの絞込みを行います。バグ一覧を取得するために「すべて」の「バグ」を対象にし、適用ボタンを押します。



※ここでは「すべて」の「バグ」を対象にしていますが、フィルタ条件はプロジェクト方針に合わせて自由に設定していただけます。

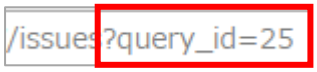
(5)　すべてのバグの一覧が表示されたら保存ボタンを押します。



(6)　新しいクエリに名前を付けて保存します。

(7)　チケット一覧右側のカスタムクエリ一覧に手順(6)で作成したクエリが表示されます。作成したクエリ名をクリックします。

(8)　URL の最後にクエリ ID が表示されるので、「query_id=xx」をコピーします。



(9)　手順(3)までで作成した URL の最後に手順(8)の「&query_id=xx」を入力します。
例)https://xxx.xxxx/projects/xxxx/issues.json?key=xxxx&query_id=xx

(10)　手順(9)でできた URL をブラウザのアドレス欄に直接入力します。json の取得が確認できたら URL を登録欄に入力し、登録ボタンを押します。

2.9. Q: チケットを CLOSE にしたのに欠陥 OPEN 数が変わらない

欠陥 OPEN 数はチケットの総数を表しているため、グラフが減ることはありません。