フロントエンドでSentryを正しく初期設定する

フロントエンド用にSentryをセットアップするためのポイントは4つあります

1. 最新のSentry SDKを利用する
2. ソースマップを設定する
3. エラーをフィルターする
4. ワークフローと統合する
5. （**POC**のみ）: エラーを100%レポートする
6. 詳細な利用法などについてのお問合せ

1. 最新のSentry SDKを利用する

Sentry SDKの最新バージョンを使用していることを確認してください

1. https://docs.sentry.io/ へ行きます
2. 利用しているプラットフォーム(React, React Nativeなど)を選択します
3. 右上のパッケージとバージョン番号を確認し、必要であればSDKをアップデートしてください

2. ソースマップを設定する

Sentryでソースマップを適切に設定することは非常に重要です。この設定が正しくできていないと、開発者の生産性が著しく低下します

ソースマップを正しく設定すると以下のような利点があります:

- エラーが発生したファイルやコード行を正確に確認することができます
- 検知したエラーをグループ化し、検索し、フィルタリングすることができます
- どのコミットでエラーが発生したかを確認できます

Sentryでソースマップを設定するにはいくつかの注意点が必要です。そのプロセスを以下にまとめます

ステージング環境においては、以下の2つのステップが必要となります

1. ソースマップを生成する
2. ステージング環境にソースマップをデプロイする

本番環境においては、以下の4つのステップが必要となります

1. ソースマップを生成する
2. ソースマップをSentryにアップロードする
3. 本番環境にはソースマップをデプロイしない
4. ソースマップのリリースバージョンをSentry.init()で設定する

© Ichizoku 2023 All Rights Reserved

ソースマップを生成するプロセスは、ご利用のフロントエンドフレームワークによって異なります

また、ソースマップを本番にデプロイしないようにするための処理も、ご利用のフロントエンドフレームワークによって異なります

Sentryには、お客様の特定の設定を支援するライブラリがあります。SentryのライブラリはNextJS、Vue、Angular、Remixなど、主要なフロントエンドフレームワークをすべてサポートしています。また、これらのライブラリは、Webpack、Vite、Rollup、TypeScriptなどのコンパイラもサポートしています

セキュリティ上の理由から、本番環境にソースマップをアップロードするべきではありません。この場合、ソースマップはSentryに直接アップロードする必要があります

Sentryにソースマップをアップロードする際、ビルドに関連するソースマップのバージョンをSentryに伝える必要があります。これは、Sentry.init()のreleaseプロパティで行います

デフォルトでは、この値はgitコミットハッシュの全体となります。この動作を手動で上書きすることもできます。 しかしここではシンプルに、デフォルトの値を使って説明をします

git のコミットハッシュをフロントエンドのコードに渡す最も簡単で一般的な方法は、環境変数を使用することです

次に、Sentry.init()で、この環境変数と同じ "release "を設定します
具体例:

```
Sentry.init({
 // ... other values
 release: process.env.GIT_COMMIT_SHA,
})
```

ステージング環境では、この環境変数を未定義のままにしておきます。Sentryはステージングサーバーから直接ソースマップにアクセスできるため、リリースバージョンを知る必要はありません。これは推奨され、期待される挙動です

ソースマップが本番環境ではなくステージング環境にデプロイされていることを確認するため、ビルドとデプロイのログを必ず確認してください

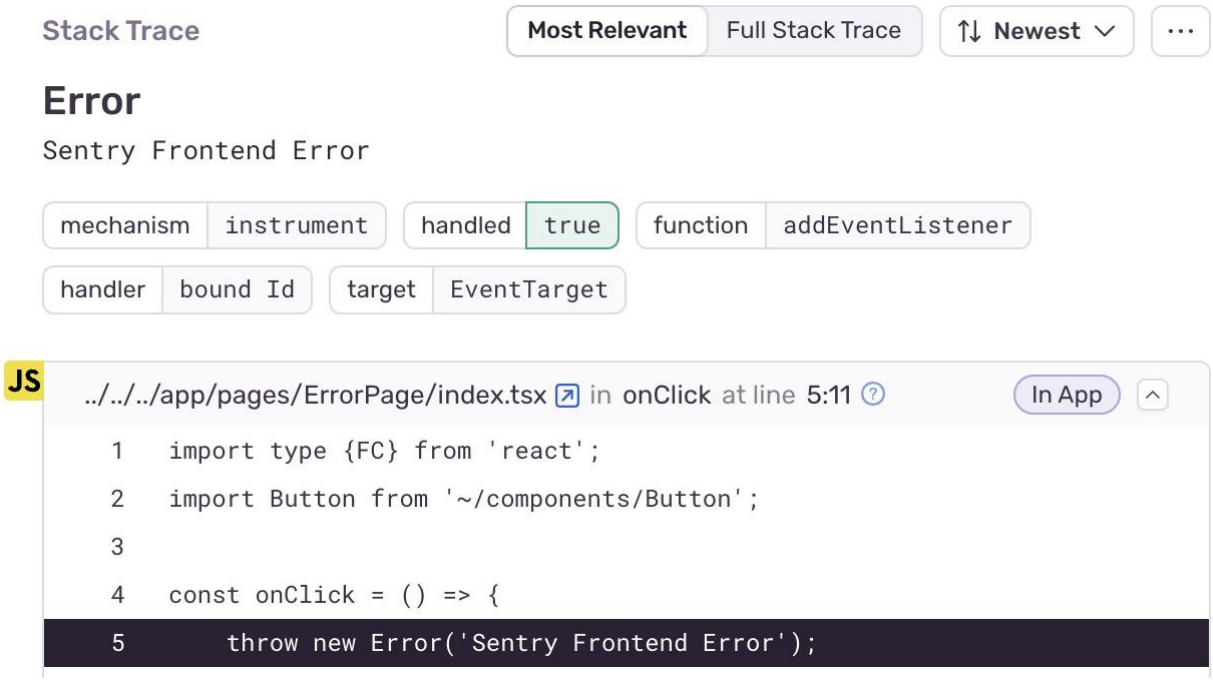
すべてが機能していることを簡単にテストする方法の1つは、エラーを返すボタンを置いた一時的なテストページを作成してデプロイし、そのボタンをクリックすることで、Sentryダッシュボードにアクセスしてエラーを見つけるというものです

```
const onClick = () => {
```


© Ichizoku 2023 All Rights Reserved

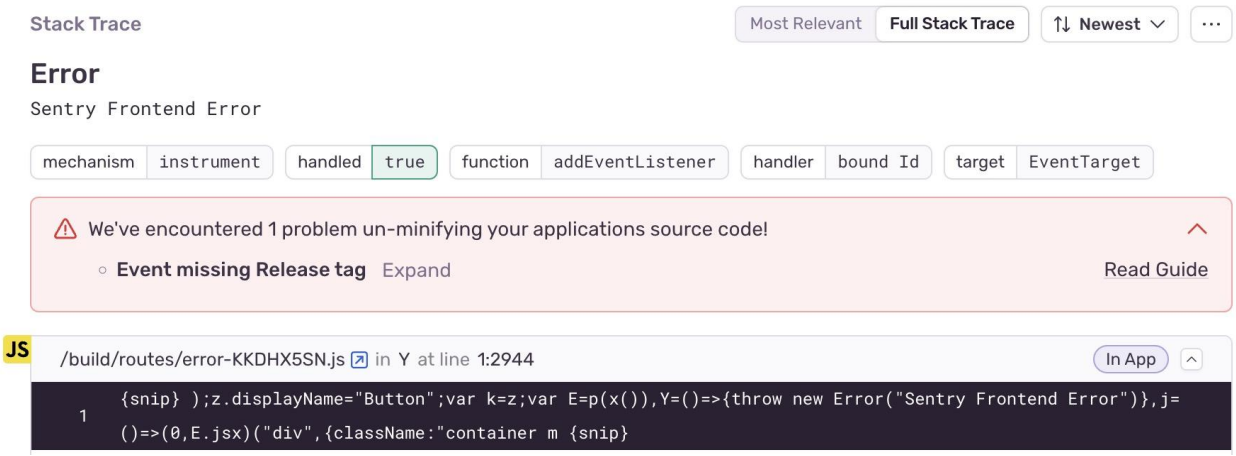
```
  throw new Error('Sentry Frontend Error');
};

<button onClick={onClick} type="button">
  Test Error
</button>
```

ステージング環境と本番環境でエラーを見ると、以下のようにソースコードが見えるはずです



© Ichizoku 2023 All Rights Reserved

ソースマップがSentryにアップロードされていない場合、またはreleaseの値が正しく設定されていない場合、赤いアラートボックスにメッセージが表示され、スタックトレースはミニファイされたコードになります



3. エラーをフィルターする

エラーのフィルタには2種類の方法があります

1. クライアントフィルタリング：Sentry.init() 内に設定します
2. インバウンドフィルタリング：Senryのアプリケーション管理画面で設定します

クライアントフィルタリング

https://docs.sentry.io/platforms/javascript/configuration/filtering/

Sentry.init() には、以下の4つのフィルター設定があります

- beforeSend
- ignoreErrors
- allowUrls
- denyUrls

インバウンドフィルタリング（Business, Enterpriseプランのみ）

https://docs.sentry.io/product/data-management-settings/filtering/

インバウンドフィルタは以下の手順に沿って設定します

1. 左カラムからProjectsをクリックします

© Ichizoku 2023 All Rights Reserved

2. Projects画面で、設定するプロジェクトの名前をクリックします
3. 右上の歯車マークをクリックします
4. 2列目の左メニューのProcessingセクションにある、”Inbound Filters”をクリックします

4. ワークフローと統合する

https://docs.sentry.io/product/integrations/integration-platform/public-integration/

Sentryをコードリポジトリや問題追跡ソフトウェアと統合することで、多くのメリットが得られます

- 問題解決までの時間を短縮する
  - Sentryは、コードベースにエラーが入った疑いのあるコミットがどれかまでを指摘できるようになります
  - エラーは、コードを書いた開発者に（自動または手動で）割り当てられるようになります
  - SentryのStacktraceは、リポジトリのエラーコードのファイルと各行に直接リンクさせることができます
- ワークロードの削減
  - チケットはSentryのイシュー画面から直接簡単に作成でき、Sentryの情報に基づいて自動的に割り当てられ、課題が解決されるとチケットも更新されます
  - 問題追跡ソフトウェアで問題を解決すると、Sentryでも解決され、コミットに関連づけられます。またその逆も同様です
- 視認性と対応速度の向上
  - Slackとの連携

Sentryは、コードリポジトリや問題追跡ソフトを統合する以外にも、Slack/Microsoft Teams連携など、多くの統合機能を備えています

Integrationsページに移動すると、利用可能なすべての統合・外部連携機能を確認することができます。左側のサイドバーで”Settings"をクリックし、内側のサイドバーで”ORGANIZATION"の下にある“Integrations"をクリックします

またSentryは、必要であればカスタムの統合・連携もサポートします。Sentryではこれらを”Internal Integrations"と呼んでいます。詳細は、以下のドキュメントをご覧ください
https://docs.sentry.io/product/integrations/integration-platform/internal-integration/

© Ichizoku 2023 All Rights Reserved

Githubとの連携

https://docs.sentry.io/product/integrations/source-code-mgmt/github

Github連携を有効にしたら、Sentryの内部で以下を設定する必要があります

Integrationsページにおいて:

- GitHubをクリックします
- メニューからConfigurationsをクリックします
- Githubのリポジトリー覧が表示されるので、そこから”Configure”の歯車をクリックします

重要な設定事項としてUser MappingsとCode Mappingsがあります

User Mappings

Sentryは、SentryユーザーをGitHubの関連するユーザーアカウントにマッピングする必要があります。各開発者のユーザーマッピングを追加します

Code Mappings

Sentryで問題を見るとき、SentryがStacktraceをGitHubの特定のファイルにマッピングできないという内容の警告が表示されることがあります。この問題は、GitHub連携設定ページの"Code Mappings"タブから解決できます

https://docs.sentry.io/product/integrations/source-code-mgmt/github/#stack-trace-linking

Jiraとの連携

詳細は以下のドキュメントをご覧ください
https://docs.sentry.io/product/integrations/issue-tracking/jira/

© Ichizoku 2023 All Rights Reserved

5.（POCのみ）: エラーを100%レポートする

POCの一環として、アプリケーションで発生するすべてのエラーの全容を把握できるエンタープライズプランをご利用いただけます

Sentry.init()を以下の設定で編集し、Sentryがすべてをレポートするように設定します

```
Sentry.init({
  // ... other values
  sampleRate: 1.0,
  tracesSampleRate: 1.0,
  integrations: [
    new Sentry.Replay(), // web only
  ],
  replaysSessionSampleRate: 0.1, // confirm with Adam
  replaysOnErrorSampleRate: 1.0,
})
```

6. 詳細な利用法などについてのお問合せ

IchizokuはSentryと提携し、日本でSentry製品の導入支援、テクニカルサポート、ベストプラクティスの共有を行なっています。Ichizokuが提供するSentryの日本語サイトについてはこちらをご覧ください。またご導入についての相談はこちらのフォームからお気軽にお問い合わせください。

© Ichizoku 2023 All Rights Reserved